Q3D Extractor® 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 2025 R1
January 2025

Copyright and Trademark Information

Disclaimer Notice

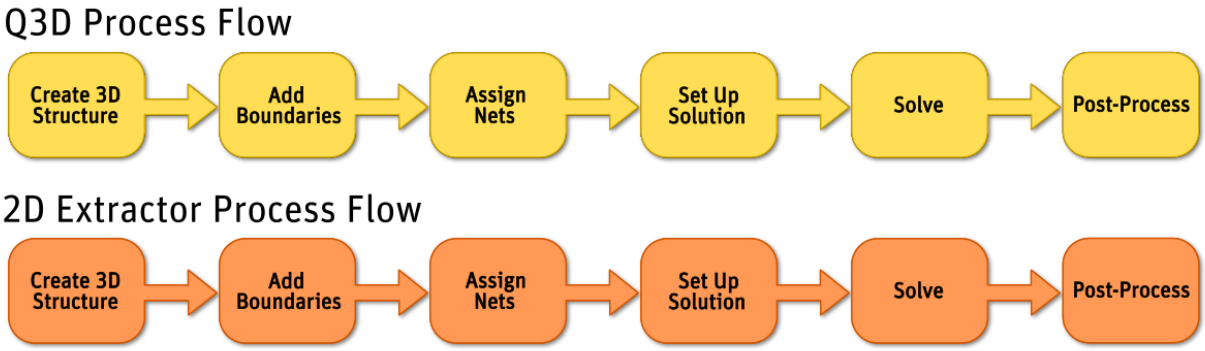
U.S. Government Rights

Third-Party Software

Table of Contents

1 - Welcome to Q3D Extractor Help

Q3D Extractor is a quasi-static 3D solver for extracting lumped RLGC parameters and Spice models. It also contains a 2D solver for extracting per-unit-length RLGC parameters of transmission lines. ***2D procedures are the same as those for 3D, unless otherwise indicated***.



Q3D Extractor Getting Started Guides

Q3D Extractor documentation includes the following Getting Started Guides:

- *Getting Started with 2D Extractor: Grounded Coplanar Waveguide Model*

  This Getting Started Guide leads you step-by-step through creating, solving, and analyzing the results of a parameterized 2D model. By following the steps in this guide, you will learn how to perform the following tasks in 2D Extractor:

  - Drawing a geometric model.
  - Modifying a model's design parameters.
  - Assigning conductors.
  - Reducing Matrices.
  - Specifying solution settings for a design.
  - Validating a design's setup.
  - Running a 2D Extractor simulation.
  - Creating a plot of results.
- *Getting Started with Q3D Extractor: PCB Via Model*

  This Getting Started Guide leads you step-by-step through creating, solving, and analyzing the results of a parameterized 3D model representing a via on a Printed Circuit

Board. By following the steps in this guide, you will learn how to perform the following tasks in Q3D Extractor:

- Drawing a geometric model.
- Setting up vias and traces.
- Modifying a model's design parameters.
- Assigning variables to a model's design parameters.
- Specifying solution settings for a design.
- Validating a design's setup.
- Running a simulation.
- Creating a plot of results.

- *Getting Started Guide: Cable Modeling Solutions*

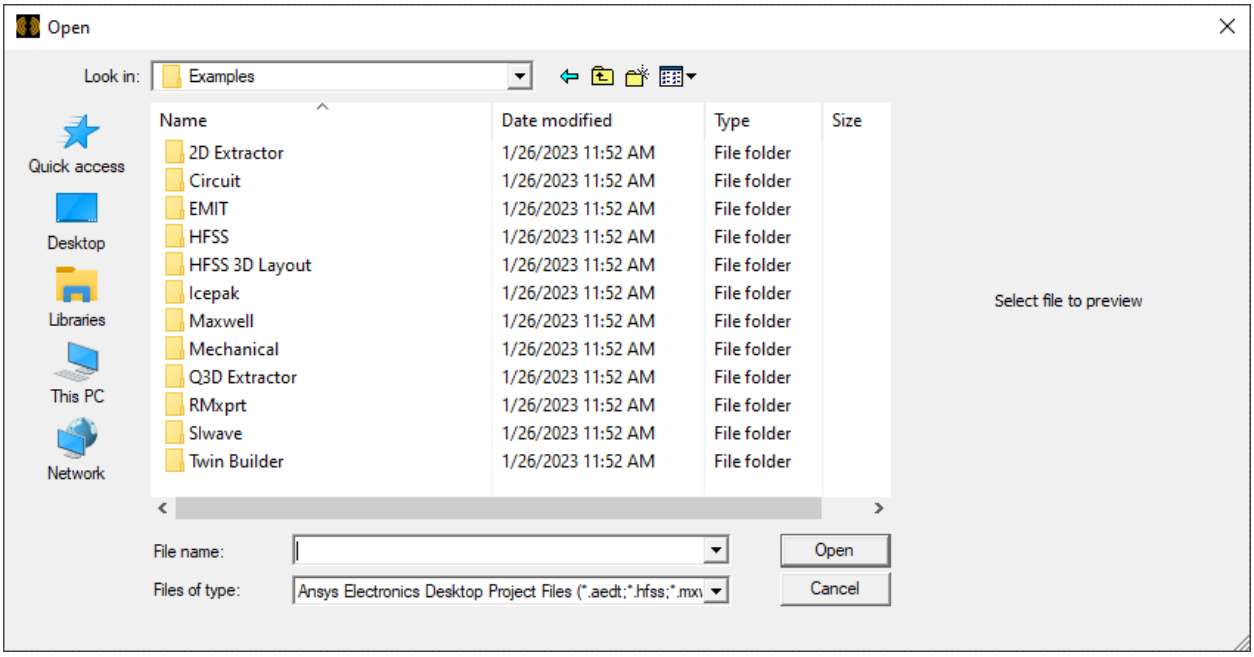
  This Getting Started Guide leads you through a cable modeling solution implemented using dynamic/data links between HFSS, 2D Extractor, and Circuit. A cable harness in HFSS is modeled as a single external field source based on quasi-static simulation of each cable cross-section in 2D Extractor and an analysis of the cable network in Circuit. The magnitude and distribution of the fields along each cable section is determined by the voltages and currents at the ends of each section, and then transmission line model is applied to propagate these fields along the cable length.

  By following the steps in this guide, you will learn the following steps and data transfers:

  - 2D Extractor
    - Solving the cable cross-section.
    - Sending a transmission line model for the cable network solution.
    - Sending transmission line modes and fields for the 3D cable solution.
  - Circuit
    - Defining a step voltage on the appropriate ports on the circuit schematic.
    - Solving the cable network.
    - Sending v and i at the ends of the cable for the 3D cable solution.
  - HFSS
    - Mapping fields onto the 3D cable, and solving the emissions from the cable assembly.

Example Projects

Your Ansys Electronics Desktop installation includes an example directory containing projects folders for several kinds of designs.



Example projects are organized by the design type.

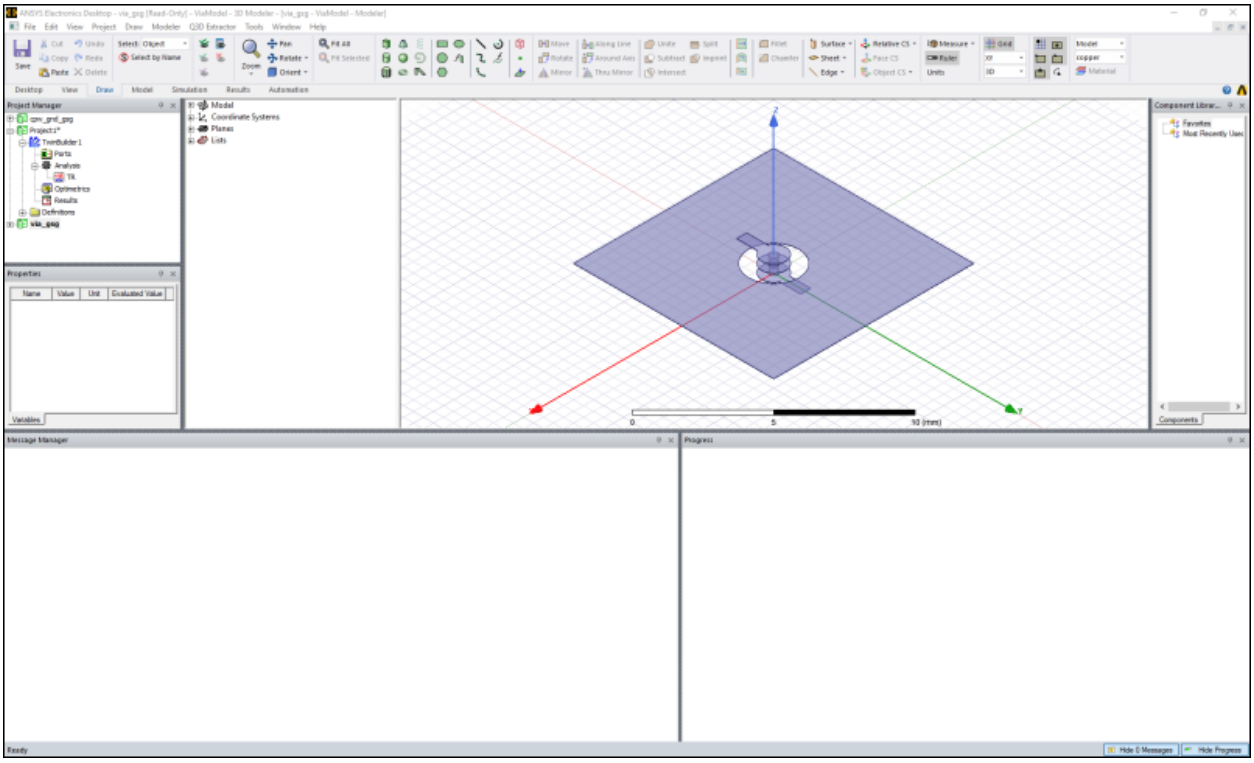
Several of these projects are associated with detailed getting started guides.

Q3D Extractor Example Project

The Examples\Q3D Extractor directory contains a ViaModel project (via_gsg.aedt), described in its associated Getting Started Guide.

To open the example, click **File** > **Open Examples**, or click **Open Examples** on the **Desktop** tab.

The Getting Started Guide is located in the "...\ANSYS Inc\v251\AnsysEM\Help\Q3DExtractor\GSG" directory.

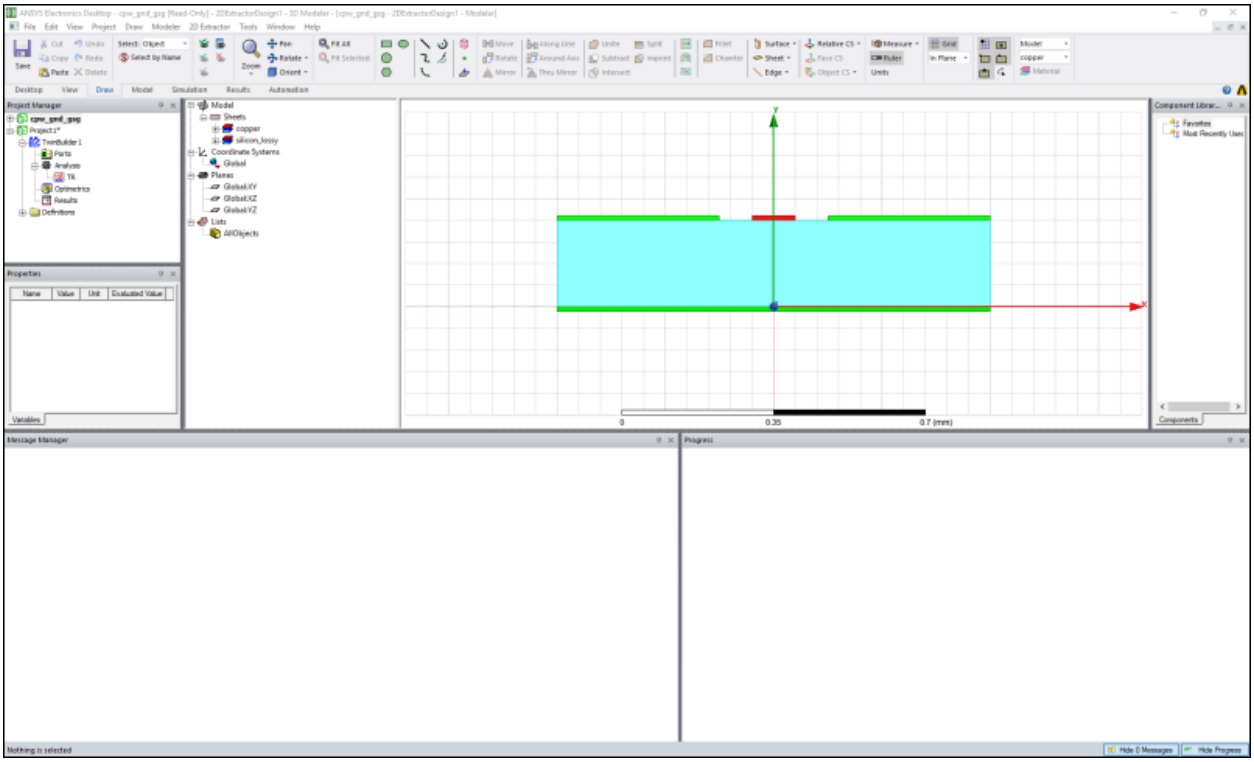


2D Extractor Example Project

The Examples\2D Extractor directory contains the cpw_gnd_gsg.aedt project, described in its associated Getting Started Guide.

To open the example, click **File** > **Open Examples**, or click **Open Examples** on the **Desktop** tab.

The Getting Started Guide is located in the "...\ANSYS Inc\v251\AnsysEM\Help\Q3DExtractor\GSG" directory.



Finding Information in the Online Help

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

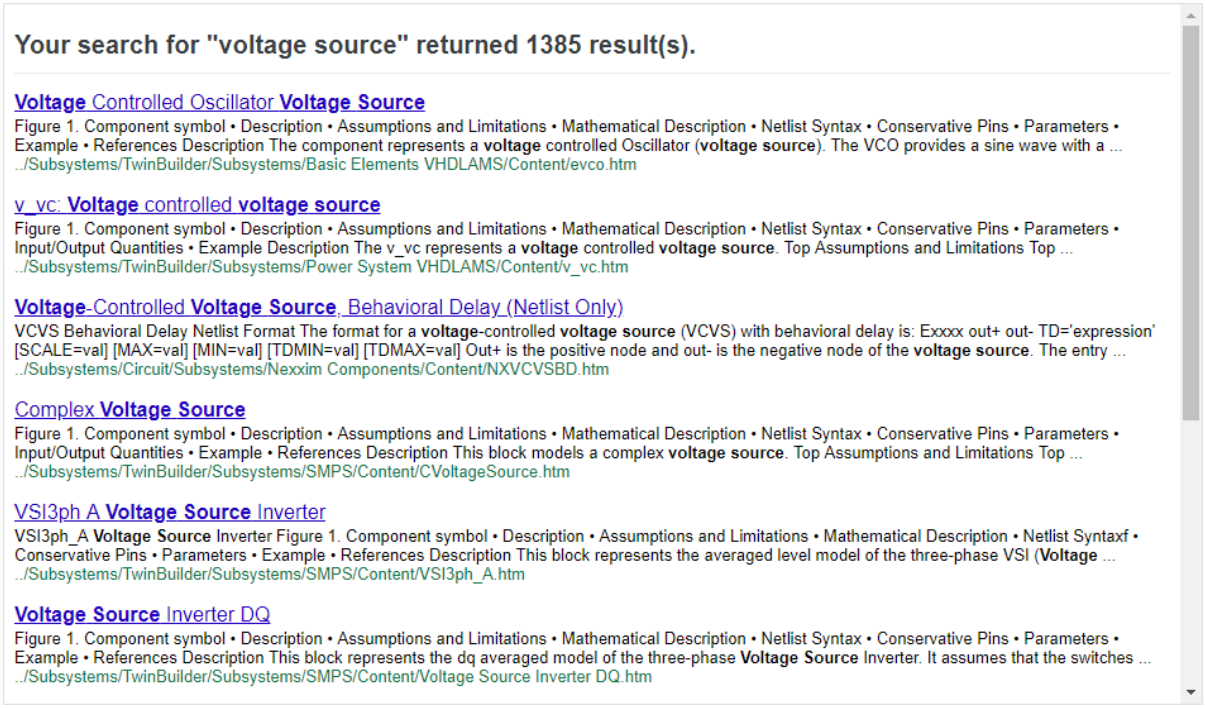
- Press **F1** on any open dialog box to open the relevant help topic.
- *A hierarchical table of contents* – Contents – Browse through the table of contents, expand entries, and close entries. Click an entry to see it in the content area.
- *A full text search* – To locate occurrences of a word or phrase that may be contained in the help, use the search function.

> **Note:**
>
> Ansys Electronics Desktop Student includes access to PDF documentation only.

Using the Search Function in the Help

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

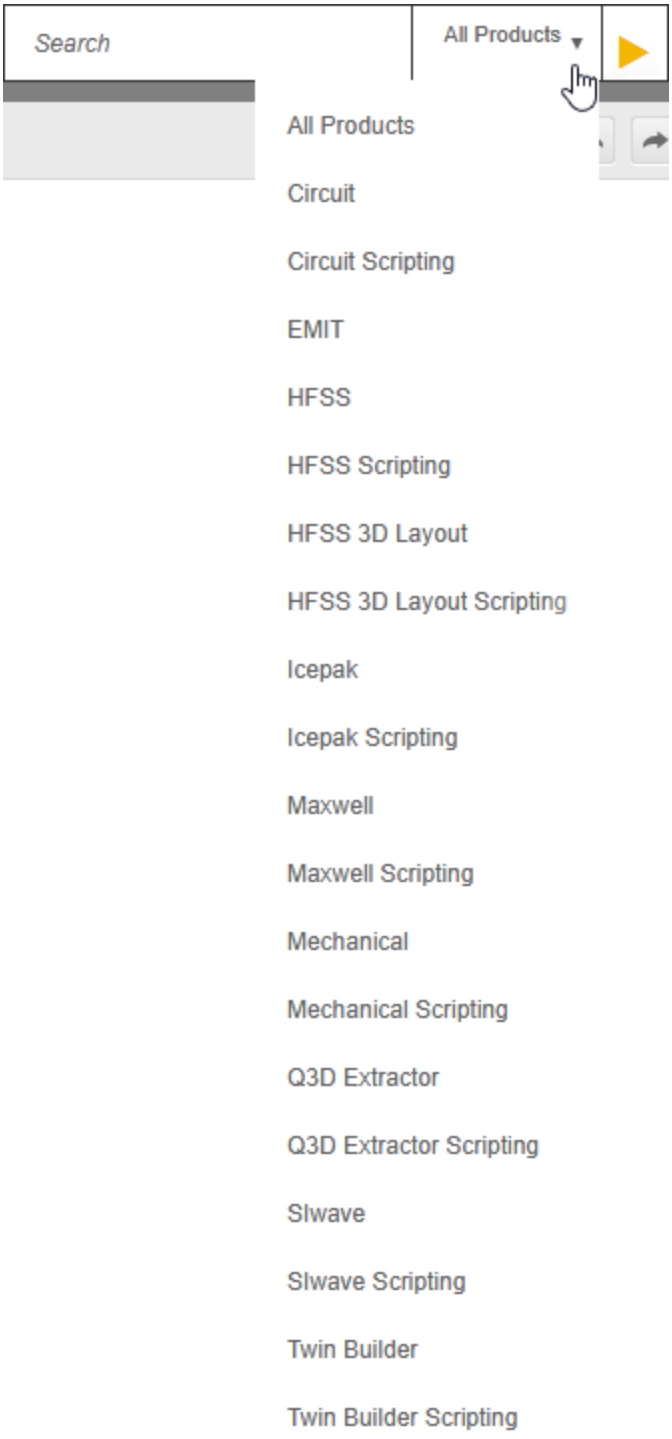
marks, the results show all topics that contain “voltage” or “source.”



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 in the search box.
    - If you are searching within the full Electronics help system, you see a search box that includes a drop-down filter for specifying a product, a product's scripting guide, or searching across all products. When you change the filter, the results dynamically reflect the selected filter.

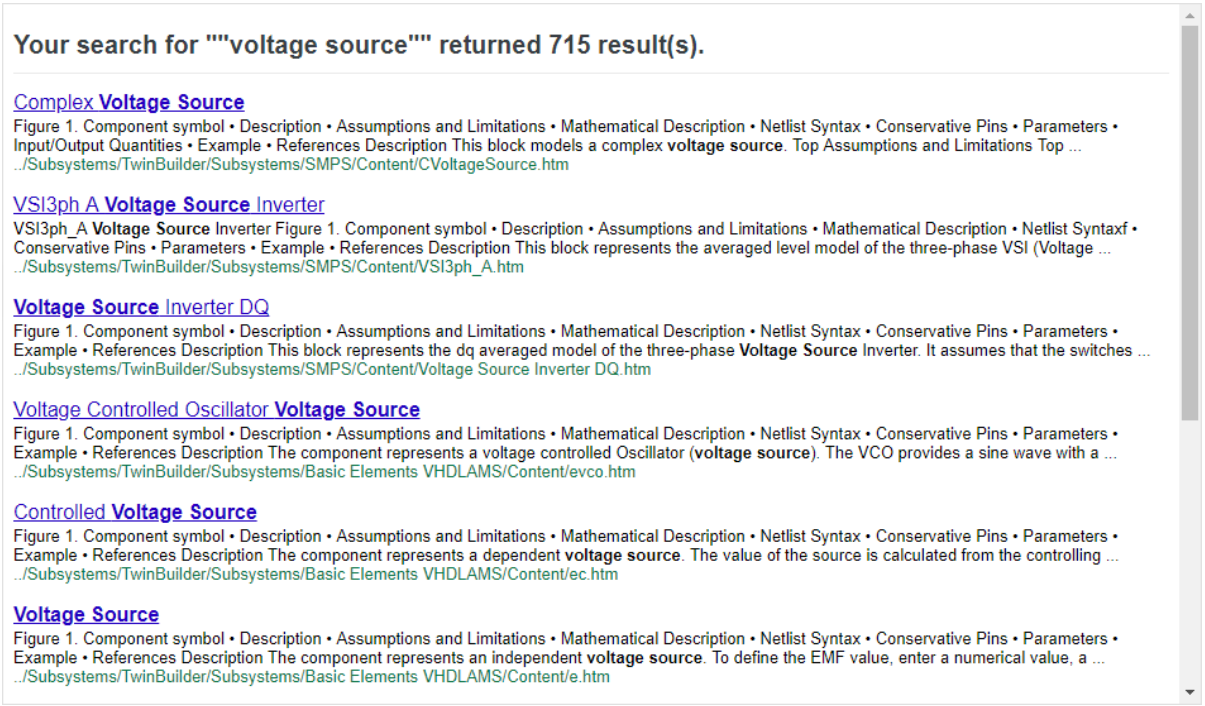


- If you accessed the help for a specific product by pressing **F1** in Electronics Desktop, the search box will have preselected that product. You can change the selection to a different product, a product's scripting guide, or all products.

2. Click on the topic you want in the results list.
    - Some topics in different products share the same title. When searching "All Products," check the URL below each link. The URL indicates the product under which the topic falls.
    - After clicking a topic, if you want to view a different topic, click your browser's back button to return to the results list.
    - To turn off highlighting on the page you are viewing, click the Remove Highlights icon ( ).

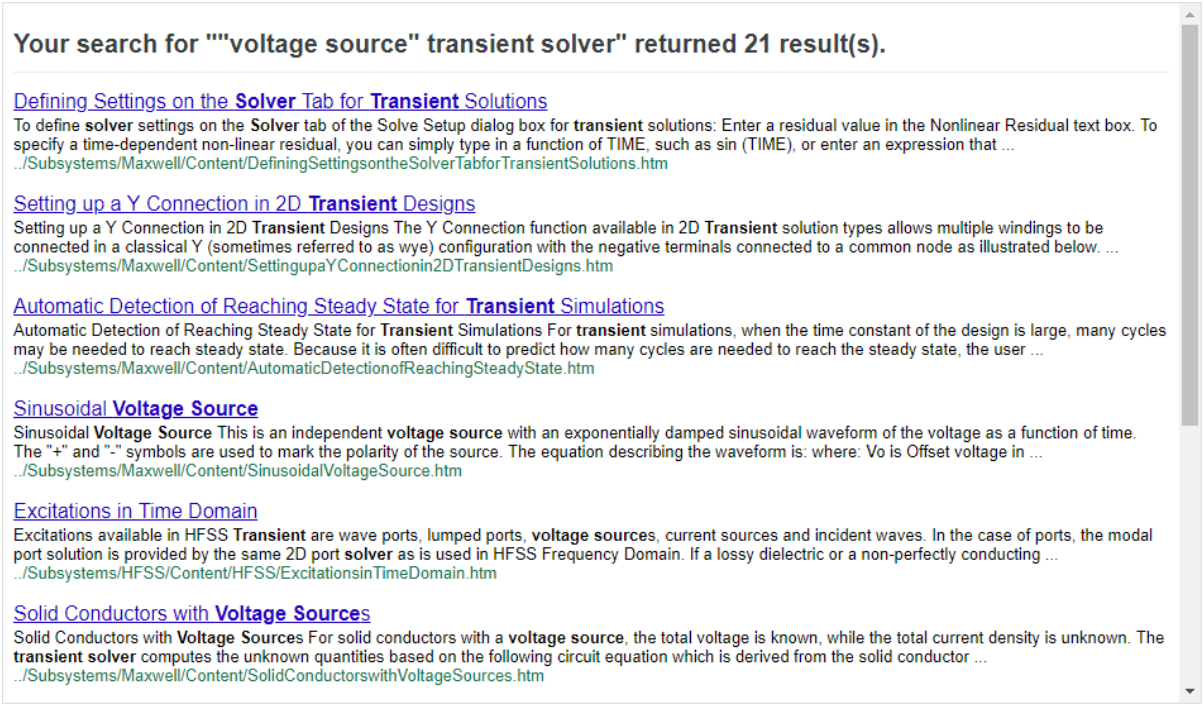
Searching with Quotation Marks

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

Note:

Functionality featured in the example(s) in this section applies to multiple design types.

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

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

Help Conventions

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

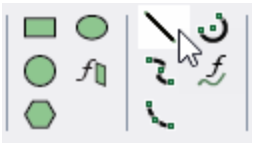
- Procedures are presented as numbered lists. A single bullet indicates that the procedure has only one step.
- Bold type is used for the following:
  - Keyboard entries that should be typed in their entirety exactly as shown. For example, “**copy file1**” means you must type the word **copy**, then 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.

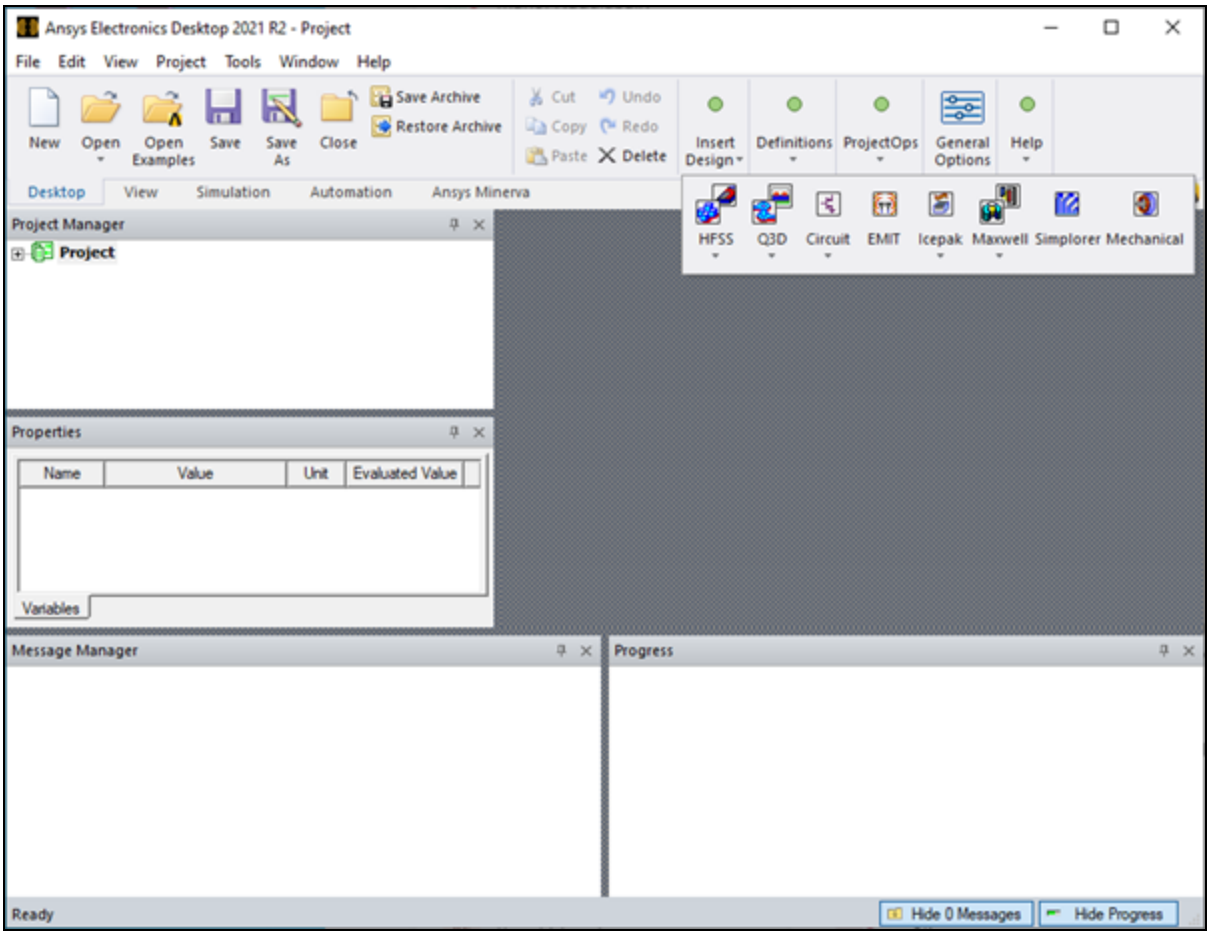
Getting Help from Ansys Technical Support

To contact Ansys technical support staff in your geographical area, please log on to the Ansys corporate website, ansys.com/support. You can also contact your Ansys EM account manager in order to obtain this information.

E-mail can work well for technical support. All Ansys EM software files are ASCII text and can be sent conveniently by email. 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.

2 - Getting Started with Ansys Electronics Desktop

The Ansys Electronics Desktop, shown in the following figure, provides a comprehensive environment for designing and simulating various electronic components and devices. The Electronics Desktop consists of a unified user interface where electromagnetic designs, thermal designs, and circuits 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 includes the following design types:

- **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.

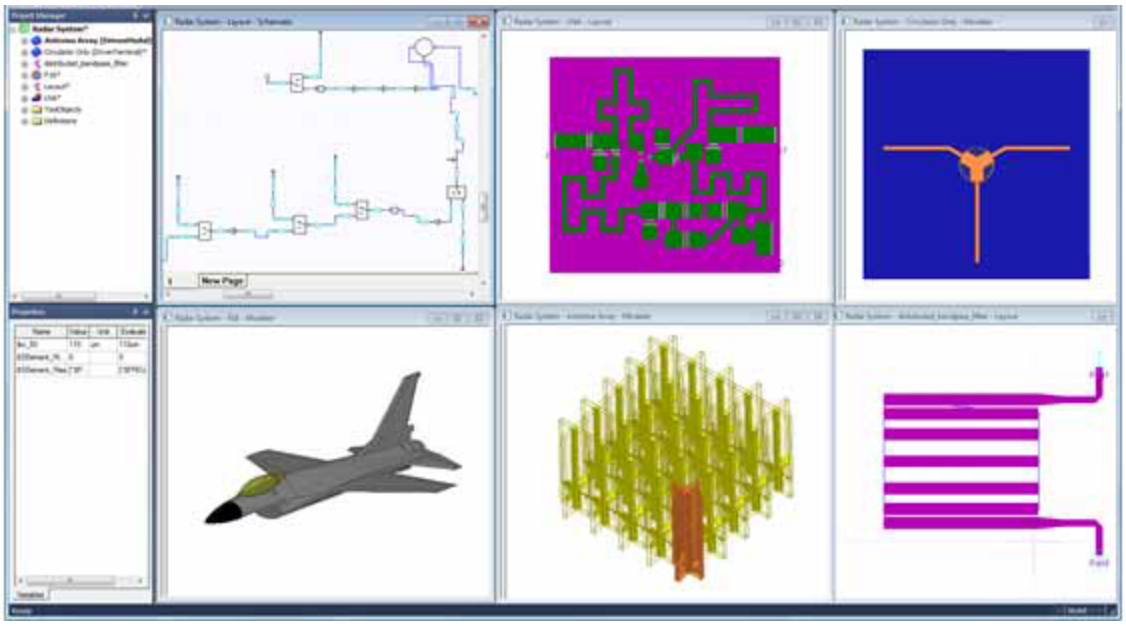
- **Q3D Extractor** – a quasi-static 3D solver for extracting lumped RLGC parameters and Spice models.
- **2D Extractor** – a 2D solver for extracting per-unit-length RLGC parameters of transmission lines.
- **Circuit** – a schematic-based interface to the Nexxim circuit simulator.
- **Circuit Netlist** – a netlist (text-based) interface to the Nexxim circuit simulator.
- **EMIT** – system simulation for predicting and mitigating radio frequency interference (RFI) in electronic devices.
- **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 Maxwell.
- **Maxwell Circuit** – sets up external circuit designs to supply excitations to coil terminals for Maxwell 2D and 3D Eddy Current and Transient designs.
- **Simplorer** – an integrated, multi-domain, mixed-signal simulator for complex technical systems. Simplorer is a subset of the Twin Builder standalone product. Please see the Twin Builder help for more information.
- **Icepak** – a fluid and thermodynamic simulator for electronic systems and components.
- **Mechanical** – perform modal analyses to determine natural vibration frequencies and thermal analyses to determine temperatures and heat flux.

You can access all of these design types and features from the **Project** menu, and any combination of 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. Ansys Electronics Desktop provides an efficient way to manage complicated projects that require several different analysis tools to model all of their 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.

You can access these design types and features from the Windows launcher. You can use the ACT Toolkit for HFSS-EMA3D Datalink to launch and use this tool.

The following illustration shows how the Ansys Electronics Desktop can be used to model different components for radar system analysis. An antenna array created in HFSS is linked to an IE design of an F16 aircraft. The low noise amplifier and bandpass filter are two important components in the receiver part of the radar module circuit design. The low noise amplifier and the filter can be modeled in HFSS 3D Layout and linked together in a circuit simulation, along with other components of the radar module connected to the antenna array. The outputs of the radar module can be used to drive the antennas using the push excitation feature, whereby the

voltages on the antenna array ports can be automatically set to correspond to those of the driving circuit. The push excitation feature enables the user to view electromagnetic fields when the array is driven by the radar module circuit.



Ansys Electronics Desktop Student

Ansys Electronics Desktop Student is a free Windows version of the Electronics Desktop that allows you to model, mesh (if applicable), solve, and post-process in HFSS, Q3D Extractor, 2D Extractor, Circuit, 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
- optiSLang and LSDSO not supported
- Integration with Ansys Workbench not supported
- Beta features not supported
- Linux version not supported

Refer to the following sections to review limitations specific to the supported design types in Ansys Electronics Desktop Student.

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
- Circuit model generation from S-parameters not supported

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
- Circuit model generation from S-parameters not supported

Circuit Limitations

- Netlist design type not supported
- Component limit: 50 components
- Command line analysis initialization not supported
- Device data generation not supported
- Access to layout functionality not supported
- Circuit model generation from S-parameters not supported

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
- Classic Icepak files (.tzr) with ECAD not supported

System Requirements

Ansys Electronics Desktop supports certain versions of Windows and Linux. For supported platforms and system requirements, go to the Platform Support website and select the following document:

Ansys 2025 R1 **- Platform Support by Application / Product (PDF)**

This document covers all Ansys products. Refer to the **Electronics Applications** section.

Limitations of Linux Installations:

While the majority of the Ansys Electromagnetics Suite applications and features are supported on both Windows and Linux platforms, some are not supported on Linux. See the following topic for details:

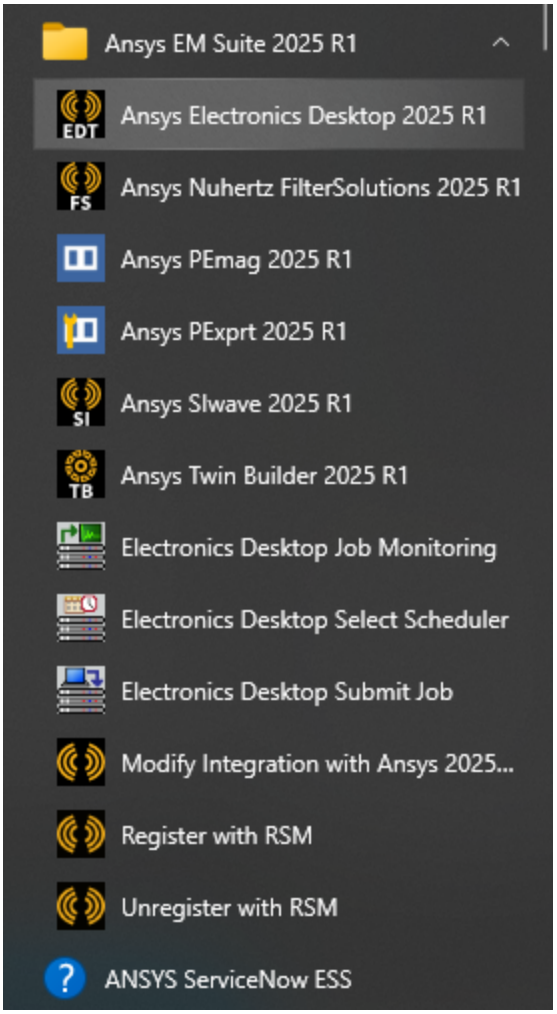
Windows vs. Linux Installations

Launching Ansys Electronics Desktop

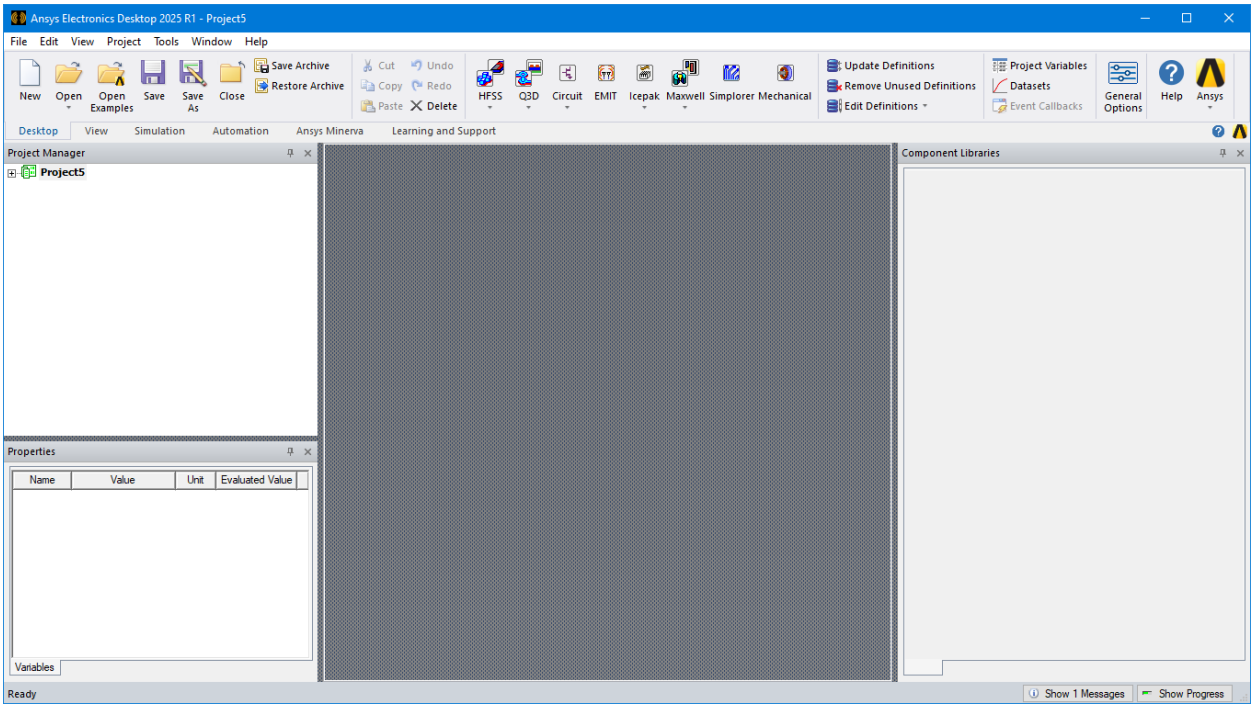
Once you have installed Ansys Electronics Desktop, start the program using one of the following methods:

- On the Windows desktop, double-click the Ansys Electronics Desktop icon.

- From the Windows Start menu, select **Ansys EM Suite 2025 R1** > **Ansys Electronics Desktop 2025 R1**.



Ansys Electronics Desktop opens.



If the program fails to start, ensure that:

- You have installed the licensing option provided to you. For detailed information on installing the software and licenses, see Ansys EM Suite Installation.

  This topic provides links to the Ansys installation guides (Windows and Linux) and product-specific platform support information.

- Ansys QA Services are not enabled. To disable, change the value of the ENABLE_ANSYS_QA_SERVICES environment variable to 0.

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 box 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 menu item **Help > Ansys Product Improvement Program** or ribbon item **Help > Ansys > Ansys Product Improvement Program**. This activates the **Ansys Product Improvement Program** dialog box. On this dialog select the **No I am not willing to participate** radio button to opt out of the program. Select the **Yes, I am willing to participate in the Ansys Product Improvement Program** radio button to opt in to the program. Click the **OK** button to accept the radio button selection. Data is no longer collected or sent if you opt out.

> **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. After installing the software, run the following command as a user with permissions to modify the installed file set:

```
UpdateRegistry.exe -set -ProductName ElectronicsDesktop2025.1 -
RegistryKey
Desktop/Settings/ProjectOptions/ProductImprovementOptStatus -
RegistryLevel install -RegistryValue 1
```

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?*

   a. Select the menu item **Help** > **Ansys Product Improvement Program** or ribbon item **Help** > **Ansys** > **Ansys Product Improvement Program**. This will activate the **Ansys Product Improvement Program** dialog box.

   b. Select the **No I am not willing to participate radio** button to opt out of the program.

   c. Click the **OK** button to accept the radio button selection.

   d. 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.

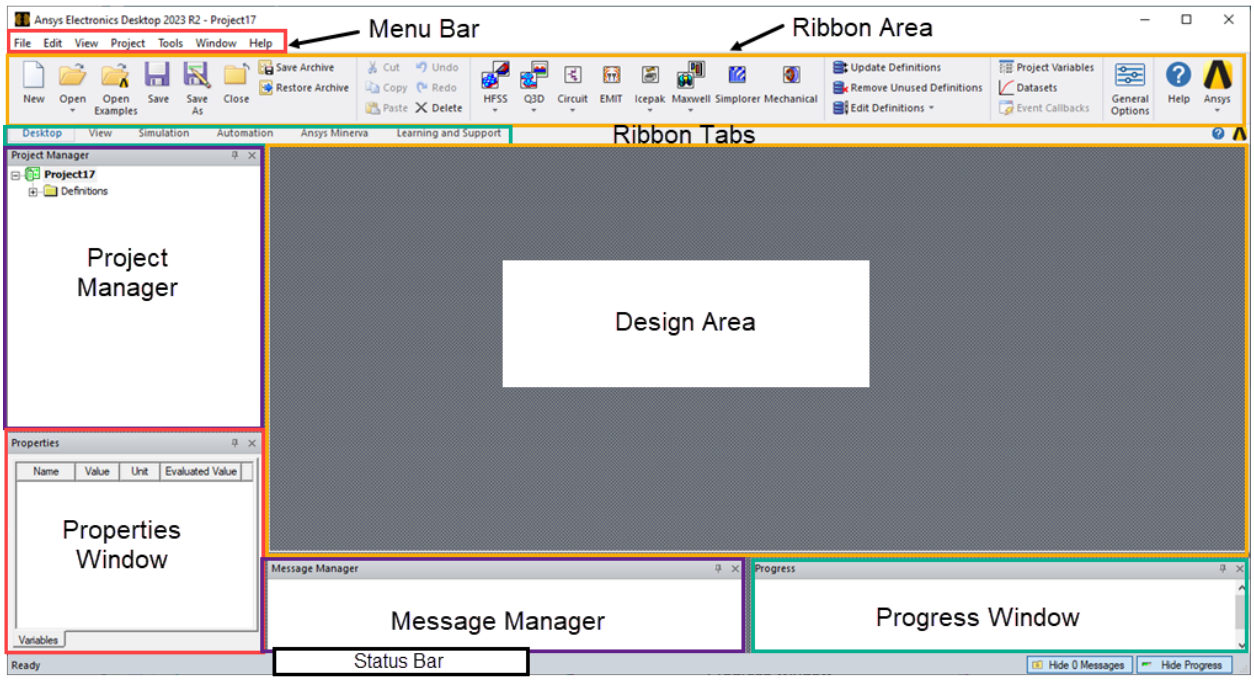
9. *How is enrollment in the Product Improvement Program determined if I use Ansys products in a cluster?*

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

Ansys Electronics Desktop Overview

The Ansys Electronics Desktop consists of several windows, a menu bar, multitab ribbon, and a status bar. You can customize the appearance of the desktop by moving, hiding, or showing

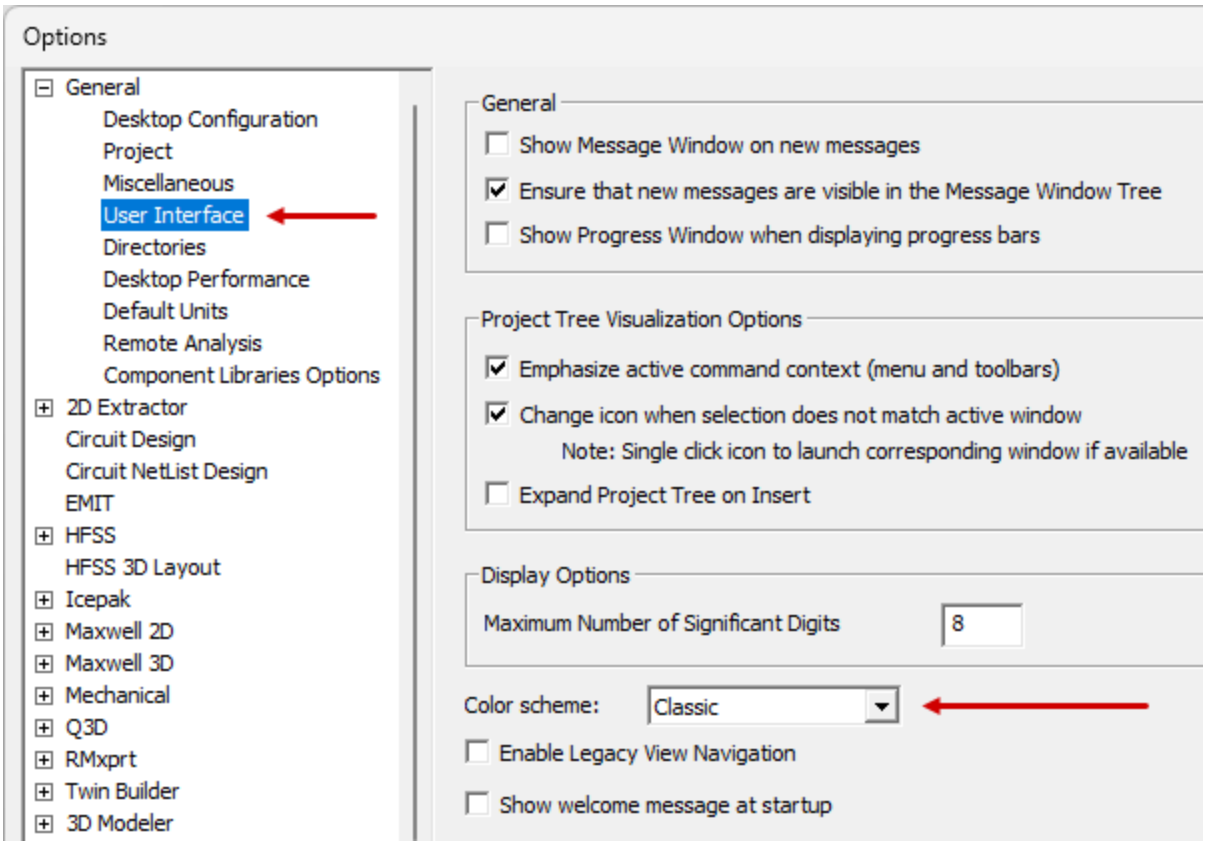
windows. Some menus, and other features, change depending on the type of project that is loaded and the editor that is active in the Design Area.



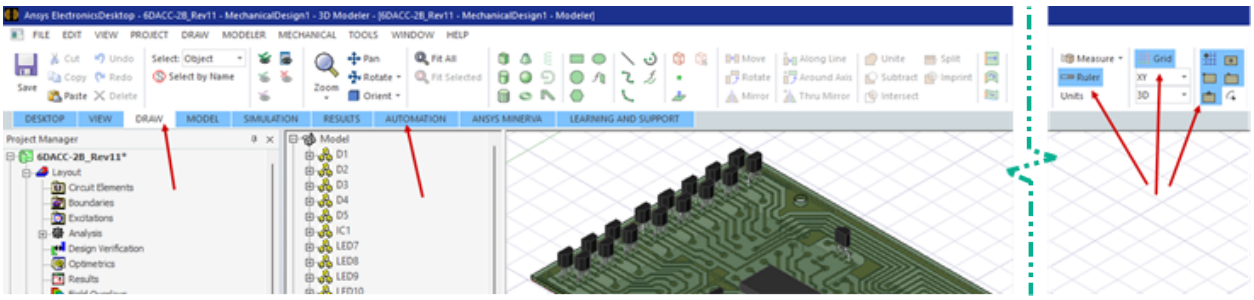
Choosing a Color Scheme [Beta]

In addition to the classic Ansys Electronics Desktop color scheme, two additional color schemes are now available as beta features. Choose the desired color scheme as follows:

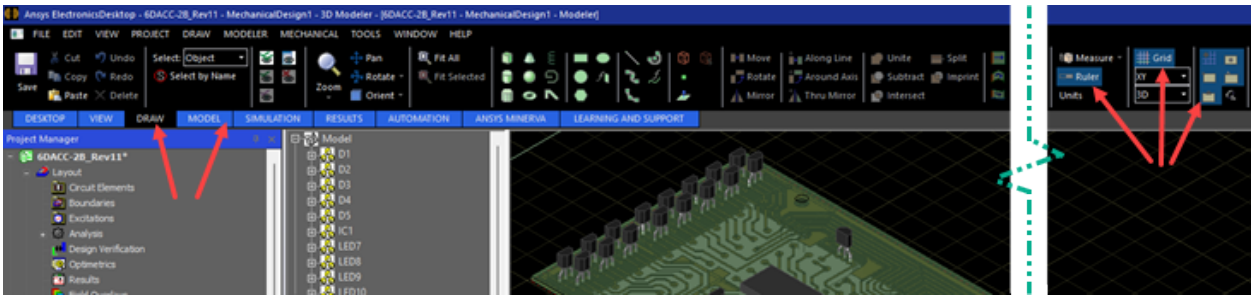
1. Access the *Options* dialog box using one of the following two methods:
   - On the **Desktop** ribbon tab, click **General Options**.
   - From the menu bar, click **Tools** > **Options** > **General Options**.
2. In the tree at the left side of the dialog box, expand **Desktop Configuration** and select **User Interface**:



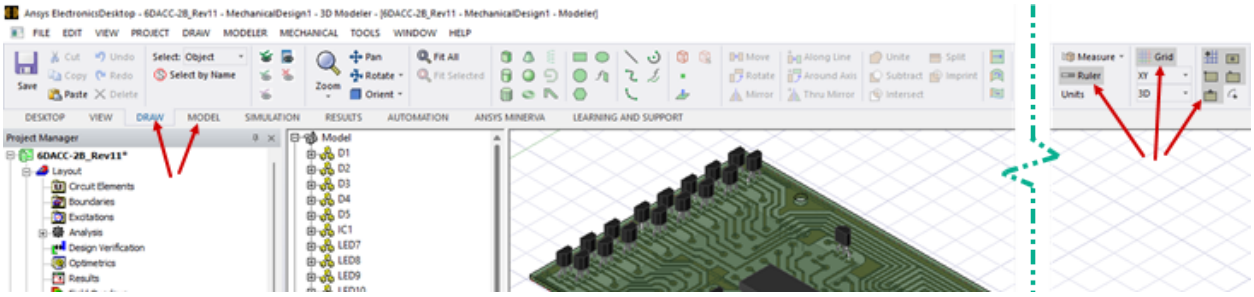
3. From the **Color scheme** drop-down menu, choose one of the following three options:
   - **Light (beta):** A light gray and white scheme with light blue highlighting for selected ribbon icons and for inactive ribbon tabs. The active ribbon has a white background (including the tab):



   - **Dark (beta):** A dark gray and black scheme with medium blue highlighting for selected ribbon icons and for inactive ribbon tabs. The active ribbon has a black background (including the tab):



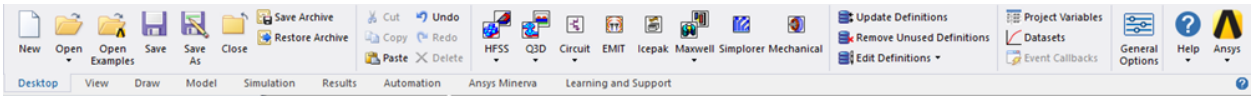
- **Classic:** A light gray and white scheme with medium gray highlighting for selected ribbon icons. The ribbon background is light gray. The name of the active ribbon tab has a blue font. A dark gray font is used for the inactive ribbon tab names:



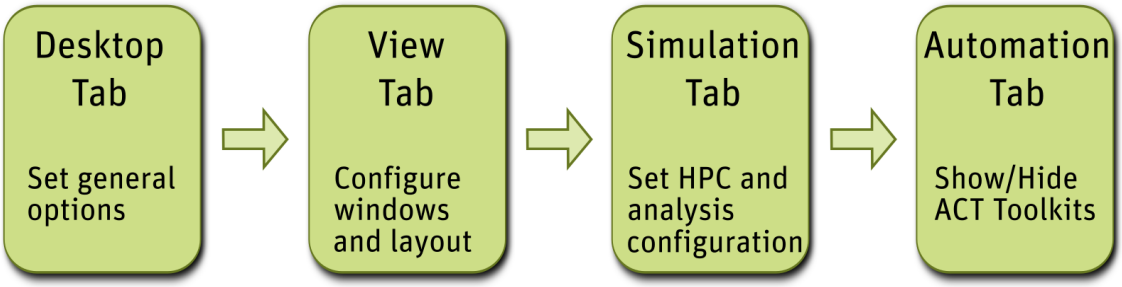
4. Click **OK** to close the *Options* dialog box.

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.



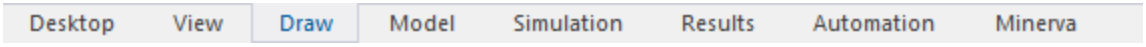
The initial workflow for local configuration and personal customization is:



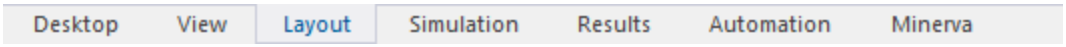
Available ribbon tabs depend on the design type. After you open or add a project and insert a design, you see additional ribbon tabs appear. The visible tabs and features are those that are appropriate for the design type and solver.

The **Desktop, View, Simulation, Automation, Minerva** and **Learning and Support** tabs appear for all design types. Additional tabs appear as follows:

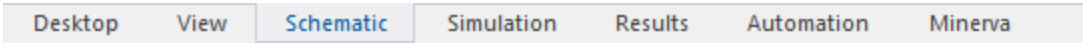
- For HFSS, Icepak, Maxwell, Mechanical, and Q3D designs, the **Draw**, **Model**, and **Results** tabs also appear:



- For an HFSS 3D Layout design, the **Layout** and **Results** tabs also appear:



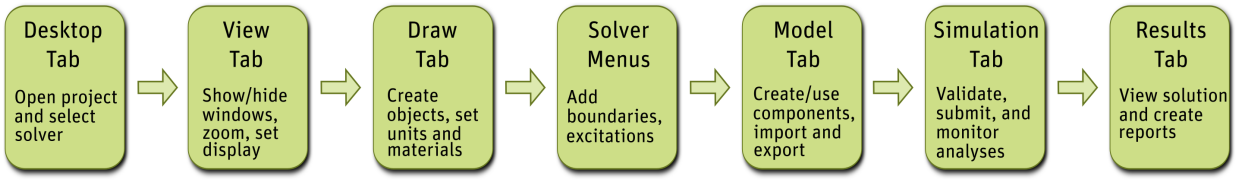
- For Circuit and Simplorer designs, the **Schematic** and **Results** tabs also appear:



In addition to **Desktop**, **View**, **Minerva** and **Learning and Support** the ribbon tabs that are applicable to Q3D Extractor designs are described as follows:

- **Draw Tab** – contains tools that allow you to draw objects, set units, and assign materials.
- **Model Tab** – contains tools to group and ungroup objects, create 3D components, and import or export geometry.
- **Simulation Tab** – contains tools that allow you to run simulations, perform validation checks, and schedule and monitor tasks.
- **Results Tab** – contains tools for creating reports and viewing solution data.
- **Automation Tab** – contains tools that allow you to run and record scripts, show or hide ACT extensions (Windows only), and install PyAEDT (Beta).

Each tab contains features specific to the design type, and the general workflow is from left to right.

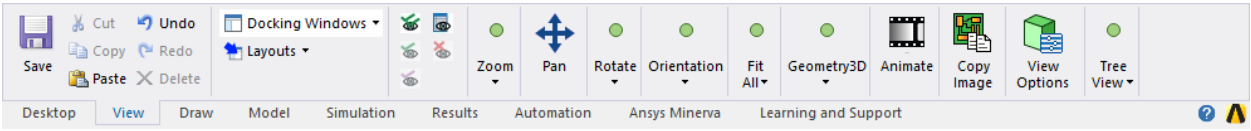


Note that workflow can vary depending on tasks at hand. For example, if you open a completed model in HFSS or use 3D components or the ACT Antenna Design toolkit (Windows only), you may not need the features of the **Draw** tab. If Validation on the **Simulation** tab identifies problems in a model, you may need to use the **Modeler** menu commands for Analysis and Heal, or use the Discovery link feature. For some tasks, such as assigning excitations or boundaries in HFSS, you must use the command menus, rather than the tabs.

Sizing the Ansys Electronics Desktop window affects the icons displayed on each tab, with priority given to the most used features.

For example, If you have inserted an HFSS project, the **View** tab displays commands appropriate for the active editor:

If you reduce the size of the Desktop window, the ribbon tabs become compressed. Fewer icons are shown, and available features are moved into drop-down menus rather than shown separately. The following example shows the **View** tab in a compressed state:



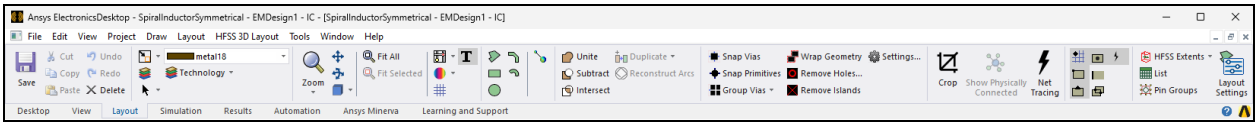
Other ribbon tabs are similarly compressed when the window size does not support the fully expanded arrangement of features.

The **Learning and Support** tab provides easy access to Technical advice, instruction, and examples on Ansys websites.

- **Ansys Innovation Courses** – opens a web page containing a wide range of Ansys Electronics Engineering courses using on-demand, self-paced video training and quizzes.
- **Ansys Learning Hub** – opens a web page with subscription based access to, virtual and self-paced learning across the Ansys Software portfolio.
- **Ansys Knowledge** – opens a web page to expert curated knowledge materials from FAQs to tutorials on simulation topics.
- **Ansys Learning Forum** – opens a web page to Ansys blog containing discussion and presentations from Ansys experts, partners and customers.
- **Customer Portal** – opens a web page to Ansys Product support.
- **About Electronics Desktop** - opens a dialog with version and release information, Ansys Electronics Desktop installed components, and licensing information.

IC Mode Layout Ribbon

In addition to exclusive tabs, HFSS 3D Layout contains two potential environments (i.e., General mode and IC mode). General mode typically opens by default. Different commands and options are available in each mode (e.g., after opening a GDSII file or selecting the **IC** layout from the **Design Settings** window (i.e., **HFSS 3D Layout** > **Design Settings** > **Design Mode** tab), the **Layout** tab will reconfigure to match the following screenshot).



The general workflow remains the same in both environments. Refer to Switching to IC Mode.

Ansys Electronics Desktop Windows

Ansys Electronics Desktop contains the following windows:

- Project Manager
- Message Manager
- Progress
- Properties

- Nets
- Components

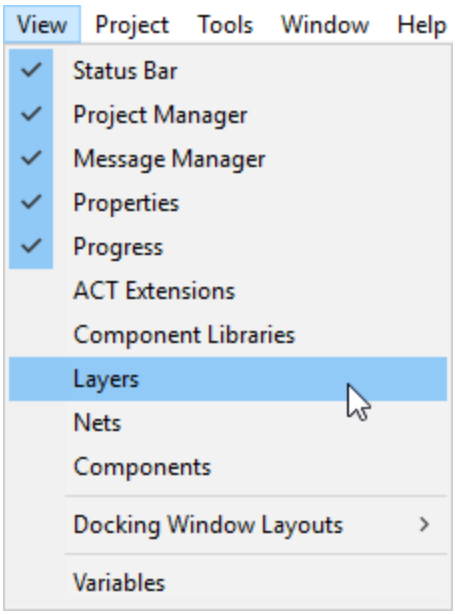
You can toggle the display of each window, auto-hide windows, or move and resize windows. Additionally, you can toggle the status bar or add ACT Extensions (Windows only).

Showing and Hiding Windows

To toggle the windows that comprise Ansys Electronics Desktop, use the **View** menu or the **View** tab on the ribbon.

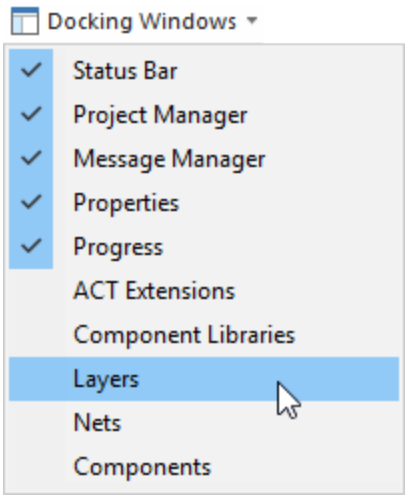
From the **View** menu:

- Click a window to show or hide it.



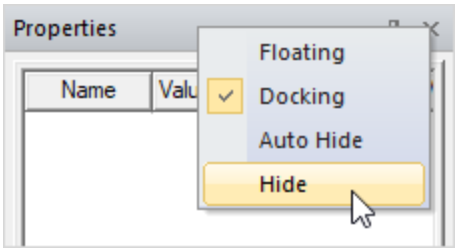
From the **View** tab:

- Use the **Docking Windows** drop-down menu to show and hide windows.



You can also hide a window directly from the window:

- Click the X in the window title bar, or
- Right-click in the window title bar and select **Hide**.



The visibility setting of a window is retained from one desktop session to the next.

Auto Hiding Windows

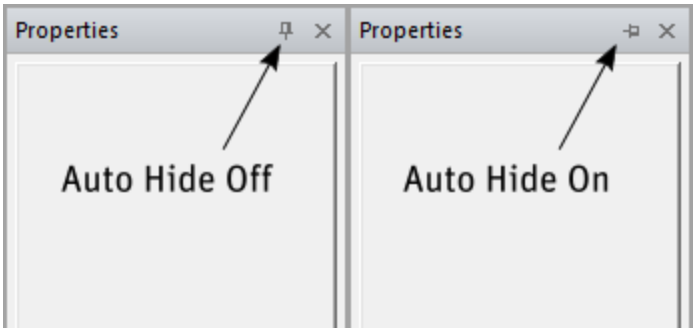
Windows can be moved out of the way so that only their title bars show. When you hover over this title bar, the window appears. This mode is called Auto Hide.

To automatically hide a window until the cursor is hovered over it:

- Click the pin icon in the window title bar, or
- Right-click in the window title bar and select **Auto Hide**.

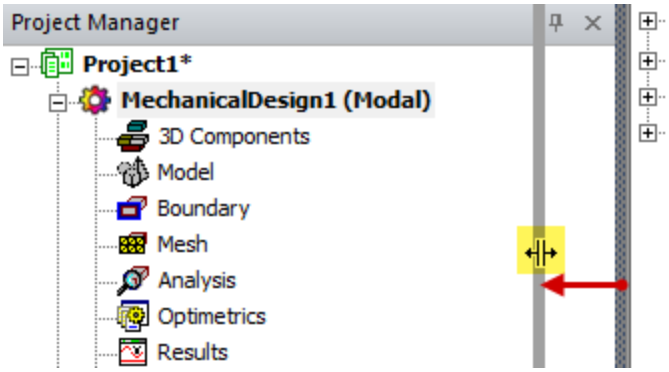
> **Note:** If a window shares vertical or horizontal space with another window, the first click will expand it to take up the entire area while hiding all other windows. Click the pin again to hide the window.

When a window is in Auto Hide mode, the pin icon flips on its side. Click the icon again to disable Auto Hide.

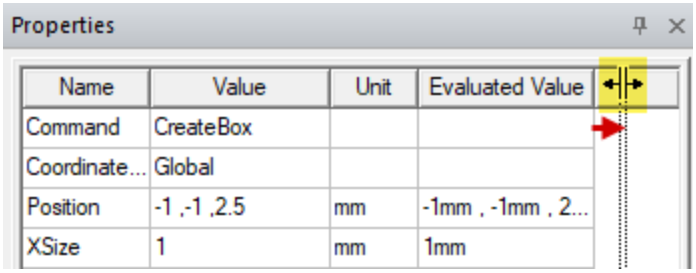


Moving and Resizing Windows

You can resize a window by clicking and dragging a vertical or horizontal window border. The cursor appearance changes to indicate when you're over a movable window border (as highlighted in yellow below):



You can also click and drag the borders of columns within certain windows containing tabular data, such as the **Properties** window:



Major Ansys Electronics Desktop windows can be docked in various locations or can float over other windows. To move a window, click and drag the window's title bar. While dragging a window's location, several docking position icons appear on the screen.

By releasing the mouse button over the following icons, you can dock a window in the following four locations along the perimeter of the user interface:

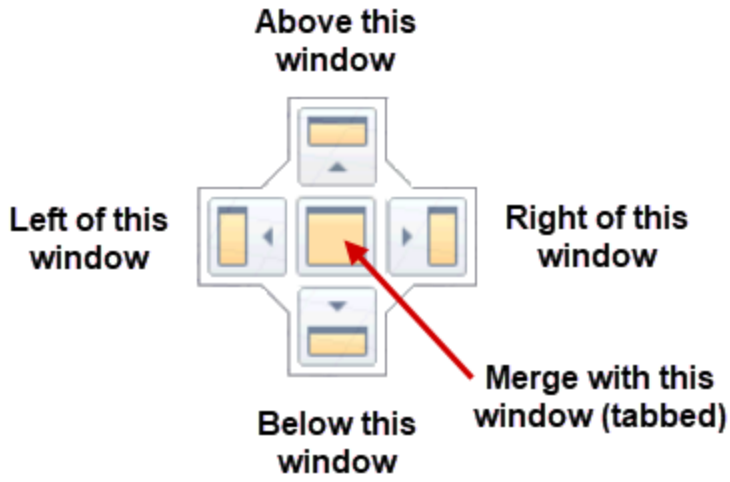
Left edge of user interface: 

Right edge of user interface: 

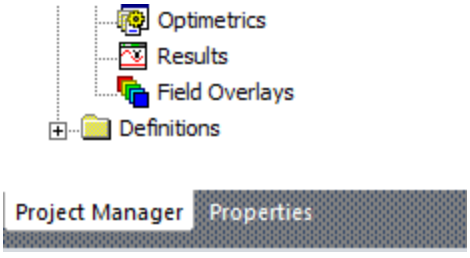
Top of user interface: 

Bottom of user interface: 

Additionally, you can use the following group of five docking locations to position the window relative to another window:



When you release the mouse button over the middle icon, the two windows are merged, with tabs at the bottom to select which one to view:

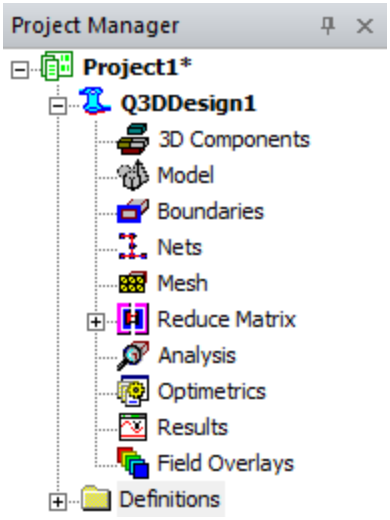


If the middle icon does not appear, it's because tabbed windows are not supported by one of the windows. For example, you cannot merge any window with the Modeler window.

If you release the mouse button while not over any of the above nine icons, the window stays in its current dragged location as a floating window.

Project Manager Window

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 Ansys Electronics Desktop projects. The tree display is specific to the design type. For example:

- For any HFSS, Maxwell, or Q3D design, each project ultimately includes a geometric model, its boundary conditions and material assignments, and field solution and post-processing information. Any 3D Components appear under the 3D Components icon.

To show or hide the **Project Manager** window, do one of the following:

- Click **View** > **Project Manager**.

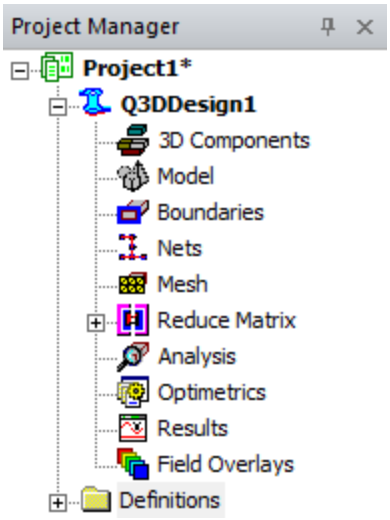
A check box appears next to this command if the **Project Manager** window is visible.

- Right-click in the toolbars area on the desktop, and then click **Project Manager** on the shortcut menu.

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

Working with the Project Tree

The project tree is located in the **Project Manager** window and contains details about all open Ansys Electronics Desktop projects, as shown below:



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

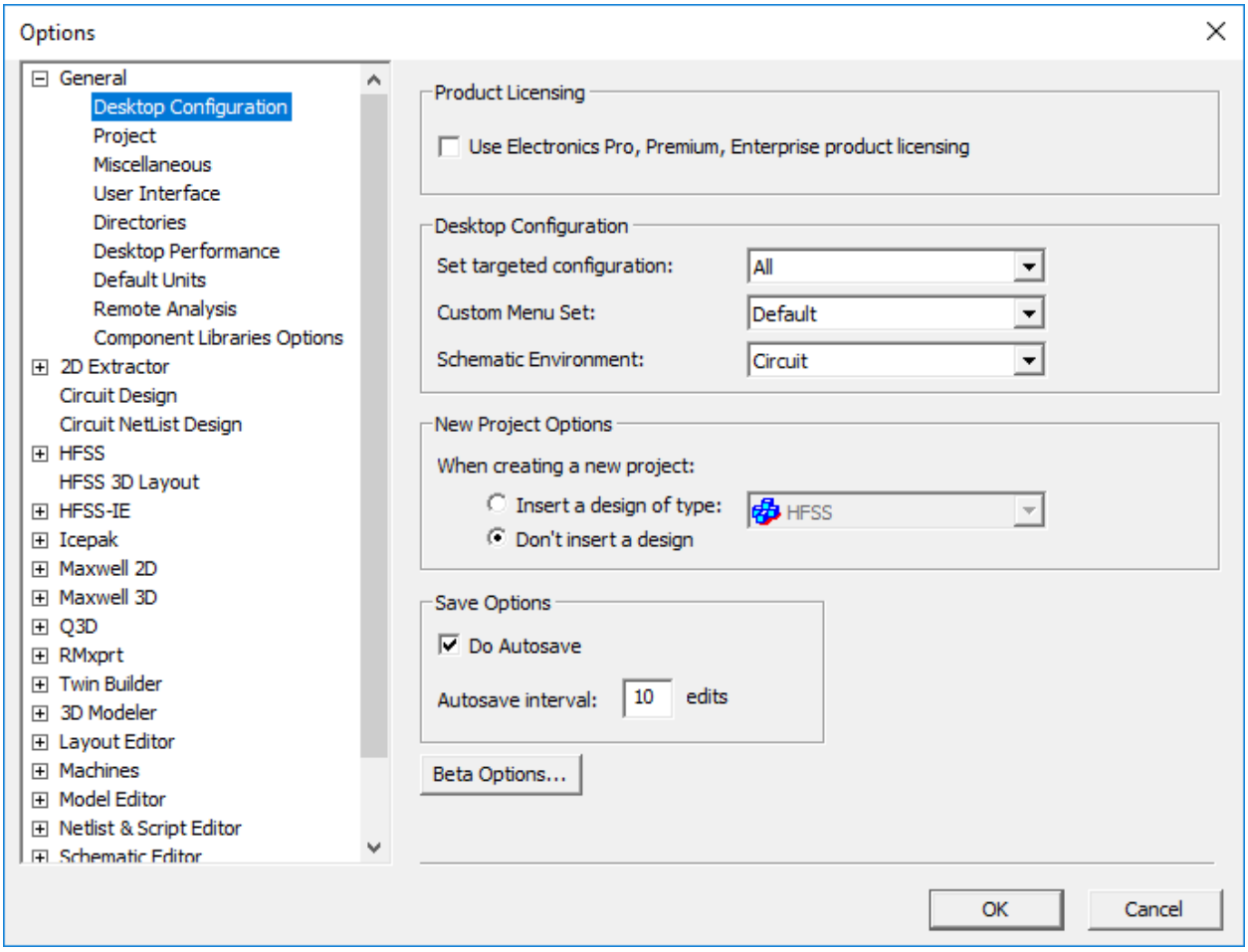
By default, the Project tree icon for the active window is highlighted. See General Options: Miscellaneous for options.

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**.

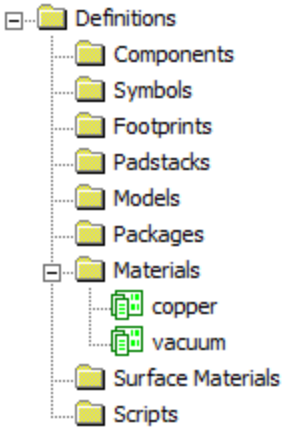
   The **Options** window appears.



2. Click **User Interface**.
3. In the **Project Tree Visualization Options** area, check the **Expand Project Tree on Insert** check box.
4. Click **OK**.

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.



Viewing Ansys Electronics Desktop Design Details

When you insert an Ansys Electronics Desktop design into a project, it is listed as the second node in the project tree. It is named Ansys Electronics DesktopModel*n* by default, where *n* represents the order in which the design was added to the project. Expand the design icon in the project tree to view all of the specific data about the model, including its boundary conditions and material assignments, and field solution and post-processing information.

The Ansys Electronics DesktopModel*n* node contains the following project details:

3D Components	Displays any 3D Components added to the design.
Model	Displays the model geometries in the design.
Boundaries	Displays the boundary conditions assigned to an Ansys Electronics Desktop design, which specify the field behavior at the edges of the problem region and object interfaces.
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 an Ansys Electronics Desktop design. A solution setup specifies how Ansys Electronics Desktop will compute the solution.
Optimetrics	Displays any Optimetrics setups added to an Ansys Electronics Desktop design.
Results	Displays any post-processing reports that have been generated.

Field Overlays	Displays fields 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.
Documentation	Displays files you have added as documentation.
Definitions	Displays definitions for Q3D Extractor, including Components, Materials, and more (as applicable).

Note:

To edit a project's design details:

- In the project tree, double-click the design setup icon that you want to edit.

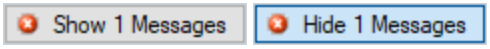
A window appears with that setup's parameters, which you can then edit.

Message Manager Window

The **Message Manager** window displays informational messages about various processes in Ansys Electronics Desktop, including messages related to simulations and any errors they may have produced.

Display or hide the **Message Manager** in one of two ways:

- Click **View** > **Message Manager**
  A check box appears next to this command if the **Message Manager** is visible.
- On the status bar, click **Show Messages** or **Hide Messages**.



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).

Right-click in the **Message Manager** window to open a pop-up menu with the following options:

- **Clear messages** for the current model.
- **Copy messages** to the clipboard. This can be helpful for sending messages to application engineers.
- **Details** – opens an information dialog that contains the project and design data for the specified message.
- **Go to Reference** – allows you to right-click on an intersection error message after running a validation check. This selects intersecting objects in the current design being validated.

Setting the Message Manager to Open Automatically

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

Showing New Messages

You can set the **Message Manager** to automatically open when a new message appears.

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

   The **Options** window appears.

2. Expand **General** and click **User Interface**.
3. In the **General** area, check the **Show Message Window on new messages** check box.
4. Click **OK**.

Automatically Expanding the Message Manager Tree

You can set the **Message Manager** Tree to automatically expand when a new message is added.

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

   The **Options** window appears.

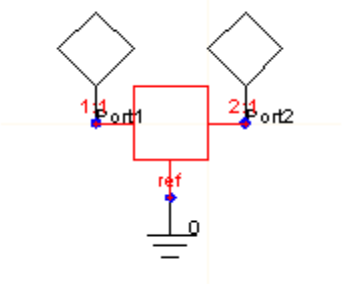
2. Expand **General** and click **User Interface**.
3. In the **General** area, check the **Ensure that new messages are visible in the Message Window Tree** check box.
4. Click **OK**.

Action Messages

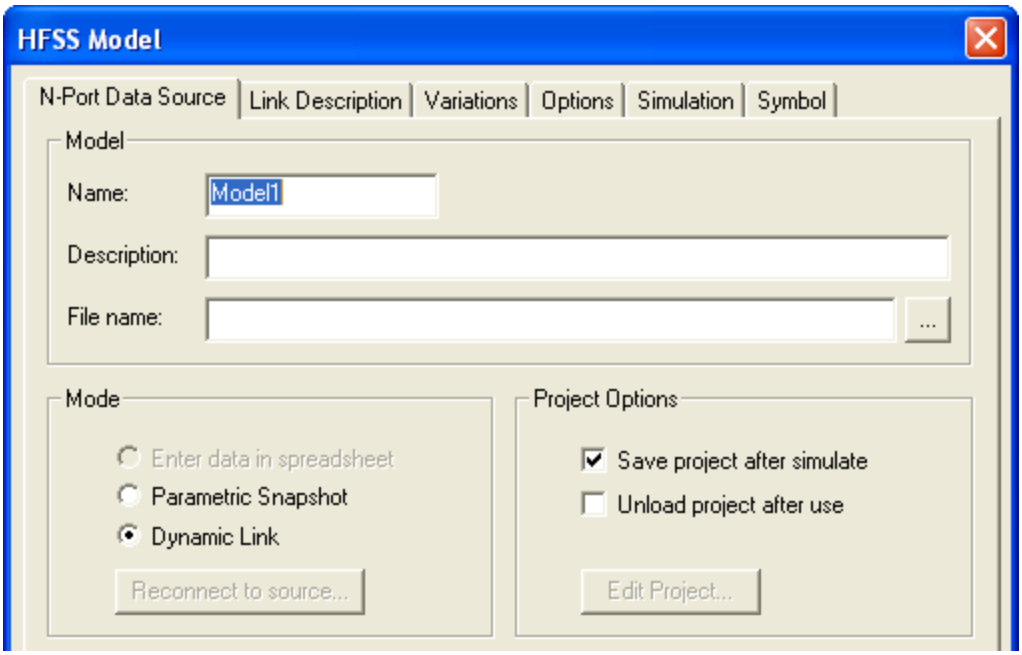
Messages displayed in the **Message Manager** can be associated with actions that can be invoked in order to address a condition you are alerted to by the message. If a message has an associated action, a magnifying-glass icon will be present to the left of the message icon. When the cursor moves over a message with an action, the cursor changes to a hand.

You can invoke a message action by clicking the magnifying glass icon, by double-clicking the message, or by right-clicking the message and selecting **Go To Reference** from the pop-up menu.

One example of a message-associated action is the selection of an associated object:



Another example is the opening of a related window:



Clearing Messages

The **Message Manager** clears at the start of each analysis. To manually clear messages for a project, right-click the message tree and select **Clear Messages for *<ProjectName>***.

Hiding the Message Manager Window until Messages Appear

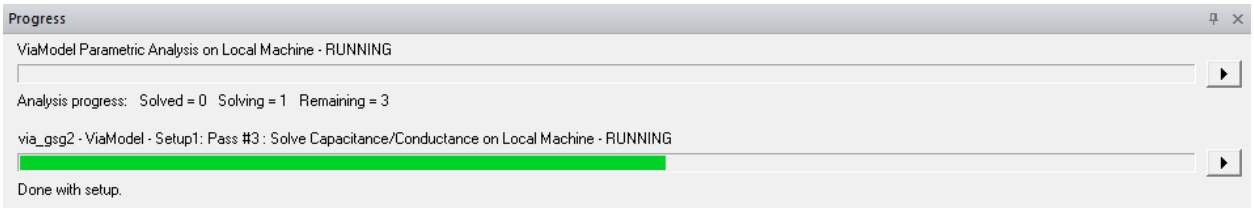
If you prefer to hide the Message Manager until a message is added:

1. Select **View** > **Progress Window** to hide the window. For more information see the View drop-down menu.
2. Click **Tools** > **Options** > **General Options**.
3. Check the **Show message window on new messages** check box.

The **Message Manager** window will re-open when Ansys Electronics Desktop reports errors, warnings, or successful completion of any simulation.

Progress Window

The **Progress** window monitors a simulation while it is running. Each simulation has its own progress bar. Right-clicking the bar allows you to abort the simulation or view simulation details.



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

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

- Click **View** > **Progress Window**.
- Right-click the history tree, and then click **Progress** on the shortcut menu.

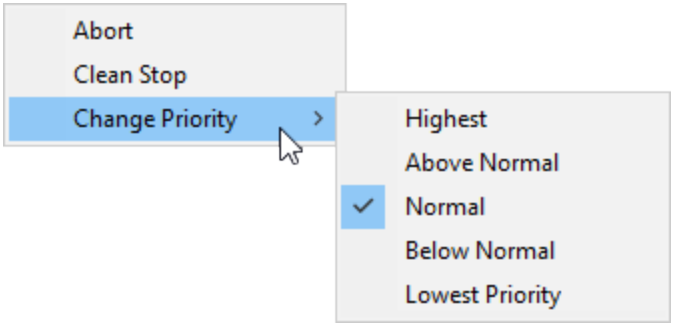
A check box appears next to this command if the **Progress** window is visible.

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

The progress window is also dockable, so you can reposition it.

Stopping or Aborting Simulation Progress

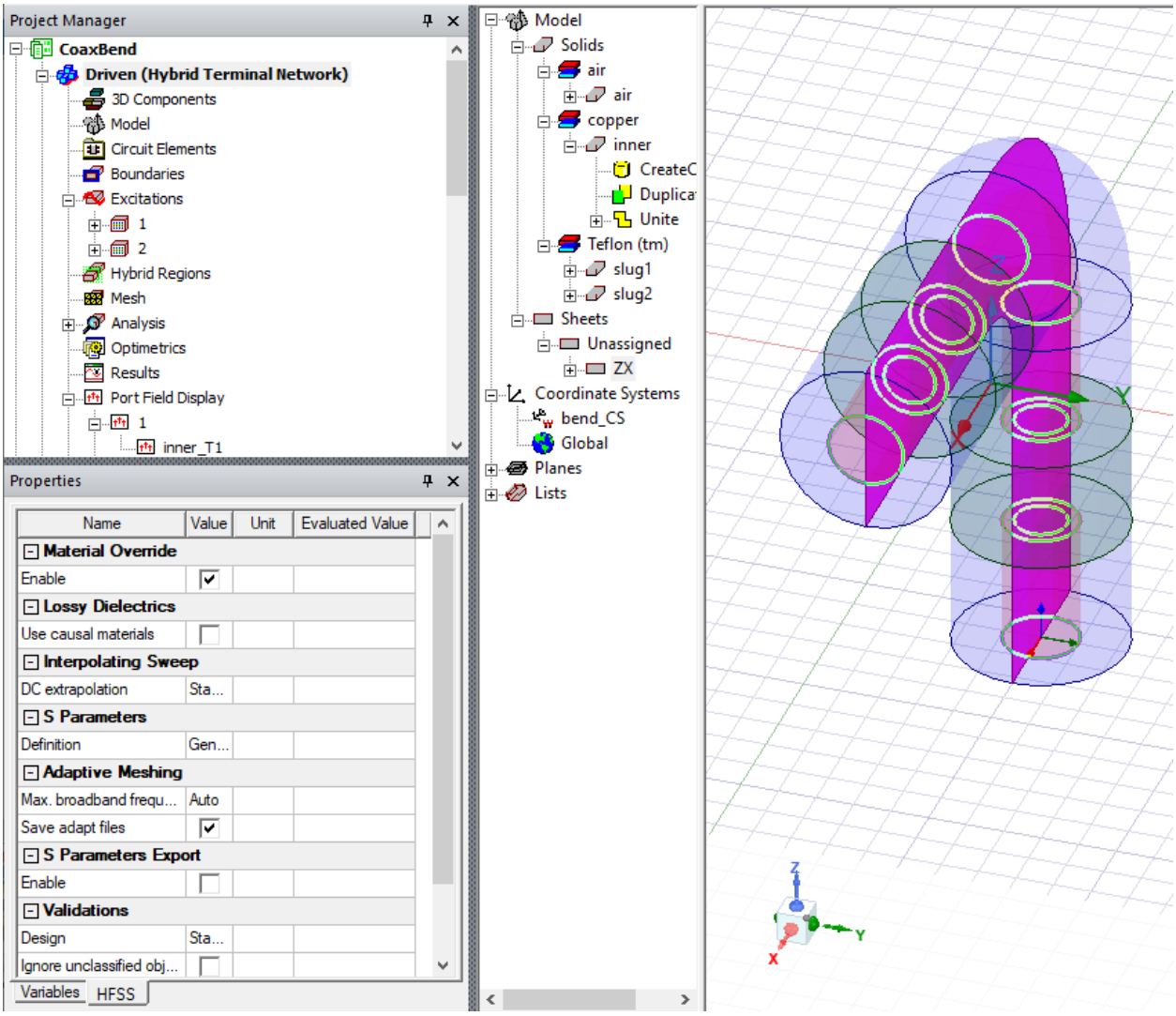
- To abort progress, right-click in the **Progress** window and select **Abort**.
- To stop the simulation cleanly between time steps, right-click in the **Progress** window and select **Clean Stop**.
- To change priority, right-click. From the **Change Priority** submenu, select from Highest, Above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



Properties Window

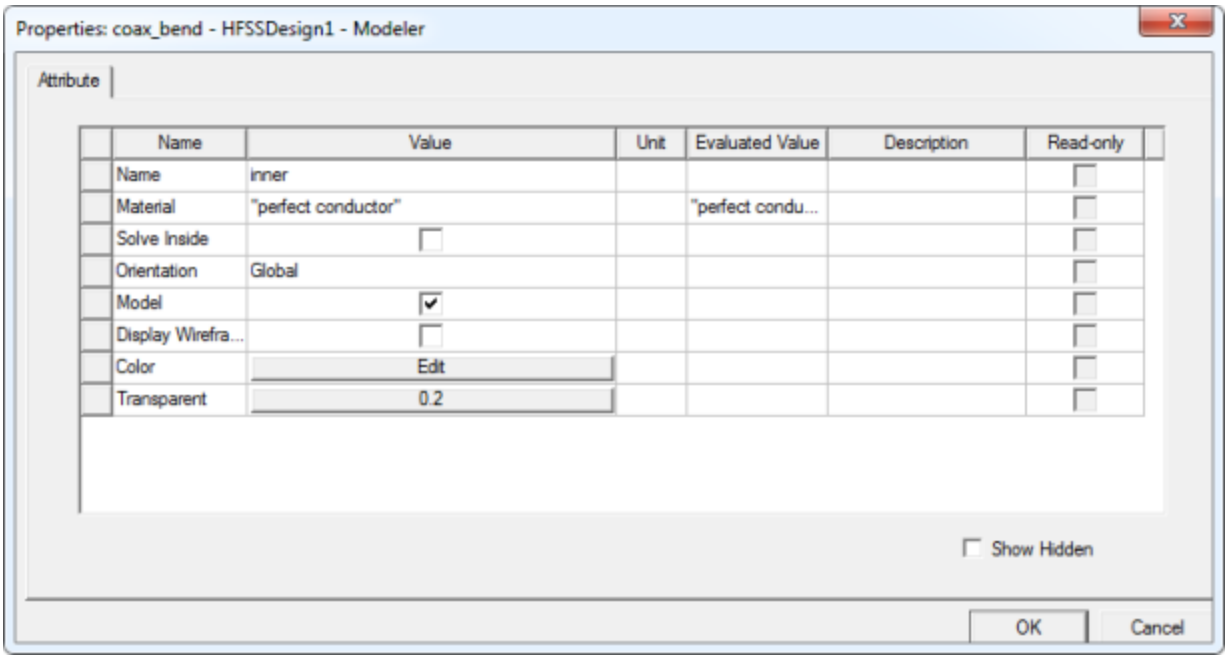
The **Properties** window displays and allows you to edit the properties of an item selected in the project tree, the history tree, or the **3D Modeler** window. The properties shown and their editability vary depending on the type of item selected.

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 the **Properties** window is docked, it displays the properties of any item you select in the Project tree, the History Tree, or the 3D Modeler window. Select **View** > **Properties** to remove the docked properties window.

Regardless of whether or not you display a docked **Properties** window on the desktop, you can open an undocked **Properties** window for any item in the project tree or history tree by double-clicking the item.



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 finished making changes, click **OK**.

Rather than opening a separate window, you can have the **Properties** window displayed within the desktop.

Setting the Properties Window to Open Automatically

To set the **Properties** window to open after an object is drawn, do the following:

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

   The **Options** window appears.
2. Expand **3D Modeler** and select **Drawing**.
3. Check the **Edit property of new primitives** check box.

Hereafter, when you draw an object in point mode the **Properties** window opens. However, if you draw an object in Dialog mode, this setting is ignored.

Modifying Object Attributes Using the Properties Window

1. Select the object for which you want to edit 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**. You can modify names, but names must include only letters, numbers and underscores. Illegal names are not accepted and generate a message in the Message Manager window.
- Click the button and edit the current settings in the window that appears.
- Click the attribute and select a new setting from the menu that appears.

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. In general, command properties you can typically modify are numeric values, such as position values, size values, and various other coordinate values. You can also modify many of the unit settings for a command property. You can modify names, but names must include only letters, numbers and underscores. Illegal names are not accepted and generate a message in the Message Manager window.

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

> **Tip:**
>
> Press and hold **Ctrl** to select multiple commands. If you select multiple commands, only the common (shared) properties will display 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 edit the current settings in the window that appears.
- Click the attribute and select a new setting from the menu that appears.

Properties Dialog Box

Like the **Property** window, the **Properties** dialog box shows the properties or parameters of selected objects, and, where appropriate, allows editing the values of these properties. It extends **Property** window functions with additional editing commands, and with settings for tuning, optimization, sensitivity, and statistical analysis, that are not available through the **Property** window. It is also used to define project variables. The **Properties** dialog box has the following tabs:

- General
- Symbol
- Property Displays

Opening the Properties Dialog Box

You can open the **Properties** dialog box in two ways:

- To open the **Properties** dialog for a component, double-click the component.
- Right-click the component and select **Properties**.

You can manipulate the **Properties** dialog box in the following ways:

- Close the **Properties** dialog box by clicking the **Esc** key.
- Resize the **Properties** dialog box by dragging its edges.
- Change the relative widths of adjacent columns by dragging the header separators between them.

General and Symbol Tabs

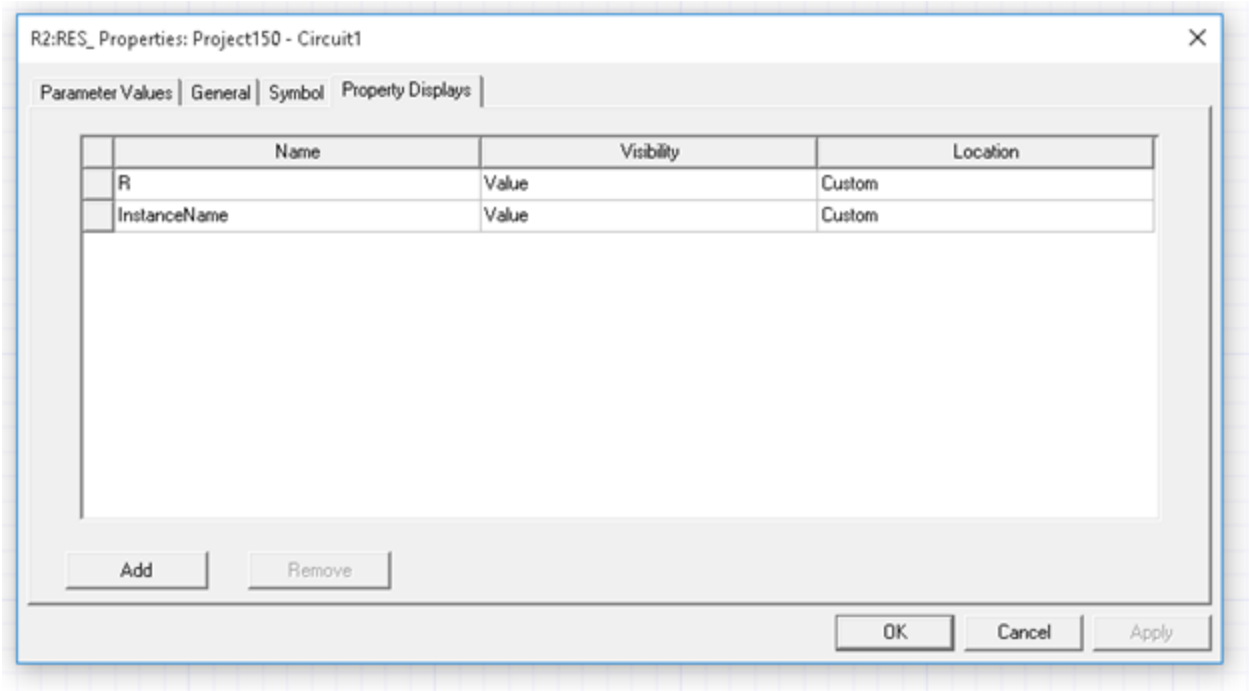
The **General** tab in the Properties Dialog Box lists the selection's name, symbol name, reference designator, and so on. These are generally not editable. The information is identical to that on the General tab of the Property Window.

The **Symbol** tab provides information on the location of the component symbol in the schematic. The information is identical to that on the Symbol tab of the Properties Window.

When the **Properties** dialog box is opened for a design (**Design Properties**), both the General and Symbol tabs are initially empty.

Property Displays Tab

The **Property Displays** tab in the Properties Dialog Box controls how the properties of the component appear on the schematic.



Click in the **Visibility** field for a parameter to select how the information for that parameter is displayed:

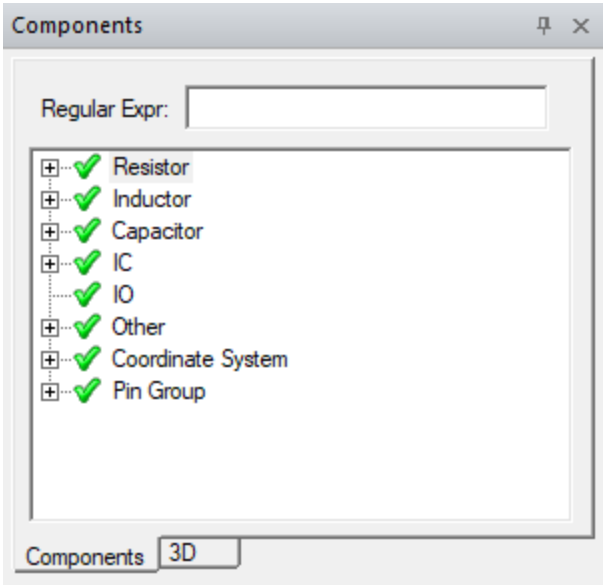
- **None** – results in no label being shown.
- **Name** – displays the parameter name (e.g., R).
- **Value** – displays the component value, which can be a single value (e.g., 10000) or an expression (e.g., 10000*5).
- **Both** – displays the parameter name and its value (e.g., R = 10000, R=10000*5).
- **Evaluated Value** – displays the evaluated value of an expression that has been used for the value of a parameter (e.g., 50000 for the expression 10000*5).
- **Evaluated Both** – displays the parameter name and the Evaluated Value (e.g., R = 50000 for the expression 10000*5).

Click in the **Location** field for a parameter to specify the location for the display. The locations are: **Left**, **Top**, **Right**, **Bottom**, **Center**. When you have set the location with the cursor in the schematic, the **Location** field has the entry **Custom**.

- To add a parameter to the display, click the **Add** button. Click on the **Name** field and select the parameter to add from the drop-down menu.
- To delete a parameter from the display, select it in the **Property Display** list and click **Remove**.

Working with the Components Window

The **Components** window is a dockable window that is used to view and configure various component settings for Layout components. Use the **Components** window to select a component or a component class, add ports to a component, and configure multiple ports across components. You can also select multiple components in the Layout editor to configure pins and ports for each simultaneously. The window can be resized and relocated.



To show or hide the **Components** window on the desktop, click **View** > **Components**. A check box appears next to the command if the **Components** window is visible.

With a component selected, the following configuration controls are available from the right-click menu:

- **Type** – reclassifies the component type to **Resistor**, **Inductor**, **Capacitor**, **IC**, **IO**, or **Other**.
- **Enable**/**Disable** - enables or disables the component.

- **Model** – opens the **Component Model** window, where you can change the component model definition, including die, solder ball, and port properties. Changing the model scope will update every reference designator that belongs to the model definition.
- **Create Ports on Component** – adds ports for the component
- **Remove Ports on Component** – removes ports for the component.
- **Fit Selected** – fits the selected component in the active view window.

> **Note:** When a component classification is selected, any changes are applied to all components within the class.

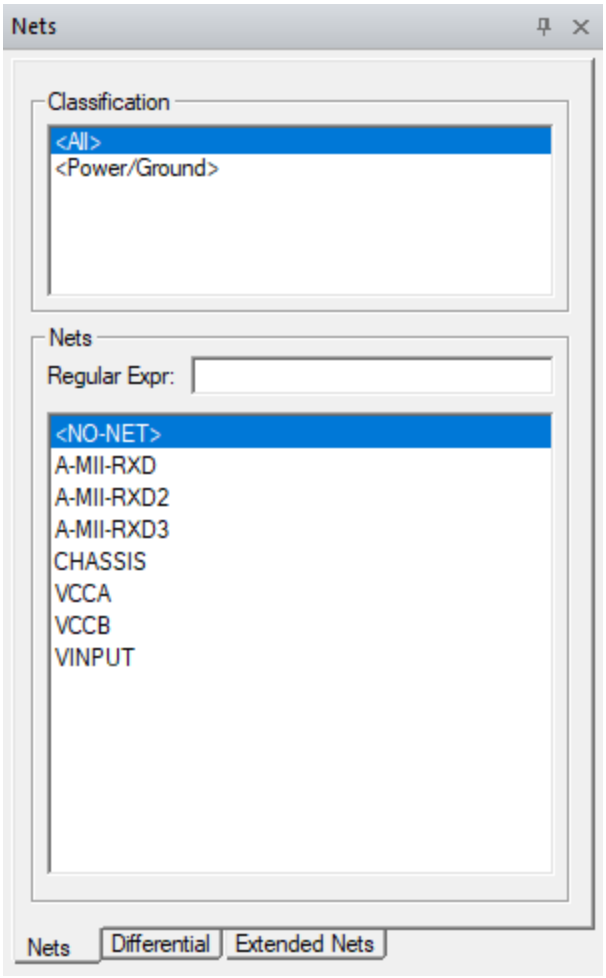
For more information, see Layout-based Component Encapsulation.

Working with the Nets Window

The **Nets** window is used to view and configure nets and collections of nets. It is a dockable window that can be resized and relocated. The **Nets** window is divided into three tabs:

- **Nets** – lists net and net classes.
- **Differential** – lists coupled positive and negative nets.
- **Extended Nets** – lists extended nets for use for SIwave SYZ with HFSS Regions simulations.

To show or hide the **Nets** window, click **View** > **Nets**. A check box appears next to this command if the **Nets** window is visible.



Nets Tab

The **Nets** tab contains two panes: an upper **Classification** pane and a lower **Nets** pane.

Classification Pane

A net classification (also called a class) is a collection of nets. Classifications are designed to group and organize nets by a common characteristic. A net can be in multiple classes at once. By default, there are two classifications: All and Power/Ground net class.

The **Classification** pane lists and manages net classes. When you select a class in the **Classification** pane, the nets assigned to that class are listed in the **Nets** pane.

The **Classification** pane's shortcut menu offers the following actions:

- **New** – opens the **Add Net Class** window for net class creation (see below).
- **Edit** – opens the **Net Class Properties** window for net class modification (see below).
- **Delete** – deletes the selected net class(es).
- **Select** – in the layout editor, selects the net classes selected in the classification pane.
- **Show** – in the layout editor, shows the net classes selected in the classification pane.
- **Show (Hide All Other)** – in the layout editor, shows the net classes selected in the classification pane and hides all other net classes.
- **Hide** – in the layout editor, hides the net classes selected in the classification pane.
- **Create Ports** – creates ports on all nets in the selected net classes.
- **Remove Ports** – removes ports from all nets in the selected net classes.

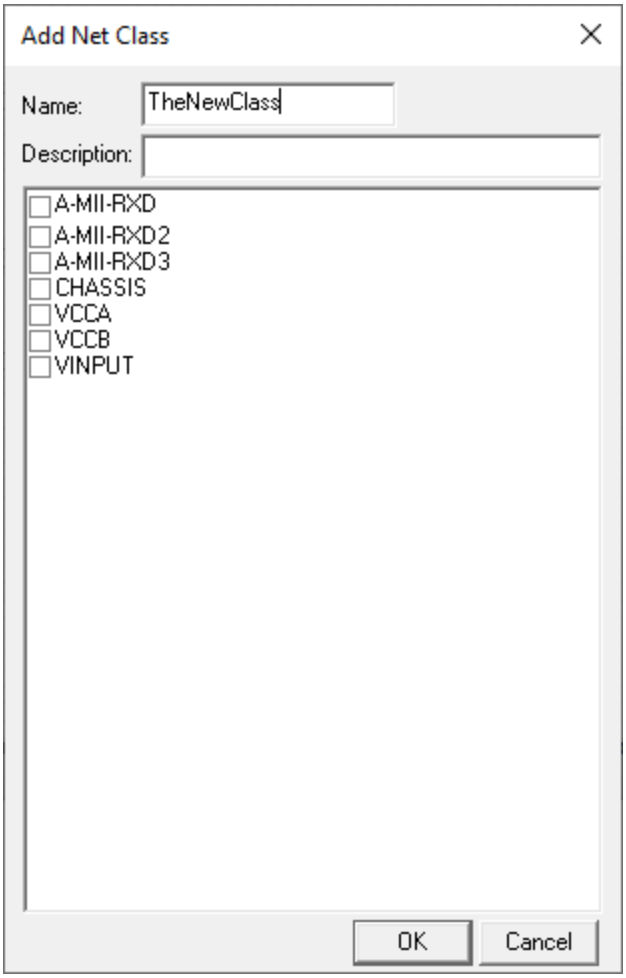
Nets Pane

The displayed nets may be filtered using Perl Regular Expression syntax. In the **Nets** pane, the right-click menu contains additional options:

- **Add to Power/Ground** – adds one or more selected nets to Power/Ground.
- **Delete** – deletes the selected net(s).
- **Lists** – opens a list of all nets in the Design List window.
- **Remove from Power/Ground** – removes one or more selected nets from Power/Ground.
- **Create Differential Pair** – with two nets selected, groups them as a differential pair (see below).

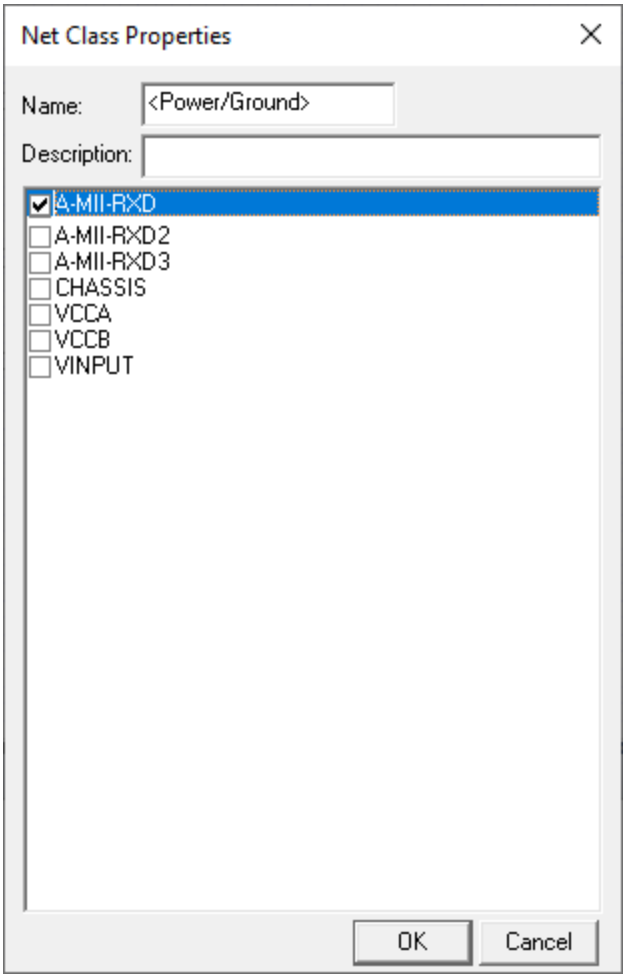
Add Net Class

Selecting **New** from the right-click menu opens the **Add Net Class** window. Enter a name and description (optional) for the new net class and click **OK**.



Net Class Properties

Selecting **Edit** from the right-click menu opens the **Net Class Properties** window. Use this window for net class modification.

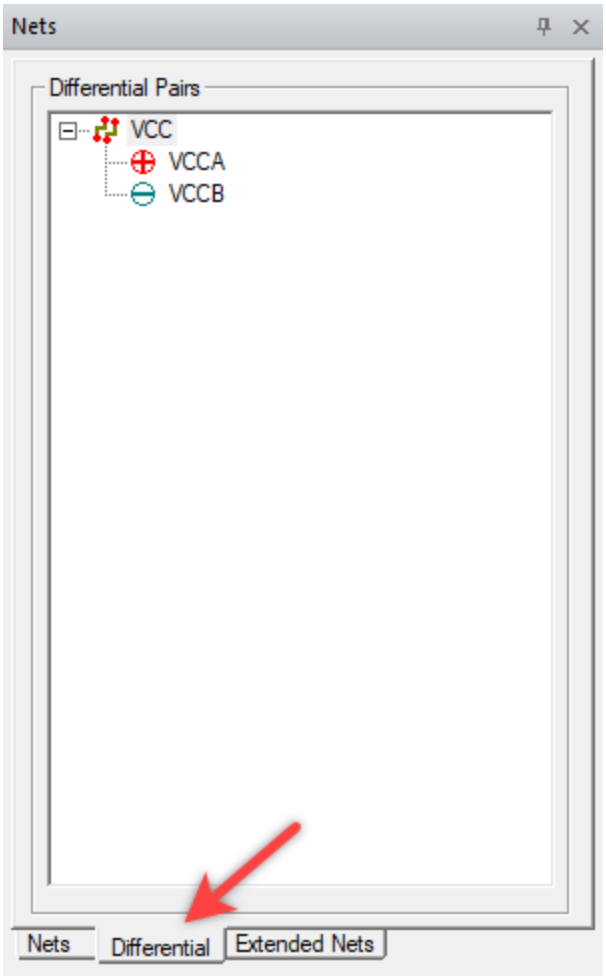


Differential Pairs

Each Differential Pair is composed of a positive net and a negative net. From the **Nets** window, you can view and create Differential Pairs.

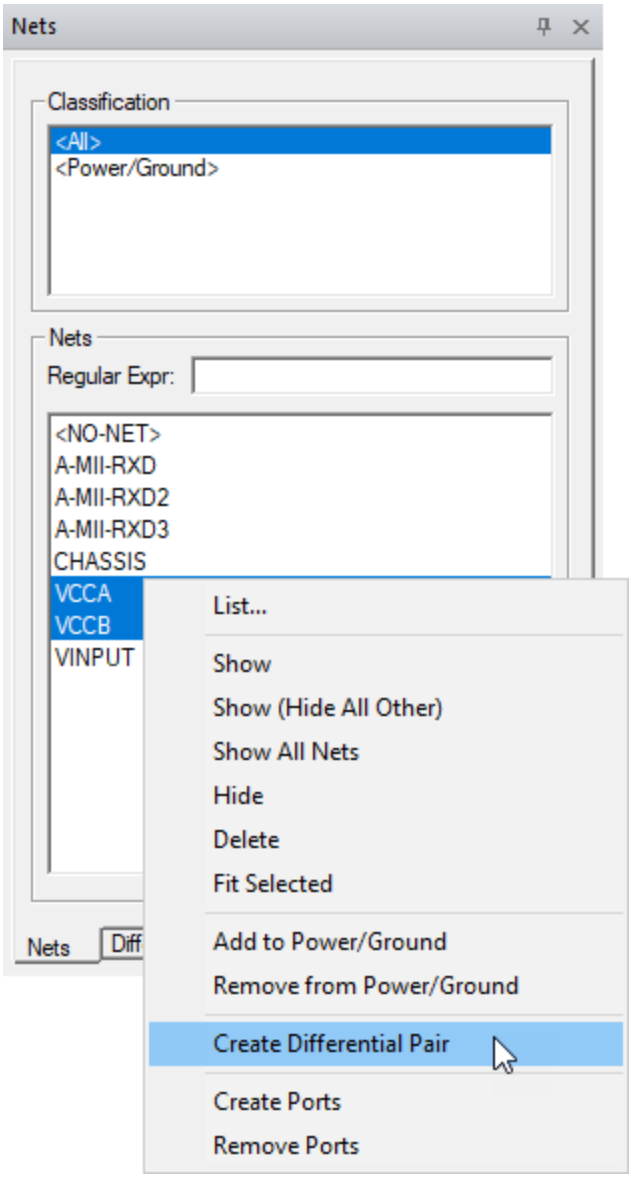
Viewing Differential Pairs

Click the **Differential** tab to display a list of Differential Pairs.



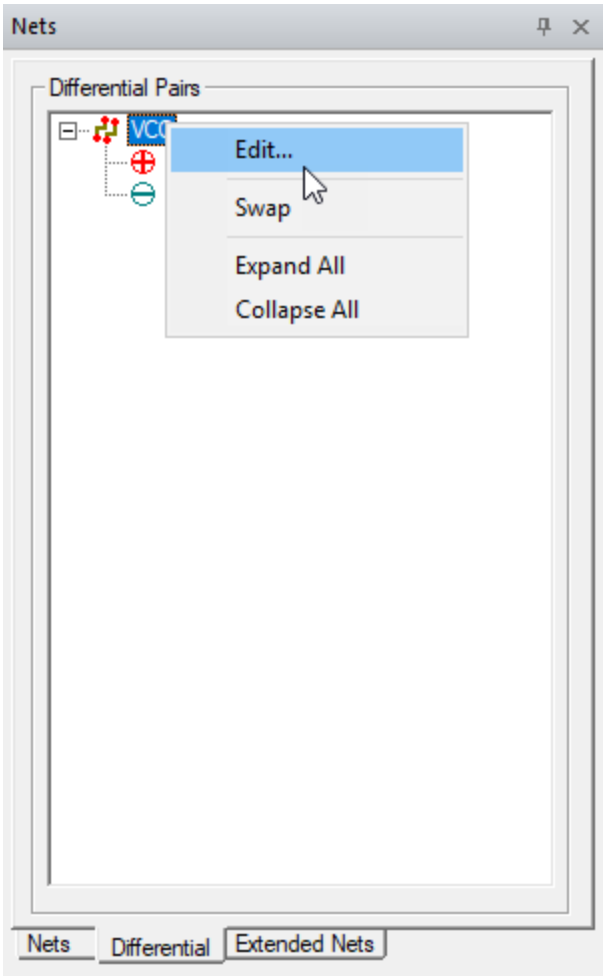
Creating Differential Pairs

Select two nets in the **Nets** list and right-click to create a new Differential Pair.



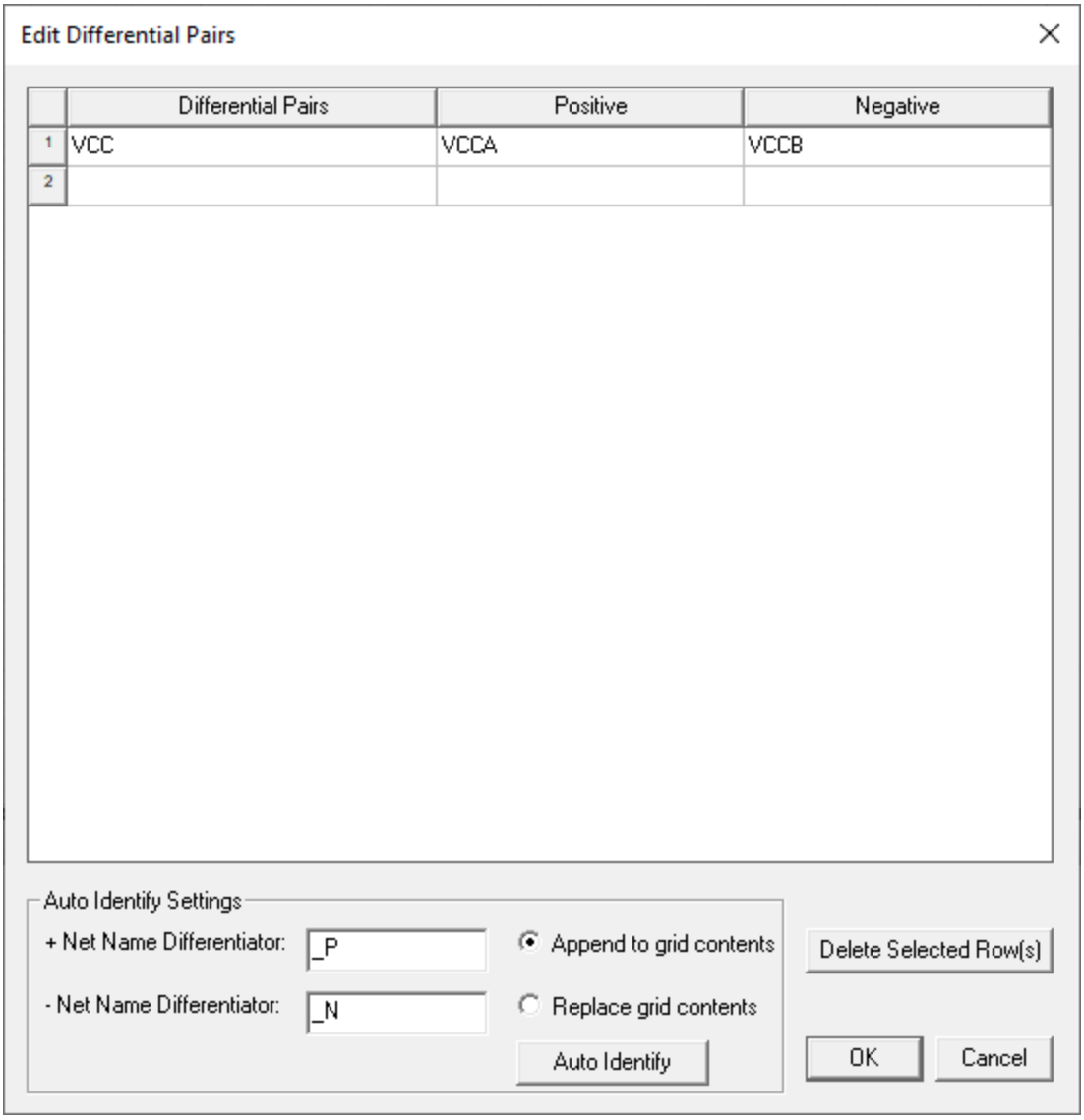
From the **Differential Pairs** list, you can select a pair, open the right-click menu and select:

- **Swap** – swaps the positive and negative net.
- **Edit** – opens the **Edit Differential Pairs** window.
- **Expand All** – expands all Differential Pairs.
- **Collapse All** – collapses all Differential Pairs.



Edit Differential Pairs Window

From the **Edit Differential Pairs** window, you can delete pairs or auto detect Differential Pair with postfixes of the positive and negative nets that are provided. Enter differentiators to identify the positive and negative parts of pairs. Use **Append to grid contents** to add any new pairs to the list in the grid and **Replace grid contents** to replace existing pair with newly identified pairs.



Extended Nets Tab

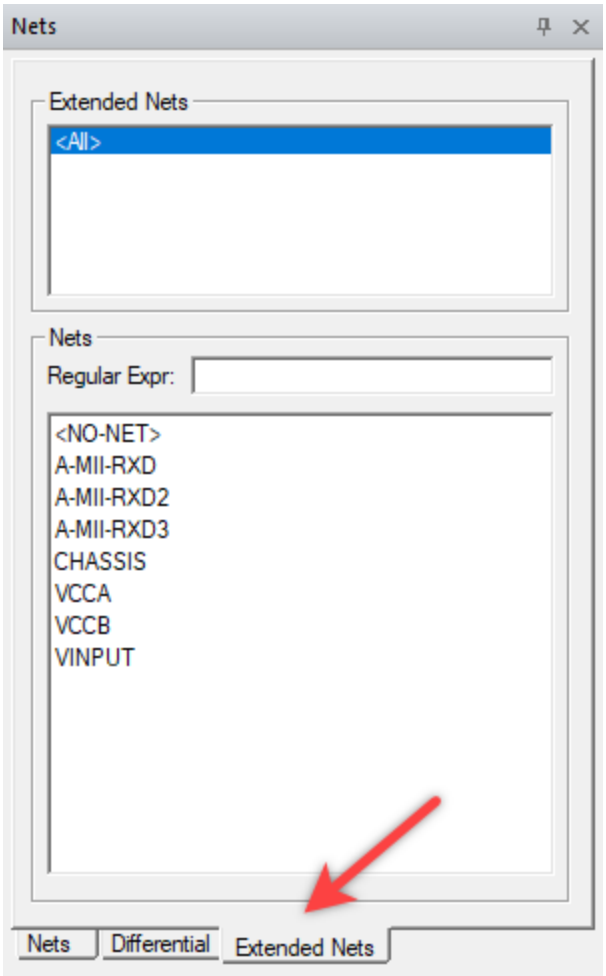
Extended nets are used to indicate a logical equivalence among multiple nets. This is useful when two nets can be considered electrically equivalent, but are not physically connected. For example, routing for a data signal may have a short trace between a series termination resistor and pin. While the net assignment for traces on either side of the resistor are different, logically they are equal and can therefore be grouped into an extended net.

Extended nets are collections of nets that are similar to net classifications but with subtle differences. A net can be in only one extended net at a time, in contrast to being in multiple

classifications simultaneously. There are also no default extended nets and no ports are associated with extended nets.

Extended nets are used for SIwave SYZ with HFSS Regions simulations. If an HFSS region contains capacitors, inductors, or resistors that connect two different signal nets, these signal nets must be defined as extended nets. SIwave needs this definition to know how to pull in required nets and auto generate ports in the region.

The **Extended Nets** tab contains two panes: an upper **Extended Nets** pane and a lower **Nets** pane.



Extended Nets Pane

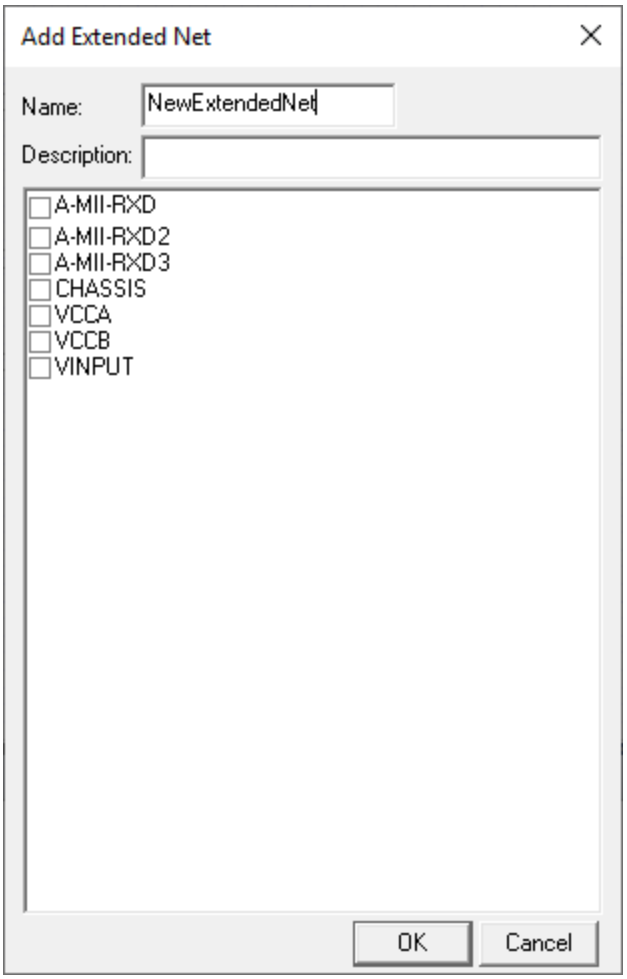
The **Extended Nets** pane lists all extended nets. Use it manage the extended nets. When an extended net is selected in the **Extended Nets** pane, the nets assigned to that extended net appear in the **Nets** pane. The entry <All> lists all nets in the **Nets** pane including entities not in a

net. <All> is not an extended net. The **Extended Nets** pane's shortcut menu offers the following actions:

- **New** – creates a new extended net.
- **Auto Identify** – automatically identifies new extended nets in the **Auto Identify Settings** window.
- **Edit** – changes the selected extended net.
- **Delete** – deletes the selected extended nets.
- **Select** – selects the members of the extended nets in the Layout editor.
- **Show** – shows the members of the selected extended nets in the Layout editor if hidden.
- **Show (Hide All Other)** – shows the members of the selected extended nets in the Layout editor while hiding all other nets.
- **Hide** – hides the members of the selected extended nets in the Layout editor.

Create an Extended Net

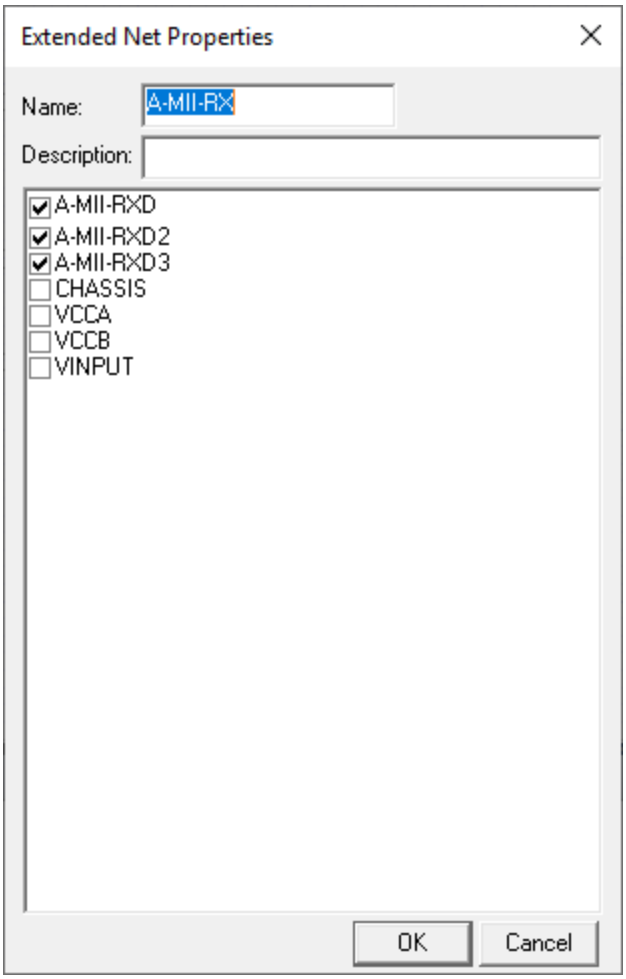
1. In the **Extended Nets** pane, right click and click **New** to open the **Add Extended Net** window.



2. Enter a name and optional description.
3. Select the nets that will make up the extended net.
4. Click **OK**.

Editing an Extended Net

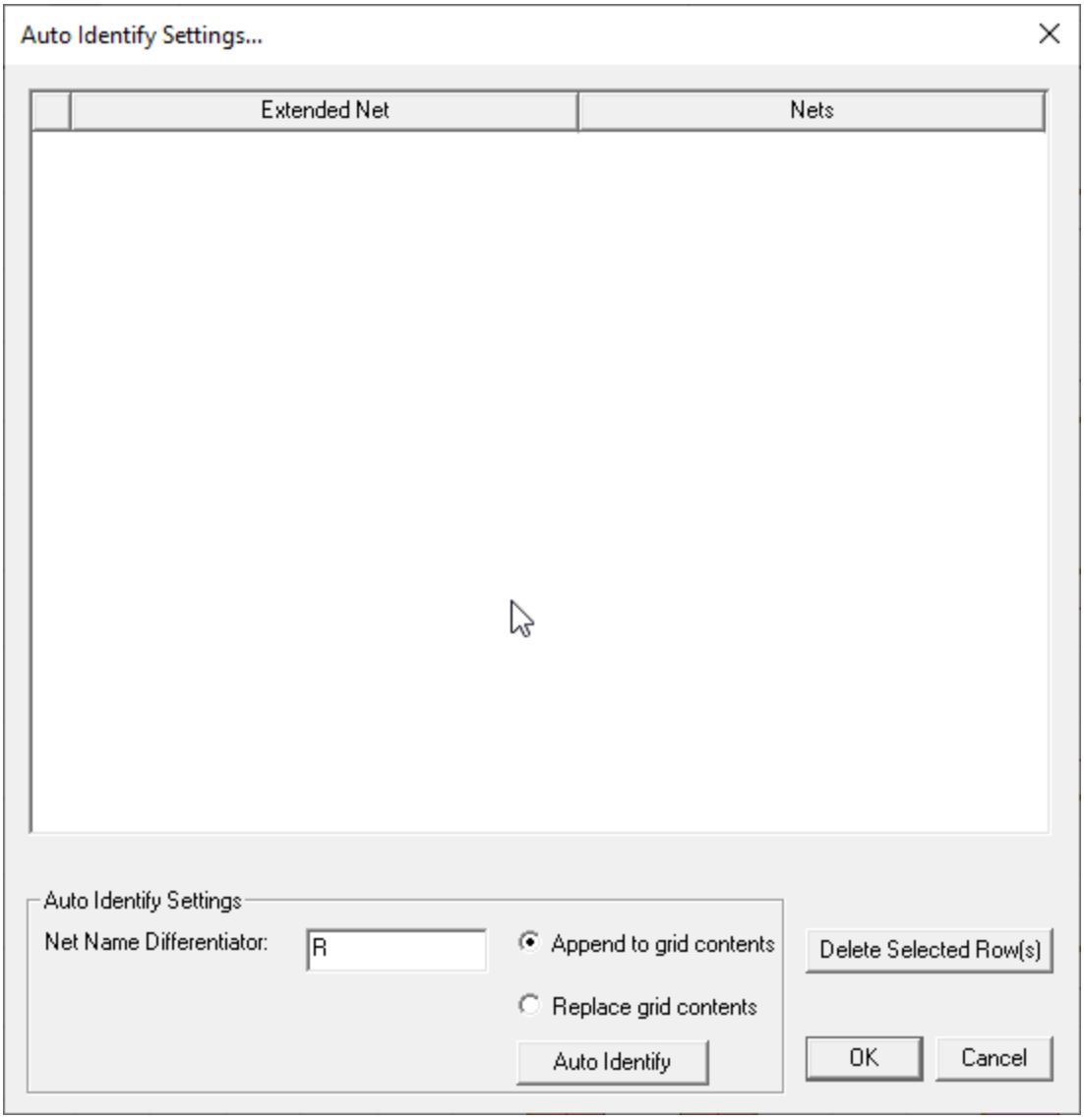
1. In the **Extended Nets** pane, right click and select **Edit** to open the **Extended Net Properties** window.



2. Change the name and optional description.
3. Select the nets that will make up the extended net.
4. Clear the nets that should not be in the extended net.
5. Click **OK**.

Auto Identify Settings window

From the **Auto Identify Settings** window, you can automatically detect and create extended nets, rename extended nets, or delete extended nets. Existing nets are listed in the grid. Enter a differentiator to identify extended nets. Use **Append to grid contents** to add any new extended nets to the list in the grid and **Replace grid contents** to replace existing extended nets with newly identified ones.



Nets Pane

The **Nets** pane displays the members of the extended nets that are selected in the **Extended Nets** pane. These nets can be filtered using regular expressions. In the **Nets** pane, the short menu offers these actions:

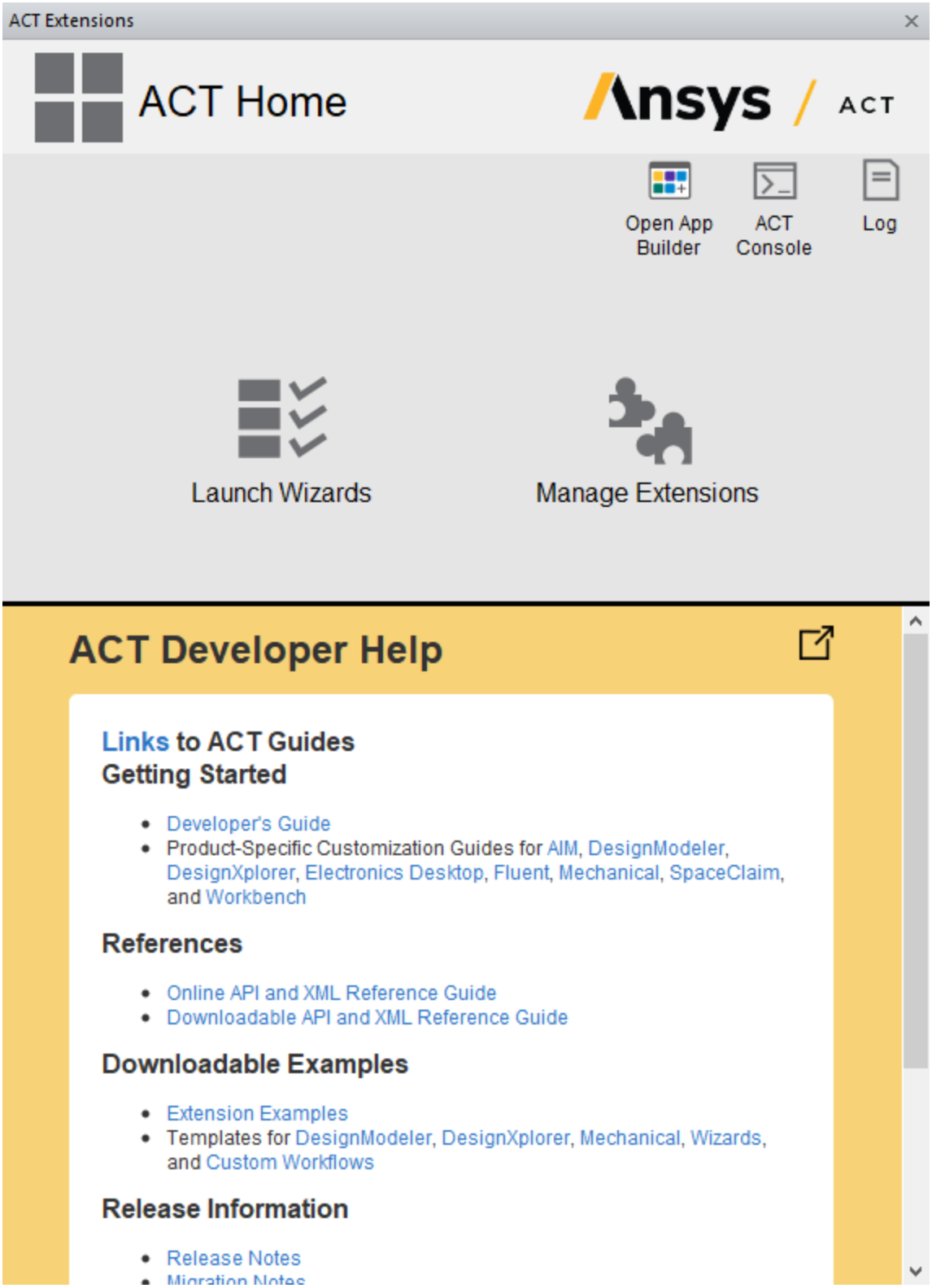
- **Show** – shows the selected nets in the Layout editor if any are hidden.
- **Show (Hide All Other)** – shows the selected nets in the Layout editor while hiding all other nets.
- **Show all nets** – shows all nets in the Layout editor.
- **Hide** – hides the selected nets in the Layout editor.

- **Delete** – deletes the selected nets from the project. Members of the net remain in the project.
- **Fit Selected** – zooms the Layout editor into show the selected nets.

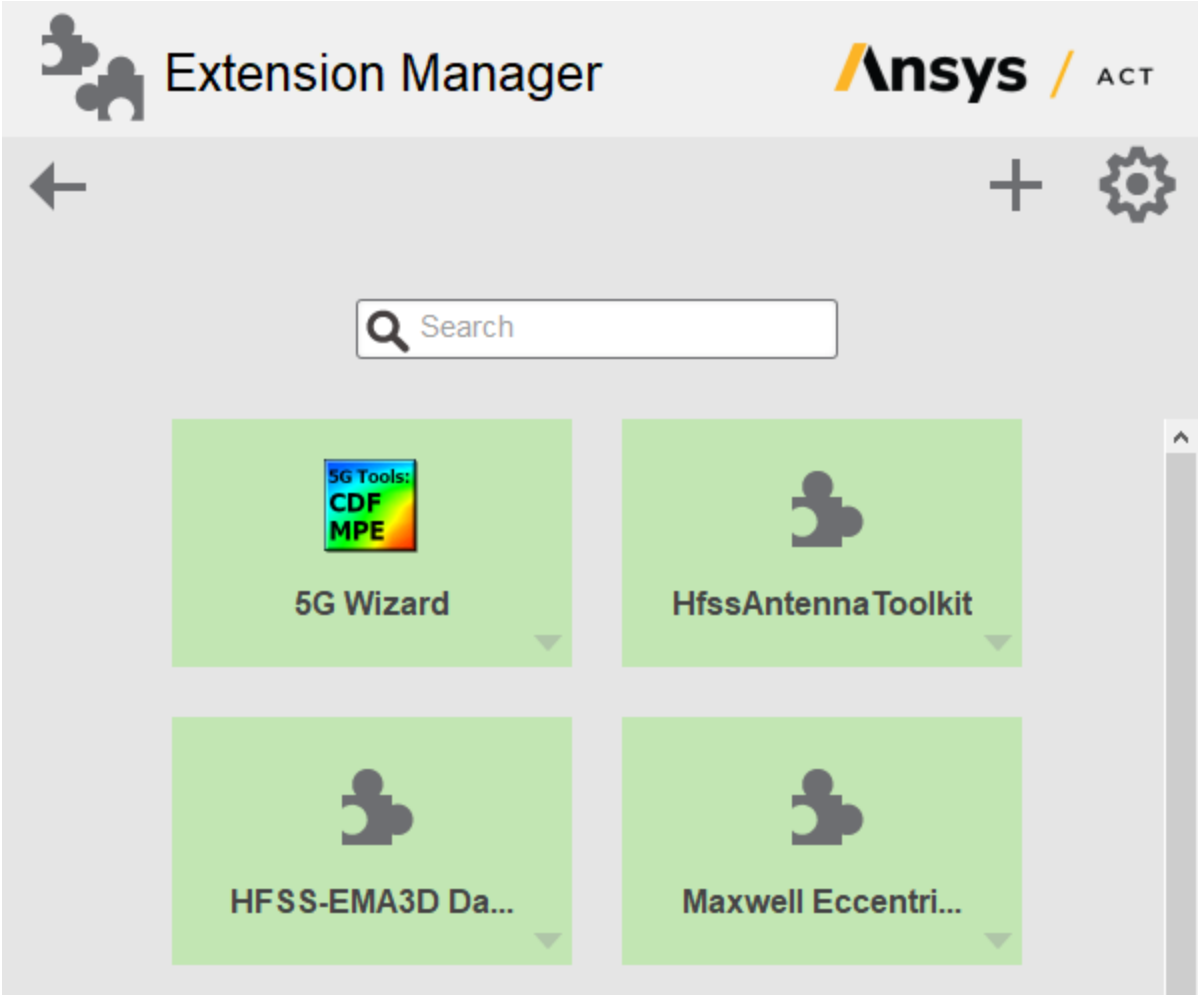
ACT Extensions Window

> **Note:**
>
> The **ACT Extensions** window and the design wizards it contains (*5G Wizard, HFSS Antenna Design Toolkit, HFSS-EMA3D Link,* and *Maxwell Eccentricities*) are only available for the Windows version of the Ansys Electronics Desktop software. These items are not available when the software is installed on a Linux platform.

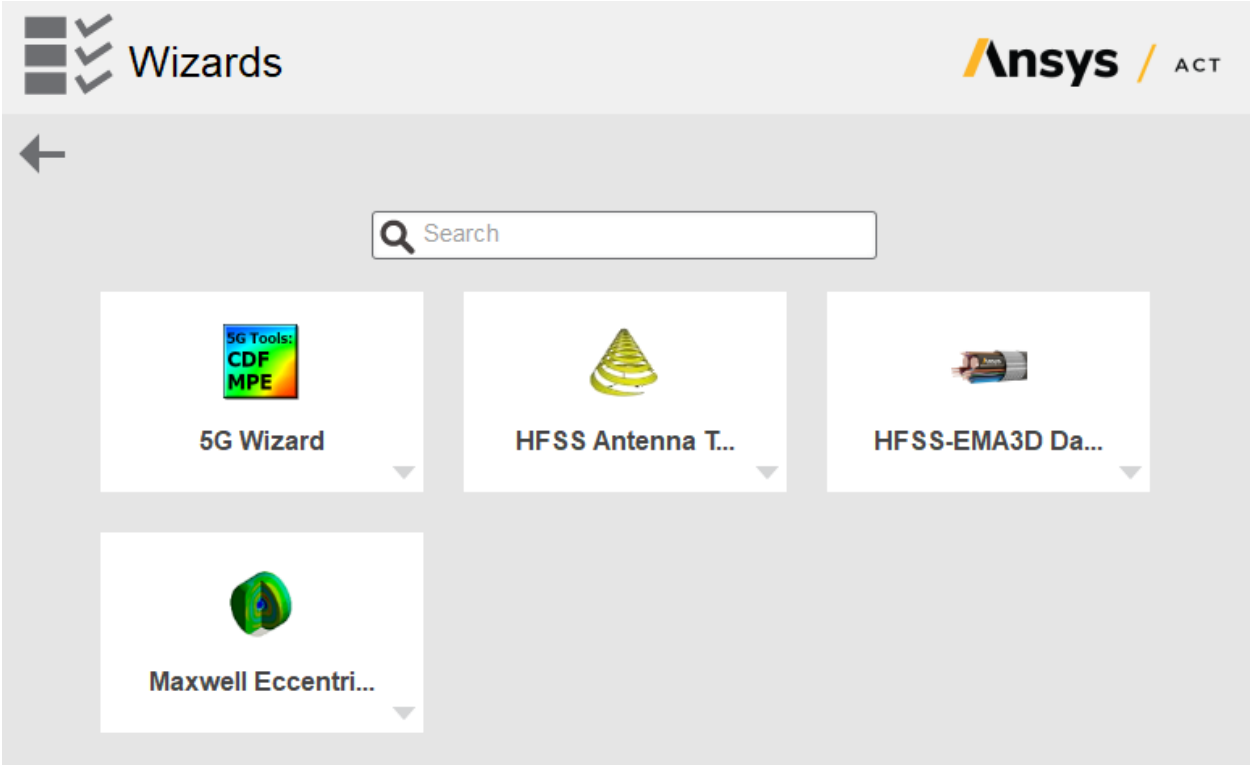
The **View** > **ACT Extensions** command opens the ACT Extensions window for toolkit extensions implemented via Ansys ACT. Depending on how you size and arrange the desktop, the appearance may vary. You may choose to detach and size the window to suit your needs.



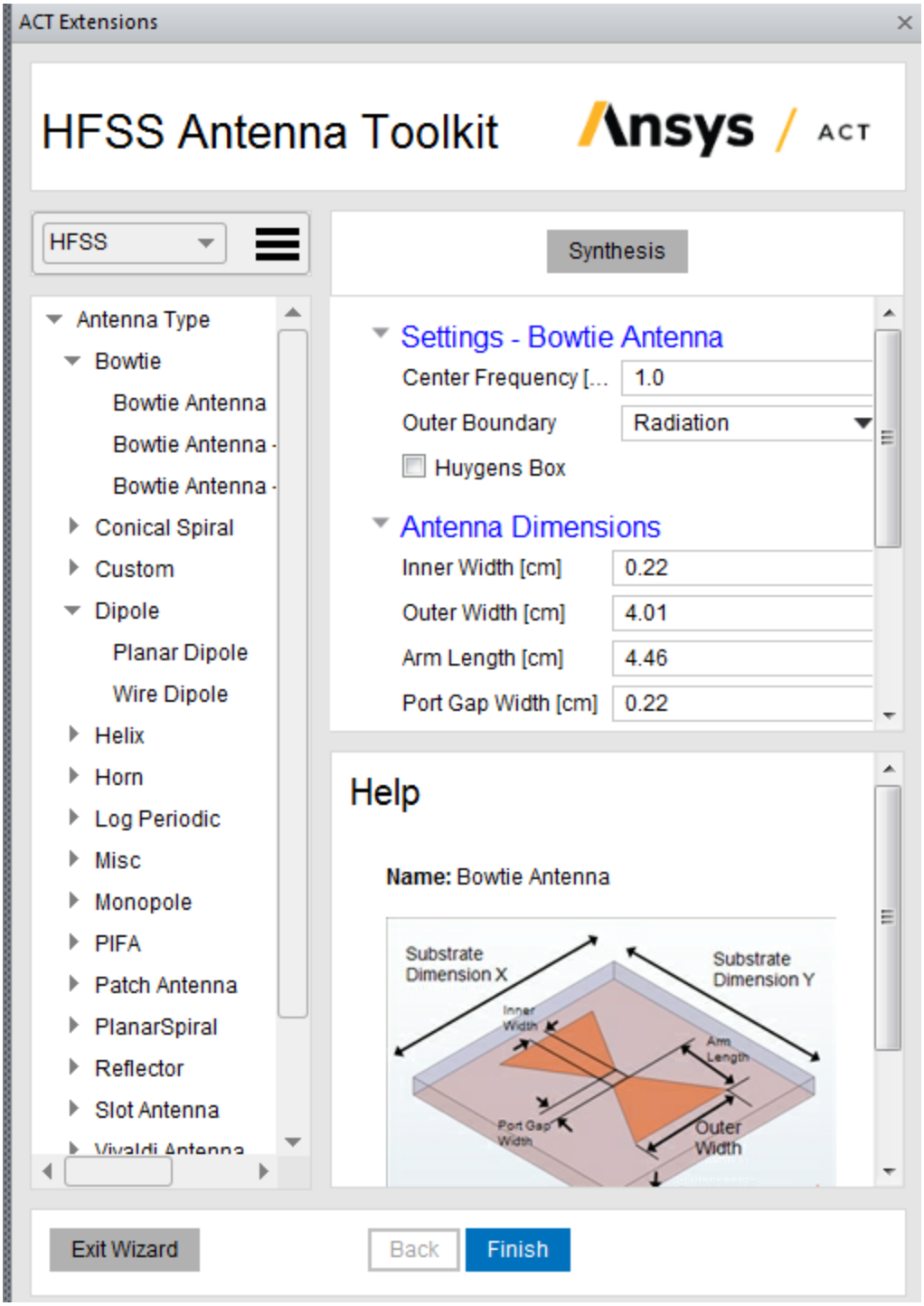
Click the **Manage Wizards** icon to load the **Extension Manager** to select from available extensions. Active extensions are highlighted. Selectable extensions show while. Scroll bars and window sizing lets you view addition extensions. You can search, or access the Ansys App Store.



Click the **Launch Wizards** icon to load existing toolkits for activation. If there are many available extensions, you can use the **Search** bar. Search is not case sensitive.



Click the desired ACT extension to load it. The selected Wizard operates in the window.



Ansys Electronics Desktop Menus

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

Ansys Electronics Desktop contains the following menus, which appear at the top of the desktop:

File	Contains commands to manage Ansys Electronics Desktop project files and printing options.
Edit	Contains commands to modify the objects in the active model and undo and redo actions.
View	Contains commands to display or hide desktop components and model objects, modify 3D Modeler window visual settings, and modify the model view.
Project	Contains commands to add a specific design type to the active project; view and define datasets, project variables, and event callbacks.
Tools	Contains commands to modify the active project's material library, arrange the material libraries, run and record scripts, update project definitions from libraries, customize the desktop's toolbars, and modify many of the software's default settings.
Window	Contains commands to rearrange the 3D Modeler windows and toolbar icons.
Help	Contains commands to access the help system and view the current Ansys Electronics Desktop version information.

Once you have inserted a design type, the menu bar also includes menus specific to that type. These may include:

Draw	Contains commands to draw one-, two-, or three-dimensional objects, and sweep one- and two-dimensional objects.
Modeler	Contains 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.
Ansys Electronics Desktop	Contains commands to set up and manage all the parameters for the active project. Most of these project properties also appear in the project tree.

Top Menu Bar

The **Top Menu** bar contains menus for controlling Ansys Electronics Desktop and the various editors and viewers. Click on a **Top Menu** bar entry to open its menu. In addition, the **Alt** key makes it easy to access the **Top Menu** bar menus.

- A menu in the **Top Menu** bar can be opened by clicking the **Alt** key and then pressing the underlined letter of the menu you wish to activate (the underlines appear when the **Alt** key is pressed).
- Clicking the **Alt** key enables you to scroll across the **Top Menu** bar, opening the menus, by pressing the left (⇐) and right (⇒) arrow keys.

Once a menu is open, you can use the down (⇓) and up (⇑) arrows to change the selection. You can use the right (⇒) arrow key to open a subordinate menu for the selected command. Operations on the menus can be executed in three ways:

- Clicking on the operation.
- Typing the underlined letter as shown in the menu (for example, typing the “n” in **New** on the **File** menu). Type all underlined letters in lower case.
- Using a shortcut key combination as shown in the menu (for example, Ctrl+N for the New command on the **File** menu. Shortcut key combinations are valid whether or not the menu is visible. See Shortcut Keys to locate topics with listings of various types of shortcuts.

Shortcut Menu in the Toolbars Area

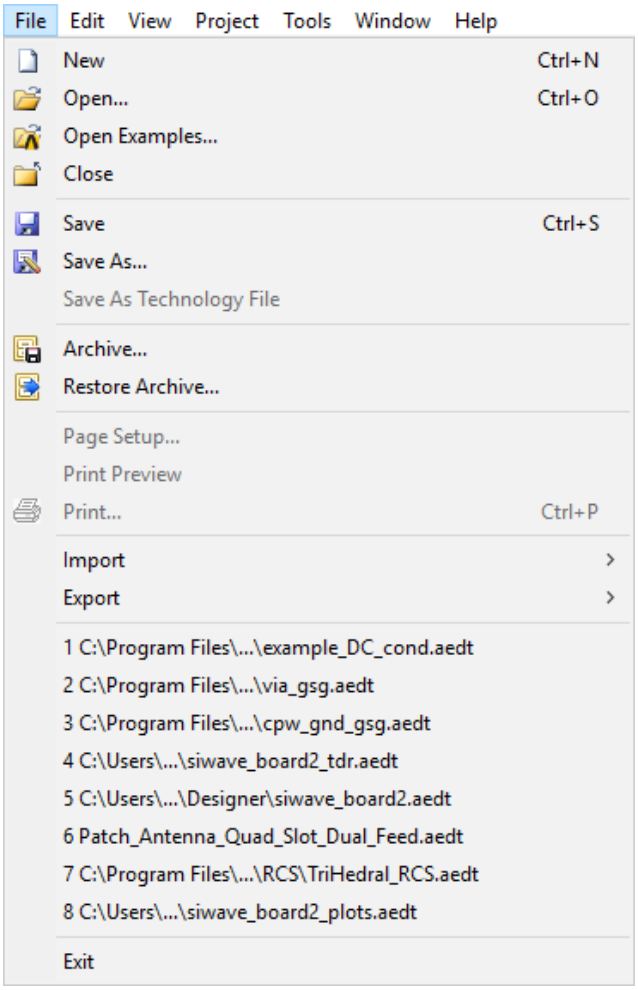
Use the shortcut menu in the toolbars area of the desktop to toggle the show/hide settings for various desktop windows and toolbars. To access the shortcut menu, right-click in the toolbars area at the top of the desktop.

When a project is loaded, from left to right, the Ansys Electronics Desktop drop-down menus are:

File Edit View Project Draw Design-Editor Tools Window Help

These menus are described in the following sections.

File Menu



File menu items are common operations on files and projects. You can search the Help to find information on any of the commands that appear on Ansys Electronics Desktop menus.

- Click **New** (or type **n**) to set up a new project.
- Click **Open** (or type **o**) to open an existing project.
- Click **Open Examples** to open an example project.
- Click **Close** (or type **c**) to close the selected project. If the project has changed since the last save, you will be prompted to save the project before closing.
- Click **Save** (or type **s**) to save the selected project.
- Click **Save As** (or type **a**) to save the selected project under a different name or in a different directory.

- Click **Save As Technology File** (or type **t**) to save the selected top-level design as a technology file (*.asty). You must have a design selected to activate this menu item. You cannot export a technology file from a project or from a subcircuit.
- Click **Archive** to archive the selected project and any selected additional files. If the project has changed since the last save, you will be prompted to save the project before closing.
- Click **Restore Archive** (or type **r**) to restore a previously archived project.
- Click **Page Setup** (or type **u**) to set up formatting to print the active window in the Design Area.
- Click **Print Preview** (or type **v**) to display a preview of the print job.
- Click **Print** (or type **p**) to print the active window.
- Click **Import** to open a submenu and select a file for import.
- Click **Export** to open a submenu and select a file for export.
- Click the name of a project (or type its number) to open a project from the listing of recently opened projects.
- Click **Exit** (or type **x**) to exit Ansys Electronics Desktop. If any project has unsaved changes, you will be prompted to save the project before closing.

Edit Menu

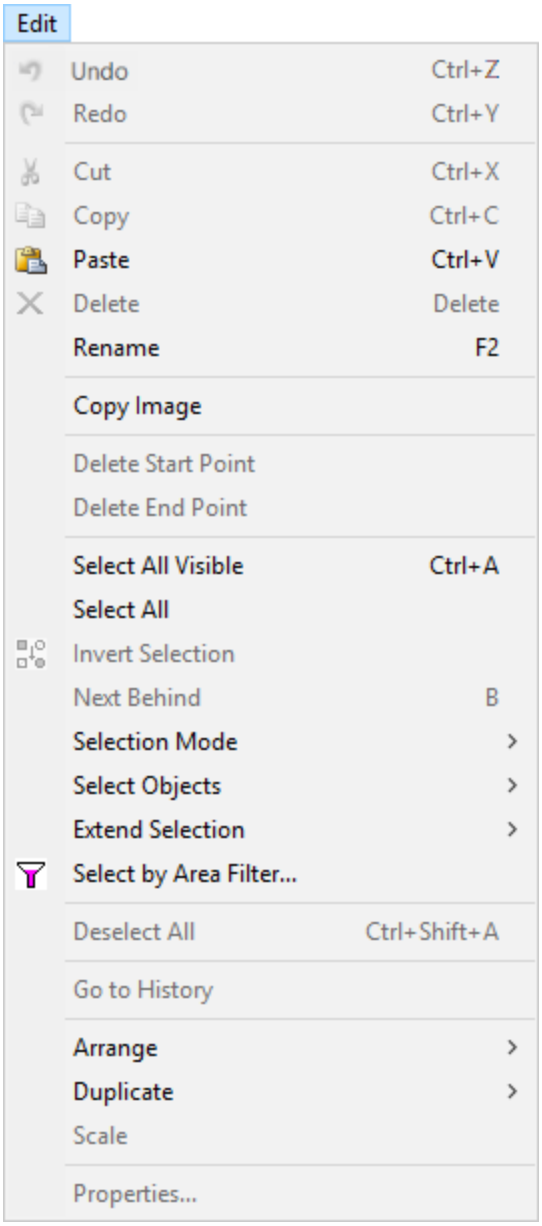
The **Edit** menu changes, depending on the active window in the Design Area.

These menus include:

- 3D Modeler
- Netlist Editor
- Report Window

Edit Menu for 3D Modeler

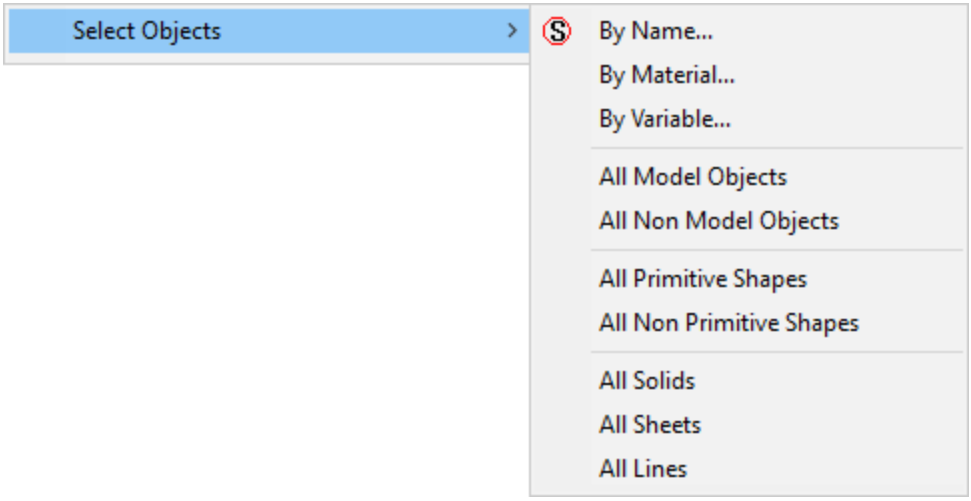
When a Modeler window is active in the Design Area, the **Edit** menu is similar to the following image:



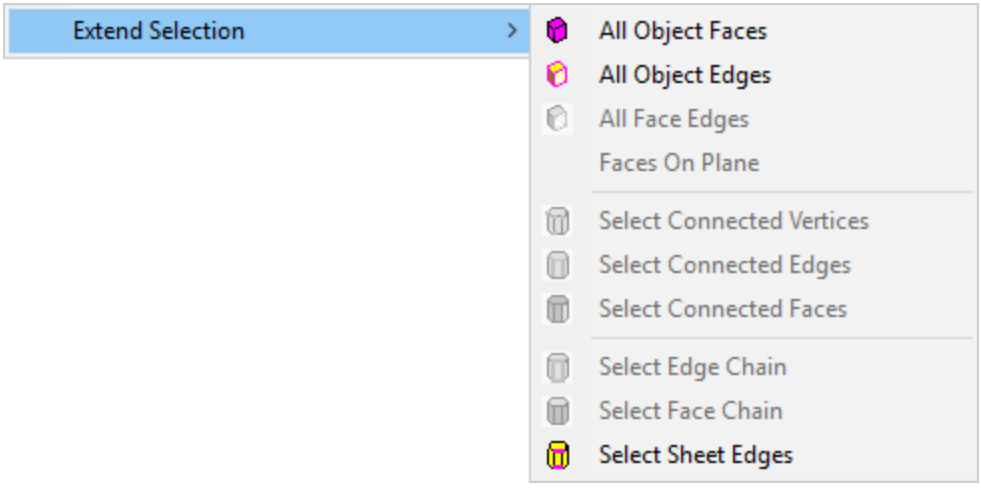
Comands will be available or not available for clicking depending on what is currently selected or on whether any operations have been performed, undone, or redone. The 3D Modeler's Edit menu contains the following options:

- **Undo** – allows you to undo the last action, which is specified after the word "Undo."
- **Redo** – allows you to redo an undone action, which is specified after the word "Redo."
- **Cut / Copy / Paste** – allows you to copy and paste elements.
- **Delete** – deletes a selected element.
- **Rename** – renames a selected element.

- **Copy Image** – copies the visible design area to the clipboard, in bitmap format.
- **Delete Start Point / Delete End Point** – allows you to delete start points and end points.
- **Select All Visible / Select All** – selects either all objects or all objects visible in the design area.
- **Invert Selection** – selects the opposite of the current selection.
- **Next Behind** – selects the object behind a selected face, edge, vertex, or object.
- **Selection Mode** – changes the selection mode.
- **Select Objects** – opens a submenu that allows you to select objects and shapes by name, material, and type.



- **Extend Selection** – opens a submenu that allows you to extend your current selection.

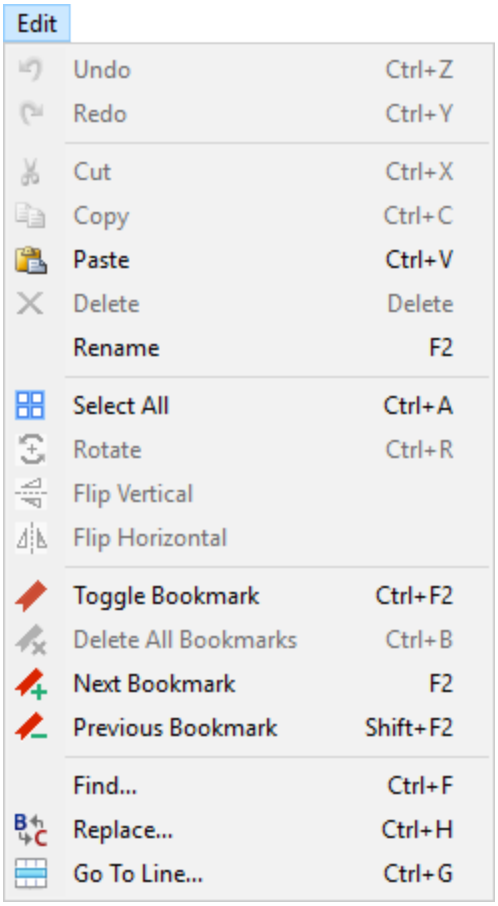


- **Select by Area Filter** – opens the **Select By Area Filters** window.
- **Deselect All** – deselects all selected objects.

- **Go To History** – selects the History Tree entry for the selected object.
- **Arrange** – allows you to move, rotate, or mirror the selected object.
- **Duplicate** – allows you to duplicate the selected object along a line, along an axis, or mirrored to the selection.
- **Scale** – allows you to scale the selected object.
- **Properties** – opens the relevant Properties window for the selected object.

Edit Menu for Netlist Editor

When the Netlist Editor is the active window, the **Edit** menu appears similar to the following:



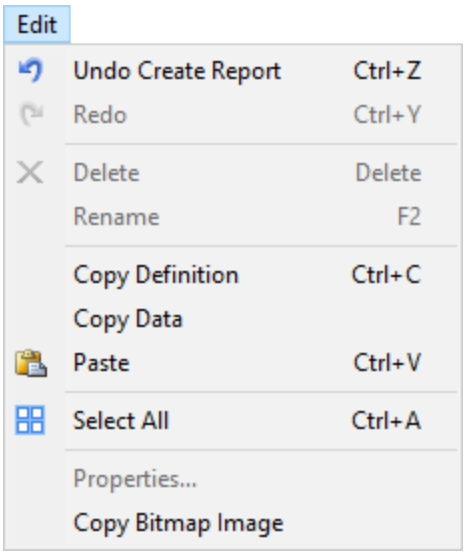
The Netlist Editor Edit menu contains the following options:

- **Undo** – allows you to undo the last action, which is specified after "Undo."
- **Redo** – allows you to redo an undone action, which is specified after "Redo."
- **Cut / Copy / Paste** – allows you to copy and paste netlist elements.
- **Delete** – deletes a selected element.
- **Rename** – renames a selected element.

- **Select All** – selects the entire netlist.
- **Rotate / Flip Vertical / Flip Horizontal** – deactivated options.
- **Toggle Bookmark** – toggles the display of bookmarks.
- **Delete All Bookmarks** – deletes all bookmarks.
- **Next Bookmark / Previous Bookmark** – allows you to navigate between bookmarked lines.
- **Find** – searches the netlist for a string of text.
- **Replace** – searches the netlist for a string of text and replaces it with different text that you specify.
- **Go To Line** – allows you to enter a line number and jump to that line.

Edit Menu for Report Window

When the Report Window is the active window, the **Edit** menu appears similar to the following:



The Report Window Edit menu contains the following options:

- **Undo** – allows you to undo the last action, which is specified after "Undo."
- **Redo** – allows you to redo an undone action, which is specified after "Redo."
- **Delete** – deletes a selected report element.
- **Rename** – renames a selected report element.
- **Copy Definition** – copies the report definition.
- **Copy Data** – copies report data.
- **Paste** – pastes the copied information at your selected location.
- **Select All** – selects everything in the report window.

- **Properties** – opens the relevant Properties window for the selected report element.
- **Copy Bitmap Image** – copies the entire report window as a bitmap image that can be pasted into another program or document.

View Menu

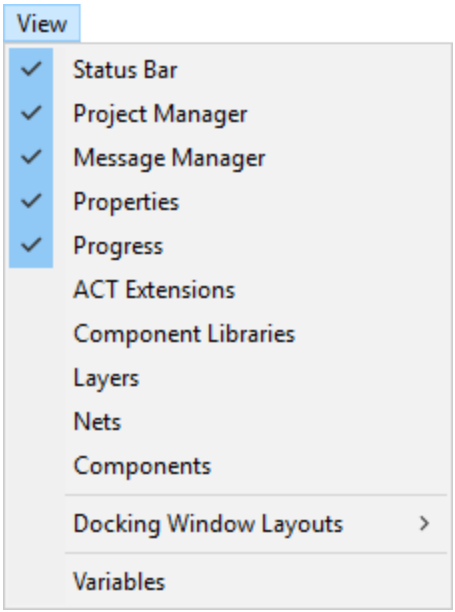
The **View** menu changes, depending on the active window in the Design Area.

These menus include:

- Basic View Menu
- Schematic Editor's View Menu
- Layout Editor's View Menu
- 3D Modeler's View Menu

Basic View Menu

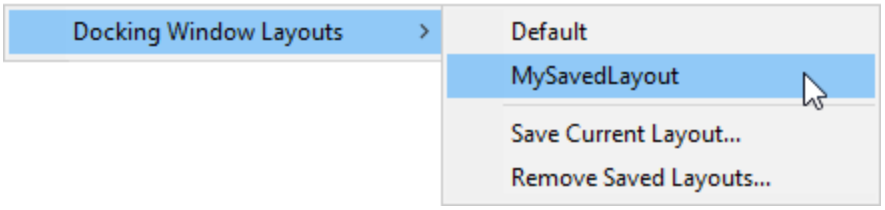
When no editor is open in the design area, or when the Netlist Editor or Report Window is active, the **View** menu appears similar to the following:



- The check boxes toggle the display of the **Status Bar**, **Message Manager Window**, **Project Manager**, **Properties Window**, and **Progress Window**. You can enable the display of **Component Libraries**, **Layers**, **Nets**, **Components** and **Variables** windows.
- The **ACT Extensions** command opens the *ACT Extensions* window, which lets you work with ACT integration tools. For more information, see the ACT Extensions Window section.

> **Note:**
>
> The **ACT Extensions** window and the design wizards it contains (*5G Wizard, HFSS Antenna Design Toolkit, HFSS-EMA3D Link,* and *Maxwell Eccentricities*) are only available for the Windows version of the Ansys Electronics Desktop software. These items are not available when the software is installed on a Linux platform.

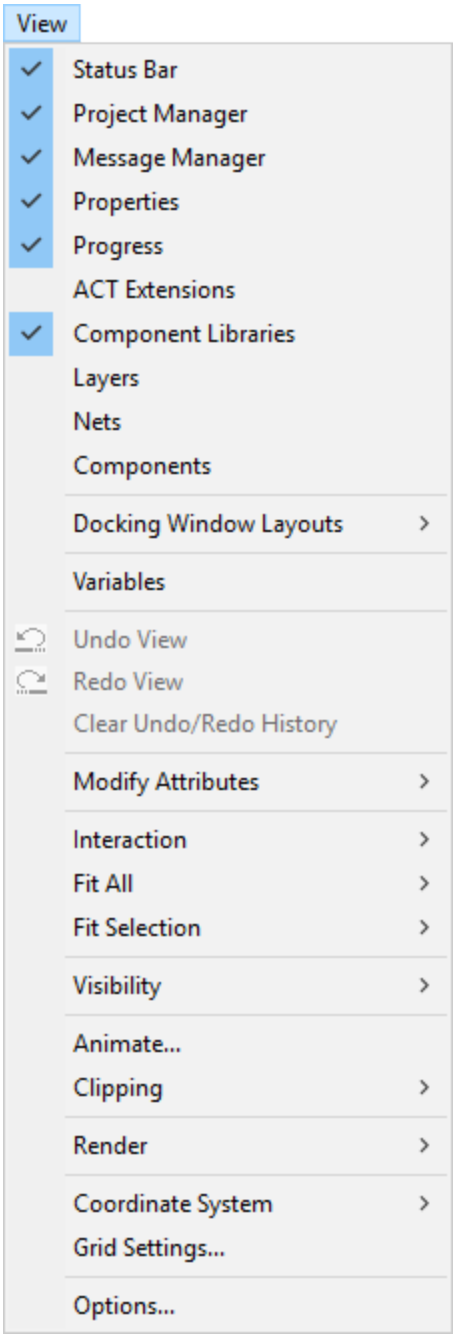
- The **Docking Window Layouts** submenu lets you select from the **Default** and any saved window layouts, **Save Current Layout**, or **Remove Saved Layouts** by selecting them from a list.



- The **Variables** option opens the **Project** and **Design** **Variables** window, allowing you to set variables.

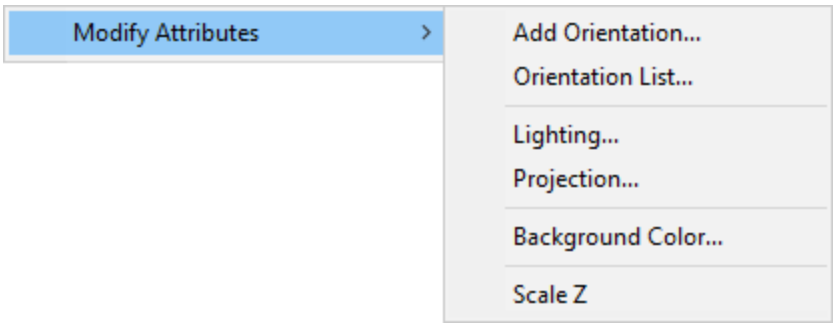
3D Modeler View Menu

When a Modeler window is active in the design area, the **View** menu's appearance is similar to the following image:

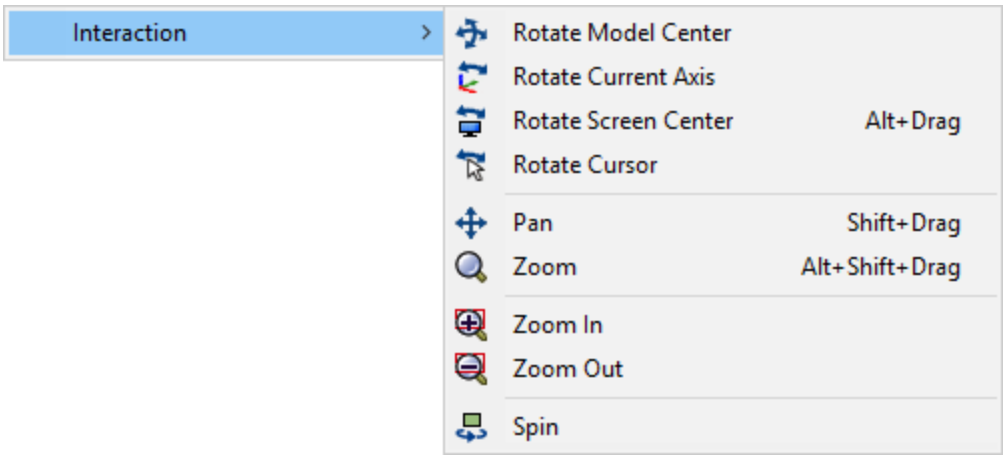


Thi View menu includes the options available on the Basic View Menu. The following additional options are available when a *Modeler* window is active. Note that a number of these options also appear in the View branch of the shortcut menu that appears when you right-click in the Modeler window.

- **Undo View** / **Redo View** / **Clear Undo/Redo History** – allow you to change the view in the design area, based on view history.
- **Modify Attributes** – offers options to **Add Orientation**; view the **Orientation List**; change **Lighting**, **Projection**, or **Background Color**; and **Scale Z**.

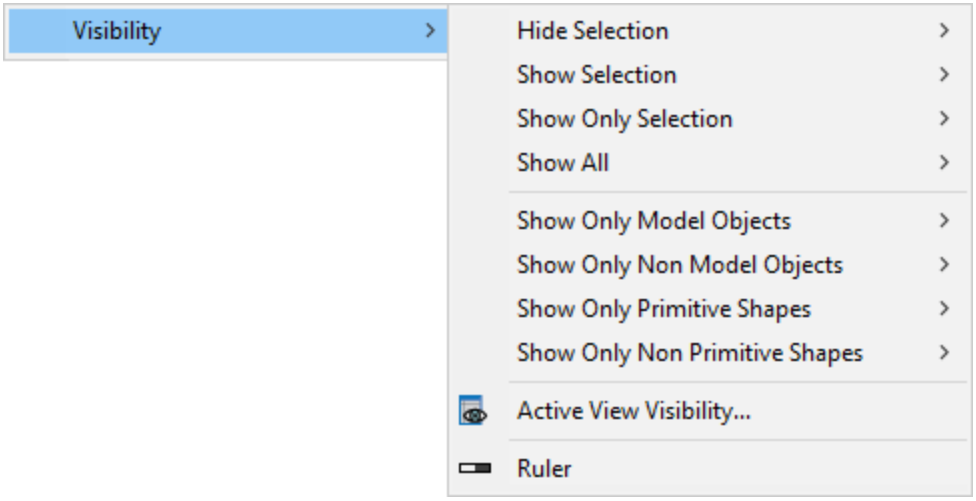


- **Interaction** – offers options to rotate, pan, zoom, and spin the drawing in the design area.

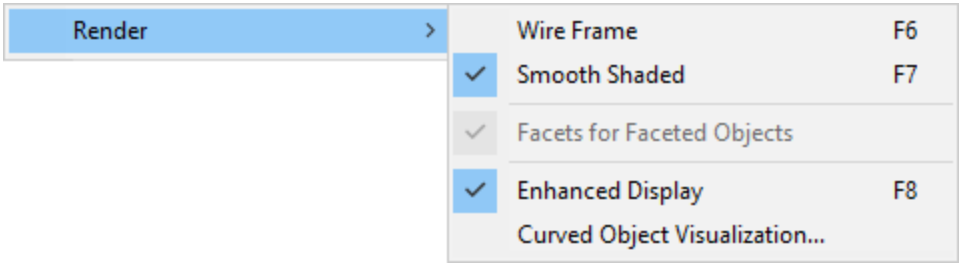


- **Fit All** – allows you to fit all objects to the view window.
- **Fit Selection** – allows you to fit selected objects to the view window.

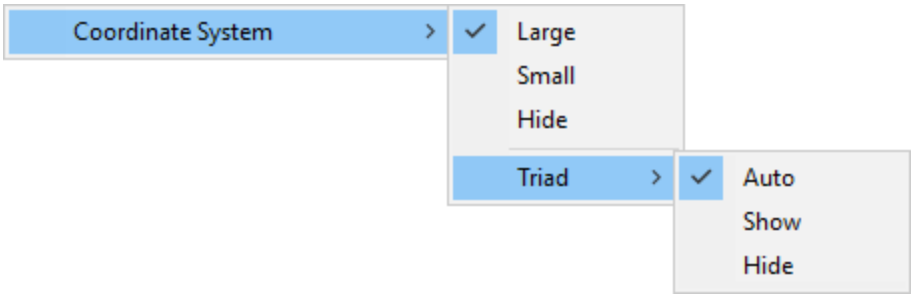
- **Visibility** – offers options to show and hide selections, objects, shapes, and the ruler.



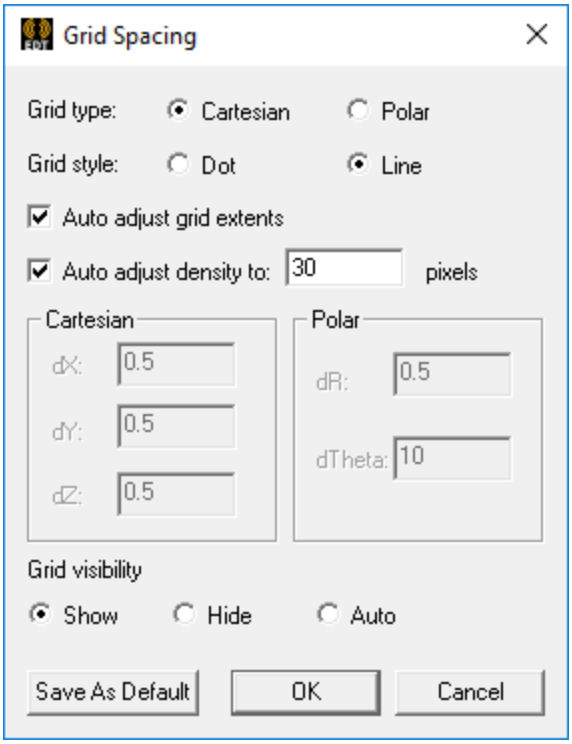
- **Animate** – opens the **Create Animation Setup** window, so that you can begin creating animations.
- **Clipping** – allows you to place a 3D coordinate system manipulator in the design area so that you can define clip planes.
- **Render** – offers options to switch between wire frame and smooth shaded render, toggle enhanced display, and change how curved objects are visualized.



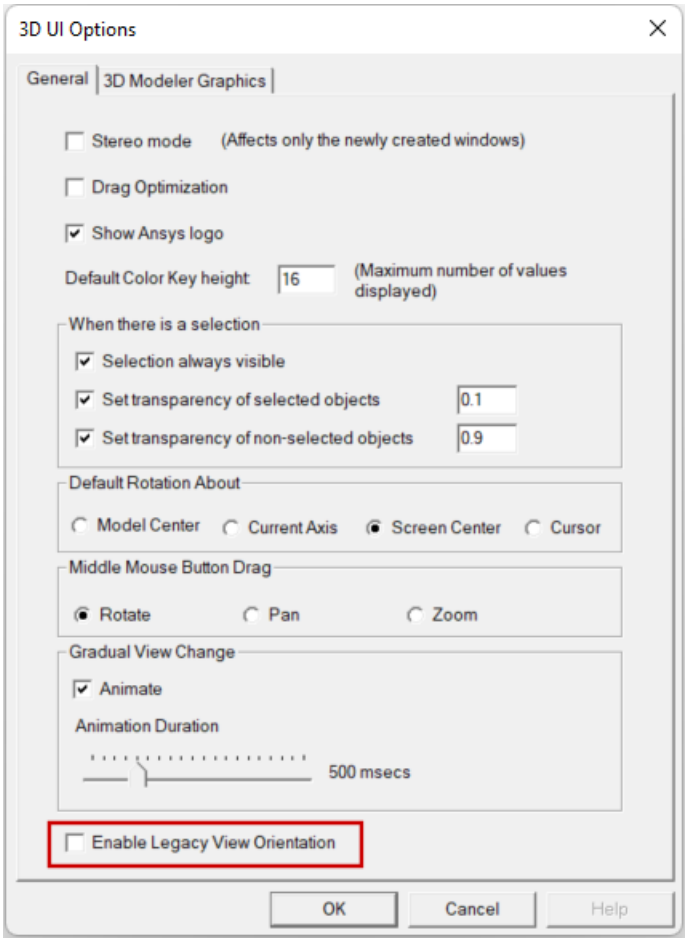
- **Coordinate System** – offers options to change how the coordinate system displays in the design area.



- **Grid Settings** – opens the **Grid Spacing** window, where you can adjust spacing.

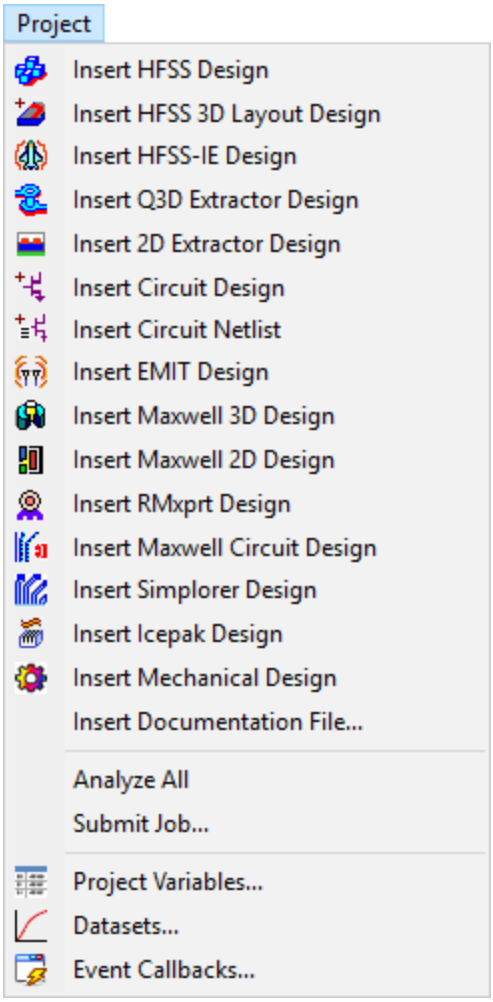


- **Options** – opens the **3D UI Options** window, where you can change additional settings.



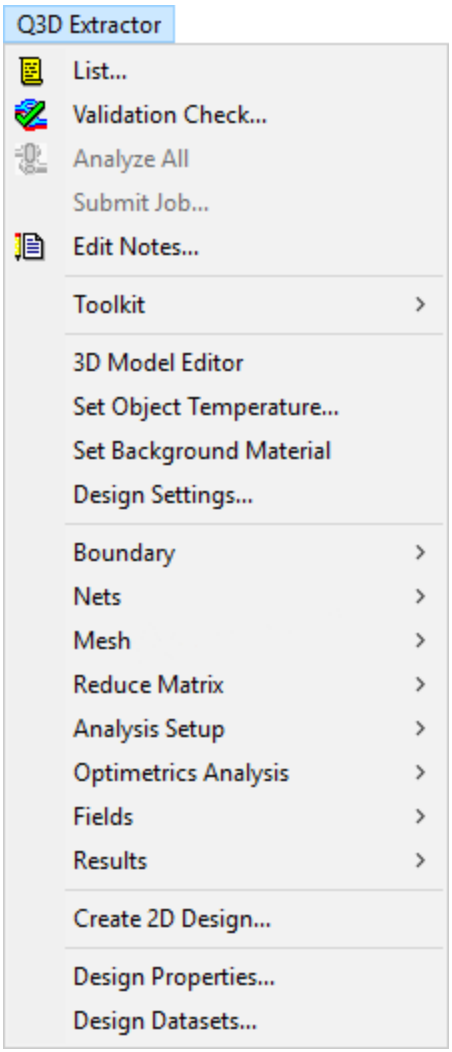
Project Menu

To create (insert) a new design, you must first open a new project folder by selecting **New** from the **File** menu. Then select a project to insert from the Project menu. Or you can click the icon corresponding to the type of design you wish to create.



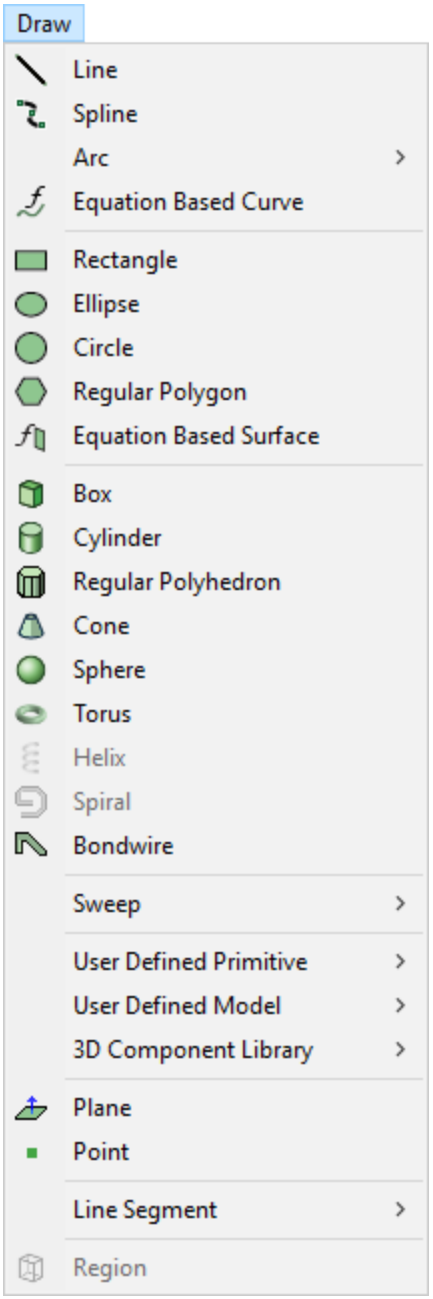
Editor and Design Specific Menus

The Top Menu bar contains editor and design menus that are specific to the editor or viewer that is active in the Design area. The specific menus correspond to the type of design that is inserted using the Project menu.



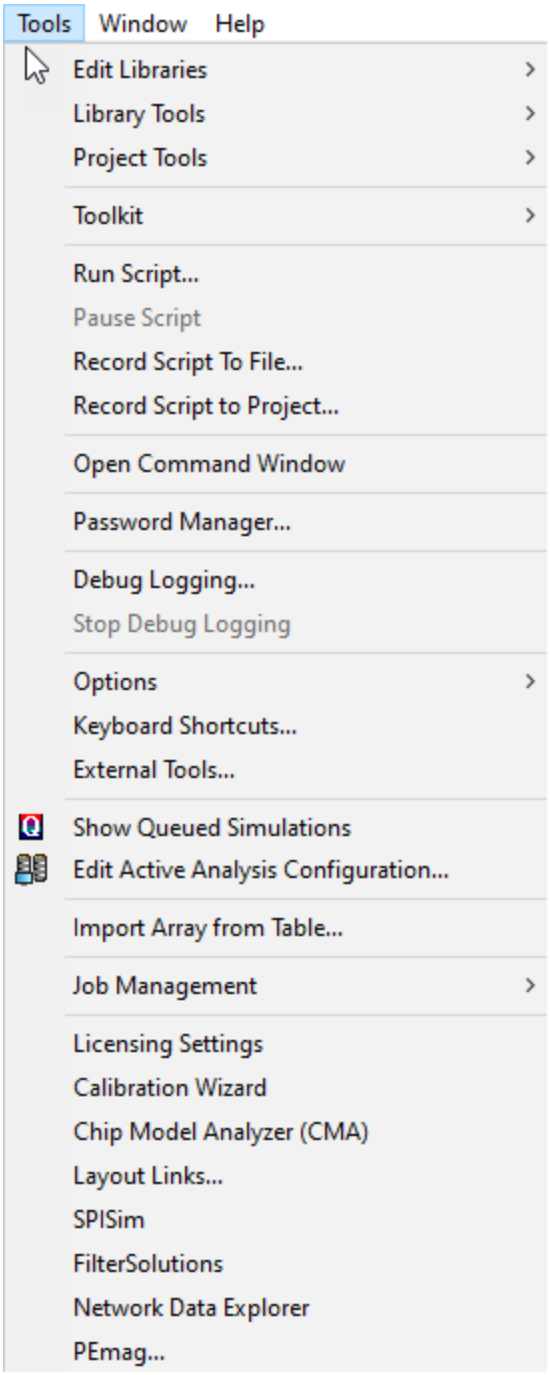
Layout Editor Draw Menu

The **Draw** menu for the **Layout Editor** is context sensitive and appears slightly different depending on the type of design that is loaded.

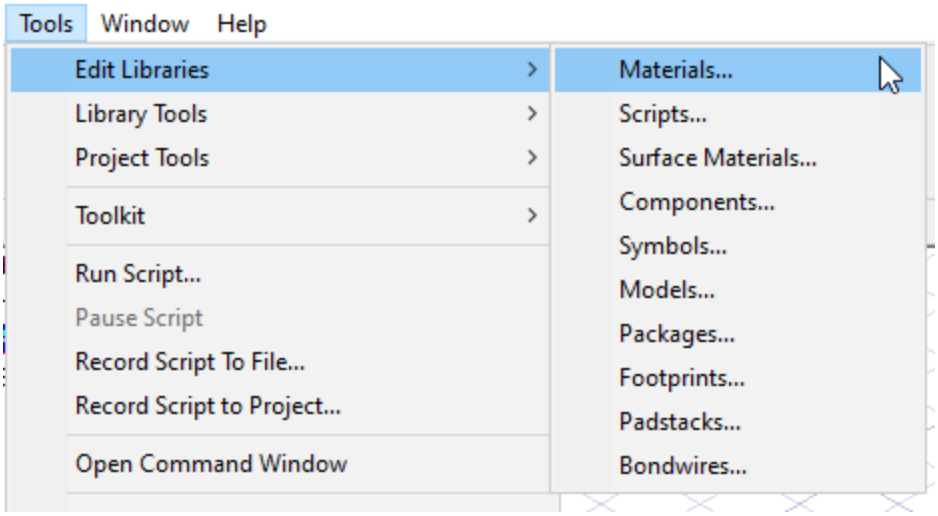


Tools Menu

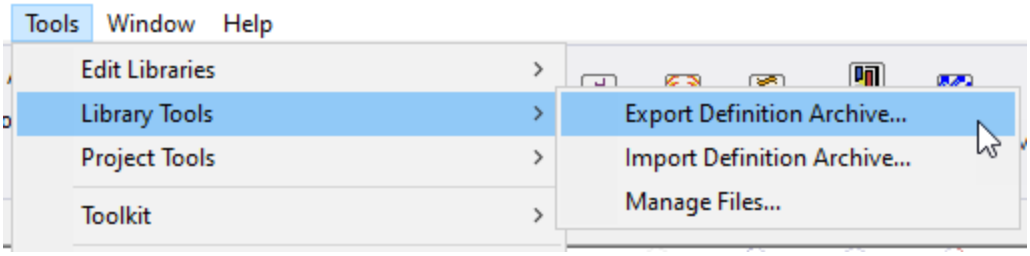
The **Tools** menu contains operations that are common to the analysis tools.



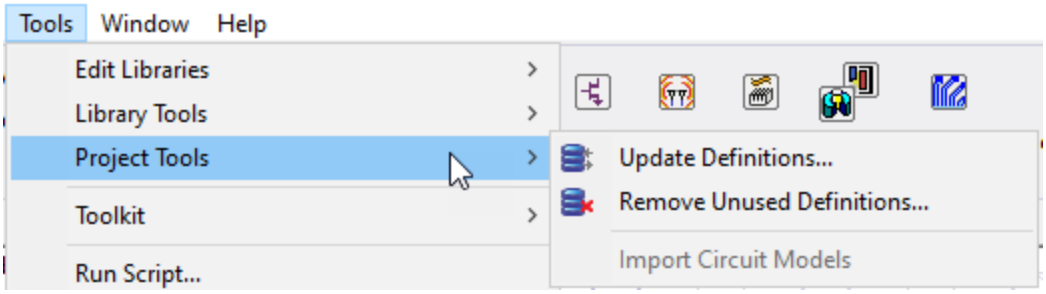
- Click **Tools>Edit Libraries** to view menus allowing access to the libraries available for the various Ansys Electronic Desktop solves, such as Materials, Scripts, and Components, Symbols and Models for Simplorer, Icepak, and Circuit.



- Click **Tools>Edit Library Tools** to view tools for importing or exporting definition archives for UserLib and Syslib and to manage files relevant for various solver libraries.



- Click **Tools>Project Tools** to manage definition libraries relevant to various solvers.



- Click **Tools>Toolkit** to install or update, or access Python and Python-based toolkits.
- Click **Tools> Runscript...** to run scripts that you have recorded. A Pause command is available. You can also **Record Script to File...** or **Record Script to Project...**. See the Scripting Help for more information.

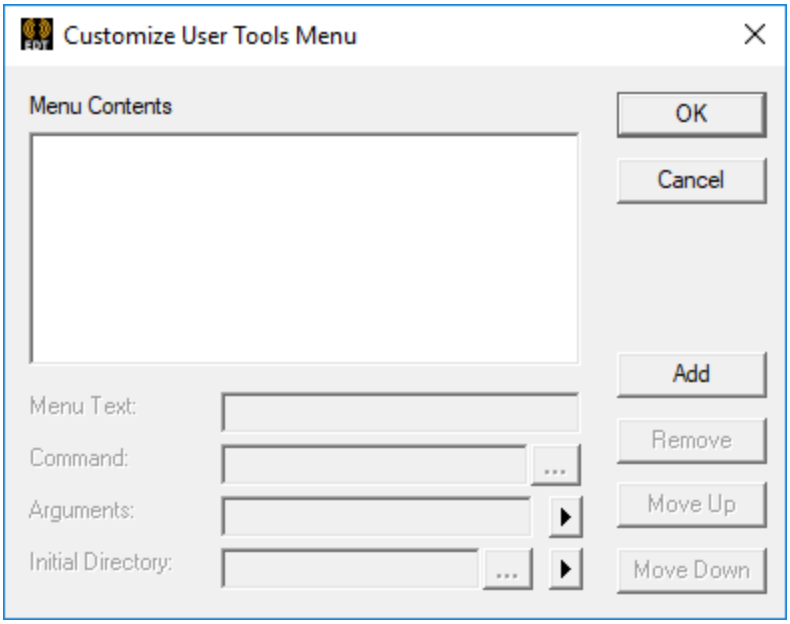
- Click **Tools> Command Window...** to open an IronPython command window. See the Scripting Help for more information.
- Click **Tools>Debug Logging** to enable debug logging via an environment variable. When enabled, the **Debug Logging** is disabled and **Stop Debug Logging** is enabled.
- Click **Tools> Options** to set General Options and HPC options. You can also Export Optionsfiles.
- Click **Tools >Keboard Shortcuts..**. to manage your keyboard shortcut behavior.
- Click **Tools> External Tools...** to manage menu access to external tools.
- Click **Tools> Show Queued Simulations** to view a dialog listing simulations in a queue and providing tools to manage the queue.
- Click **Tools>Edit Active Analysis Configuration** to view a dialog that lets you view and manage the current configuration.
- Click **Tools>Import Array from Table** to import an array definition from a .csv file.
- Click **Tools>Job Management...** to select a scheduler, and to submit and monitor remote simulations.
- Click **Tools>License Settings** to launch the Licensing Settings tool.
- Additional Tools listed include Calibration Wizard, and Network Analyzer, and other tools used to help you with Ansys Electronics Desktop projects and solvers.

Adding External Tools to the Tools Menu

To add an executable to the Tools menu:

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

   If a **User Tools** menu item has been defined, its contents are displayed. Command buttons let you Add new commands and Delete selected commands, and Move Up and Move Down commands. You can specify the command line arguments to the program and the directory from which it will run.



3. To add a custom Tools menu entry, click **Add**.

   This enables the following fields:

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

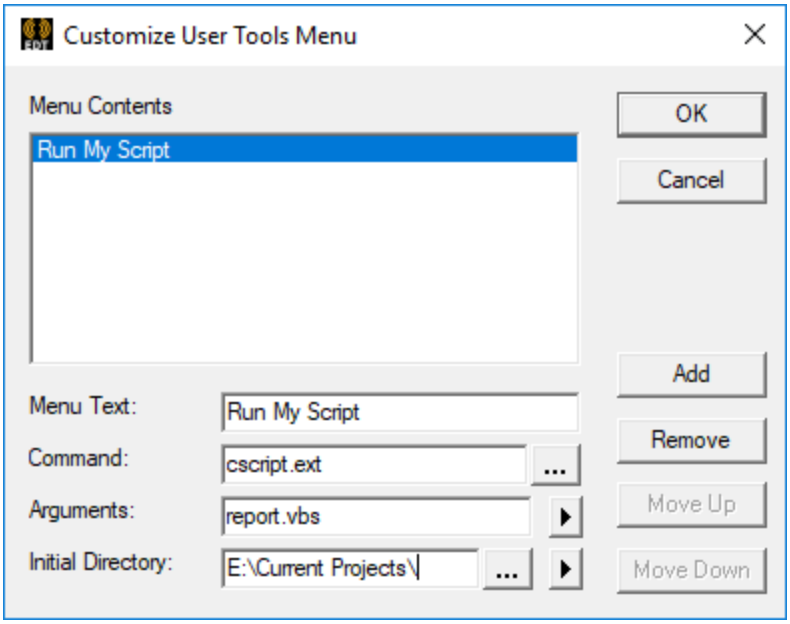
   Command field – displays the external executable. An ellipsis button [...] lets you navigate to the file location.

   Arguments 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 – 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 Q3D Extractor or Cancel to close the dialog without changes.

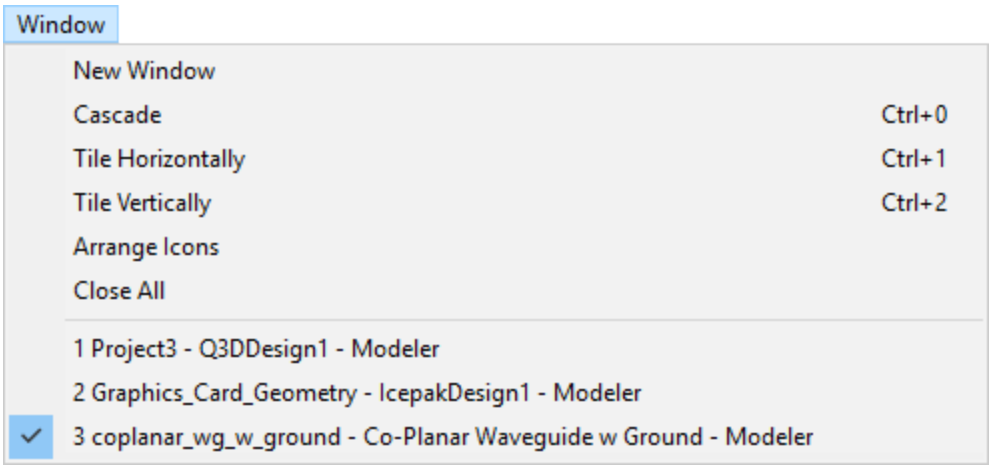
   You can also add scripts to the **Tools** menu. Assuming you have a script to generate custom reports called report.vbs, use the cscript.exe program to execute your script.



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

Window Menu

The **Window** menu contains common window control operations, as well as a list of projects that are currently open.



- Click **New Window** (or type **n**) to open a new window in the Design Area. The new window will show the active design.
- Click **Cascade** (or type **c**) to arrange the open design windows in overlapping sequence:

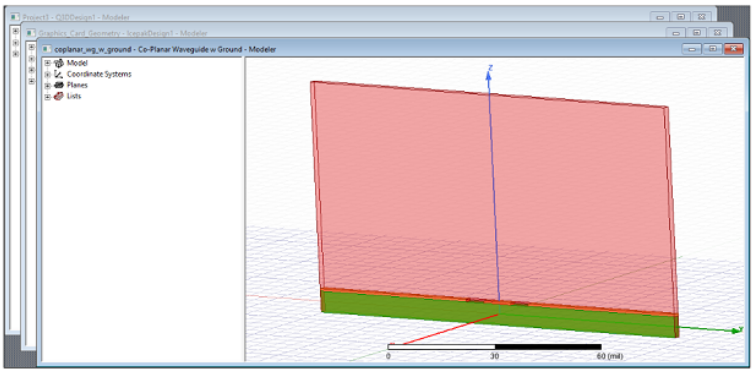
You can use the **Cascade** operation to restore all windows to their default sizes after one or more of them have been maximized (enlarged to fill the entire Design Area).

- Click **Tile Horizontally** (or type **h**) to arrange the open design windows in a top-to-bottom sequence:
- Click **Tile Vertically** (or type **v**) to arrange the open design windows in a side-to-side sequence:

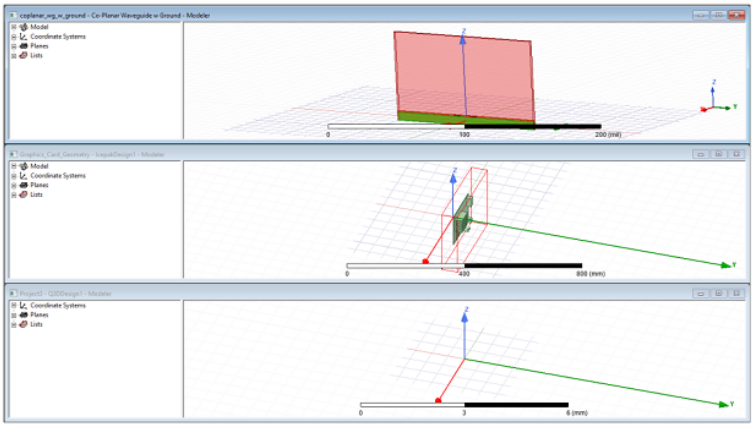
> **Tip:**
>
> You can use the shortcut key sequences Ctrl+0, Ctrl+1, and Ctrl+2 to execute the Cascade, Tile Horizontal, and Tile Vertical operations, respectively. The shortcuts can be used at any time, bypassing the Window menu.

- Click **Arrange Icons** (or type **a**) to restore iconized windows to a neat row at the bottom of the Design Area, after the icons have been repositioned manually. See Working with Editor Windows for details on iconizing editor windows and on repositioning the iconized windows.
- Click **Close All** (or type **l**) to close all the editor windows in the Design Area.

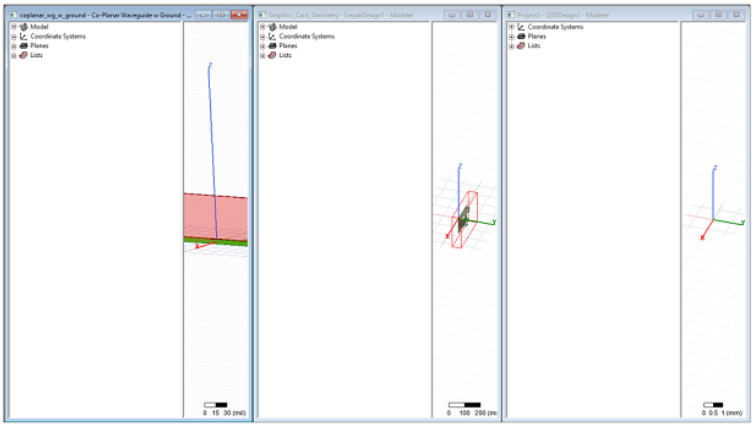
Cascade



Tile Horizontally



Tile Vertically



Help Menu

Ansys Electronics Desktop features extensive help documentation, including text search and context-sensitive help items. To access the main help system, click **Help** on the top menu bar.

The Help Menu displays different selections depending upon the type of design inserted to the active project. The basic selections for Q3D Extractor are:

> **Note:** Ansys Electronics Desktop Student includes access to PDF documentation only.

- **Q3D Extractor Help** – opens the Q3D Extractor help within the Electronics help system. You can also access PDF versions from within the help system.
- **Q3D Extractor Scripting Help** – opens the Q3D Extractor scripting help.
- **Q3D Extractor Getting Started Guides** – opens a list of links to the Q3D Extractor Getting Started Guides, which walk you through projects that demonstrate product features.
- **Q3D Extractor PDFs** – provides access to PDFs for Q3D Extractor, including the main help, scripting guide, and Getting Started Guides.
- **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 2025 R1.
- **Ansys Product Improvement Program** – opens a window describing the Product Improvement Program option.
- **License Settings** – Opens help for the License Settings tool.
- **About Ansys Electronics Desktop** – opens a dialog box that displays the Ansys Electromagnetics Suite release number and contains tabs that show information about the **Installed Components** and **Client License Settings**.
- **Ansys Innovation Courses** – opens a web page containing a wide range of Ansys Electronics Engineering courses using on-demand, self-paced video training and quizzes.
- **Ansys Learning Hub** – opens a web page with subscription based access to, virtual and self-paced learning across the Ansys Software portfolio.
- **Ansys Knowledge** – opens a web page to expert curated knowledge materials from FAQs to tutorials on simulation topics.
- **Ansys Learning Forum** – opens a web page to Ansys blog containing discussion and presentations from Ansys experts, partners and customers.
- **Customer Portal** – opens a web page to Ansys Product support.
- **About Electronics Desktop** - opens a dialog with version and release information, Ansys Electronics Desktop installed components, and licensing information.

Context-Sensitive Help

To access context-sensitive help from the Ansys Electronics Desktop user interface, press **F1** while your cursor is on an item. The help system specific to the product opens.

Obtaining Information about the Software and Release

To obtain information about the software and release:

1. Click **Help** > **About Ansys Electronics Desktop**
   The **About Ansys® Electromagnetics Suite [release number]** dialog box appears, listing information about the product.
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
   - Ansys License 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** drop-down menu is already specified as **Export (*.txt)**.
   d. Click **Save**.
5. Click **OK** to close the **About Ansys® Electromagnetics Suite [release number]** dialog box.

Shortcut Menus

A variety of shortcut menus — menus that appear when you right-click a selection or in a window — are available in the **3D Modeler** window, in the **Project Manager** window, in the **History Tree**, and in the **Progress** window.

Shortcut menu in the 3D Modeler window	Use the shortcut menu in the **3D Modeler** window to select, magnify, and move objects (zoom, rotate, etc.); change the view; perform boolean

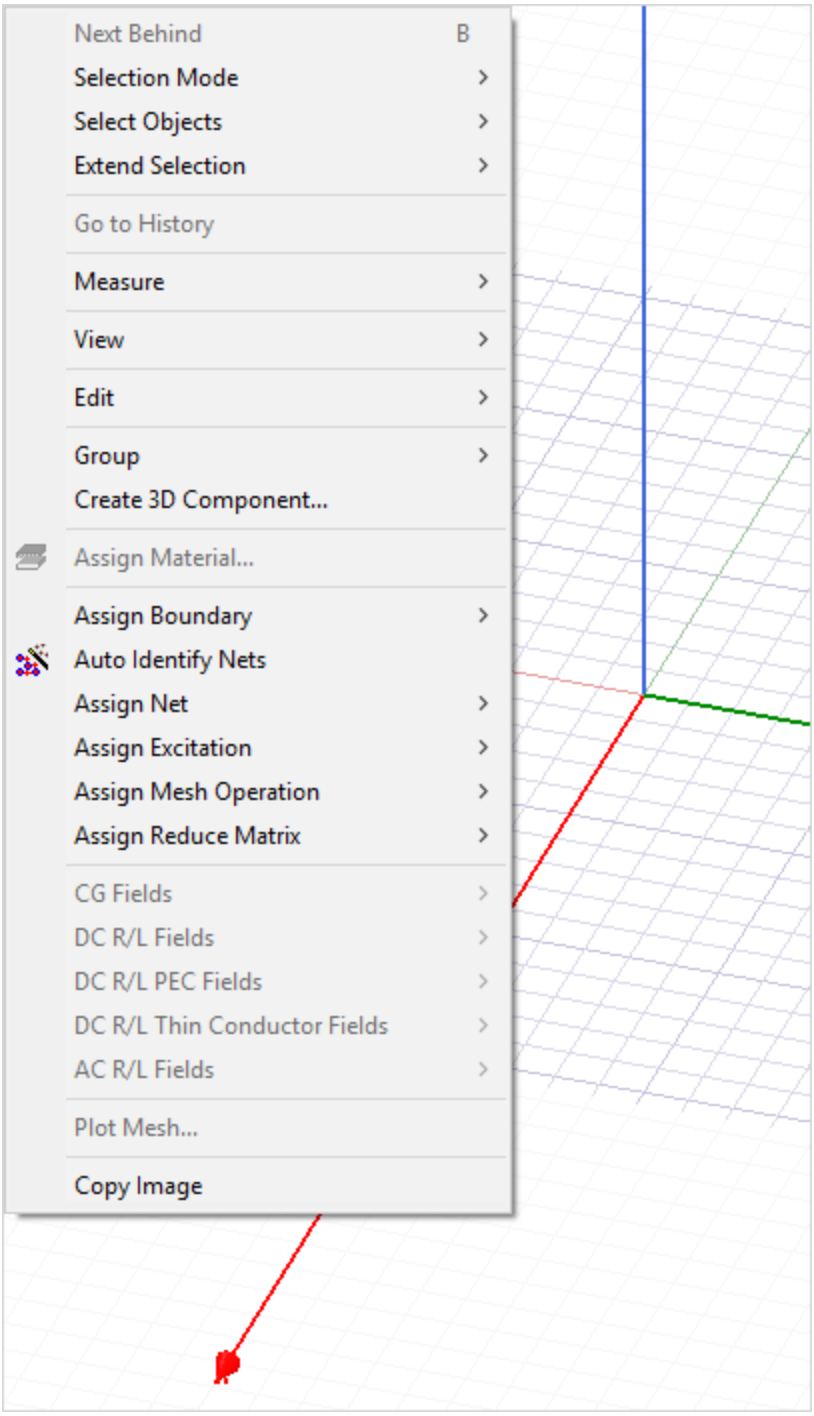
	operations; assign materials, boundaries, excitations, or mesh operations to objects; and work with field overlays.
Shortcut menus in the Project Manager window	Use the shortcut menus in the **Project Manager** window to manage Ansys Electronics Desktop project and design files and design properties; assign and edit boundaries, excitations, and mesh operations; add, analyze, and manage solution setups; add Optimetrics analyses; create post-processing reports; insert far- and near-field radiation setups; edit project definitions; and run Maxwell SPICE.
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(s).
Shortcut menus in the Progress window	Use the shortcut menus in the **Progress** window during a simulation to **Abort** or **Clean Stop**.

> **Note:**
>
> All commands available on shortcut menus are also available from the menu bar.

Shortcut Menu in the 3D Modeler Window

Use the shortcut menu in the **3D Modeler** window to select, magnify, and move objects (zoom, rotate, etc.); change the view; perform boolean operations; assign materials, boundaries, excitations, or mesh operations to objects; and work with field overlays.

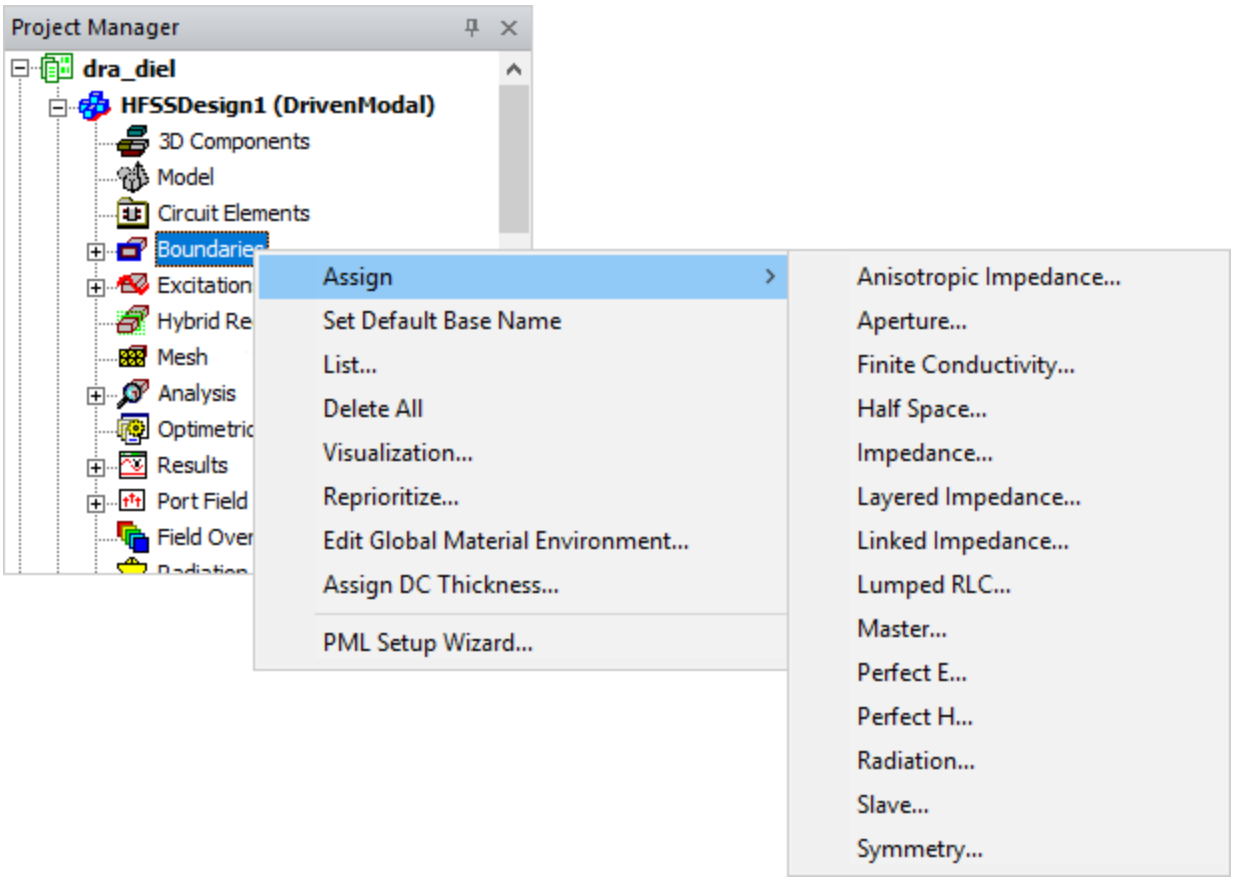
To access the shortcut menu, right-click in the 3D Modeler window.



Shortcut Menus in the Project Manager Window

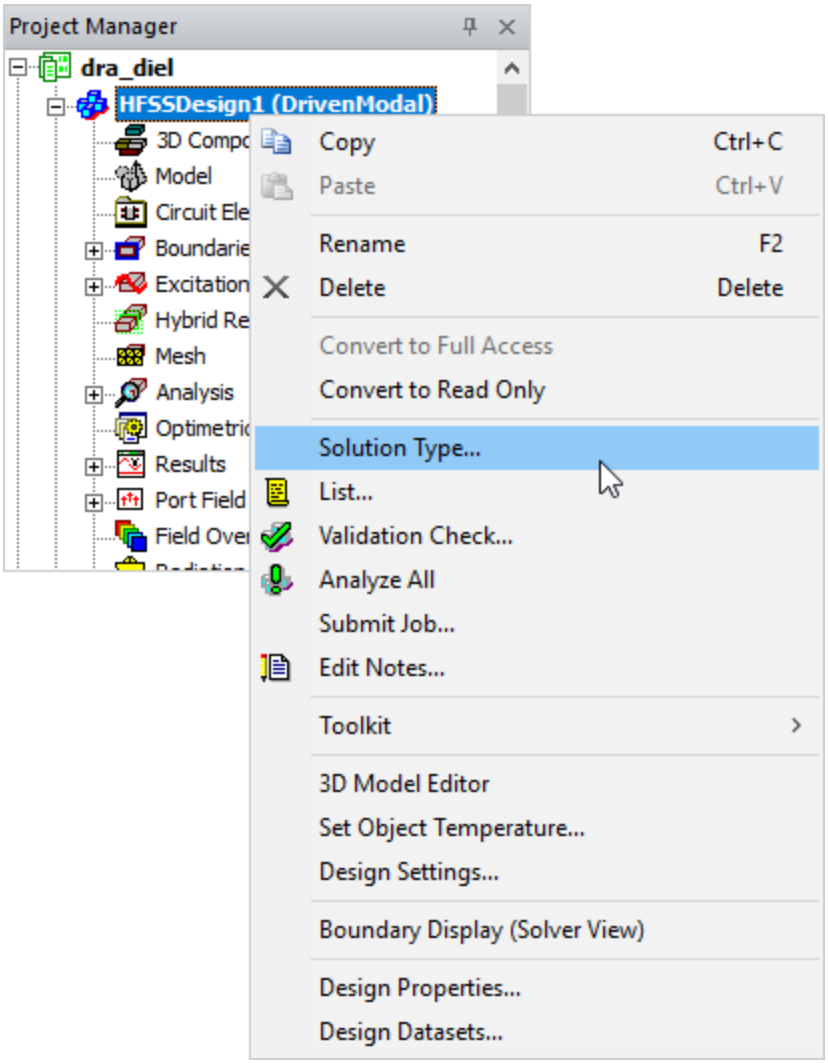
Each node, or item, in the project tree has a shortcut menu. For example, from the shortcut menu for the **Boundaries** icon, you can assign boundaries to selected objects; review

information for all the boundary assignments for the active design; remove all boundary assignments; show or hide a boundary's geometry, name, or vectors; change the priority of a previously assigned boundary; and use the PML Setup wizard to create a perfectly matched layer (PML) boundary.



Other nodes have shortcut menus appropriate for the context.

For example, the following figure shows the shortcut menu when you right-click the design.



Customizing Ansys Electronics Desktop Menus

Ansys Electronics Desktop comes with a default Top Menu Bar. You can modify the top menu bar by using customized User Interface (UI) setups to add, remove, rename, and relocate commands. You can also add commands that execute external scripts or configure right-click menus for folders and items in the Project Tree, which allows for extensive customization of the Ansys Electronics Desktop Menus.

Customized setups are implemented using named subfolders that contain XML files used to configure customized menu settings. The subfolders included with Ansys Electronics Desktop are:

- config/UI/ElectronicsDesktop/EM
- config/UI/ElectronicsDesktop/RF

- config/UI/ElectronicsDesktop/RF.0
- config/UI/ElectronicsDesktop/SI
- config/UI/ElectronicsDesktop/SI1.0
- config/UI/ElectronicsDesktop/SI2.0
- config/UI/ElectronicsDesktop/Twin Builder

Each customized UI type has its own subfolder. Within these subfolders, XML files hold the menu configurations as well as the right-click menus for that UI type.

Select Different UI Types

To change the menu display from the default UI to a different UI type:

- Click **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**.
- Choose the UI type that you wish to use from the **Custom Menu Set** drop-down menu and click **OK**.

To switch from a customized UI type to the default UI:

- Click **Tools** > **Revert To Default UI**.

Add a New Customized UI Type

A new folder needs to be added to the config/UI/ElectronicsDesktop folder for any new customized UI type. All XML files for this UI type must go in this folder. The new UI type will appear in the **Custom Menu Set** drop-down menu (**Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**).

For any UI type, Ansys Electronics Desktop displays the default UI menus for any products/contexts that are not in the xml files. If there is error processing any XML file, the default UI menu is displayed for that product. Check the message window for the names of problematic XML files and suggestions on how to fix them.

XML files are only processed once when you first switch to that UI type. After you make changes to any XML file, in order for it to be reprocessed, navigate to **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration** and reselect the UI type from the **Custom Menu Set** drop-down menu. Then click **OK**.

Names of the XML Files

Below is the list of XML files that can be placed in a folder for a new UI Type. If any of these XML files does not exist in this folder, the default menu setting displays for that product.

- 2D Extractor.xml – used for 2D Extractor projects.
- Circuit Design.xml – used for Circuit Design projects.
- Circuit Netlist.xml – used for Circuit Netlist projects.
- EMIT.xml – used for EMIT projects.
- HFSS 3D Layout Design.xml – used for HFSS 3D Layout Design projects.

- HFSS.xml – used for HFSS projects.
- HFSS-IE.xml – used for HFSS-IE projects.
- Icepak.xml – used for Icepak projects.
- Maxwell 2D.xml – used for Maxwell 2D projects.
- Maxwell 3D.xml – used for Maxwell 3D projects.
- Maxwell Circuit.xml – used for Maxwell Circuit projects.
- NoDesignUI.xml – used for menu settings when no project is selected.
- Q3D Extractor.xml – used for Q3D projects.
- RightClickMenu.xml – used for all right-click menu settings.
- Twin Builder.xml – used for Twin Builder projects.

Valid XML Elements and Attributes

1. Root Element is **DesignerMenu**:
    - **xmlns** – required attribute which needs to be set to the following:

      <DesignerMenu xmlns="http://www.ansys.com/uiConfigMenu">

    - **UseProjectWindowSelectionContext** – child of DesignerMenu, appears zero or one time.

      > **Note:**
      >
      > This is only meaningful if it is used in NoDesignUI.xml. If this is set to true, then clicking in the Project Window on the Project icon or the Definitions icon (or a subitem) will show “Project” context in NoDesignUI.xml instead of active design context. No setting or setting this to false makes this UI type behaves the same as the default behavior of Ansys Electronics Desktop. The menus and toolbars are always shown for the active design unless there’s no design at all.

    - **Context** – child of DesignerMenu. It appears at least one time and has a required “name” attribute. For details on setting the context name, see the “Context Name” section below.
2. Child elements of **Context**:
    - **TopMenu** – For RightClickMenu.xml, do not use this element. You can specify child elements of TopMenu listed below under Context. For any other xml files, you need at least one TopMenu child element for any Context.

      Child elements of TopMenu:

- **MenuName** – required, appears only one time.
- **popupMenu** – may appear one or multiple times; child elements are the same as those of Topmenu.
- **LeafMenu** – may appear one or multiple times.

  Child elements of **LeafMenu**:

  - **MenuName** – optional string.

    Pay attention to the character reference "&" in XML. In order for the name of the menu to be displayed as “Tools”, the XML syntax must be:

    <MenuName>&Tool</MenuName>

    “&“ is the character reference for ”&” in XML, while “&Tool” tells Ansys Electronics Desktop that Alt+T is the shortcut key for this menu.

  - **MenuID** – optional number.

    For a list of valid MenuIDs, please see ***Command IDs for Customizing AEDT Menus.xlsx*** under <*install_dir*>/v<*release_number*>/[Win64 or Linux64]/Help.

  - **ShowBitMap** – optional, “Yes” or “No” (default)
  - **Accelerator** – optional string (Example: Ctrl+N)

  > **Note:**
  >
  > To add a new LeafMenu, both “MenuName” and “MenuID” are required.

- **CustomMenu** – may appear one or multiple times; used to add a customized menu to run an external script (vbs or python).

  Child elements of **CustomMenu**:

  - **MenuName** – see LeafMenu for Usage
  - **ShowBitmap** – see LeafMenu for Usage
  - **Accelerator** – see LeafMenu for Usage
  - **ScriptPath** – required string; used to supply path to the script (use of $PROJECTDIR, $PERSONALLIB, $USERLIB, $SYSLIB variables is also allowed).

    Example:

    <ScriptPath>C:/Users/jwei/Python/HelloWorld.vbs</ScriptPath>

or

<ScriptPath>$PERSONALLIB/HelloWorld.py</ScriptPath>

> **Note:**
>
> To add a new CustomMenu, both MenuName and ScriptPath are required.

- **Separator** – may appear one or multiple times; has no child element.

3. All elements have an optional attribute of **action**.

   Valid values are:

   - **add** – can be applied to all elements other than “DesignerMenu” or “Context” . This element and all its child elements will be added to its parent. An optional attribute of “position” can be followed here to specify the position of this newly added menu. Position starts from 1. If no position is specified, this menu will be appended to its parent menu.
   - **useDefault** – default menus will be used for this menu(matched by MenuName or MenuId). Any of it’s child menu will be processed according to its action setting.
   - **delete** – can be applied to all elements other than “DesignerMenu” or “Context” . This menu (matched by MenuName or MenuID) will be deleted from its parent.

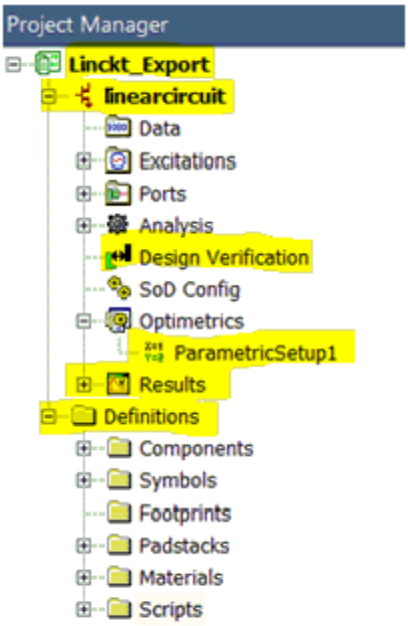
Note:

- For “DesignerMenu” and “Context”, only no action setting or “useDefault” is valid.
- The “DesignerMenu” action setting is significant because it is how you specify whether you want to construct your own menus or modify existing default menus. If this is set to “useDefault”, default menu settings are used for any menus not listed in your XML files. If no action is set, only the “File”, “Window” and “Help” menus display; no other default menus display. Menus listed in the XML file display after the “File” menu and before the “Window” menu. In this case, no action is needed for any other elements in the XML file. If any action is specified for any element, you will receive an error message and a default menu will be displayed for that context. A “Revert To Default” menu will be appended to the “Tools” menu. If no “Tools” menu is specified as a TopMenu, a “Tools” menu is created before the “Window” menu with one “Revert To Default” menu item.
- For any other elements, a missing action setting means that the parent action will be used.
- If any element’s action is set to “add”, then no action is needed for any of its child elements. If any action is specified, it will be ignored. This menu and all of its child menus will be added to the default menu.
- If any TopMenu or pop-upMenu’s action is set to “delete”, you don’t need to list any of its child menus. The menu will be deleted with all child menus.

Context Name

1. The “name” attribute specifies the name of the context for this product. You can have an "All" context if you set the action of DesignerMenu to UseDefault to specify a menu setting that you want to apply to all contexts for that product.
2. Valid context name for different xml files:
    - 2D Extractor.xml — “All”, “3d modeler”, “report2d”
    - CircuitDesign.xml — “All”, "SchematicEditor", “Layout”, “Netlist”, “report2d”
    - Circuit Netlist.xml —“All”, “Design”, “Netlist”
    - HFSS.xml — “All”, “3d modeler”, “report2d”
    - HFSS 3D Layout Design.xml — “All”, “Design”, “Layout”, “Layout3D Editor”, “report2d”
    - HFSS-IE.xml — “All”, “3d modeler”, “report2d”
    - NoDesignUI.xml — “All”, “No Context”, “project”, “FilterDesign”
    - Q3D Extractor.xml — “All”, “3d modeler”, “report2d”

- RightClickMenu.xml — the following figure illustrates valid context names and how they display in Ansys Electronics Desktop.



  - **Project Folder** – "Project"; in the above figure, “Project” represents the “Linckt_Export” project folder.
  - **Design Instance Folder** – “Circuit Design”, “HFSS 3D Layout Design”, “HFSS”, “Q3D Extractor”, “Circuit Netlist”, “HFSS-IE”, “2D Extractor”
  - **Definitions Folder** – “Definitions”; in the above figure, the Definitions folder is highlighted.
  - For any Folder or Item under the Design Instance or Definitions folder, use the “/” notation to specify the path to the Folder or Item for context name. In the above figure: Use “Circuit Design/Results” for the Results folder. Use “Circuit Design/Design Verification” for the Design Verification folder.
    - For any folder or item under this, use the context name for the folder appended with “/Item”. In the above figure, “Circuit Design/Optimetrics/Item” represents ParametricSetup1.

> **Note:**
>
> For the Report folder under Results, please use the context name for the Results folder appended with "Report"; for example: "Circuit Design/Results/Report".
>
> For the Trace folder under Report, please use the context name for the Report appended with "Trace"; for example: "Circuit Design/Results/Report/Trace"

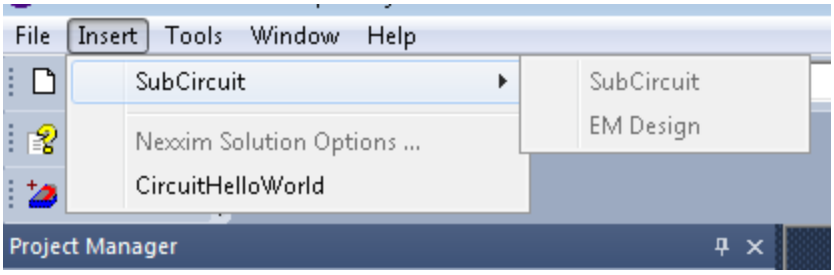
Sample XML Files

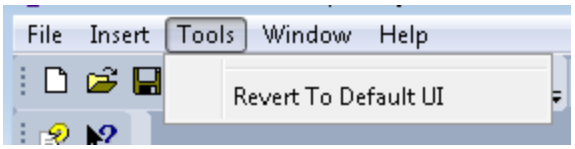
1. **Circuit Netlist.xml Sample**

```
<DesignerMenu xmlns="http://www.ansys.com/uiConfigMenu">
   <Context name="Design">
      <TopMenu>
         <MenuName>Insert</MenuName>
         <pop-upMenu>
            <MenuName>S&amp;ubCircuit</MenuName>
            <LeafMenu>
               <MenuName>&amp;SubCircuit</MenuName>
               <MenuID>55500</MenuID>
               <ShowBitMap>No</ShowBitMap>
               <Accelerator></Accelerator>
            </LeafMenu>
            <LeafMenu>
               <MenuName>&amp;EM Design</MenuName>
               <MenuID>55502</MenuID>
               <ShowBitMap>No</ShowBitMap>
               <Accelerator></Accelerator>
            </LeafMenu>
         </pop-upMenu>
         <Separator></Separator>
         <LeafMenu>
            <MenuName>Nexxim Solution &amp;Options ...</MenuName>

            <MenuID>38460</MenuID>
            <ShowBitMap>No</ShowBitMap>
            <Accelerator></Accelerator>
         </LeafMenu>
         <CustomMenu>
            <MenuName>CircuitHelloWorld</MenuName>
            <ScriptPath>$USERLIB/HelloWorld.vbs</ScriptPath>
            <ShowBitMap>No</ShowBitMap>
         </CustomMenu>
      </TopMenu>
```

```
    </Context>
</DesignerMenu>
```

Menus created by processing the above sample, Circuit Netlist.xml



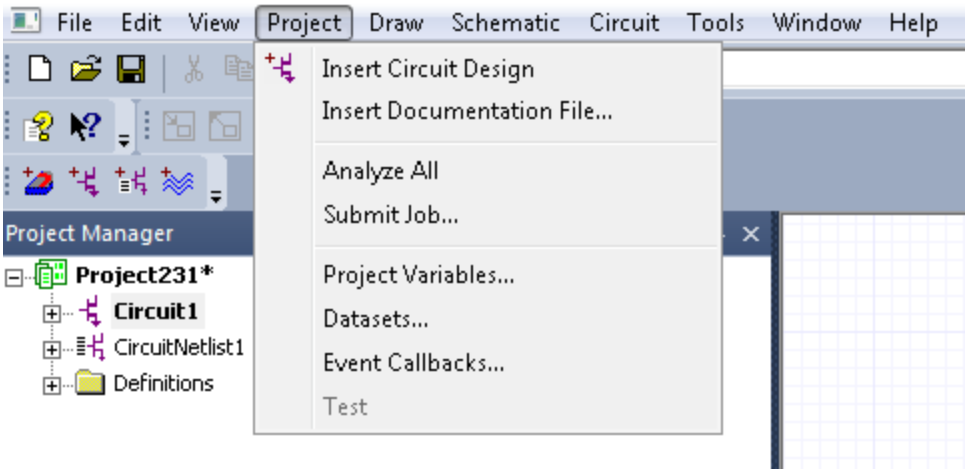


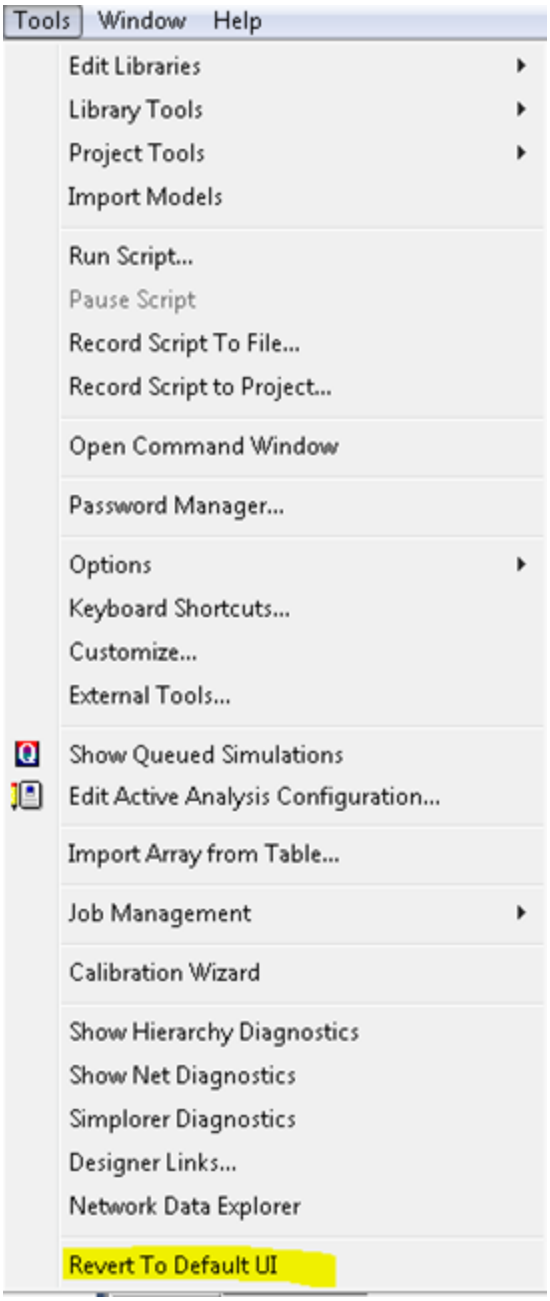
2. **Circuit Design.xml Sample**

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<DesignerMenu xmlns="http://www.ansys.com/uiConfigMenu"
action="useDefault">
  <UseProjectWindowSelectionContext>false</UseProjectWindowSelectionC
  ontext>
  <Context name="All">
    <TopMenu>
          <MenuName>&amp;amp;Project</MenuName>
      <LeafMenu action="delete">
                <MenuName>Insert HFSS Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
                <MenuName>Insert HFSS 3D &amp;amp;Layout
        Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
                <MenuName>Insert HFSS-IE Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
                <MenuName>Insert Q3D Extractor Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
```

```
                <MenuName>Insert 2D Extractor Design</MenuName>
        </LeafMenu>
        <LeafMenu action="delete">
                <MenuName>Insert Circuit &amp;Netlist</MenuName>
        </LeafMenu>
        <LeafMenu action="add">
                <MenuName>Test</MenuName>
                <MenuID>3333</MenuID>
        </LeafMenu>
      </TopMenu>
   </Context>
</DesignerMenu>
```

Menus created by processing the above XML sample, Circuit Design.xml



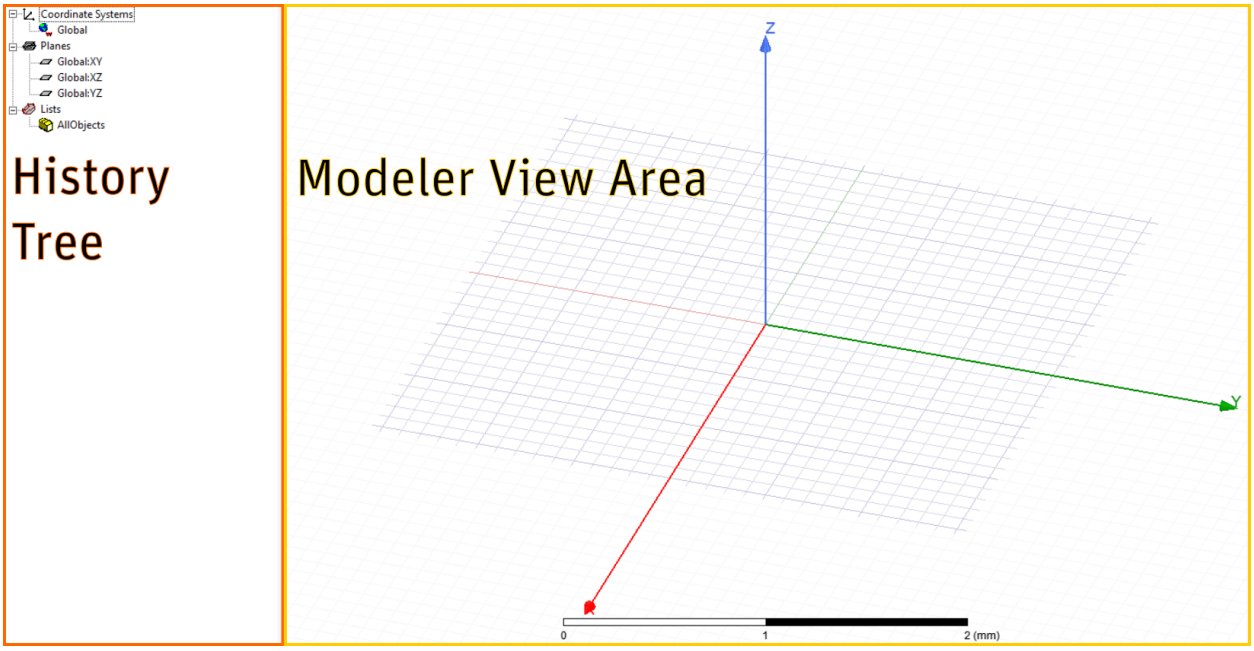


Ansys Electronics Desktop Design Area

The Design Area of the desktop can display one or more editor windows and report windows, depending on the type of designs you create or load. See the Desktop Windows topic for ways to manipulate the windows in the Design Area.

3D Modeler Window

The **3D Modeler** window is the area where you create the model geometry. It appears to the right of the **Project Manager** window after you insert an Ansys Electronics Desktop design into a project. The **3D Modeler** window consists of the modeler view area, or grid, and the History Tree. The two areas are highlighted in the following figure:



You can open a new 3D Modeler window in one of two ways:

- Insert a new Ansys Electronics Desktop design into the current project, or
- Double-click an Ansys Electronics Desktop 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 **3D Modeler** window. You can create 3D objects using Ansys Electronics Desktop's Draw menu commands, or you can draw 1D and 2D objects and manipulate them to create 3D objects. See: Drawing a Model.

You can modify the view of objects in the **3D Modeler** window without changing their actual dimensions or positions. Besides menu and icon commands, you can also use **Alt+click**, mouse buttons, and keyboard shortcuts. See: Modifying the Model View.

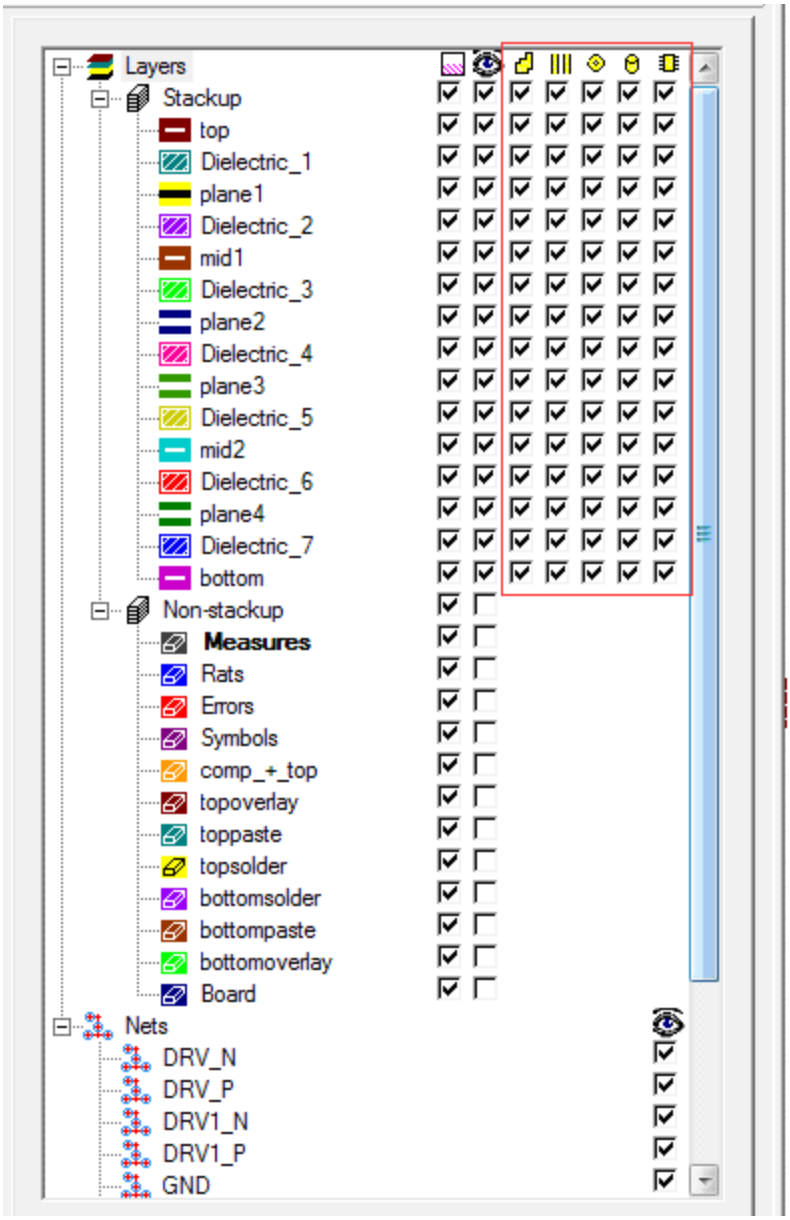
Layout Window

The Layout Window is a dockable Ansys Electronics Desktop window that can be resized and relocated, and can be used to view and configure various layout settings. Use the right-click

menu of the Layout Window to configure the following:

- **Set Active** – makes the current layer the active layer.
- **Show This Layer Only** – makes the current layer the only visible layer.
- **Show All Dielectrics** – makes all Dielectric layers visible.
- **Hide All Dielectrics** – makes all Dielectric layers invisible.
- **Show All Signals** – makes all Signal layers visible.
- **Hide All Signals** – makes all Signal layers invisible.

Use the **Layout** window to alter the following controls for setting visibility by layout-object type.



The following controls are available:

Controls the visibility of shapes

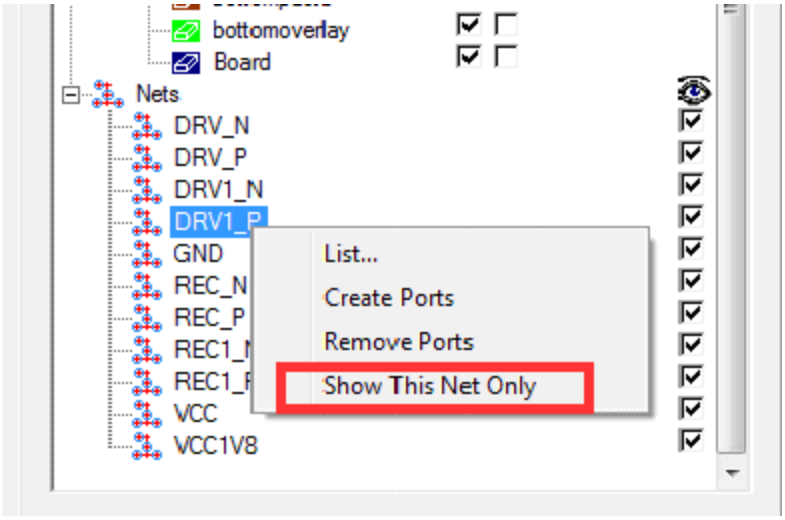
Controls the visibility of lines (paths)

Controls the visibility of pads

Controls the visibility of holes

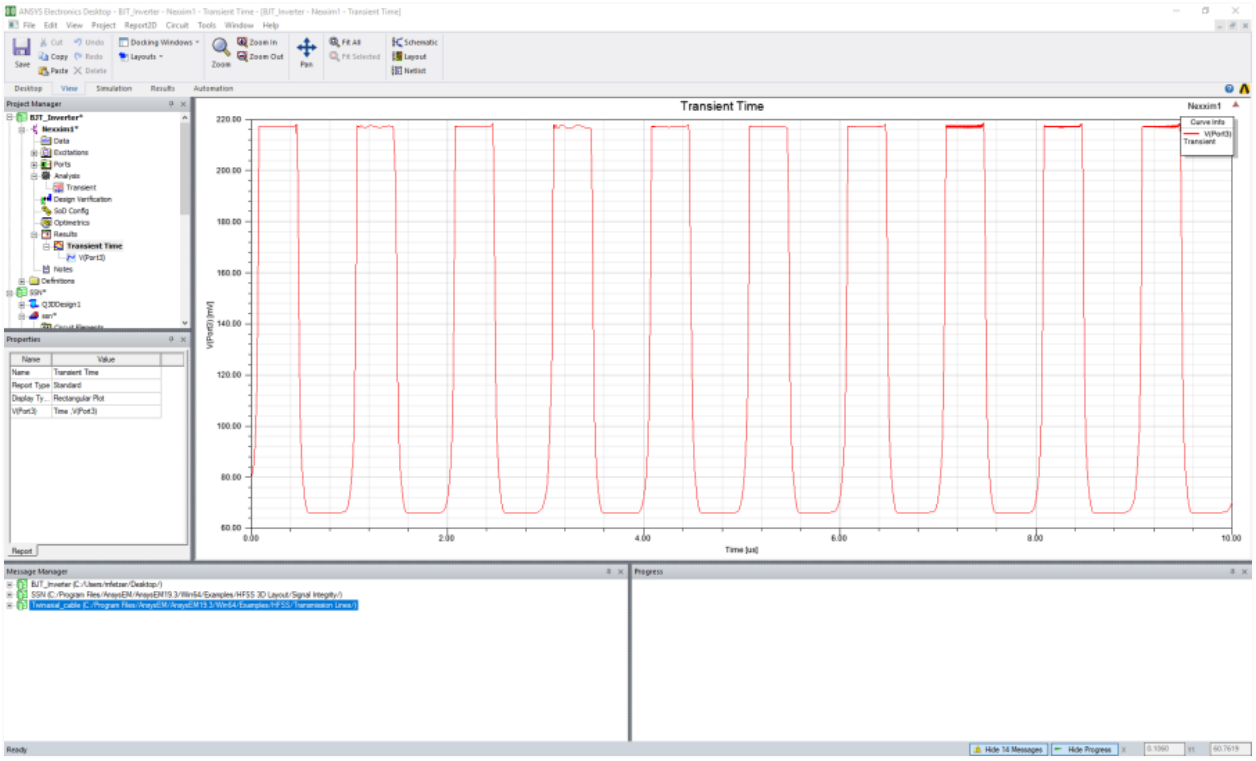
Controls the visibility of components

You can also turn visibility off for all nets but the selected net by right-clicking and selecting **Show This Net Only**.



Report Window

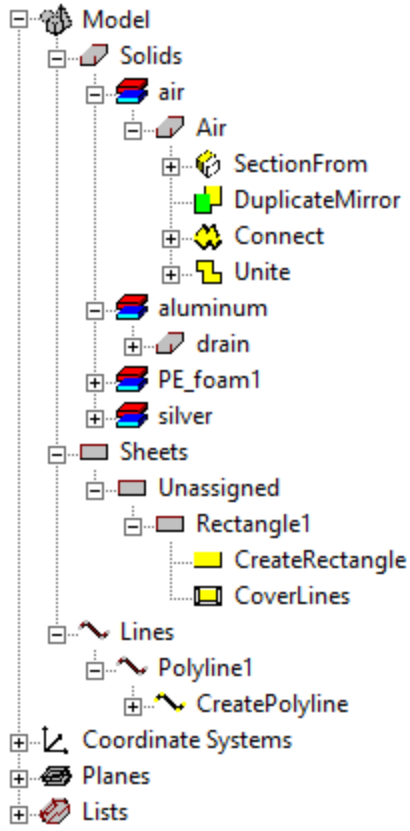
When a design has been successfully simulated, you can generate a report of results in a wide variety of forms, including XY graphs, polar graphs, 3D graphs, Smith charts, and data tables. Various attributes of each can be customized to your liking. The following shows a 2D report window in Electronics Desktop:



For more information, see Post Processing and Generating Reports.

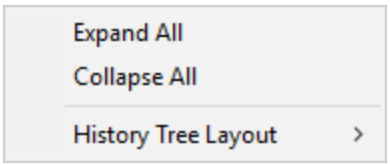
Ansys Electronics Desktop History Tree

The history tree in the **3D Modeler** window lists the active model's structure and grid details.



Expand or collapse items in the tree using the **+** and **-** boxes.

Expand or collapse all items in a category by right-clicking to open a shortcut menu:



> **Note:**
>
> Some objects have more extensive shortcut menus than others, but all of them have these three options.

Selecting any item in the History Tree also selects it in the design area and populates the **Properties** window with its properties.

You can perform the following tasks with the History Tree:

- Organize the History Tree
- Control the View of Objects in the History Tree
- Use Group Commands for Modeler Objects
- Select Objects in the History Tree
- View and Edit Commands on History Tree Objects

The History Tree can contain the following model details:

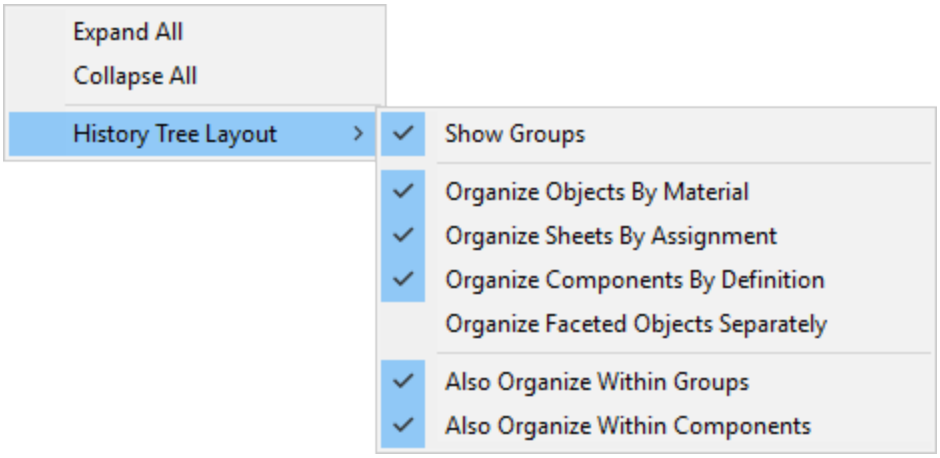
Invalid	Lists all invalid objects.
3D Component Names	Lists any 3D components added to the design. The History Tree lists their Solids, Coordinate Systems, and Planes separately from objects drawn in the modeler or imported to the modeler.
Solids	Lists the model's solid objects and a history of the commands carried out on each object.
Sheets	Lists the sheets in the model 3D design area.
Lines	Lists all line objects included in the active model. See Drawing a Straight Line for information on how to draw a line object.
Points	Lists all point objects included in the active model. See Drawing a Point for information on how to draw a point object.
Groups	Lists folders for Groups you have created or imported. See Group Commands for Modeler Objects.
Coordinate Systems	Lists all the coordinate systems for the active model. See Setting Coordinate Systems for more information on this model detail.
Planes	Lists 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	Lists the object or face lists for the active model. By default, a list called “AllObjects” appears. You can add a specific group of objects or surfaces for field plot or for calculation. See: Creating an Object List and Creating a Face List.

Note:

While objects created in Q3D Extractor can always be classed in the History Tree as a solid, sheet, or wire, some imported objects may have mixture of these. Q3D Extractor places such objects in an Unclassified folder.

Organizing the History Tree

To organize and sort objects in the History Tree, right-click any object in the history tree to display the shortcut menu and select **History Tree Layout**.



The shortcut menu contains the following commands:

- **Show Groups**
- **Organize Objects By Material:** This option is applicable to 3D objects and also to sheet objects that have the *Shell Element* attribute selected and a material selection specified (Mechanical - Thermal designs only). Imported solid objects that have no material specified yet are listed in a group named *Unassigned*. Non-model objects are listed in a *Non Model* group.
- **Organize Sheets By Assignment:** This option is applicable to sheet objects only (including Mechanical – Thermal shell objects with a boundary or excitation assigned). Sheet objects are grouped according to the boundary or excitation assigned. When a sheet is assigned both a boundary and an excitation, it is grouped according to the last assigned excitation (that is, the group name is the last assigned excitation). When multiple boundaries are assigned to a sheet, the sheet is grouped according to the boundary with the highest priority, and that boundary is used as the group name. The boundary priority depends on the design type, as follows:
  - **HFSS and Maxwell:** Priority is specified by the user.
  - **Icepak, Mechanical, and Q3D:** The last created boundary has the highest priority
- **Organize Components By Definition**
- **Organize Faceted Objects Separately**
- **Also Organize Within Groups**
- **Also Organize Within Components**

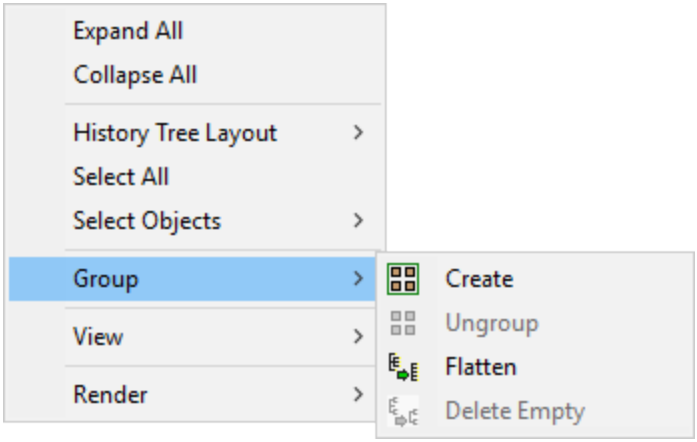
Controlling the View of Objects in the History Tree

To control the view and visibility of an object such as a box or PML, right-click an object in the history tree to display the shortcut menu and select **View**. The shortcut 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**

Group Commands for Modeler Objects

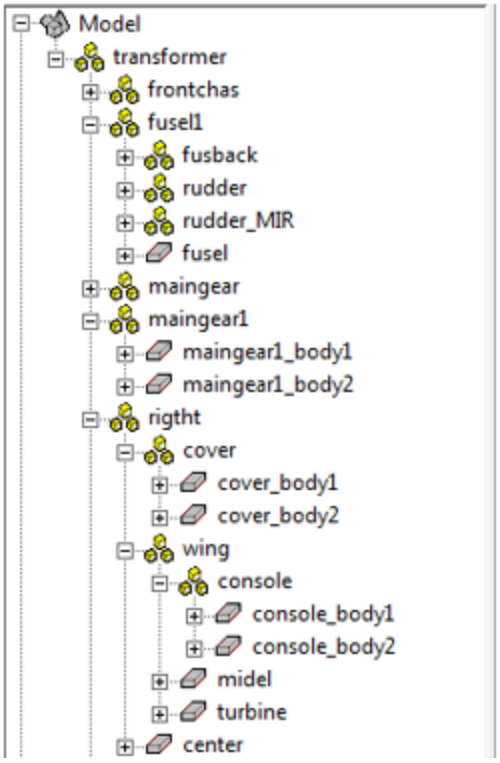
The 3D Modeler allows you to group objects in the History tree. Besides predefined folders under **History Tree Layout** for solids, sheets, and material assignments, a set of **Group** commands allows you to create groups, ungroup objects, flatten a selected group's hierarchy, and delete empty groups. Groups are not compatible with 3D components, although, you can organize 3D components by definition in the History tree.



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 history tree. They do not affect 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 following important distinctions 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. Delete of an object in one group could cause object in another group to be also deleted.

Accessing Group Commands

You can access Group commands in several ways: via **Modeler > Group**, via the right-click shortcut menu in the History tree, via the right-click shortcut menu in the Modeler window, and via the Group icons on the Model tab.

Group Command Descriptions

Group > **Create** works with or without a selection. If the History Tree Layout>Show Groups command is disabled, a dialog box reports that groups are currently hidden, and that if you OK to continue, groups will be shown. If there is no selection of objects for new group, an empty group is created under Model. If there are selections, 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 the 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** is enabled when you select a group. You can select multiple groups at different levels. All of the selected group's contents (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. If the History Tree Layout > Show Groups command is disabled, a dialog box reports that groups are currently hidden, and that if you OK to continue, groups will be shown.

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. If the History Tree Layout > Show Groups command is disabled, a dialog box reports that groups are currently hidden, and that if you click **OK** to continue, groups will be shown.

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 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

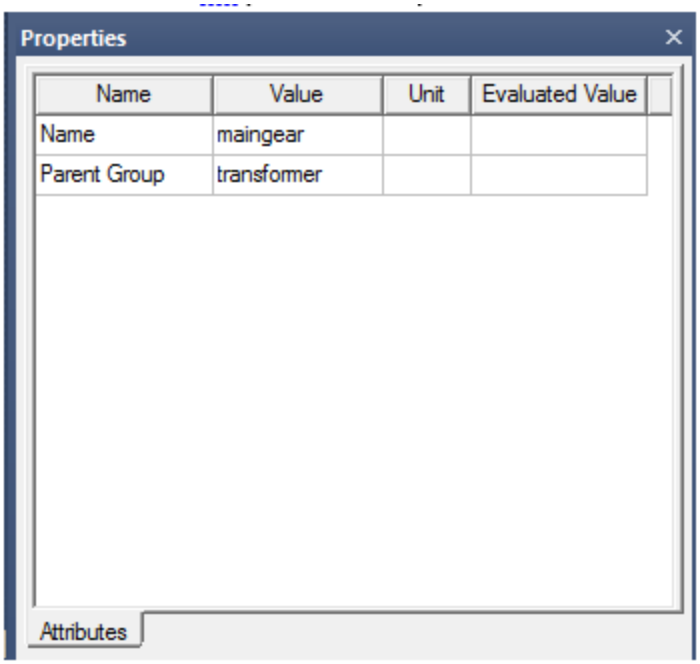
You can select groups to enable duplicate operations. Afterward, the History tree shows a duplicate operation under every object of that group. Newly created objects appear under a new group. There will be new group for every instance of duplicate.

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.

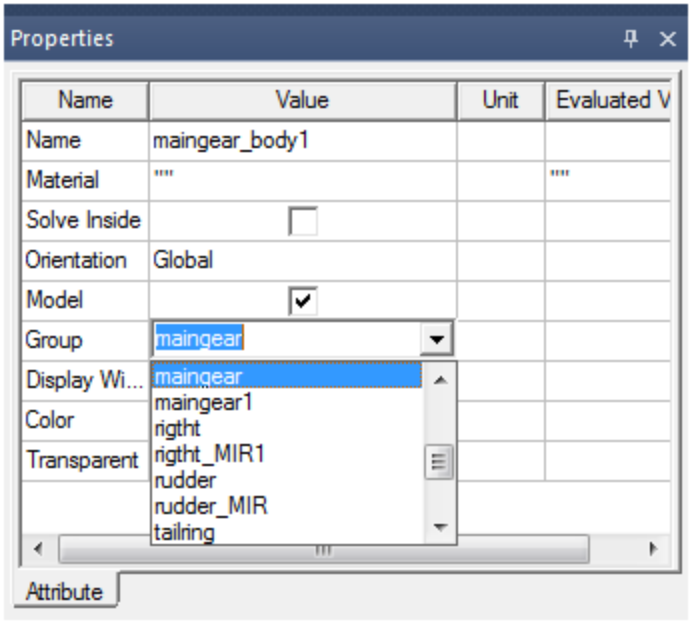
Group Properties

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.



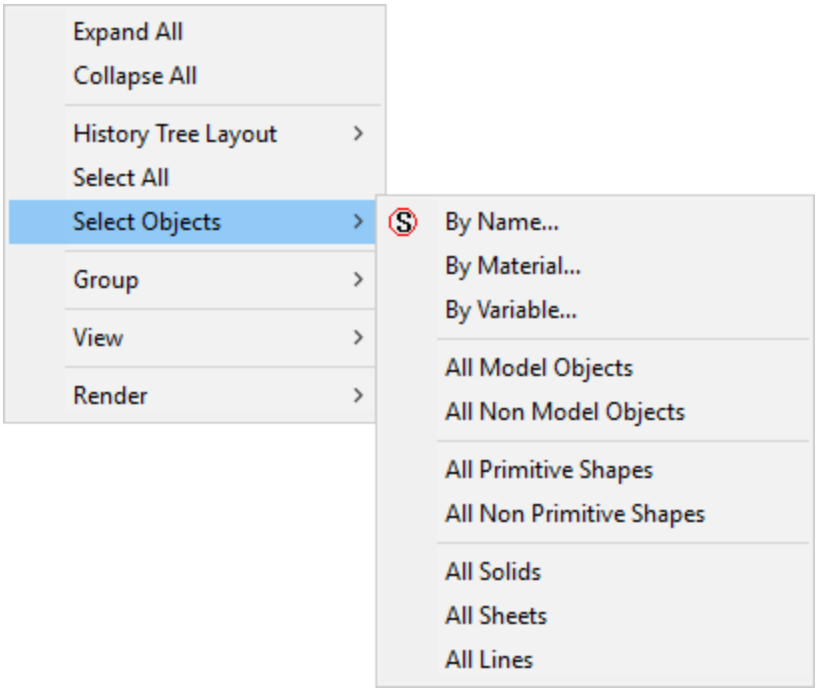
Group assignment could be changed in Object's property window

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 objects of interest by name or material because the object is inside or behind others.

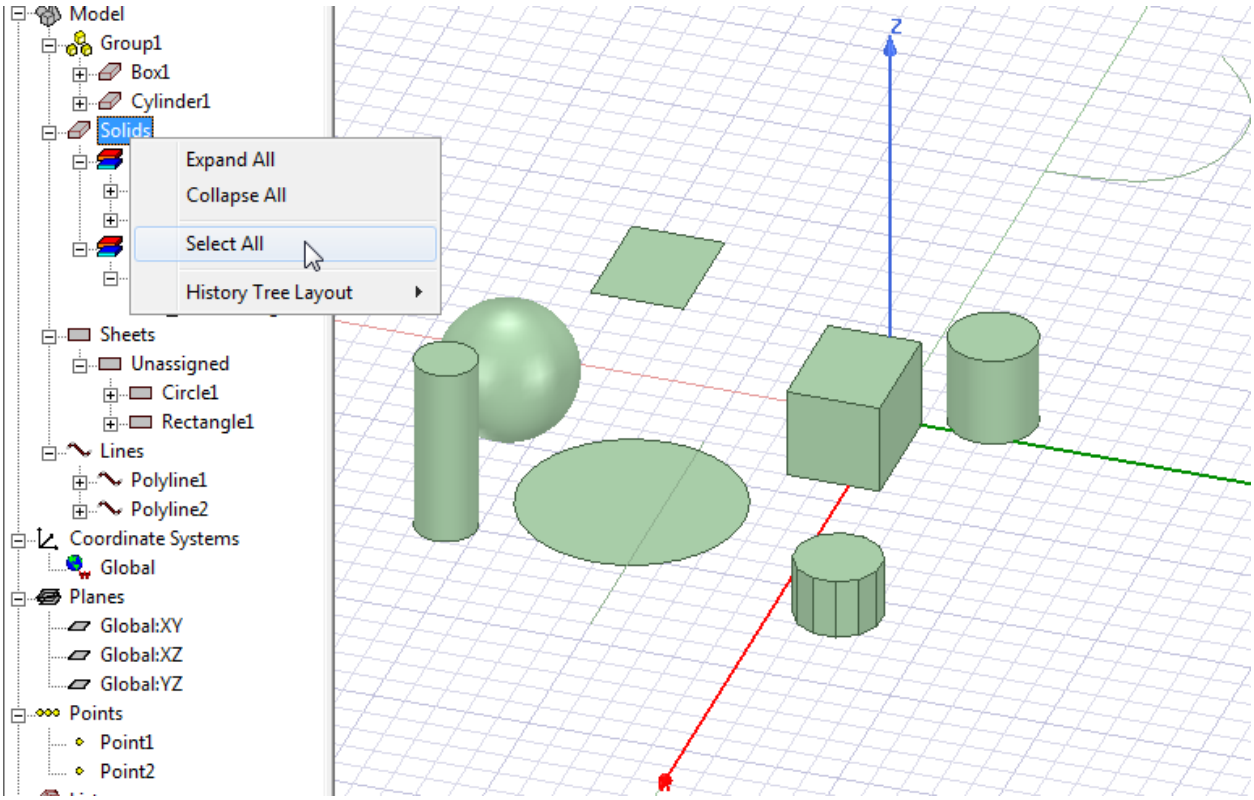
Ctrl+click to make multiple selections in the History Tree. You can select a range of objects by clicking the first and using **Shift+click** to select all in the range. You can also click and drag the mouse to make rubber band selections. Only visible objects are selected. That is, if the hierarchy is closed under the selection, any operand parts are ignored and do not interfere with cut and paste operations.

Right-click **Model** in the History Tree to see additional selection options:



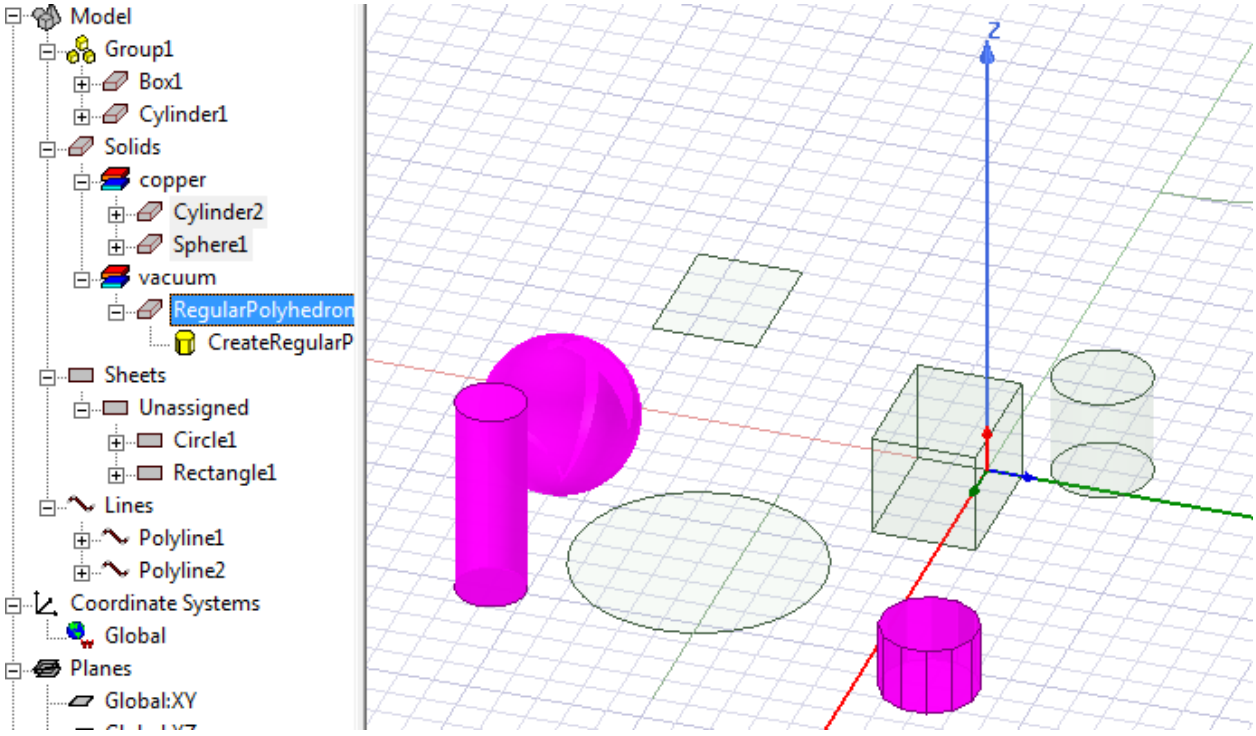
You can right-click any History Tree object or grouping to display a shortcut menu with a **Select All** option. This lets you select all non-grouped solids, sheets, lines, non-model objects, or all unclassified objects, as well as all non-grouped objects assigned to specific materials.

For example, right-click 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.

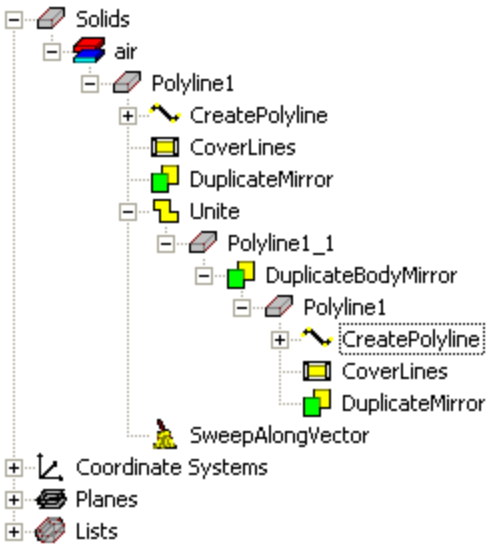


The right-click for Model and **Select All** does select the members of Group1 because they are part of the model.

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.

Viewing and Editing Commands on History Tree Objects

The history tree also lists the history of all commands carried out on 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 project:

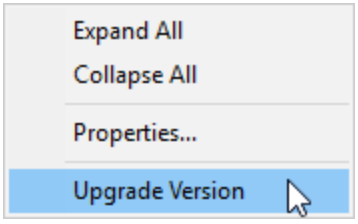


Selecting a command in the history highlights the object in the Modeler window and shows that object's properties (if available) in the docked **Properties** window. You can look at the fields in the **Properties** window to see any editable fields for that command, such as coordinate system, line type,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

For some commands, such as **Edit** > **Arrange** > **Move** or boolean operations, selecting them in the history tree enables the delete icon (**X**) on the toolbar 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.

Upgrading Version in the History Tree

Right-click an operation in the History Tree to see the **Upgrade Version** command.

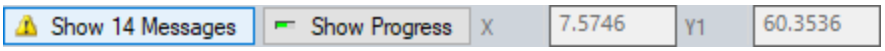


By default, the Modeler opens legacy projects using the modeling kernel version under which the project was saved, so that users don't see side effects from slight topology changes between versions. The **Upgrade Version** operation forces Electronics Desktop to use the latest modeling

kernel. Typically, this command would only be used when a needed fix is available for the newest version.

Ansys Electronics Desktop Status Bar

The status bar is located at the bottom of the application window. By default, it contains buttons to Show or Hide the Message window and Progress window.

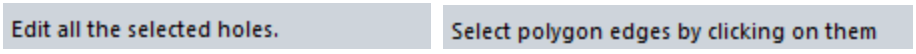


Note that the Show Messages button indicates the number of messages.

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

The status bar also displays helpful information about the current selection or command.

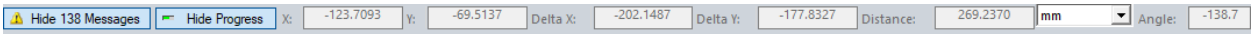
Directions for inputs appear on the left side of the status bar:



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

- **X**, **Y**, and **Z** coordinate boxes
- A drop-down menu for entering absolute, relative, cartesian, cylindrical, or spherical coordinates
- The model's units of measurement

These coordinate boxes and drop-down menus appear on the right side of the status bar:



To toggle status bar display:

- Click **View** > **Status Bar**.

A check mark next to this command indicates that the status bar is visible.

Keyboard Shortcuts

The following keyboard shortcuts apply to Q3D Extractor in general:

F1	Help (context-sensitive)
Ctrl + F4	Close window
Alt + F4	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

See the subtopics in this Help branch for additional information concerning shortcuts and view navigation.

To customize the shortcut assignments, use **Tools** > **Keyboard Shortcuts**. Not all shortcuts are customizable.

Desktop Shortcuts

Modifier + key	Hold down the modifier, such as Shift or Ctrl, and press the key.

The following Ansys Electronics Desktop shortcut key combinations are available at any time:

Ctrl + N	New
Ctrl + O	Open
Ctrl + P	Print
Ctrl + S	Save
Ctrl + 0	Cascade windows
Ctrl + 1	Tile windows horizontally
Ctrl + 2	Tile windows vertically

Delete	Delete
F1	Open help

3D Modeler Window Shortcuts

The following shortcut key combinations are available when the Modeler window is active. View navigation shortcuts are also applicable to 3D Plot windows. The table below is organized into four sections, with the shortcuts separated according to the following categories:

- Commands and options used for the selection of entities
- Choosing the entry mode to use when creating geometry
- View navigation (rotating, zooming, panning, choosing a view orientation)
- Choosing the geometry rendering mode (wireframe or shaded)

> **Note:**
>
> In the following table, **LMB** is an abbreviation for *Left Mouse Button*, and **MMB** means *Middle Mouse Button*.
>
> **Wheel** refers to rotation of the mouse wheel that is integrated with the middle button.

Entity Selection Commands and Options:	
B	Select face or object **behind** current selection
F	Select **faces** mode
O	Select **objects** mode
E	Select **edges** mode
V	Select **vertices** mode
M	**Multi-select** mode
Ctrl + A	Select **all** visible objects
Ctrl + Shift + A	Deselect all objects
Geometry Properties Entry Mode:	
F3	Switch to the **point** entry mode (using the mouse or coordinate text boxes to draw objects). Switching takes effect only after the drawing of an object has started.
F4	Switch to the **dialog box** entry mode (draw objects by specifying properties in the Command and Attribute tab of a dialog box). Switching takes effect only after the drawing of an object has started.
View Navigation:	
Ctrl + D	**Fit** view

Ctrl + '+'	**Zoom in** (screen center)
Ctrl + '-'	**Zoom out** (screen center)
Alt + double-click	Change the view orientation to one of nine predefined views depending on where you double-click within the Modeler window. See Changing the Model View with Alt+Double-Click Areas for more information.
Alt + *arrow keys*	**Rotate** the view about a vertical or horizontal axis. Each press of an arrow key rotates the view 2.5°.
Ctrl + *arrow keys*	**Pan** the view vertically or horizontally. Each press of an arrow key pans the view 10 screen pixels.
MMB + drag (no modifier)	Freely **rotate** the view orientation (except when clicking and dragging near the window border). When clicking near the window border, rotation is constrained about the screen-normal axis.
MMB + X + drag	Constrained **rotation** of the view orientation about the **X** axis
MMB + Y + drag	Constrained **rotation** of the view orientation about the **Y** axis
MMB + Z + drag	Constrained **rotation** of the view orientation about the **Z** axis
Wheel	Rotate the mouse wheel to **zoom** in and out. Rotate the top of the wheel forward (away from the mouse) to zoom **in** closer, and rotate it rearward to zoom **out**.
Shift + MMB + drag	**Zoom** in or out. Moving the cursor upward zooms in closer. Moving it downward zooms out. (* see note below)
Ctrl + MMB + drag	**Pan** the view (* see note below)
(*) **Note:** The listed actions for **Shift** and **Ctrl** keys in combination with the **MMB** are the view navigation assignments for 2025 R1. Simultaneously, legacy hot keys using **Shift** or **Alt + Shift** in combination with the **LMB** for these functions continue to be supported by default. Optionally, you can choose to disable the legacy shortcuts that were applicable to 2023 R2 and earlier versions. See Choosing the View Navigation Options for more information.	
Geometry Rendering Modes:	
F6	Render the geometry as wireframes (no shading of faces)
F7	Render the geometry with smooth, shaded faces

To customize shortcut assignments, use **Tools** > **Keyboard Shortcuts**. Not all shortcuts are customizable.

Report Shortcuts

Modifier + key	Hold down the modifier (such as Shift or Ctrl) and press the key.

The following shortcut key combinations are available when a Report window is open:

Ctrl + A	Select all traces
Ctrl + C	Copy selected trace
Ctrl + V	Paste
Ctrl + X	Cut selected trace
Ctrl + Y	Redo insert report
Ctrl + Z	Undo insert report
Delete	Delete

Choosing the View Navigation Options

New mouse button and modifier key assignments (hotkeys) have been added for navigating the model view (rotate, pan, and zoom functions). Simultaneously, legacy hotkeys for these functions continue to be supported by default. Optionally, you can choose to disable the legacy view navigation shortcuts that were applicable to version 2023 R2 and earlier, as follows:

1. Access the *Options* dialog box using one of the following two methods:
   - From the menu bar, click **Tools** > **Options** > **General Options**.
   - From the **Desktop** ribbon tab, click  **General Options**.
2. In the tree at the left side of the dialog box, select **General** > **User Interface**. Then:
   - To use either the legacy or current view navigation buttons and hotkey combinations, select **Enable Legacy View Navigation**. This option is selected by default.
   - To use only the current view navigation behavior and disable legacy button/hotkey support, ensure that **Enable Legacy View Navigation** is cleared.
3. Click **OK**.

   The navigation option change becomes effective immediately (no program restart required).

Current vs. Legacy View Navigation

The following table summarizes changes in mouse-button and hotkey assignments between the current scheme (version 2024 R1 and newer) and the legacy scheme. "*Legacy*" assignments are applicable to version 2023 R2 or earlier and when the *Enable Legacy View Navigation* option is selected in newer versions):

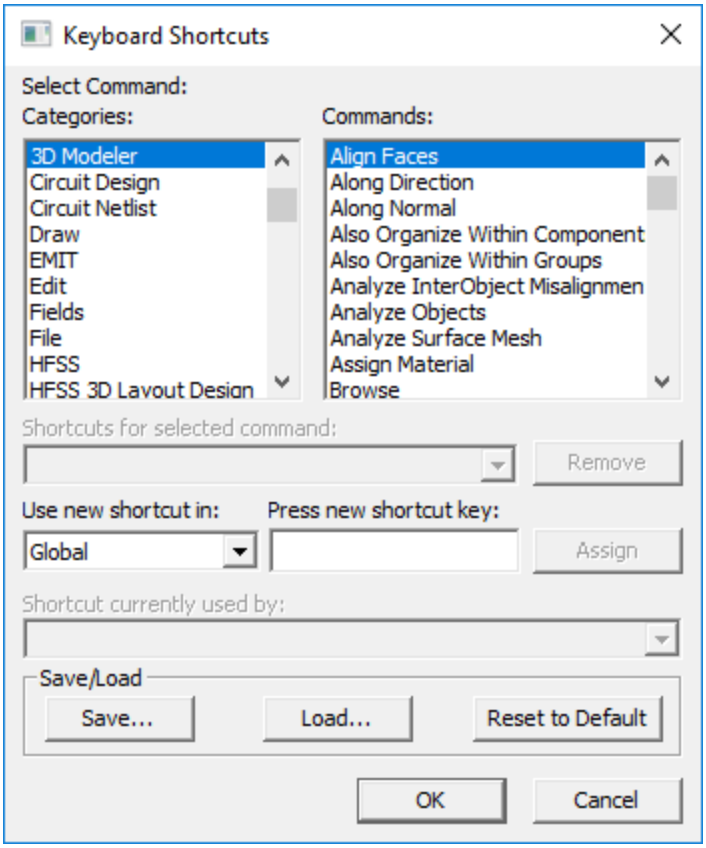
> **Note:**
>
> The *"Current"* assignments are available whether or not the *Enable Legacy View Navigation* option is selected:

Function	Key / Mouse Button Assignment	
	Current	**Legacy**
Pan (all contexts)	Ctrl + MMB + drag or Ctrl + *arrow keys*	Shift + LMB + drag or Ctrl + *arrow keys*
Rotate (3D contexts)	MMB + drag or Alt + *arrow keys*	MMB + drag or Alt + LMB + drag or Alt + *arrow keys*
Zoom (all contexts)	Wheel or Shift + MMB + drag or Ctrl + "+", Ctrl + "-"	Wheel or Alt + Shift + LMB + drag or Ctrl + "+", Ctrl + "-"

- "**LMB**" = Left Mouse Button
- "**MMB**" = Middle Mouse Button
- "**Wheel**" refers to rotation of the mouse wheel.
- In some 2D contexts, where rotation is not applicable, you can use **MMB + drag** to pan the view for either scheme.
- For **Alt + *arrow keys***, rotation is about a vertical or horizontal axis, regardless of the model view orientation.
- For zooming, if using the "=\+" key, instead of the "+" key on the numeric keypad, do **not** press Shift.

Custom Keyboard Shortcuts

Click **Tools** > **Keyboard Shortcuts** to open the **Keyboard Shortcuts** window. Here, you can view existing assignments, create new shortcuts, and save or load assignment files.



Selecting a **Command Category** populates the **Commands** list with the available commands for that category. If the command has an assigned shortcut, it is displayed in the **Shortcuts for the 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 unavailable.

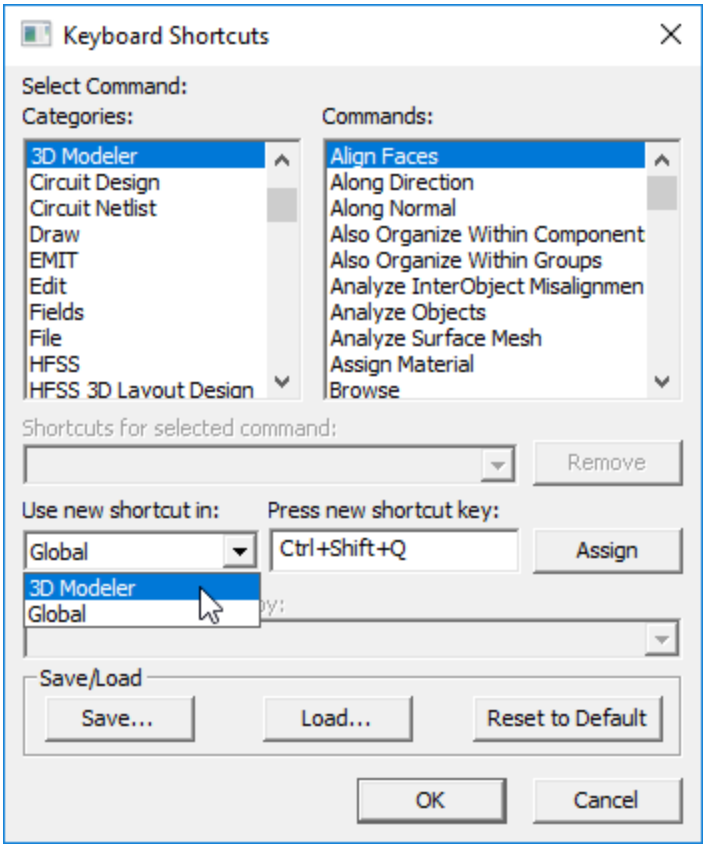
To create a new shortcut key:

1. Select the **Category** and **Command**.
2. If you want to 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 **Assign** button becomes available. If you combine keystrokes, these are displayed with a plus sign (+) between them. For example, Ctrl+P or Alt+O.

   If the shortcut you select is currently in use, that information displays in the **Shortcut currently used by** field.

4. The **Use new shortcut in** drop-down list displays **Global** by default, which means that the shortcut will apply to all applicable contexts. If a limited context exists, the menu offers a selection.



5. When you have made your desired assignments, you can save them to a named file.

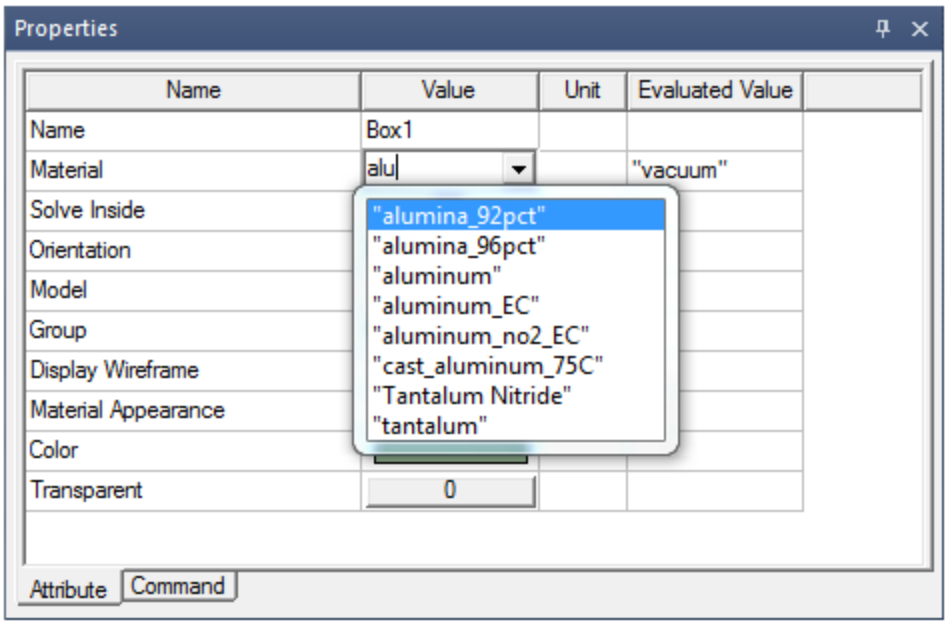
   Click **Save...** to save the assignments to your desired location in Ansoft Keyboard Shortcut (*.aks) format.

   If you have an existing *.aks file, use the **Load...** button to locate and select the file.

6. Click **OK** to save the current settings, or **Reset to Default**.

Auto-Complete for Variables and Properties in Electronics Desktop

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

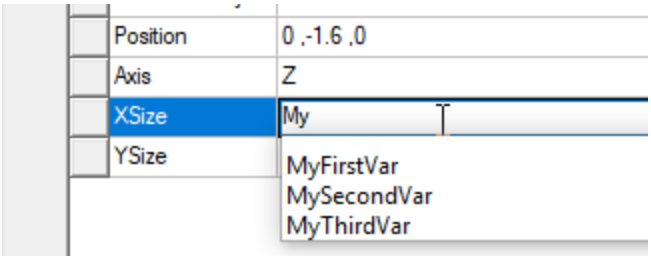


Using Auto-Complete

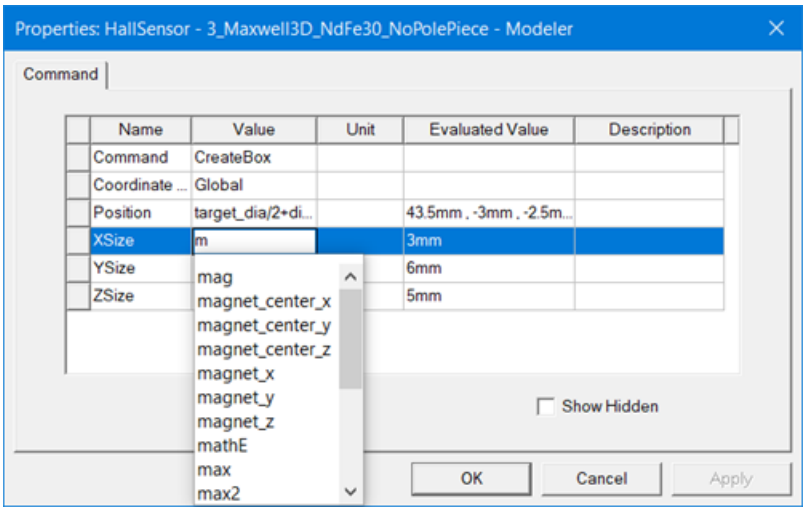
Certain commonly used text fields have auto-complete configured. When you start typing in these cases, matches display in a list below the text field. If there is no matching text, then no list displays. The list is automatically sized, but you can resize it. Electronics Desktop remembers the new size when doing additional auto-complete matching in the same text field. Switching to a different text field resets the sizing.

Hitting **Tab** or **Enter** accepts the current selection for auto-complete, which replaces the text typed with the full auto-complete match and hides the list. Otherwise, hitting the escape key **Esc** hides the auto-complete list. Typing more letters causes auto-complete matching to resume.

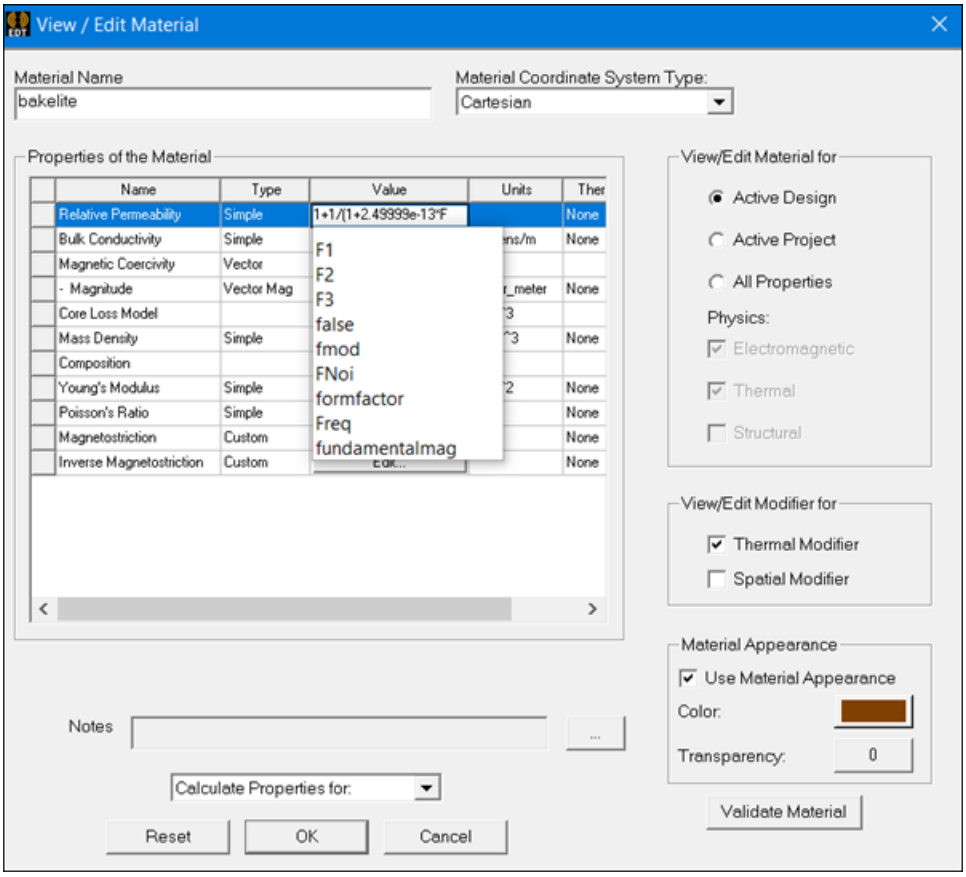
Where you select design properties, auto-complete works with expressions for the values, providing matches for the names of constants, intrinsic variables, functions, project variables (prefixed with “$”), and design properties. When project variables are displayed (that is, you select the project in the Project tree), auto-complete for value expressions also works, except without matching design properties. The variable auto-complete list does not include Separator Variables or Hidden variables since these could cause invalid formulas.



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). The following shows a design property with auto-complete matching:

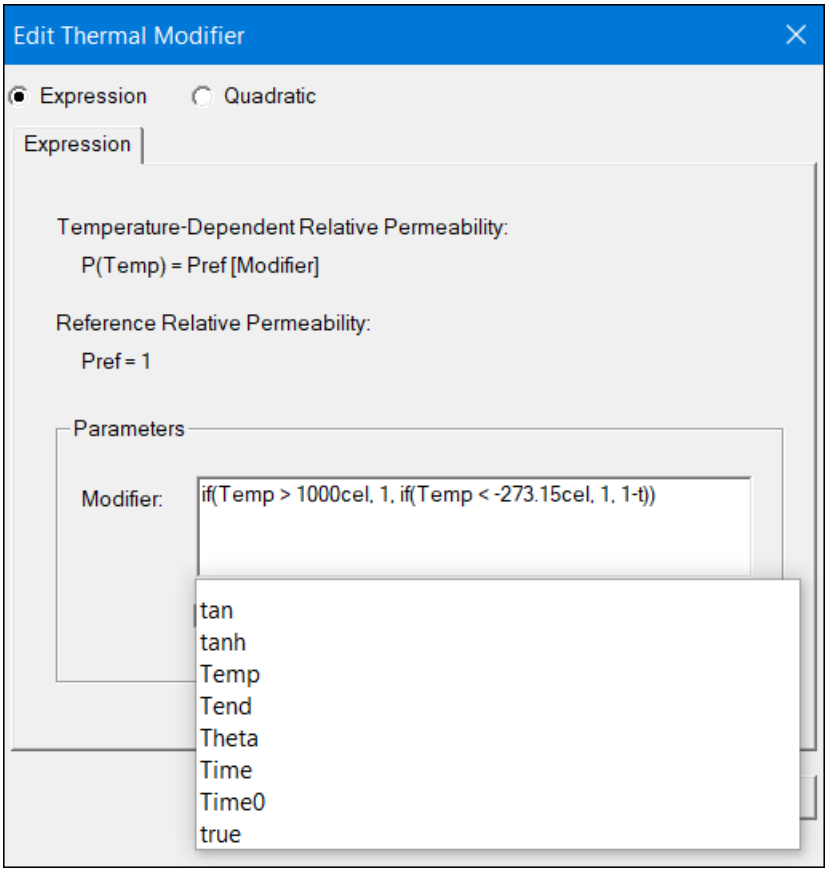


Auto-complete also works with value expressions for material properties:



Note the scroll bar, indicating that there are more matches than those currently displayed.

Auto complete also works with thermal modifier expressions for Materials:



Using the Password Manager to Control Access to Resources

Ansys Electronics Desktop lets you specify library resources that require password access, and encryption of those resources. For convenience, the same password can apply to multiple resources. To access the **Password Manager,** click **Tools** > **Password Manager**. This displays the **Password Manager** window.

To Specify a New Password Protected Resource

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

   This opens the **New Encrypted Resource** dialog box.

3. Specify the name of the resource that you want to protect and click **OK**.

   This displays the **Enter Passwords** window, which contains radio buttons to let you:

- Enter Password and confirm for **Full Access** or for **Execute Only Access**.
- **Use Ansys Password** (for execute only). This option does not require you to enter a password, but still encrypts the library.

4. Once you have selected a radio button and specified any necessary passwords correctly, click **OK**.

   The **Password Manager** updates to show the resource.

To Encrypt a Resource

1. Click **Tools** > **Password Manager**.
2. The **Password Manager** window appears.
3. In the Password Manager, select an existing resource to highlight it.
4. Click **Encrypt File**.

   A file browser window appears.

5. Use the drop-down menu to select the appropriate **Files of Type** filter.

   Any existing resources in the selected directory will appear.

6. When you have selected the appropriate resource, click **OK**.

   This encrypts the resource.

> **Note:**
>
> The **Expire** resource on option lets you select the date when the password expires for the relevant resource.

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.

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. You can run this command with the **-Iconic**, **-Logfile**, and **-ng** options.

  If no migration is necessary (*e.g.* a project containing a circuit design): The original file will be overwritten with the new version, and no backup will be made.

  Special behavior when opening backed up files from the user interface: Files located in the AnsysEM_Backup directory would be the original non-migrated files that were backed up during an earlier –batchsave or open. Opening a file located in the AnsysEM_Backup directory via the open file dialog box will invoke the Parasolid migration, and the migrated file will be saved in the parent folder of the AnsysEM_Backup folder with a unique name so as not to overwrite any existing files.

  If a ***file*** is given, **-batchsave** will run on the specified file.

  If a ***folder*** is given, **-batchsave** will run on all files underneath the specified folder.

  .aedt, .aedtz, and .a3dcomp files will be processed.

  Allowed options:

  norecurse: only process files in top level of input folder

  forcearchive: project files are converted, then saved as archive files.

  previewonly: don't perform actual conversion, instead write preview of filenames to batchlog.

  outputfolder=<outputfolder>: save converted files to <outputfolder>, preserving subfolder structure.

- **-BatchSolve <***project file name***>**

  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 quotation marks. 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***>:<***TasksOnMachine***>:<***CoresOnMachine***>:<***RAM Limit in %***>:<***GPUsOnMachine***>**

    Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. RAM% must be of the form "<num>%" where num is an integer between 1 and 99. Please be sure to leave sufficient RAM for the operating system and applications. If -auto is specified with a machine list, the number of tasks for each machine must be -1.

    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***>:<***TasksOnMachine***>:<***CoresOnMachine***>:<***RAM Limit in %***>:<***GPUsOnMachine***>**

  Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. RAM% must be of the form "<num>%" where num is an integer between 1 and 99. Please be sure to leave sufficient RAM for the operating system and applications. If -auto is specified with a machine list, the number of tasks for each machine must be -1.

  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 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.

  If the list contains any whitespace, it must be enclosed in quotation marks. For example:
  `"includeTypes=Frequencies,Mesh Assembly"`
- **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.

  If the list contains any whitespace, it must be enclosed in quotation marks. For example:
  `"excludeTypes=Frequencies,Mesh Assembly"`
- **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.

-Auto [**NumDistributedVariations=<***num***>**]

This flag enables automatic HPC settings and must be used with one of the following options:

- **-machinelist list=**<*machine list*>, with tasks for each machine set to -1
- **-machinelist numcores=**<*num*>, under a scheduler

All design types being solved must support **-auto** or the solve will be aborted.

The NumDistributedVariations option can be used to specify the number of optimetrics variations to solve simultaneously. The default is to solve optimetrics variations sequentially.

Arguments with **-auto** in a scheduler environment:

- **numcores=**<*total number of cores*>

  Total number of cores, and can be used only with -auto and in a scheduler environment.

- **file=**"<*tmachine list file path*>"

  The specified file can contain line delimited machine specifiers as described above.

- **num=**<*tnum distributed tasks*>

  This is the total number of tasks and used only in a scheduler environment.

- **numgpus=**<*tnumber of GPUs to use*>

  This is the total number of GPUs and used only in a scheduler environment. numgpus must be combined with either **num=** or **numcores=**.

  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. If -auto is specified with a machine list, the number of tasks for each machine must be -1.
- **-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 file name***> <***project file name***>** – 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.

> **Important:**
>
> - **-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 file name***>** – 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 file name***>** – 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.
- **-grpcsrv** – For external cpython scripting support. If not specified with **ansysedt.exe**, the Start GRPV server listens on a port in the default range 50051:51051. If specified, Start GRPV server listening on the the specified port number. An error is issued if the port is already used. For example:

```
ansysedt.exe // Start GRPV server listening on a port in
the default range 50051:51051.
ansysedt.exe -grpcsrv portnumber // Start GRPV server
listening on the the port number. error if the port
already used.
ansysedt.exe -grpcsrv 50051:50150 // Start GRPV server
listening on a port range 50051:50150.
ansysedt.exe -grpcsrv 50051:100 // Start GRPV server
listening on a port range 50051:50151.
```

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\ANSYS Inc\v251\AnsysEM\ansysedt -distributed -
machinelist list="255.255.1.1,255.255.1.2" -batchsolve
myDesign:Optimetrics "C:\myProject.aedt"
```

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

> **Note:** The example is for an HFSS project. The path, project and design names will vary depending on what the user specifies.

```
ansysedt -ng -batchextract exportToFile.py "C:\Program Files\ANSYS
Inc\v251\AnsysEM\Examples\ElectronicsDesktop\HFSS\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.
- **repackageresults** – Add batchsolve results back to 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 Direct 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.
- **-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 quotation marks. For example:

  ```
  ansysedt.exe -scriptargs "Design1 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 Design1
  msgbox AnsoftScript.Arguments(1) // Returns Setup1
  ```

  In either case, **Design1** is taken into Electronics Desktop 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 Electronics Desktop.

-Batchoptions

All options that are specified through **Tools** > **Options** go into the user-level registry.

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 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

You can override the option 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 the following products:

HFSS, HFSS-3DLayout, Icepak, Q3D, Q2D, Maxwell3D, Maxwell2D, and Mechanical.

When this option is set, only the initial mesh and manual mesh operations are completed for the batch solution.

> **Note:**
>
> No adaptive meshing occurs for any Icepak or Mechanical solutions. Therefore, after solving, the final mesh will be identical to the starting mesh for these two design types.

This example enables CreateStartingMesh for HFSS, and runs a batch solution of 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 2025 R1 installed on shared directories or network drives. See: Example Uses for Export Options Features.

Examples and Further Explanations of -batchoptions Use

This section provides further examples and 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 HFSS-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
'HFSS/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

```
ansysedt.exe -batchsolve -batchoptions <filename>
projectname.aedt
```

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

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

This command overrides the registry values of the NumberOfProcessors (Desktop/Settings/ProjectOptions key), NumCoresPerDistributedTask (HFSS 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 from the command line with **-batchoptions**, see:

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

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 and/or EnableGPUForEye – Applicable to Circuit Design, EMIT, HFSS, HFSS 3DLayout, and Maxwell 3D (also EnableGPUForSBR for HFSS only)
- <*design_type*>/MPIVendor (either “Intel” or “Microsoft”) – Applicable to HFSS, HFSS 3DLayout, Icepak, Maxwell 2D, Maxwell 3D), and Q3D

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	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 “Documents” directory	Directory pathname	Directory PersonalLib is appended if final 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

For -batchoptions Use: TempDirectory

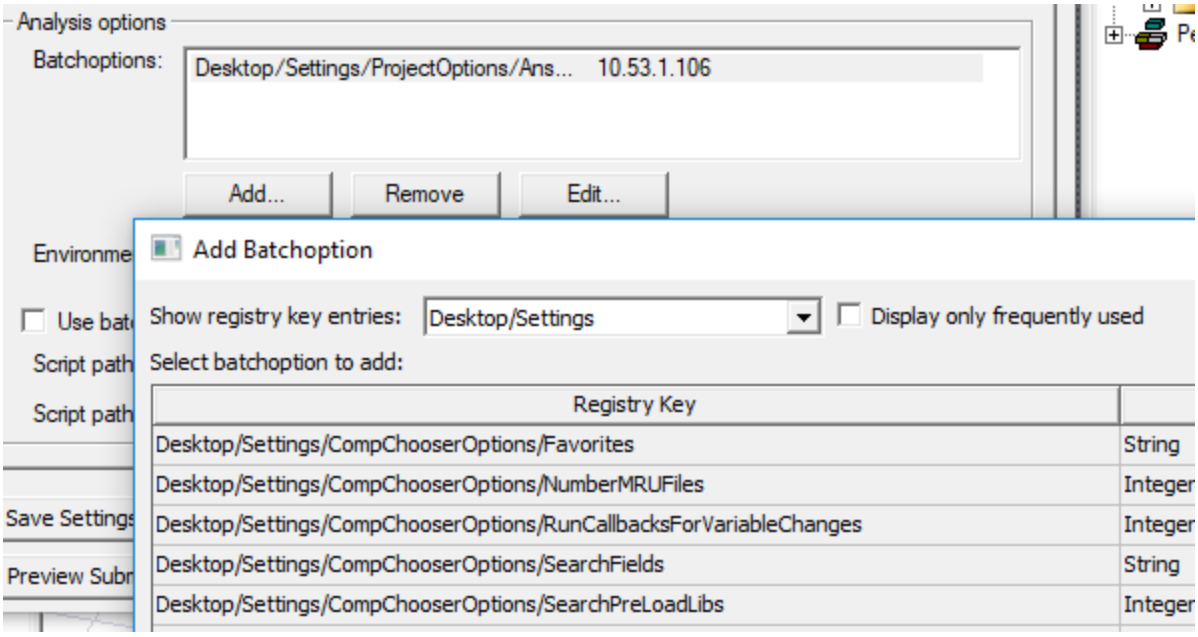
Registry Key	Default Value	Units or Values	Description
TempDirectory	Set by installer	-	Directory for temporary files

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

Registry Key	Default Value	Units or Values	Description
Desktop/Settings/ProjectOptions/ExpandMessageTreeOnInsert	1 (true)	0 (false) or 1 (true)	Ensure that new messages are visible in the message window tree
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/SavePreviewImagesInProject File	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.

Running Ansys Electronics Desktop from a Windows Remote Terminal

When running Q3D Extractor from a remote terminal, there are some performance and behavior issues to consider. These issues are due to the interaction of bandwidth/opengl drivers/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.

Using Windows HPC Commands

HPC Integration allows you to submit jobs directly using Ansys Electromagnetics command line arguments for batch solves. 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.

For information concerning the supported HPC software, go the the Platform Support webpage and click the links to the following topics:

- *Ansys 2025 R1 - Job Schedulers and Queuing Systems Support*
- *Ansys 2025 R1 - Message Passing Interface Support for Parallel Computing*

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

Customizing Ansys Electronics Desktop with Ansys ACT

With Ansys ACT, you can create custom applications or extensions to customize supported Ansys products, including Electronics Desktop.

> **Note:**
>
> The **ACT Extensions** window and the design wizards it contains (*5G Wizard, HFSS Antenna Design Toolkit, HFSS-EMA3D Link,* and *Maxwell Eccentricities*) are only available for the Windows version of the Ansys Electronics Desktop software. These items are not available when the software is installed on a Linux platform.

An ACT guided process extension enables you to leverage both the functionality of Electronics Desktop and the scripting capabilities of the Workbench/AIM framework API. You can manipulate existing features and simulation components, organizing them as needed to produce a custom automated process. A guided process extension is exposed in Electronics Desktop as a wizard that provides step-by-step simulation guidance within the application workflow.

To access ACT functionality in Electronics Desktop, open the ACT Home page by clicking **View** > **ACT Extensions**.

For more information, see ACT Simulation Wizards and Electronics Desktop Wizard in the *Ansys ACT Developer's Guide*.

Installing PyAEDT (Beta)

PyAEDT is a Python library that interacts with the AEDT API to make scripting simpler for the end user. It supports all AEDT 3D products (HFSS, Icepak, Maxwell 3D, and Q3D Extractor), 2D tools, Ansys Mechanical, EMIT, Circuit tools like Nexxim, system simulation tools like Twin Builder, and layout tools like HFSS 3D Layout and EDB. Additionally, it enables the end user to have a CPython interface with AEDT. Its class and method structures simplify operation for the end user, enabling more Pythonic code while reusing information as much as possible across the various APIs.

Documentation for PyAEDT can be found online at: https://aedt.docs.pyansys.com/version/stable/

Installing PyAEDT adds three items to the **Tools** > **Toolkit** > **PersonalLib** menu and to the **Automation** tab:

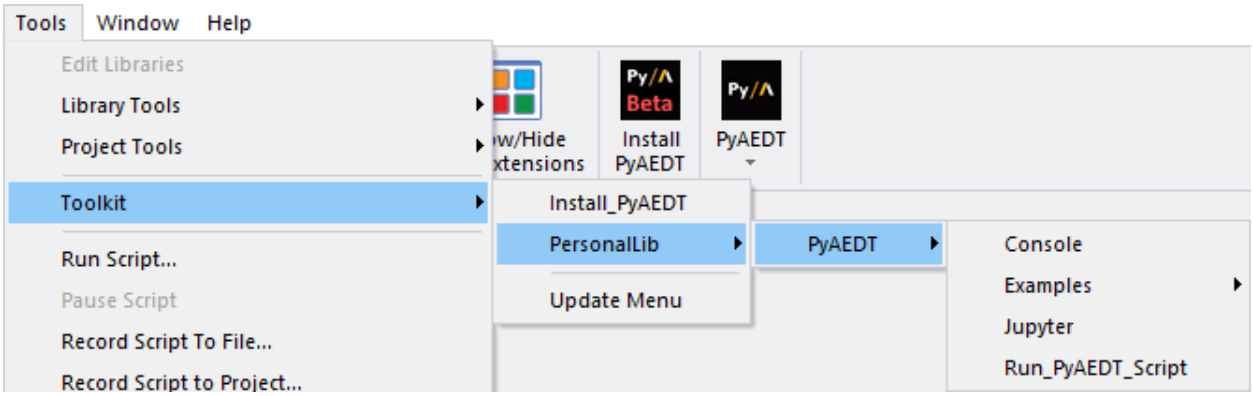
- **Console** – launches the PyAEDT console.
- **Jupyter Notebook** – launches Jupyter Notebook (a computational notebook) in an internet browser.
- **Run PyAEDT Script** – launches a file browser allowing you to select a Python script to run via PyAEDT.

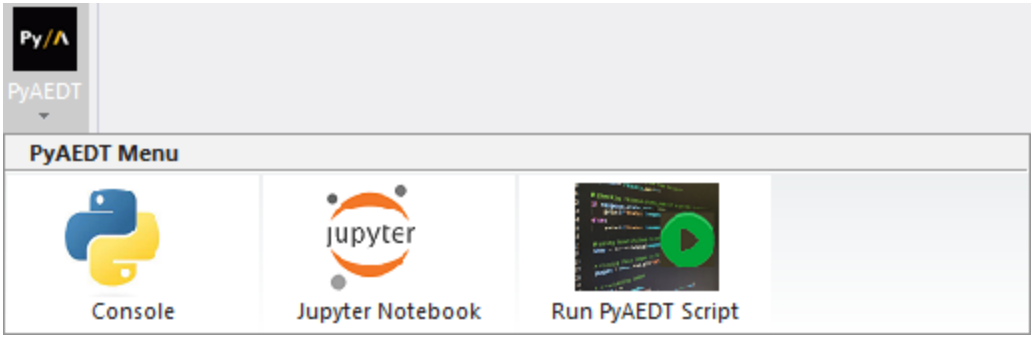
To install PyAEDT:

- From the **Automation** tab, click **Install PyAEDT**.

A web browser launches, and takes you to detailed installation instructions.

When installation is complete, the **Tools** > **Toolkit** > **PersonalLib** menu and the **Automation** tab update to display PyAEDT menu options:





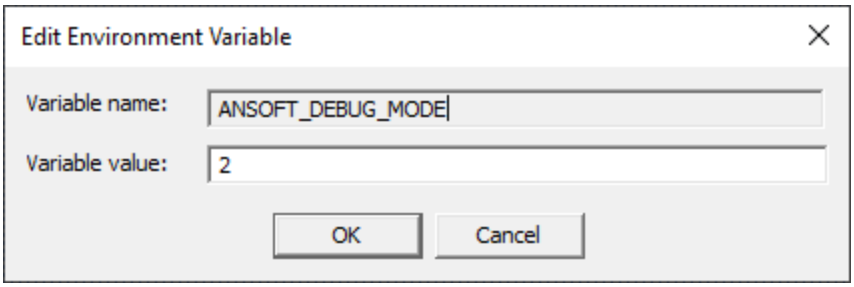
Debug Logging

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.

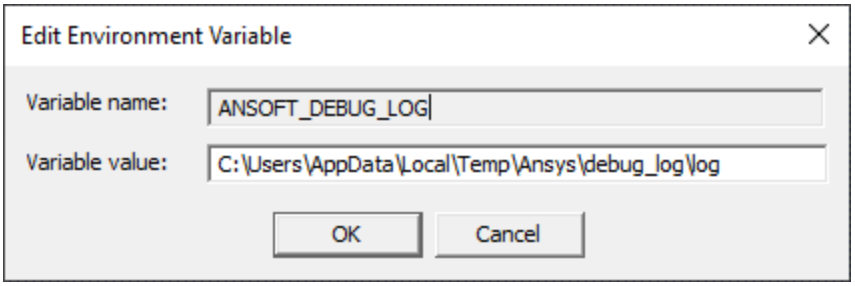


2. Enter a **Variable value**.

   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**.

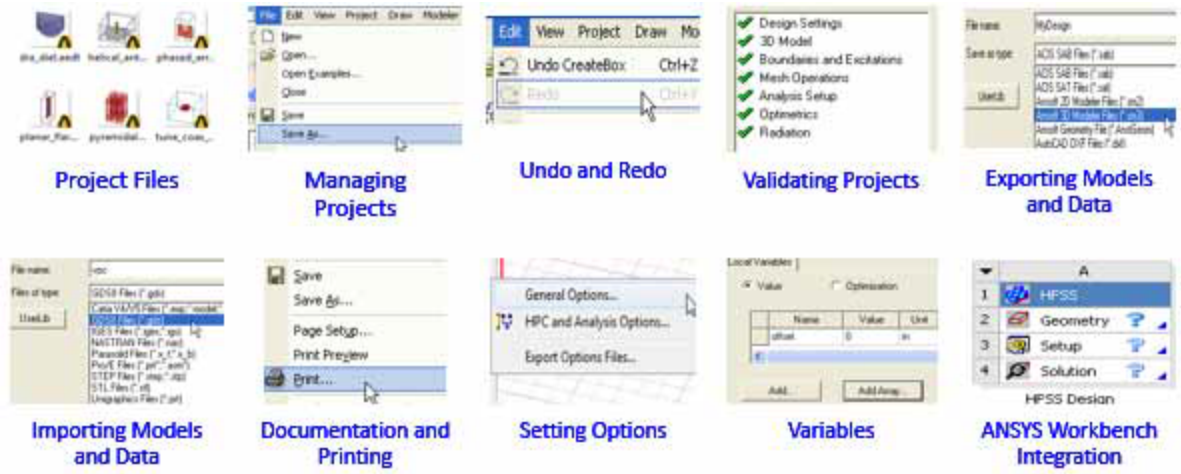
   Errors are logged to the specified folder until you disable logging.

To stop logging errors:

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

3 - Working with Ansys Electronics Desktop Projects

An Ansys Electronics Desktop project is a folder that includes one or more models, or *designs*. Each design ultimately includes a geometric model, its boundary conditions and material assignments, and field solution and post-processing information, or a schematic or netlist.



A new project called *Projectn* is automatically created when the software is launched. By option, a design named Design*n* is automatically created for a 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.

> **Note:**
>
> Not all options and capabilities documented in this section apply to all design types.

Ansys Electronics Desktop Files

When you create any project in Ansys Electronics Desktop, it is given an .aedt file extension and stored in the directory you specify. For Legacy projects, the transition is handled as follows:

- When opening a legacy project in interactive mode, a dialog box appears, informing you that the legacy file extension is no longer supported and that if you continue, the project will be converted to the new .aedt extension. If you agree, the project and results directory are renamed/moved to the new file extensions immediately. The read continues with the standard code for reading previous version projects.

- The Ansys Electronics Desktop 2025R1 version cannot open projects created with 2022R2 or earlier that were using ACIS modeler.
- Attempts to run batch solve or non-graphical with a legacy project return an error. There is no automatic/hidden conversion of file extensions. Note that the existing BatchSave command can be used to convert many projects to the new extension and version.
- Workbench integration has not been modified. It continues to open legacy projects without a warning and copies results from legacy to the new extension.

Getting Started with Q3D Extractor

Q3D Extractor is an interactive software package that electrically characterizes three-dimensional interconnect structures, such as those found in connectors, Printed Circuit Boards (PCBs), Ball Grid Arrays (BGAs), and Multi-Chip Modules (MCMs).

Q3D Extractor can solve for:

- Capacitance and conductance matrices
- DC analyses:
  - Partial inductance and resistance matrices
  - Partial resistance matrices
- AC analyses:
  - High-frequency, partial inductance, and resistance matrices

You choose the matrices you want to generate, draw the structure, specify material properties for each object, identify conductors, and specify source excitations. Q3D Extractor then generates the necessary circuit parameters. From these matrices, you can generate lumped equivalent circuit models in any of several SPICE formats.

Setting Up a Design in Q3D Extractor

To set up a design, consult the following procedure.

After you insert a design, you do not need to perform the steps sequentially, but you must complete all steps before a solution can be generated.

1. Insert a Q3D Extractor design into a project.
2. Set the model 's units of measurement.
3. Draw the model geometry and assign material characteristics to objects.
4. Identify nets.
5. Specify how Q3D Extractor will compute the solution.
6. Reduce the matrices to modify conductor definitions and generate simpler matrices.
7. Export the circuit to generate a circuit equivalent of the model.

8. Run the simulation.
9. View solution results, post-process results, view reports and assign field overlays.

> **Note:**
>
> After being idle for 10 minutes, Q3D gives up its license. A renewal of activity automatically requests a license. Such idle notifications do not occur during solves.

Inserting a Design in Q3D Extractor

Insert a Q3D Extractor design in one of three ways:

- Click **Project** > **Insert Q3D Extractor Design**.
- On the Desktop tab, click the **Q3D** icon.
- On the Desktop tab, click the **Q3D** drop-down and select **Q3D Extractor**.

Insert a 2D Extractor design in one of two ways:

- Click **Project** > **Insert 2D Extractor Design**.
- On the Desktop tab, click the **Q3D** drop-down and select **2D Extractor**.

The new design is listed in the project tree. It is named Q3DExtractorDesign*n* or 2DExtractorDesign*n*by default, where *n* is the order in which the design was added to the project.

> **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.

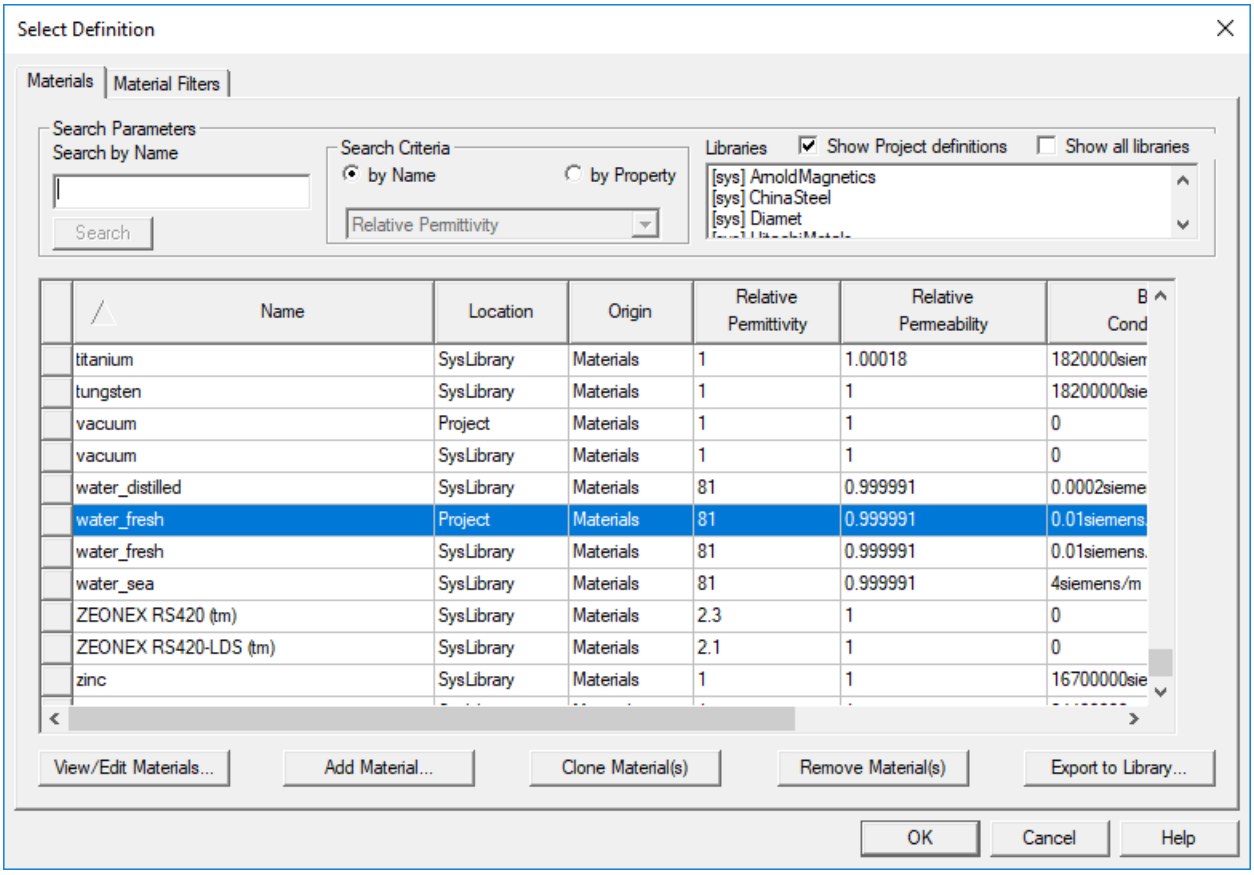
Setting the Background Material in Q3D Extractor

By default, "vacuum" is used as the background material.

To change the background material:

1. Depending on your design, click the **Q3D Extractor** or **2D Extractor** menu and select **Set Background Material**.

   The **Select Definition** window appears. By default, it lists all of the materials in the global material library as well as the project 's local material library.



2. Select a material from the list.

   If the material you want to assign is not listed, add a new material to the global or local material library, and then select it.

   If there is a specific material or material property value that you want to assign to an object, you can search the materials in the **Select Definition** window by name or by material property.

3. Click **OK**.

   The material you chose is assigned to the background.

Getting Started with 2D Extractor

2D Extractor is an interactive software package that can electrically characterize two-dimensional structures, such as transmission lines and connector cross-sections. By modeling the field patterns in a single cross-section, 2D Extractor can analyze field patterns in an entire device.

2D Extractor can:

- Compute capacitance and inductance for lossless transmission structures, and compute impedance and admittance matrices for lossy transmission structures.
- Compute the characteristic impedance of the transmission line structure, its modal velocities, and its modal transformation matrices.
- Compute forward and backward crosstalk coefficients.
- Display field patterns in a transmission line given a particular distribution of charge, potential, or current.
- Export a SPICE equivalent circuit to a file that can be read into any other software package that uses a compatible version of SPICE.
- Perform variational analyses of designs by varying solution frequencies, model dimensions, material properties, and other transmission line parameters.

You can draw or import a geometric model of your structure's cross-section into 2D Extractor and then specify:

- All relevant material characteristics
- Boundary conditions describing field behavior
- Sources of charge, current, or voltage
- Solution criteria

After 2D Extractor computes the requested circuit parameters, you can export circuit equivalents, and view and analyze the computed matrices or fields.

Setting Up a Design in 2D Extractor

To set up a design, consult the following procedure.

After you insert a design, you do not need to perform the steps sequentially, but you must complete all steps before a solution can be generated.

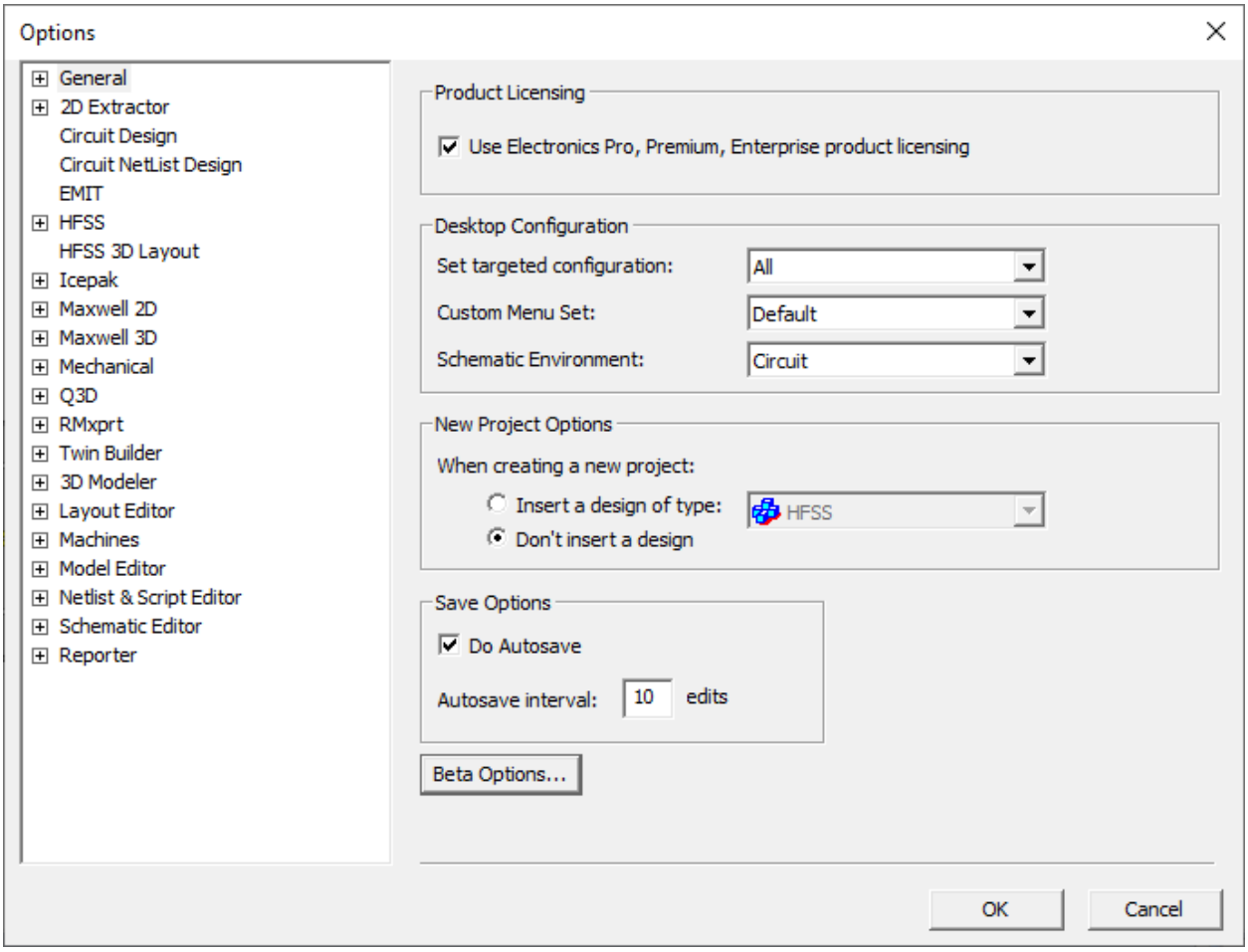
1. Insert a 2D Extractor design into a project.
2. Set the model 's units of measurement.
3. Draw the model geometry and assign material characteristics to objects.
4. Specify how 2D Extractor will compute the solution.
5. Reduce the matrices to modify conductor definitions and generate simpler matrices.
6. Export the circuit to generate a circuit equivalent of the model.
7. Run the simulation.

8. View solution results, post-process results, view reports and assign field overlays.

> **Note:**
>
> After being idle for 10 minutes, Q3D gives up its license. A renewal of activity automatically requests a license. Such idle notifications do not occur during solves.

Setting Project Options

Tools > **Options** > **General Options** opens the **Options** window.



The left pane provides access to the following sets of options:

- General
- 2D Extractor
- Circuit Design

- Circuit Netlist Design
- HFSS
- HFSS 3D Layout
- Icepak
- Maxwell 2D
- Maxwell 3D
- Mechanical
- Q3D
- RMxprt
- Twin Builder
- 3D Modeler
- Machines
- Model Editor
- Reporter

Use **Tools > HPC and Analysis Options** to specify active configuration per design type, queuing, distributed memory vendor, HPC licensing, select a MPI vendor, and whether to enable GPU for transient solves.

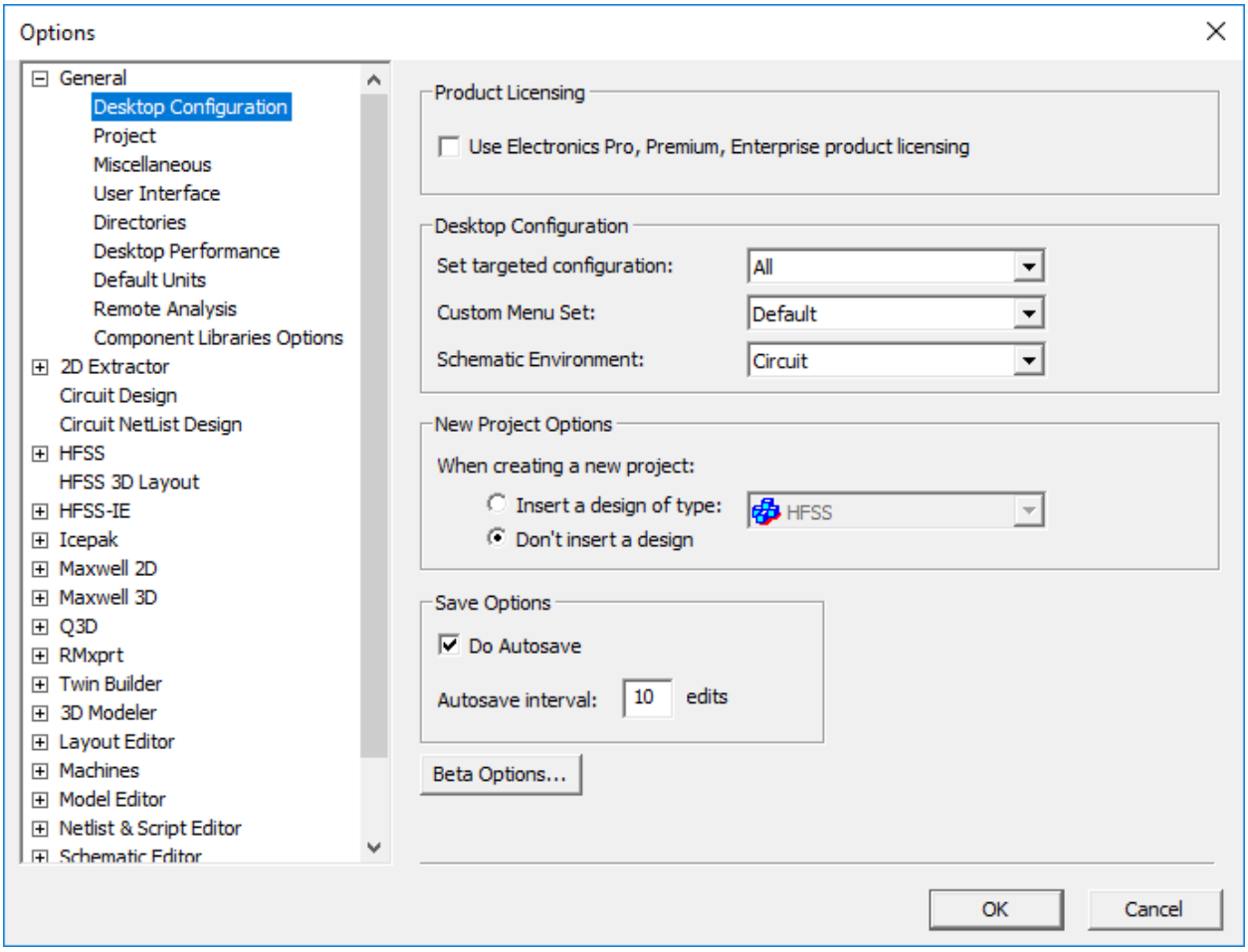
The **Tools > Options > Export Options Files** command writes xml files containing the Options settings at all levels to the specified directory. The **Tools** > **Options** > **Export Options** feature is intended to make it easier for different users to use Ansys Electromagnetics tools installed on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

Setting General Options

To set general options in Ansys Electronics Desktop:

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

   The **Options** window opens with the Desktop Configuration options selected by default.



In the left pane, select the entries below **General** to display the associated options:

- Desktop Configuration
- Project
- Miscellaneous
- User Interface
- Directories
- Desktop Performance
- Default Units
- Remote Analysis
- Component Libraries Options

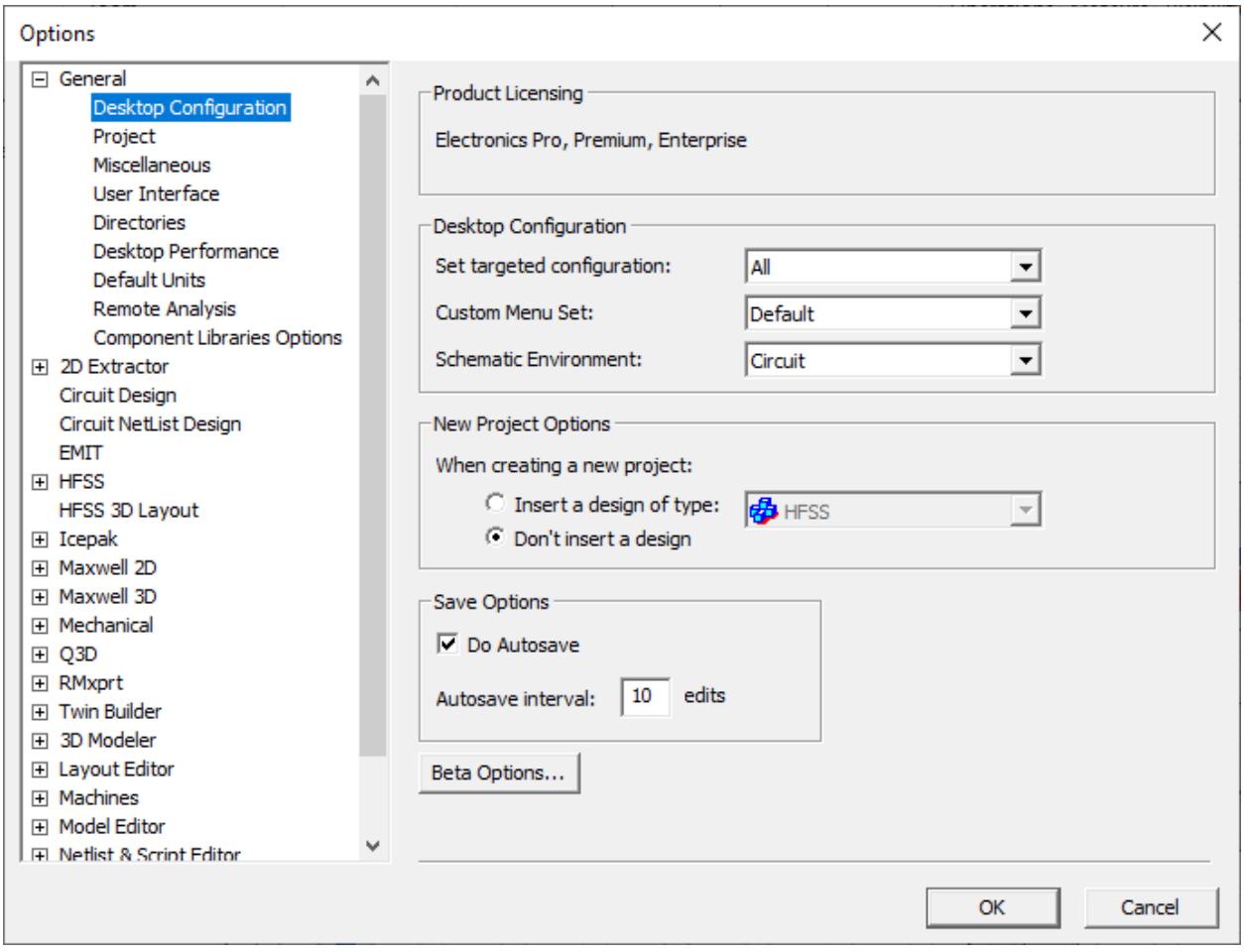
2. Click each entry and make the desired selections.

3. Click **OK**.

> **Note:**
>
> Not all option settings are applied immediately in the current design. Some settings take effect in newly created designs.

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 menus, new projects, save intervals, and beta features.



Ansys Electronics Desktop now uses Electronics Pro, Premium, Enterprise (PPE) product licensing. Legacy product licensing and DSO are no longer supported.

With PPE, HPC licensing is used to enable all cores, GPUs, and distributed tasks.

For more details, see Setting HPC and Analysis Options.

In the **Desktop Configuration** area, options include:

- **Set Targeted Configuration** – Choosing a Targeted Configuration changes the Custom Menu Set and Schematic Environment for your area of focus.

  Choices are:

  - **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 Set to** 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 everything to default to **Twin Builder**.
- **Custom Menu Set** – Changes which menu options are available in Electronics Desktop.

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

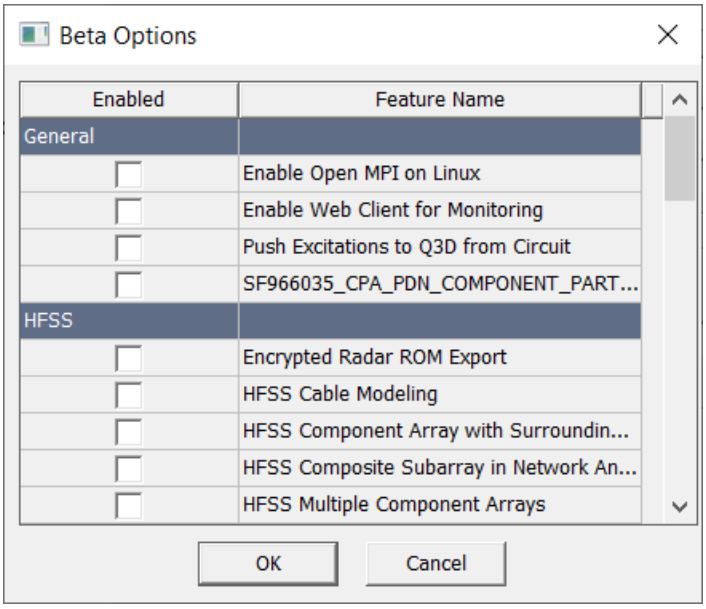
In the **New Project Options** area, options include:

- **When Creating a New Project** – Select whether to insert a design when creating a new project. If you choose to insert a design by default, use the drop-down menu to select the default design type.

In the **Save Options** area, options include:

- **Do Autosave** – Electronics Desktop has autosave enabled by default. Deselect this option to disable it.
- **Autosave Interval** – When **Do Autosave** is enabled, select the number of edits at which Electronics Desktop autosaves.

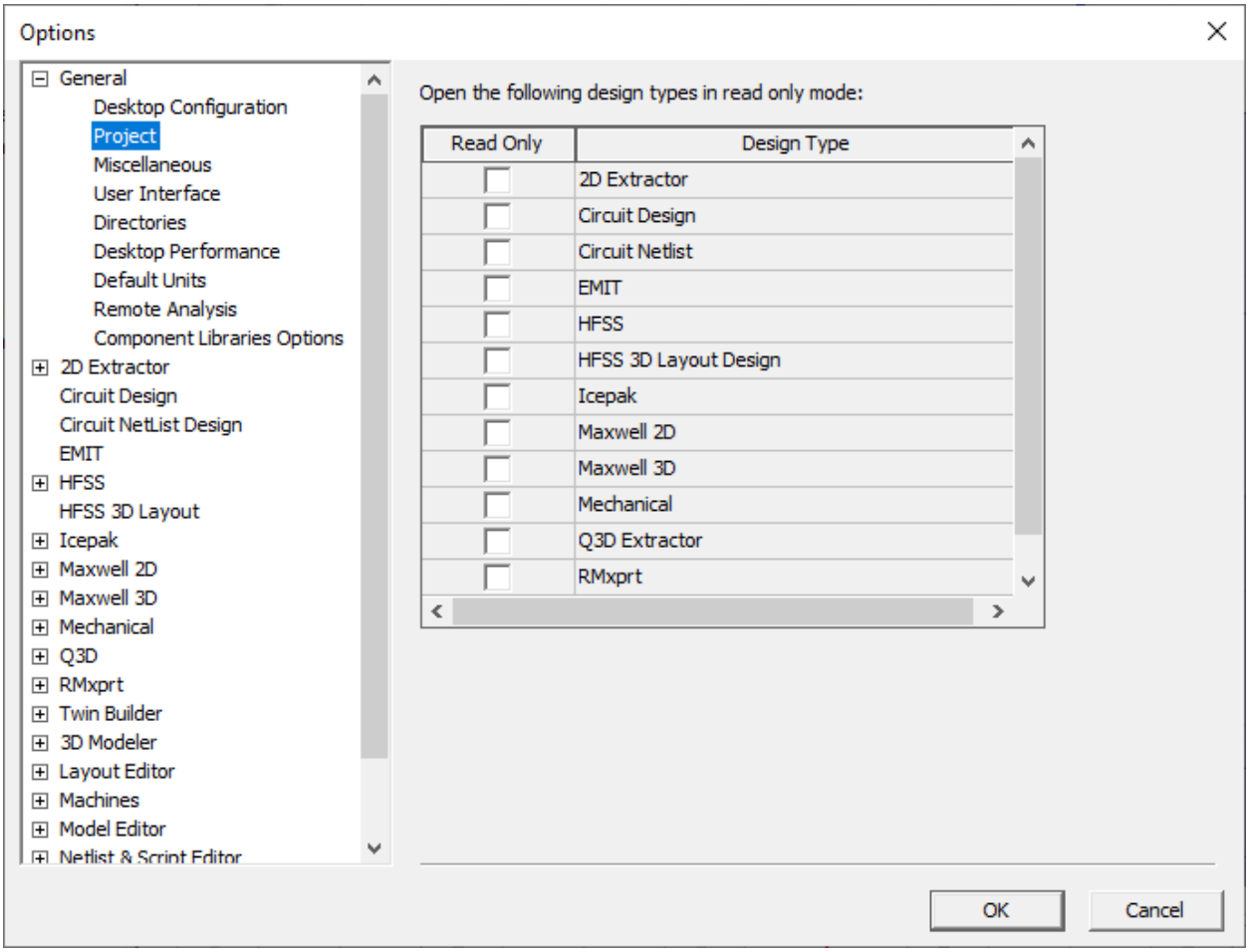
Click the **Beta Options** button to open a window listing beta options. You may need to scroll or size the window to view all options.



From there, enable or disable options, and click **OK**. You will need to restart to enable your selections.

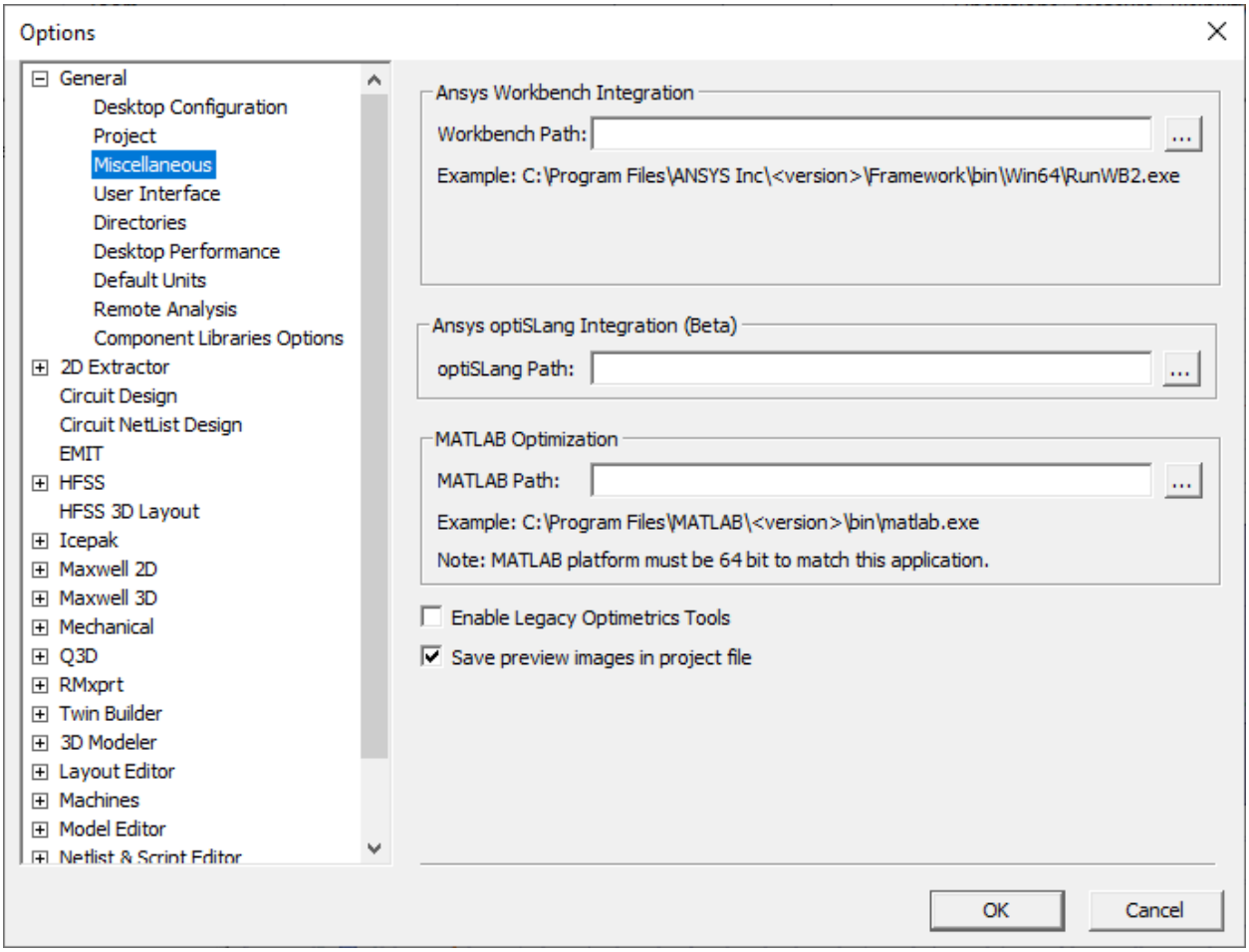
General Options: Project Options

Under General Options, **Project** options allow you to select design types to open in read-only mode. This can help prevent accidental changes.



General Options: Miscellaneous Options

Under General Options, **Miscellaneous** options allow you to specify paths for Ansys Workbench Integration, optiSLang path, MATLAB Optimization and to enable Legacy Optimetrics Tools.



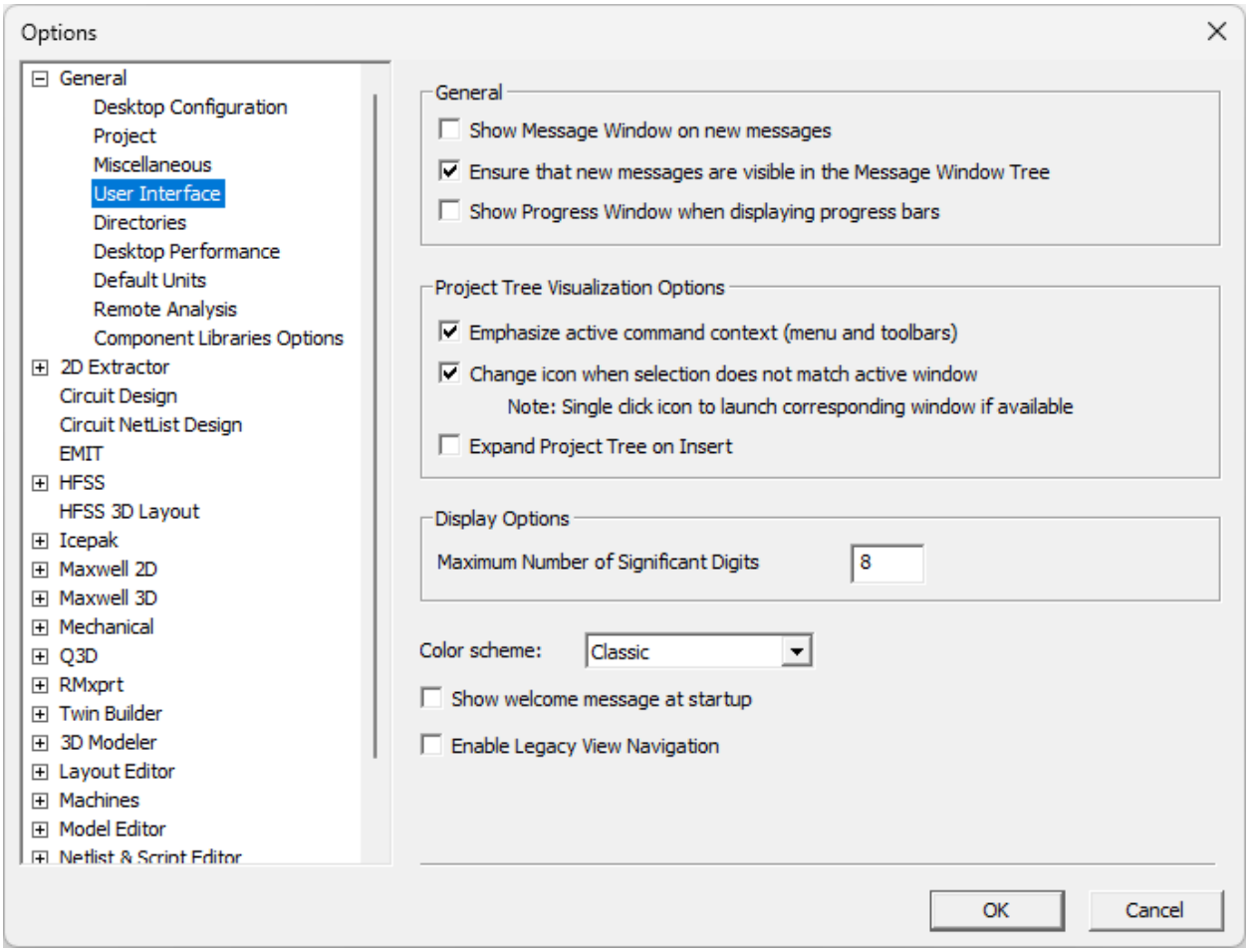
Ansys Workbench Integration allows you to specify the path to your Ansys Workbench installation. This path can be used in Optimetrics for connecting to the Design Xplorer.

MATLAB Optimization allows you to specify the path to your installation of MATLAB. MATLAB can be used as an Optimizer.

> **Note:**
>
> The platform for MATLAB and Ansys Electronics Desktop must match (for example, a 64 bit version of MATLAB).

General Options: User Interface Options

Under General Options, **User Interface** options allow you to change how Electronics Desktop displays messages, command text, the project tree, welcome messages, and more.



In the **General** area, options include:

- **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.

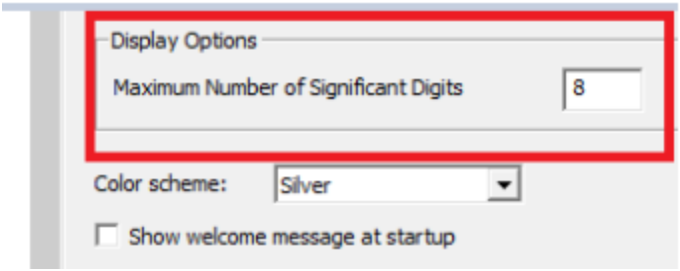
In the **Project Tree Visualization Options** area, options include:

- **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

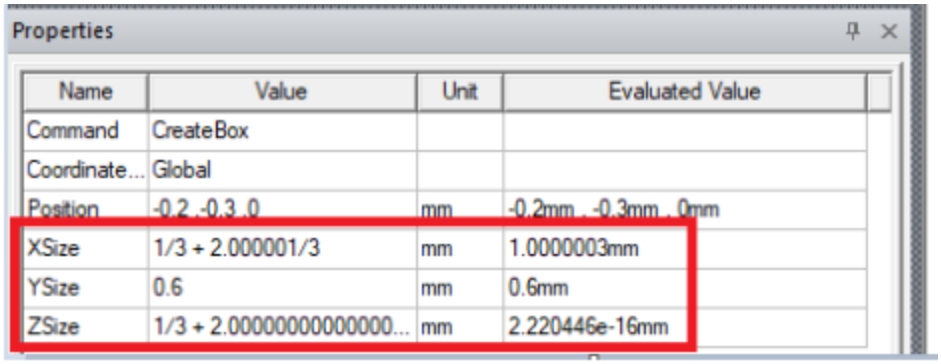
selected project item (data affecting the same model is considered to be related). Clicking the icon opens the window and brings it into focus.

- **Expand Project Tree on Insert** – when selected, the Project Tree automatically expands when you insert a new design.

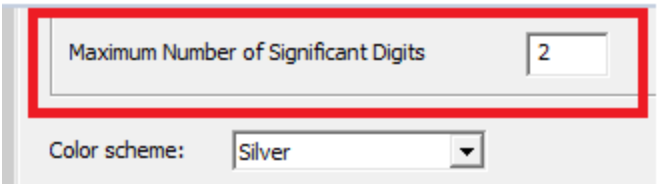
In the **Display Options** area, specify the **Maximum Number of Significant Digits** to display. The default is 8 and the maximum is 20. This affects the digits displayed in the **Solutions** dialog box, evaluated variable values, Animation dialog boxes, Optimetrics, Reports, and so forth. For example, here is the default.



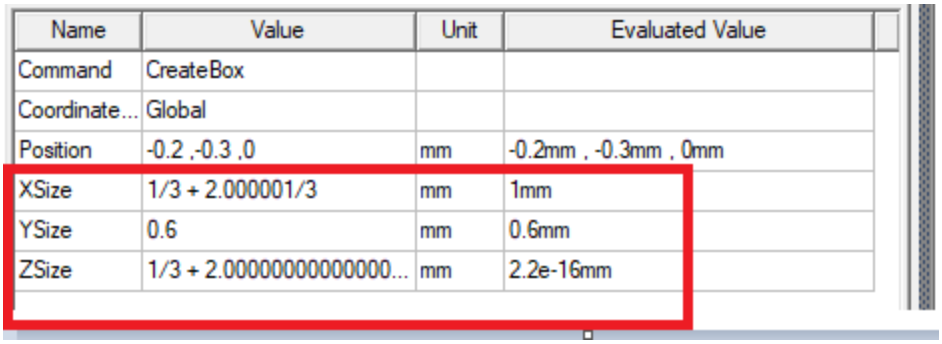
So some evaluated variable values with significant digits appear as follows.



Change Precision = 2. This lets you display near-zero values in the Properties window as 0.

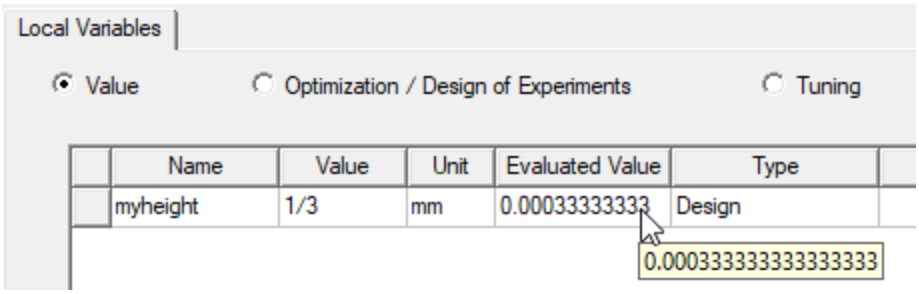


You can see that XSize is now rounded off to cleanly display 1mm. The decimal part of ZSize also shows cleaner value, but it is not rounded off to zero.

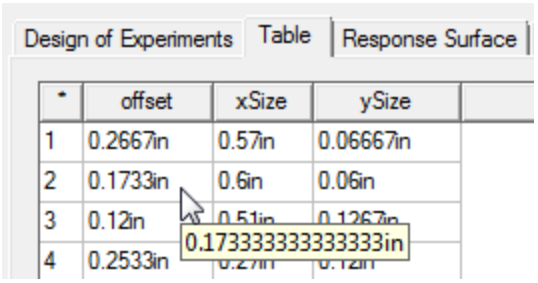


Name	Value	Unit	Evaluated Value
Command	CreateBox		
Coordinate...	Global		
Position	-0.2 ,-0.3 ,0	mm	-0.2mm , -0.3mm , 0mm
XSize	1/3 + 2.000001/3	mm	1mm
YSize	0.6	mm	0.6mm
ZSize	1/3 + 2.00000000000000...	mm	2.2e-16mm

You can still see the full precision values in tooltips by holding a cursor over the displayed value.

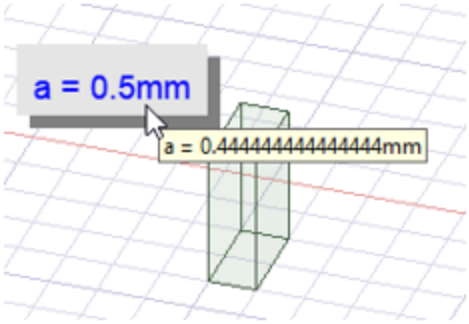


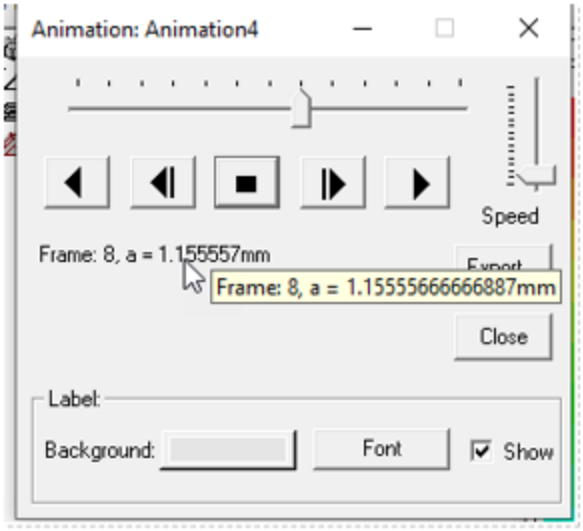
In the case of variable values, if you have assigned more significant digits, you will see these when editing the variable value. In the case of table displays of values, the tooltip display shows all available digits when the mouse pointer is over a result:



If you set the **Maximum Number of Significant Digits** to a lower value, Change Precision = 2. You can see that XSize is now rounded off to cleanly display 1mm. The decimal part of ZSize also shows cleaner value, but it is not rounded off to zero.

The tooltip functions to show internal digits in throughout the Ansys Electronics Desktop interface.





Select a **Color Scheme**. The choices are **Light (Beta)**, **Dark (Beta)**, or **Classic**: See Choosing a Color Scheme for more information and samples of each scheme.

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

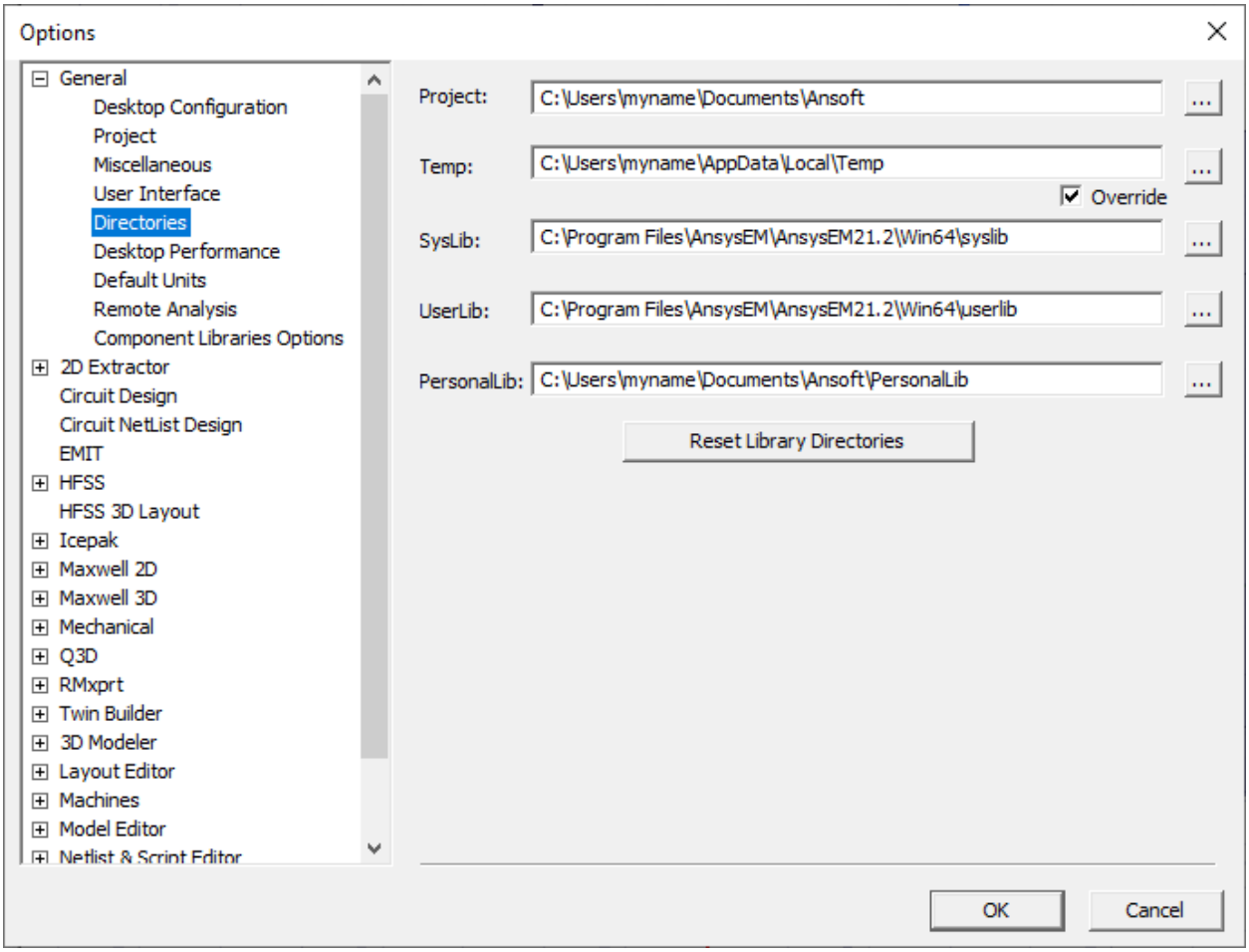
Enable legacy view navigation enables you to choose between two schemes for view navigation keyboard shortcuts and mouse button assignments, as follows:

- **Cleared:** Use only the view navigation mouse-button and hotkey assignments introduced in Ansys Electronics Desktop version 2024 R1 and applicable to subsequent versions too. Legacy (2023 R2 and earlier) mouse-button and hotkey assignments will *not* be recognized when this option is cleared.
- **Selected:** Both the legacy (2023 R2 and earlier) and current (2024 R1 and newer) mouse-button and hotkey assignments are supported, and either can be used for view navigation.

See Choosing the View Navigation Options for more information, including a detailed comparison of the two schemes.

General Options: Directories Options

Under General Options, **Directories** options allow you to get the paths for Project, Temp, SysLib, UserLib, and PersonalLib directories.



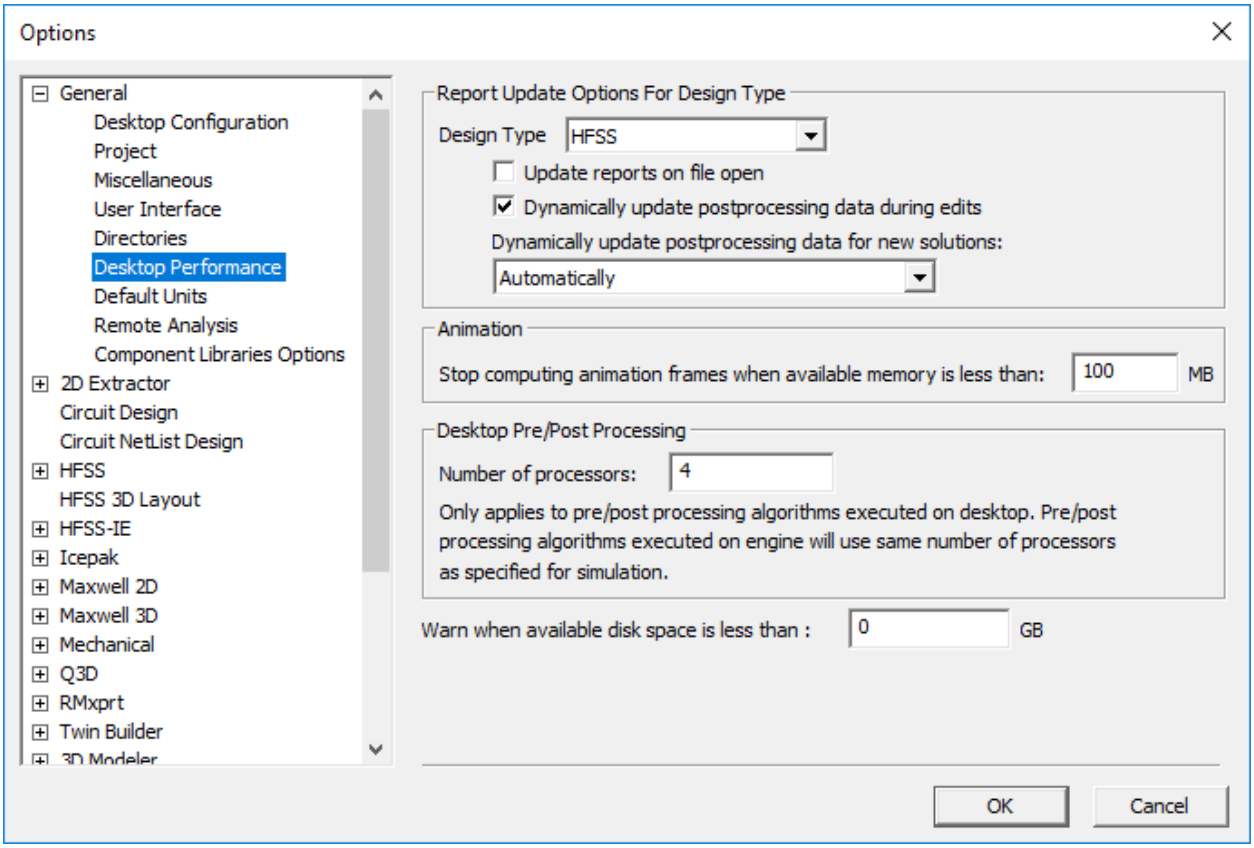
Use the **[...]** buttons to browse to paths and click **OK**. You will need to select **Override** to change the Temp directory.

If you need to restore defaults, click **Reset Library Directories**.

> **Note:**
>
> Your changes will be reflected in the User Defined Primitives menu on next startup.

General Options: Desktop Performance

Under General Options, **Desktop Performance** options allow you to change settings for animation, pre/post-processing, disk space warnings, and updating reports.



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

In the **Report Update Options for Design Type** area, select a **Design Type**. For each type, you can set the following options:

- **Update Reports on File Open** – when selected, reports are automatically updated whenever an existing file with solution data is opened for viewing/editing.
- **Dynamically Update Postprocessing Data During Edits** – when selected, report plots and overlays update as you edit their parameters.
- **Dynamically Update Postprocessing Data for New Solutions** – Because updating reports during analysis can impact solution time, you can specify how often your reports are updated:
  - **After Each Variation** – updates reports after analysis of each variation has been completed. Used for an Optimetric or parametric analysis.
  - **Automatically** – balances report and field plot updating with solution time.

    For Adaptive Passes, plots update at the end of each solution pass. For Last Adaptive or Transient, plots update at the end of the transient or adaptive solution.

For example, reports may be updated after each adaptive pass but field plots will not be updated until the solution is complete.

- **Immediately** – updates reports and plots as soon as data comes from the solver.

  This option will have the greatest impact on overall solution time, but affords the fastest updates to 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 mesh size increases.

- **Never** – updates reports only upon manual intervention. This prevents updates from impacting solution time.
- **On Completion** – updates reports once, when the solution completes.

> **Note:**
>
> Updates done on completion are done after the solve has been completed, and the time for that update is not included in the simulation profile.

In the **Animation** area, you can elect to **Stop computing animation frames when available memory is less than** a specified value, in MB (the default is 100MB). This setting is used to prevent problems related to low memory should an animation require large memory allocation.

In the **Desktop Pre/Post Processing** area, you can specify the **Number of Processors**. This option only affects pre- and post-processing (not solve or simulation). Pre-processing algorithms can take advantage of multiple processors for visualization and faceting of 3D models, model validation for 3D products, auto net identification for Q3D, and more. The default value is determined by the number of logical processors on the machine running Electronics Desktop:

- 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

You can also elect to **Warn when available disk space is less than** a specified value, in GB.

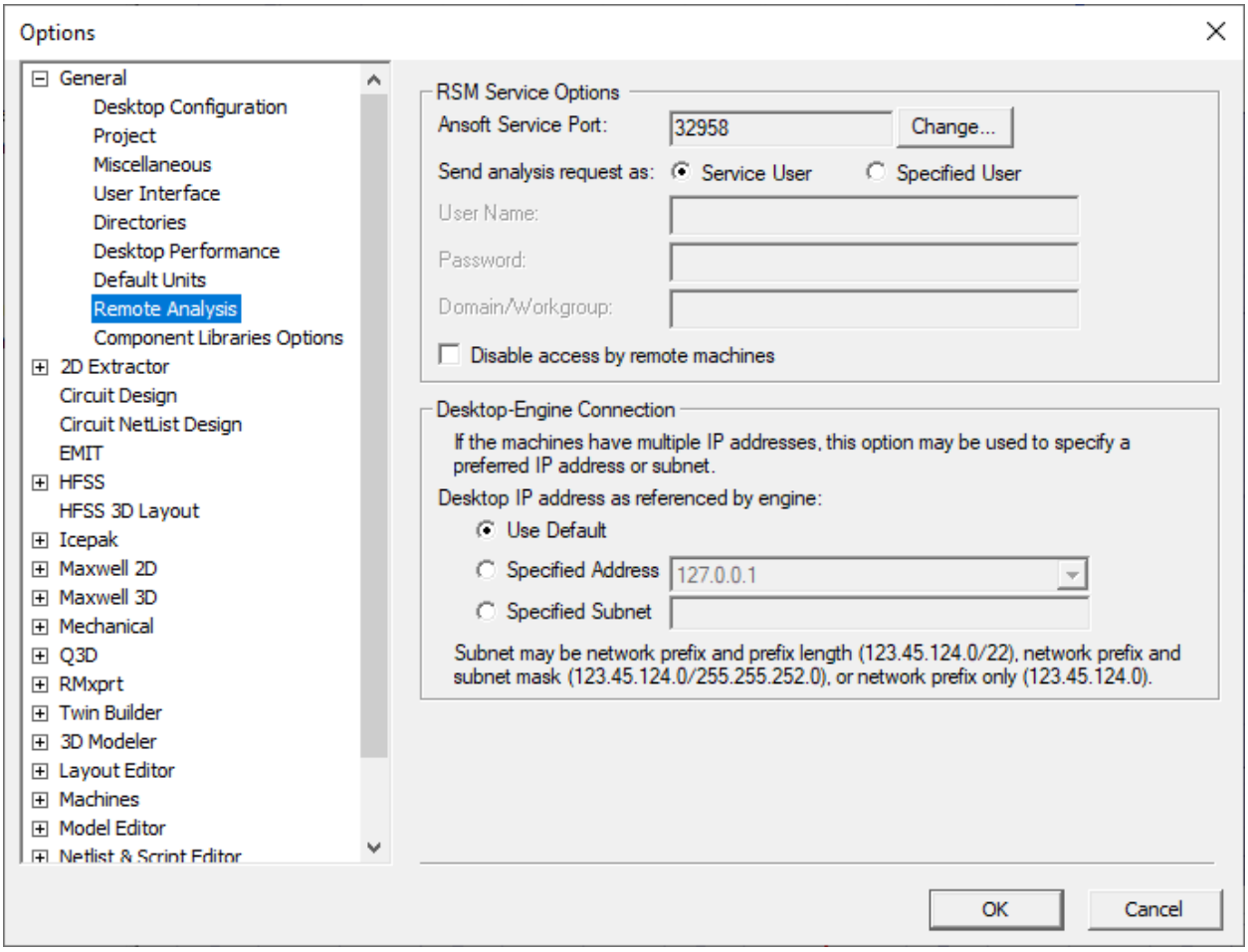
General Options: Default Units

Under General Options, **Default Units** options allow you to set default units for the following:

- Length
- Angle
- Time
- Temperature
- Torque
- Magnetic Induction
- Pressure
- Frequency
- Power
- Voltage
- Current
- Speed
- Mass
- Conductance
- Resistance
- Inductance
- Capacitance
- Force
- Angular Speed
- Magnetic Field Strength

General Options: Remote Analysis Options

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 Linux-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).

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:

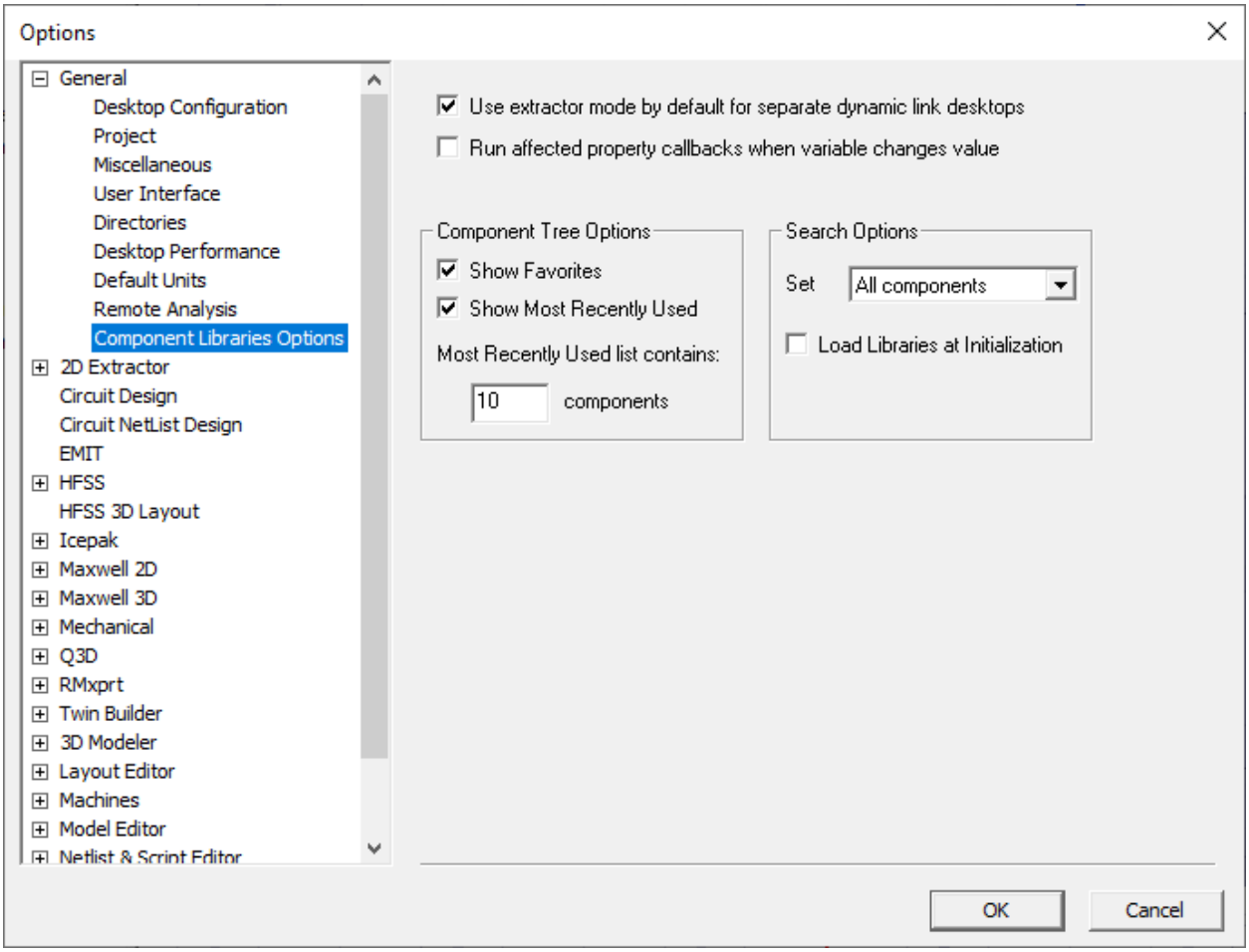
Specify the ListenPort within a 'CommDetails' block, which must be within a 'Default:CommDetails' block, which must be within the top level block of the file (the 'AnsoftCOMDaemon' block). The following example changes the listen port from 32958 to 32957, with these blocks at the beginning of the file:

```
$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.

General Options: Component Libraries Options

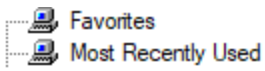
Under General Options, **Component Libraries** options allow you to change how Electronics Desktop handles components.



You can elect to **Use extractor mode by default for separate dynamic link desktops**, or to **Run affected property callbacks when variable changes value**.

In the **Component Tree Options** area, options include:

- **Show Favorites** – enables the **Favorites** folder in the **Component Libraries** tree.
- **Show Most Recently Used** – enables the **Most Recently Used** folder in the **Component Libraries** tree. Use the field to specify the number of components shown in the **Most Recently Used** folder. The default is 10.



In the **Search Options** area, options include:

- **Set** – specifies the search set when searching the Component Libraries. Choose to search **All components**, **Current list only**, or **Append to current list**.

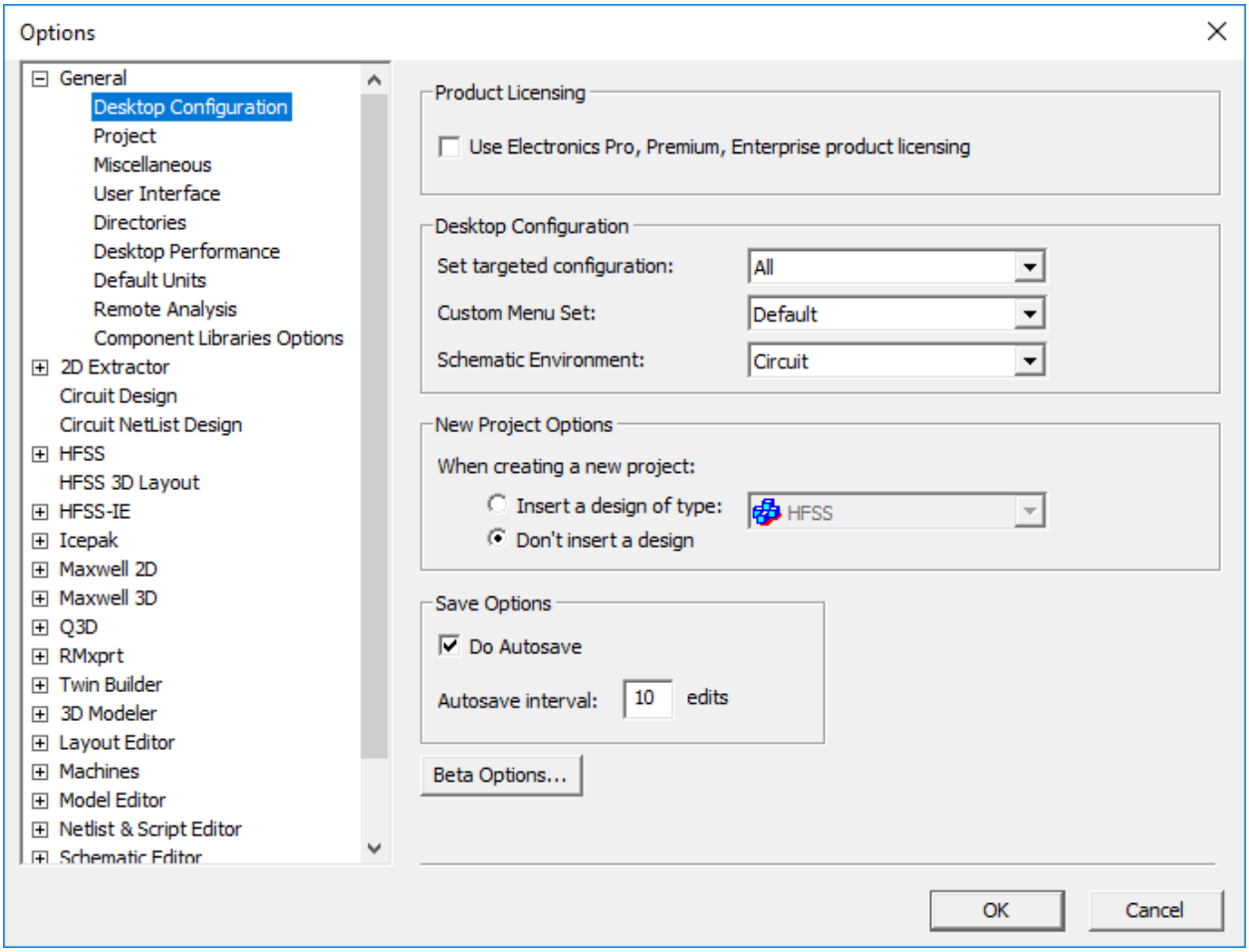
- **Load Libraries at Initialization** – this option can slow initialization.

Setting 2D Extractor Options

To set 2D Extractor options in Electronics Desktop:

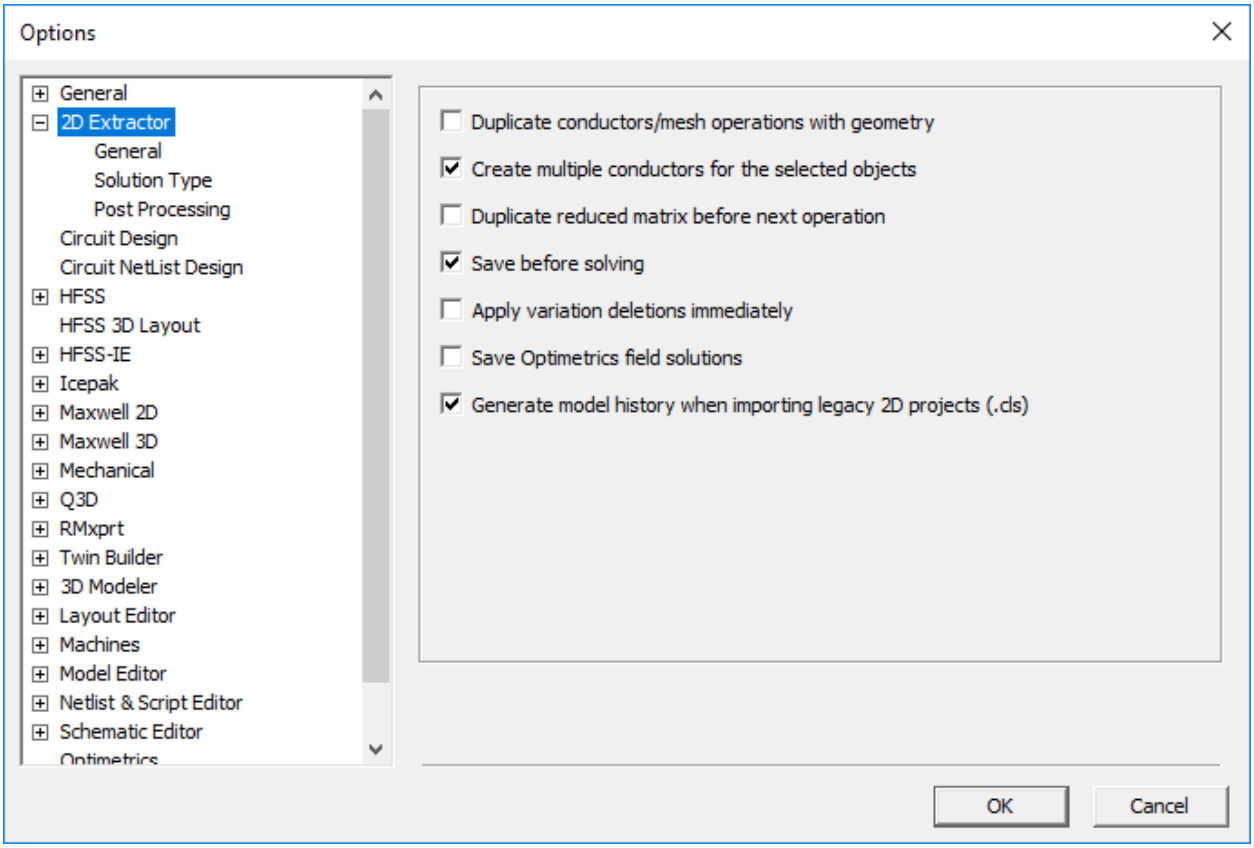
1. Click **Tools** > **Options** > **General Options**.

   The **Options** window appears, displaying **General** options.



2. In the left pane, expand **2D Extractor**.

   The **2D Extractor** options expand. The 2D Extractor **General** options display by default.



Use the tree in the left pane to select a group of options.

In 2D Extractor **General** options, you can set the following:

- **Duplicate conductors/mesh operations with geometry** – when enabled, conductors are copied along with the geometry.
- **Create multiple conductors for selected objects** – when enabled, if you select multiple objects and assign a conductor, multiple conductors are created (one for each body). When disabled, a single conductor is created and assigned to multiple objects.
- **Duplicate reduced matrix before next operation** – enables duplicating reduced matrices. See: Reduced Matrices.
- **Save before solving** – forces a full save before running the solver. If a simulation is started while another is running, you will be asked if you want solve without saving first. You can run multiple simulations, and if you have not edited the project in between solutions, crash recovery will work. In any case, you can start a new solve while running another without having to abort the current simulation.
- **Apply variation deletions immediately** – select to save disk space.

- **Save Optimetrics field solutions** – when enabled, saves the field solution data for every solved design variation.
- **Generate model history when importing legacy 2D projects (.ds)** – when enabled, Electronics Desktop generates a model history for legacy projects.

In 2D Extractor **Solution Type** options, select a **Solution Type**:

- **Open** – a surface ground does not enclose the design, and a balloon boundary condition is applied internally. In this case, you cannot create the surface ground conductor.
- **Closed** – you must create a surface ground conductor to enclose the design. No balloon boundary condition is applied. This solution type expects a surface ground to be present in the design. However, if there is no surface ground, you can still create an ideal ground, which allows your design to be validated.

  See: Balloon Boundary.

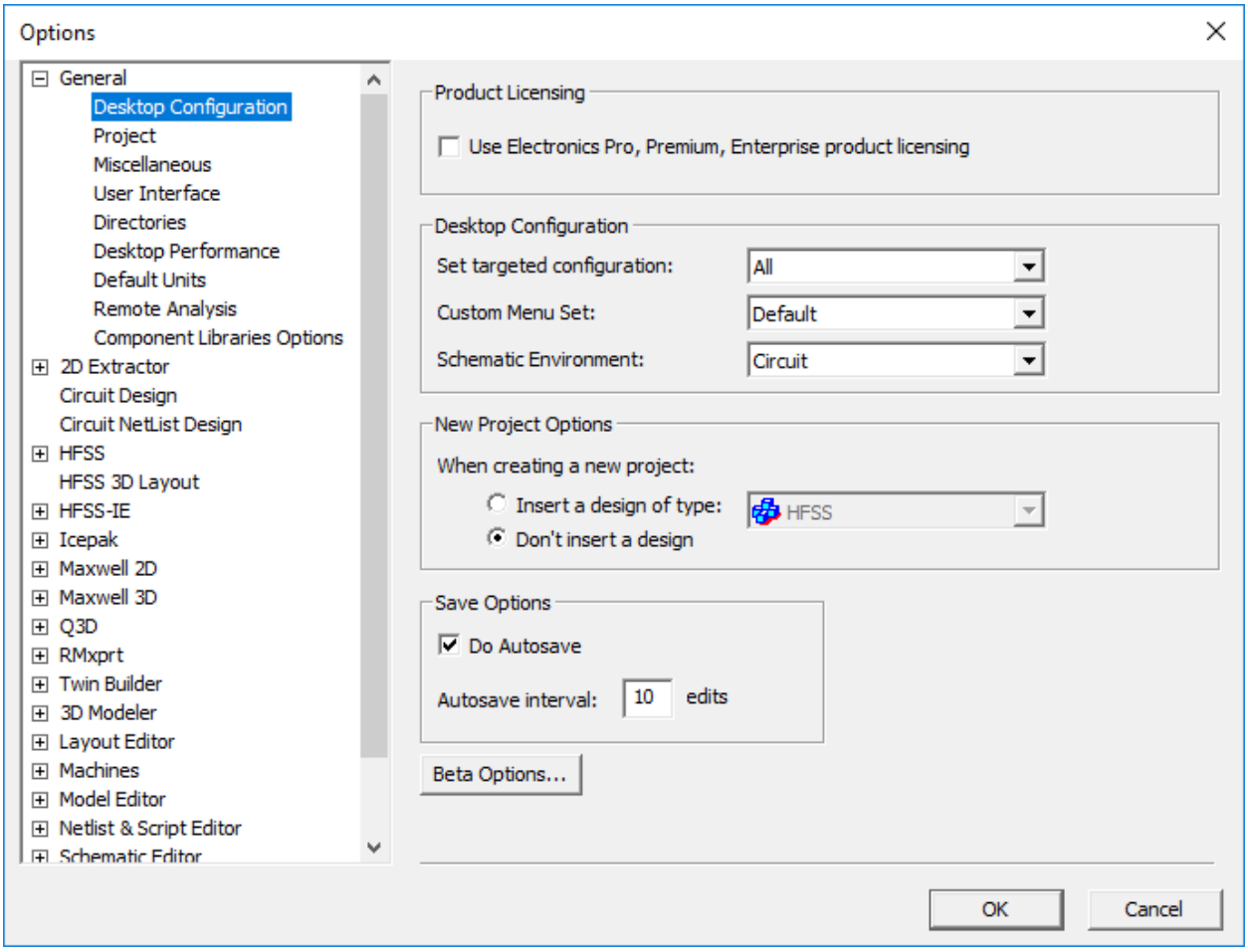
In 2D Extractor **Post Processing** options, select a **Default matrix sort order** of either **Ascending alphanumeric** or **User specified (default to creation order)**.

Setting Q3D Extractor Options

To set Q3D Extractor options in Electronics Desktop:

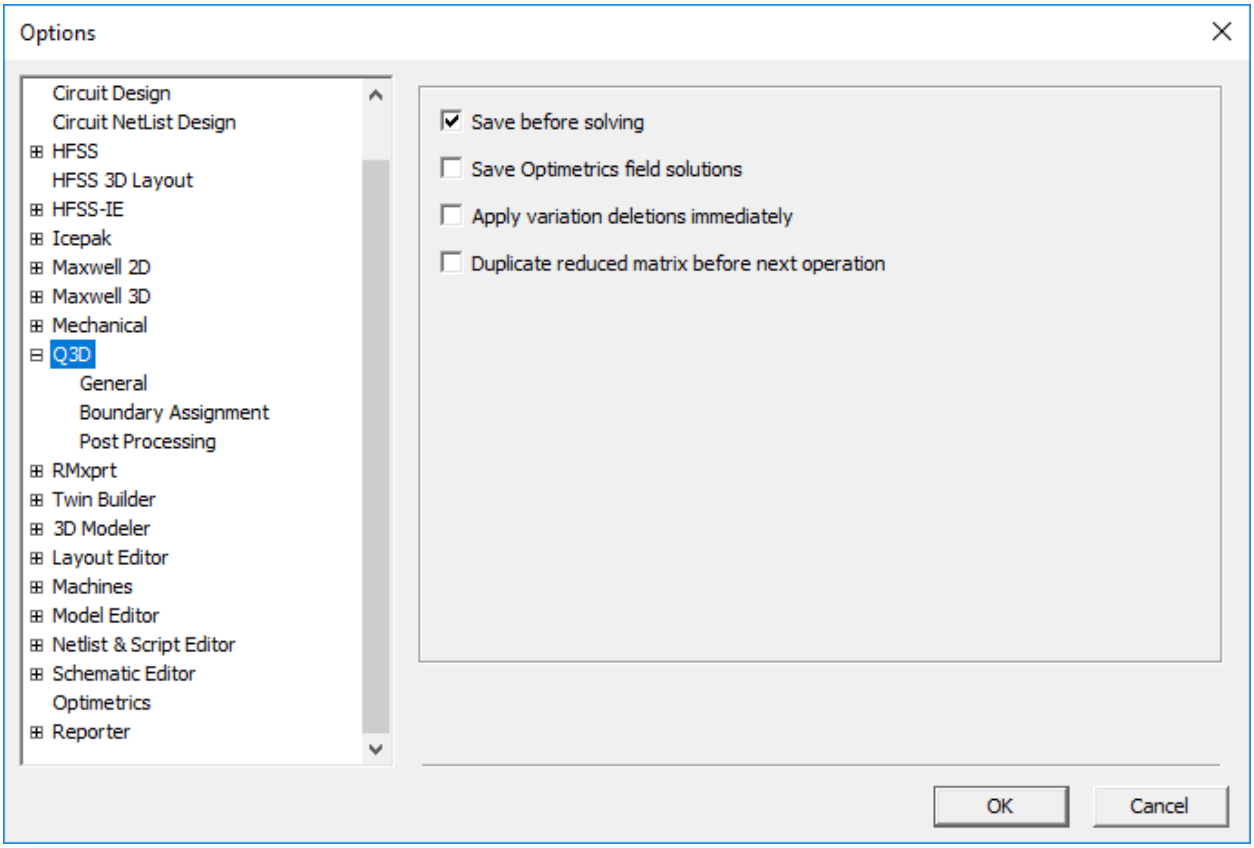
1. Click **Tools** > **Options** > **General Options**.

   The **Options** window appears, displaying **General** options.



2. In the left pane, expand **Q3D**.

   The **Q3D** options expand. The Q3D Extractor **General** options display by default.



Use the tree in the left pane to select a group of options.

In Q3D Extractor **General** options, you can set the following:

- **Save before solving** – forces a full save before running the solver. If a simulation is started while another is running, you will be asked if you want solve without saving first. You can run multiple simulations, and if you have not edited the project in between solutions, crash recovery will work. In any case, you can start a new solve while running another without having to abort the current simulation.
- **Save Optimetrics field solutions** – when enabled, saves the field solution data for every solved design variation.
- **Apply variation deletions immediately** – select to save disk space.
- **Duplicate reduced matrix before next operation** – enables duplicating reduced matrices. See: Reduced Matrices.

In Q3D Extractor **Boundary Assignment** options, you can set the following:

- **Duplicate conductors/mesh operations with geometry** – when enabled, conductors are copied along with the geometry.
- **Visualize Net Geometries** – when enabled, net geometries are visible.

In Q3D Extractor **Post Processing** options, select a **Default matrix sort order** of either **Ascending alphanumeric** or **User specified (default to creation order)**.

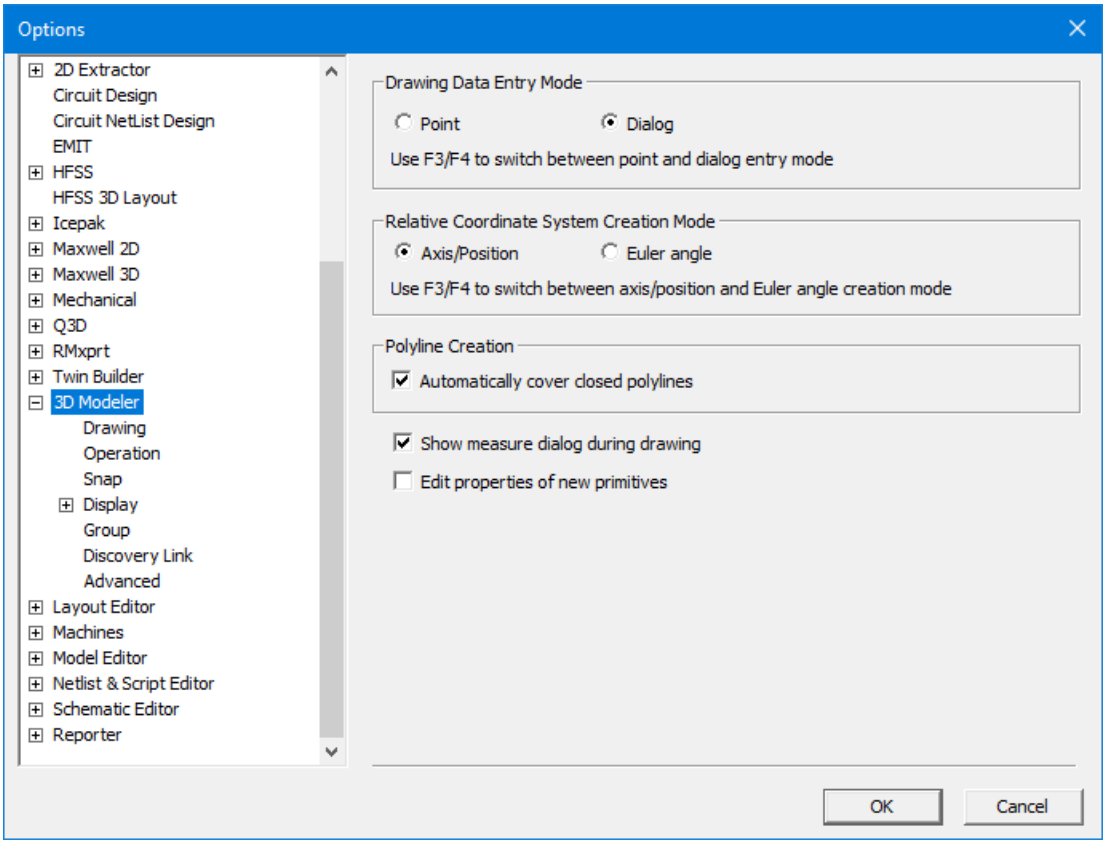
Setting 3D Modeler Options

To set 3D Modeler options in Ansys Electronics Desktop:

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

   The **Options** window opens with the Desktop Configuration options selected by default.

   In the left pane, select 3D Modeler to view modeler options:



   In the left pane, select the entries below **General** to display the associated options:

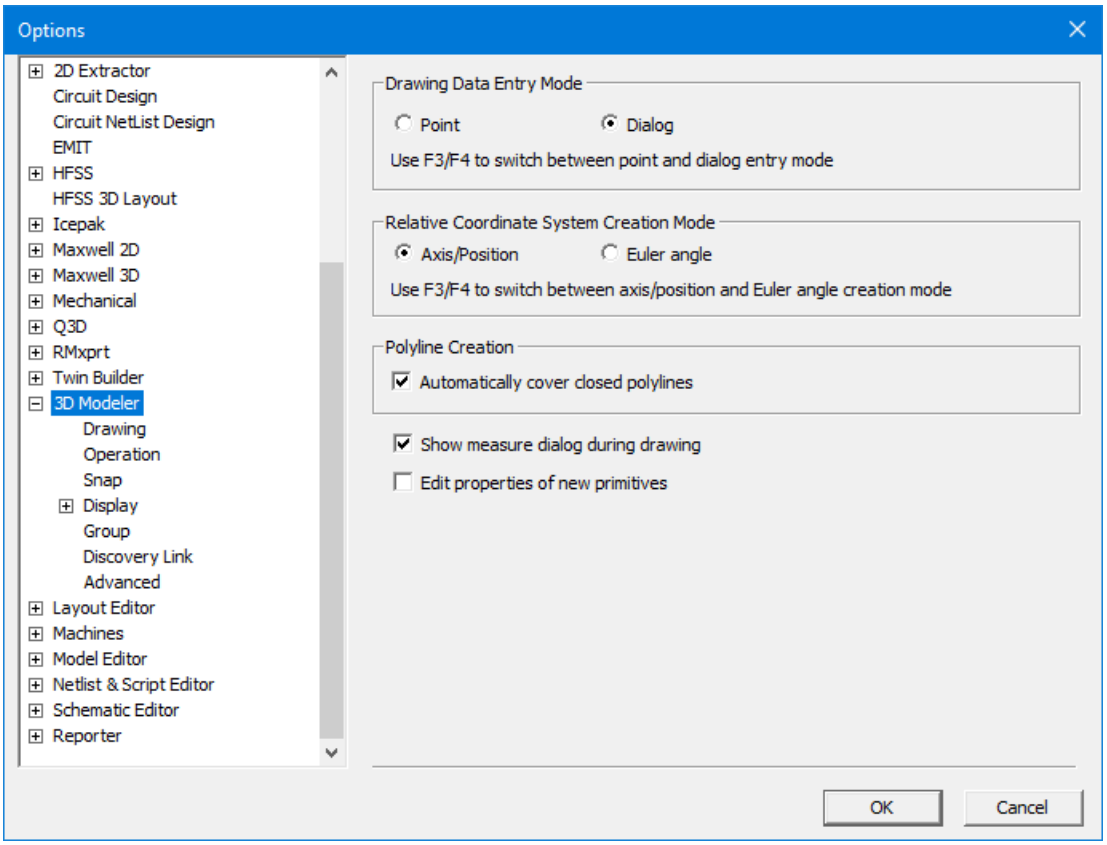
   - Drawing
   - Operation
   - Snap
   - Display
   - Group

- Discovery Link
- Advanced

2. Click each entry and make the desired selections.
3. Click **OK** to apply your preferences.

3D Modeler Options: Drawing Options

Under 3D Modeler options, **Drawing** options allow you to decide how Electronics Desktop draws objects.



In the **Drawing Data Entry Mode** area, select **Point** to draw new objects directly via the mouse, or select **Dialog** to prompt a **Properties** dialog box that allows you to enter object dimensions. Dialog mode works with the equation-based line and all two- and three-dimensional objects.

In the **Relative Coordinate System Creation Mode** area, select **Axis/Position** or **Euler angle**. This determines how the relative coordinate system is created from the **Modeler** > **Coordinate System** > **Create** > **Relative CS** menu. When **Axis/Position** is selected, relative coordinate systems are created in Draw mode. When **Euler angle** is selected, they are created from values entered in a dialog box.

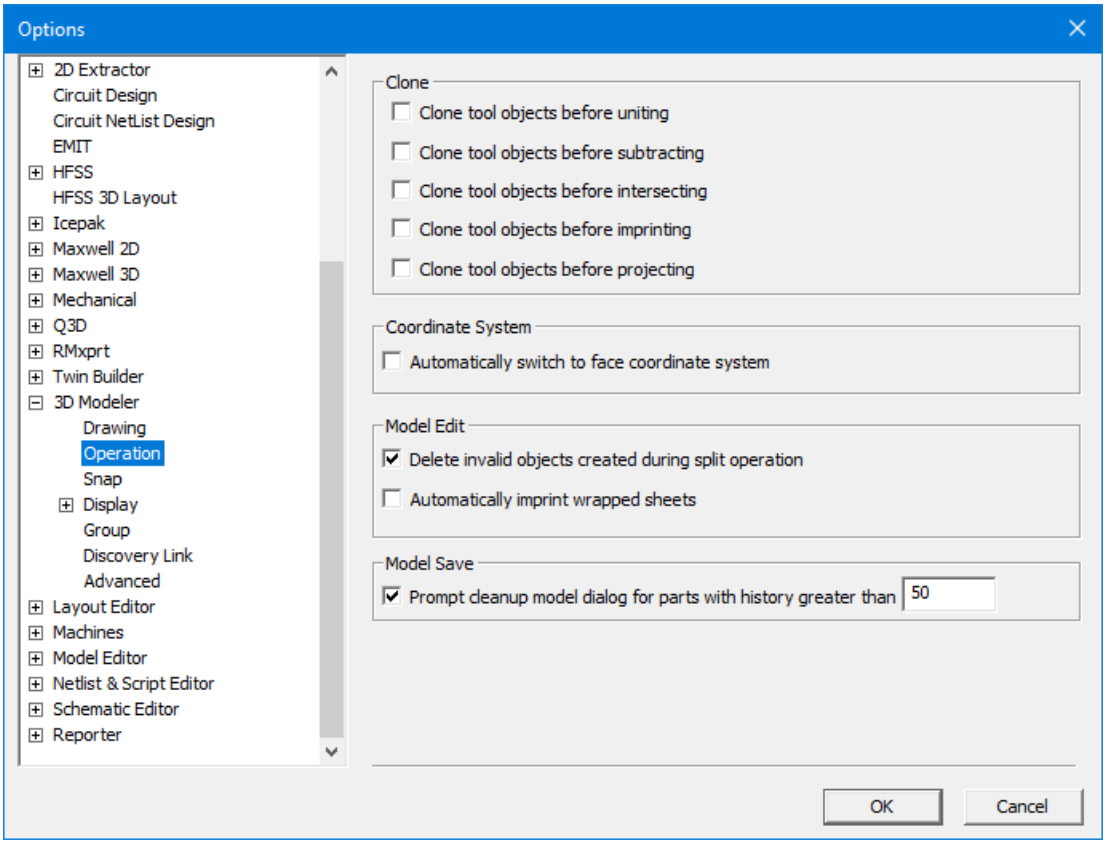
In the **Polyline Creation** area, enable or disable **Automatically cover closed polylines**. When enabled, polylines become sheet objects and are listed as such in the History tree. When deselected, closed polylines are left as uncovered objects and listed as lines in the History tree.

Other options are:

- **Show measure dialog during drawing** – specifies whether a **Measure** dialog box appears upon creation of a new primitive. The dialog box shows the coordinates of the current cursor position.
- **Edit properties of new primitives** – specifies whether a **Properties** dialog box display when you create a new object in the modeling workspace.

3D Modeler Options: Operation Options

Under 3D Modeler options, **Operation** options allow you to select options for clone, coordinate system, model editing, and model save operations.



In the **Clone** area, enable or disable options. By default, the modeler deletes tool objects when performing operations on objects. You may specify that the modeler make a clone (copy) of the tool object before these operations:

- **Clone tool objects before uniting**
- **Clone tool objects before subtracting**
- **Clone tool objects before intersecting**
- **Clone tool objects before imprinting**
- **Clone tool objects before projecting**

In the **Coordinate System** area, decide whether or not to **Automatically switch to face coordinate system**.

By default, the modeler operates within the user-selected coordinate system. If this option is enabled, when you select a face and create a new object, the modeler first creates a face coordinate system consistent with the selected face and the new object is created within the face coordinate system. When this option is disabled, you must manually create a Face Coordinate System before creating an object related to it.

In the **Model Edit** area, enable or disable the following options:

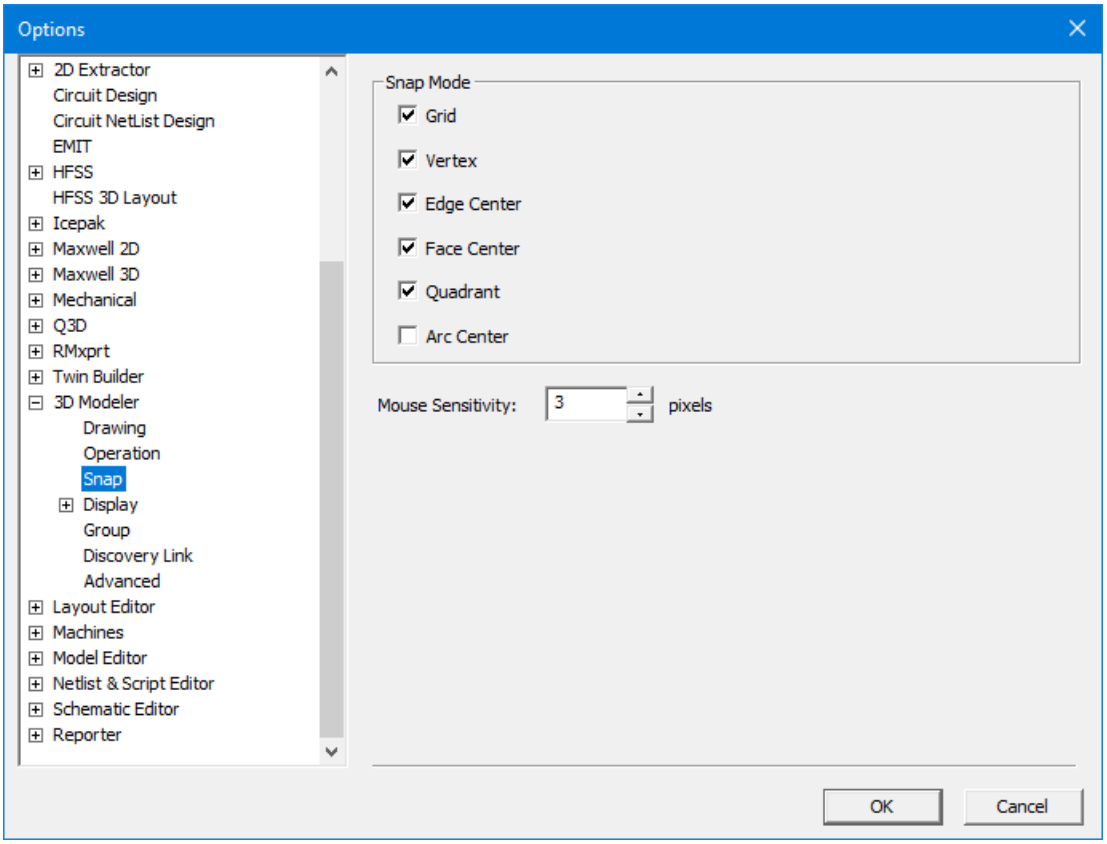
- **Delete invalid objects created during split operation** – When selected, the modeler deletes invalid objects created during split operation. When deselected, invalid objects can be freely created.
- **Automatically imprint wrapped sheets** – Imprinting is a boolean operation, whereas wrapping is not.

  See: Imprinting an Object and Wrap Sheet Command.

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 Options

Under 3D Modeler options, **Snap** options allow you to decide how Electronics Desktop snaps objects. By default, most objects are set to snap (adhere) to the nearest point on the grid. The coordinates of this point are used, rather than the exact location of the mouse.



In the **Snap Mode** area, select the points you would like the cursor to snap to:

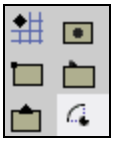
- Grid
- Vertex
- Edge Center
- Face Center
- Quadrant
- Arc Center

Use the **Mouse Sensitivity** option to select how many pixels you must move the mouse before a different snap point is selected. Values between 1 and 20 are accepted, with 1 being the most sensitive and 20 being the least sensitive. The default is 3.

You can also access snap settings:

- **From the Modeling Workspace** – While in draw mode, right-click and select **Snap Mode**.
- **From the Toolbar** – Select **Modeler** > **Snap Mode**.

- **From the Draw Ribbon** – Select and deselect the snap icons.

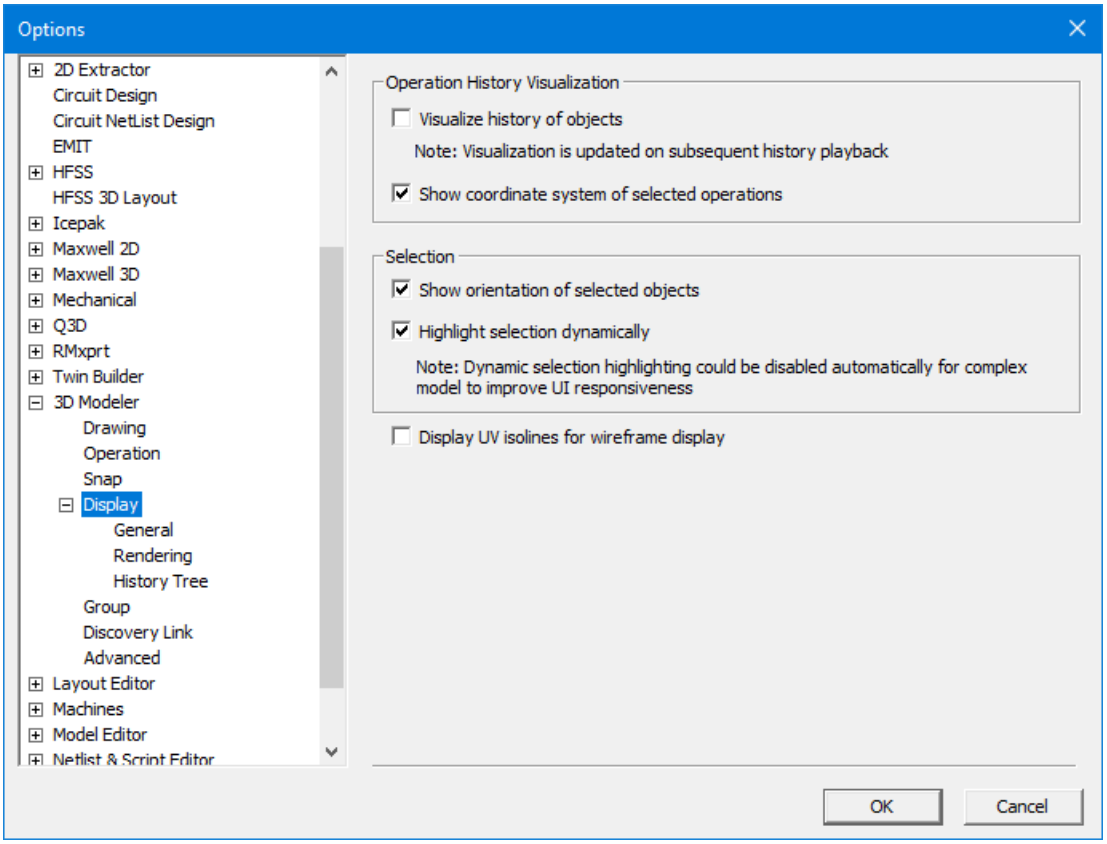


Hover over each icon for a tooltip indicating its equivalent menu option.

> **Important:**
>
> In general, you should select at least one of the snap options . If none is selected, the software is in "free mode" and selects whatever point you click, regardless of its coordinates. This can cause problems creating closed objects because the point you select may appear to be the vertex point of an open object, but may not be the exact coordinates.

3D Modeler Options: Display Options

Under 3D Modeler options, **Display** options allow you to change how Electronics Desktop displays your project tree, history, and objects.

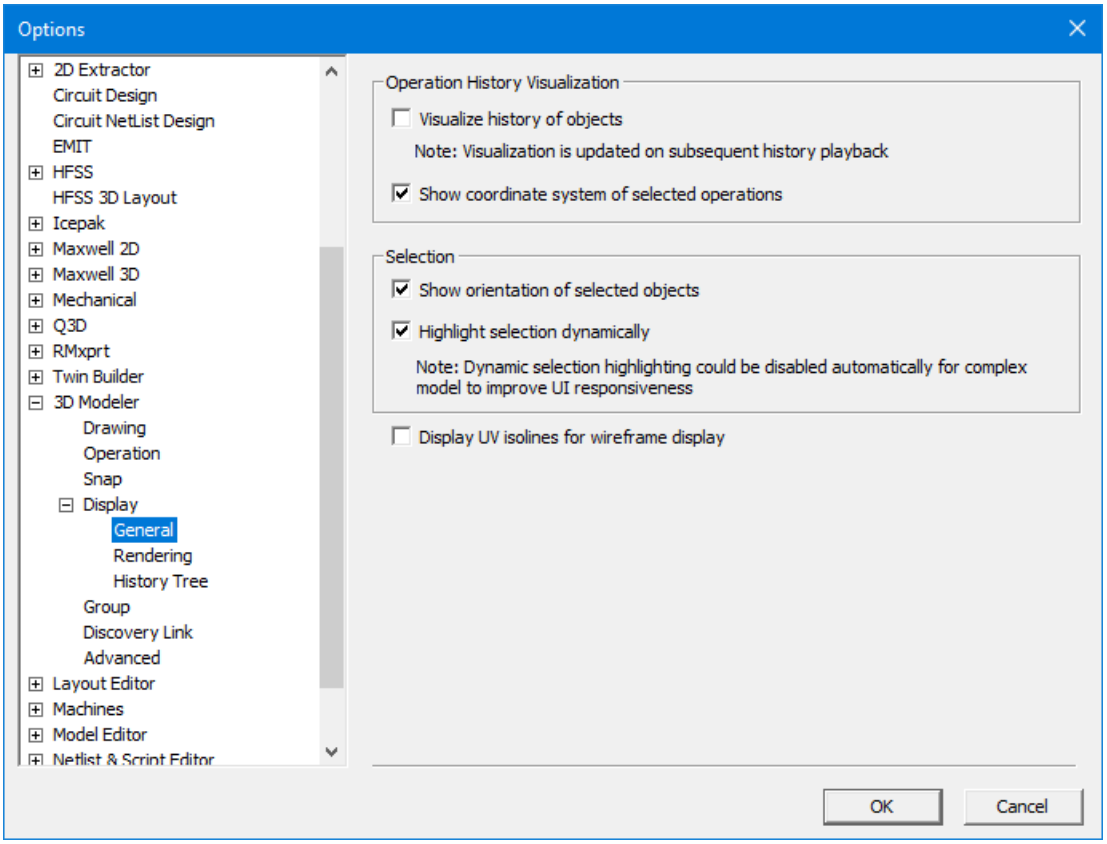


In the left pane, select the entries below **Display** to display the associated options:

- General
- Rendering
- History Tree

Display Options: General Options

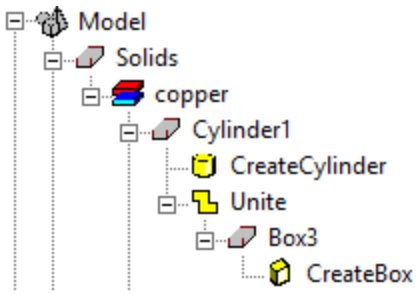
Under Display Options, **General** options allow you to change how objects are selected and how their history is visualized.



In the **Operation History Visualization** area, set the following options:

- **Visualize history of objects** – When selecting an object in the history tree, this option lets you view an outline of each part that comprises the object. This can help you visualize an object that has been merged with another object. Clearing this selection removes visualization of objects that are part of the model history. For large models, this is faster and uses less memory.

  Visualized history:



> **Note:**
>
> You must restart Electronics Desktop for this option to take effect.

- **Show coordinate system of selected operations** – Select or clear.

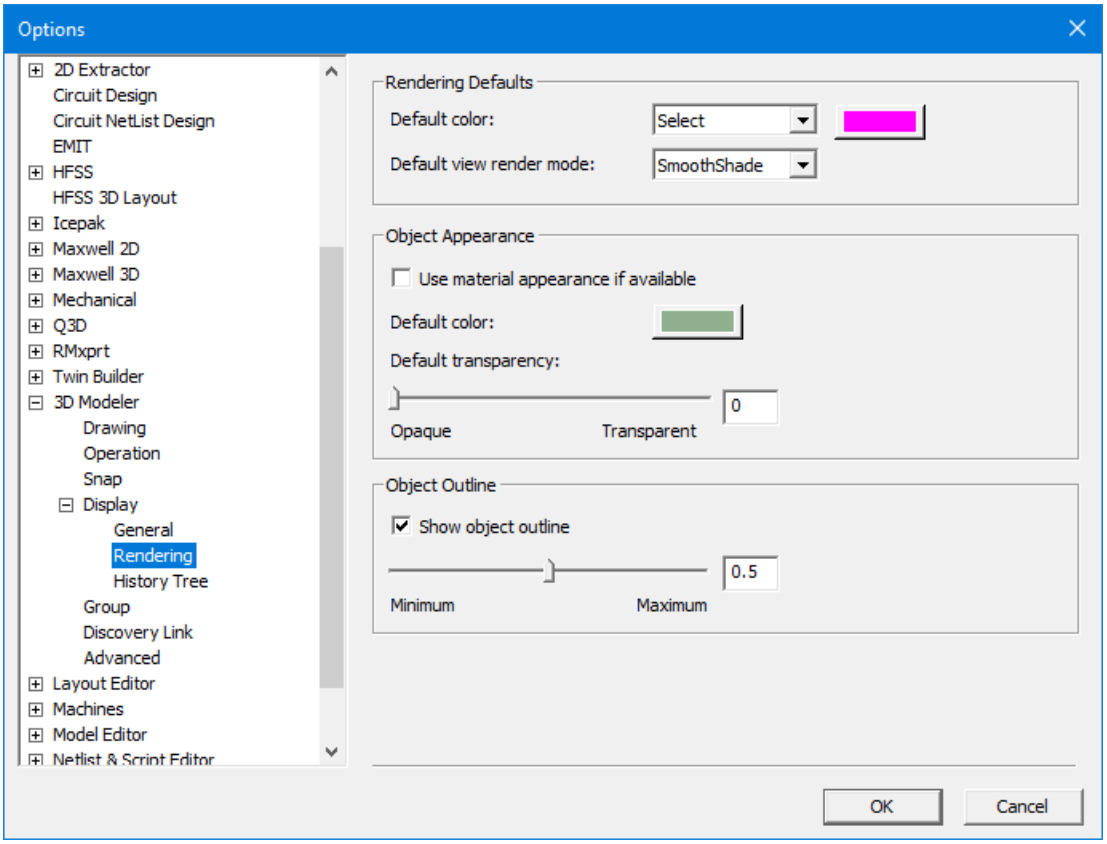
In the **Selection** area, set the following options:

- **Show orientation of selected objects** – Select or clear.
- **Highlight selection dynamically** – 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. If Electronics Desktop identifies the model to be large enough, dynamic highlighting is automatically ignored. In this case you will see a message about this.

You can also **Display UV isolines for wireframe display**. For models with curved faces, you may prefer to clear this option to simplify the wire-frame display and speed up rendering.

Display Options: Rendering Options

Under Display Options, **Rendering** options allow you to change rendering and object appearance defaults.



In the **Rendering Defaults** area, set the following options:

- **Default color** – Use the drop-down menu to choose the **Select**, **Highlight**, or **Rubberband** action. Click the color button to select a color for each action.
- **Default view render mode** – Select either **WireFrame** or **SmoothShade**. When dealing with complicated geometries, choose **WireFrame** rendering for speed.

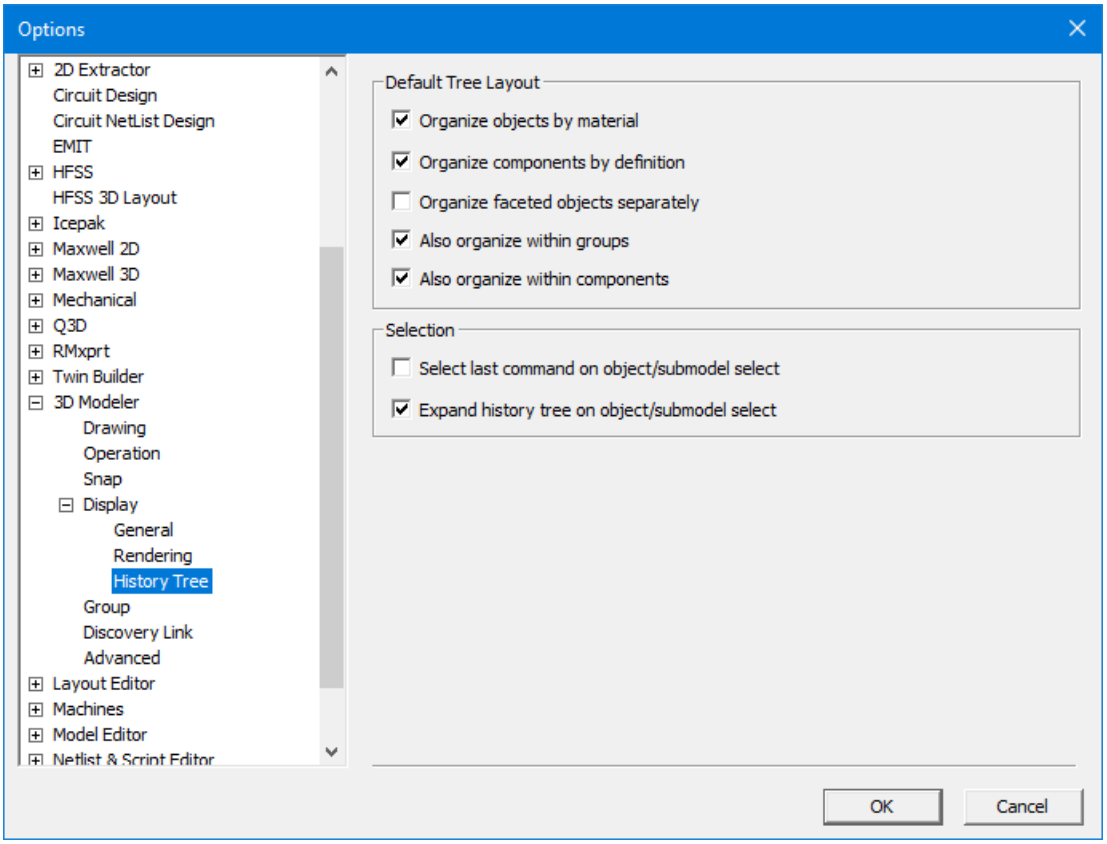
In the **Object Appearance** area, set the following options:

- **Use material appearance if available** – When this option is selected, default color and transparency used for newly created objects come from material settings. Some materials may not have default appearance specifications; in those cases, default color and transparency specified under **Rendering Defaults** are used.
- **Default color** – Click to select the default color.
- **Default transparency** – Use the slider to select the default transparency.

In the **Object Outline** area, select or clear **Show object outline**. If you elect to show the object outline, use the slider to determine the outline thickness. Value must be between 0 and 1.

Display Options: History Tree Options

Under Display Options, **History Tree** options allow you to change the history tree display and selection response.



In the **History Tree Layout** area, change the organization of the history tree as desired:

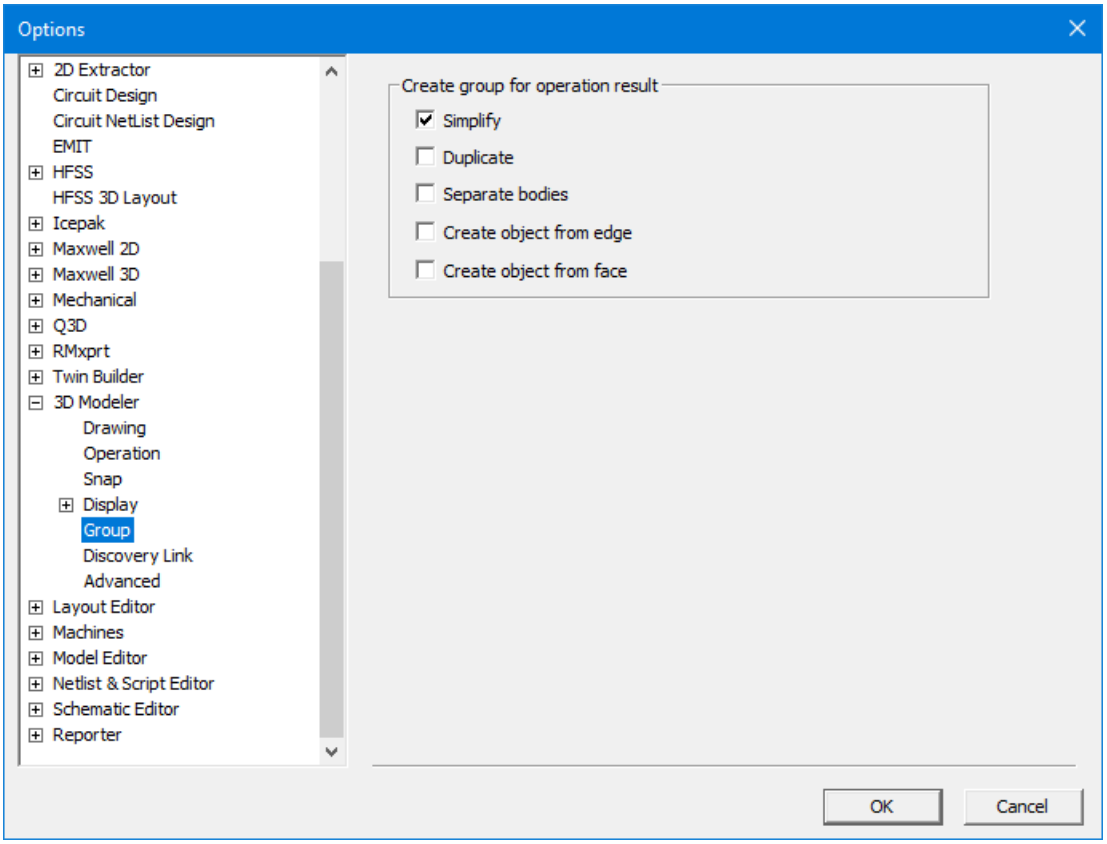
- **Organize objects by material**
- **Organize components by definition**
- **Organize faceted objects separately**
- **Also organize within groups**
- **Also organize within components**

In the **Selection** area, set the following options:

- **Select last command on object/submodel select** – When selected, the history tree is expanded after operations on object properties, even if the tree is collapsed for the item. The command is selected.
- **Expand history tree on object/submodel select** – When selected, selecting an object automatically opens the history tree.

3D Modeler Options: Group Options

Under 3D Modeler options, **Group** options determine how groups are created based on operations.



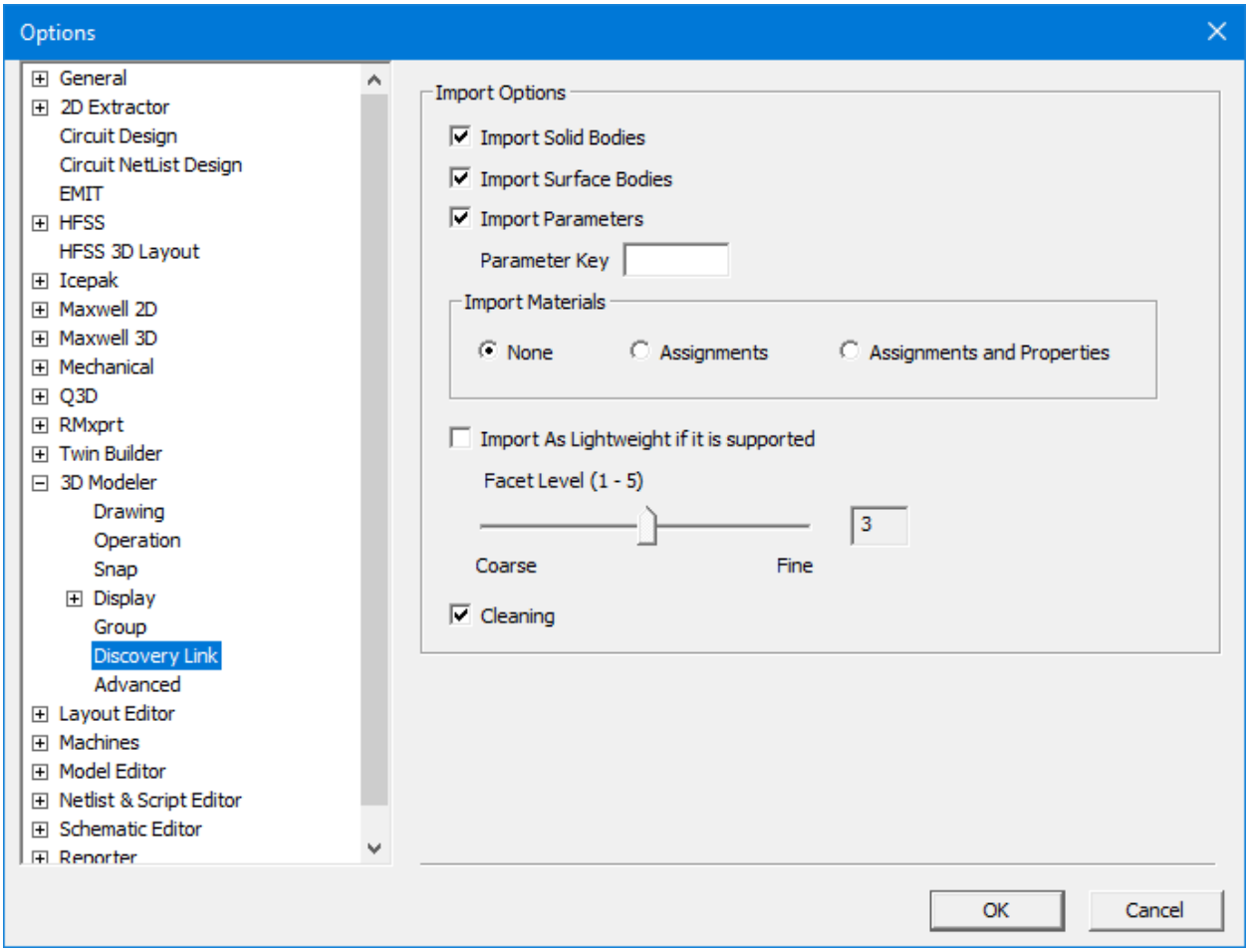
In the **Create group for operation result** area, select or clear the following actions:

- **Simplify**
- **Duplicate**
- **Separate bodies**
- **Create object from edge**
- **Create object from face**

When any selected action is performed, a group will be created in the History Tree.

3D Modeler Options: Discovery Link Options

Under 3D Modeler options, **Discovery Link** options determine how linked Discovery projects are imported.



To specify Discovery link settings, select or clear the following check boxes in the **Discovery Link** section:

- **Import Solid Bodies** – Enabled by default.
- **Import Surface Bodies** – Enabled by default.
- **Import Parameters** – Enabled by default.
- **Parameter Key** – A string used to filter out parameters. By default no string is entered and all Discovery parameters are imported.
- **Import Material Assignments** – **None** by default.
    - **Assignments** – Imports the material names without the material properties. The material properties will be obtained from a material with the same name that exists in the user's project. With this option, the system will first search for the material within the project's materials. If no matching material is found, the search will extend to the sys libraries. (See Working with Material Libraries for details on libraries.) If a material with the same name exists in a sys library, it will be imported into the

project from the library. If a material with the same name cannot be found in the project, a violation will be issued after the Discovery link is created.

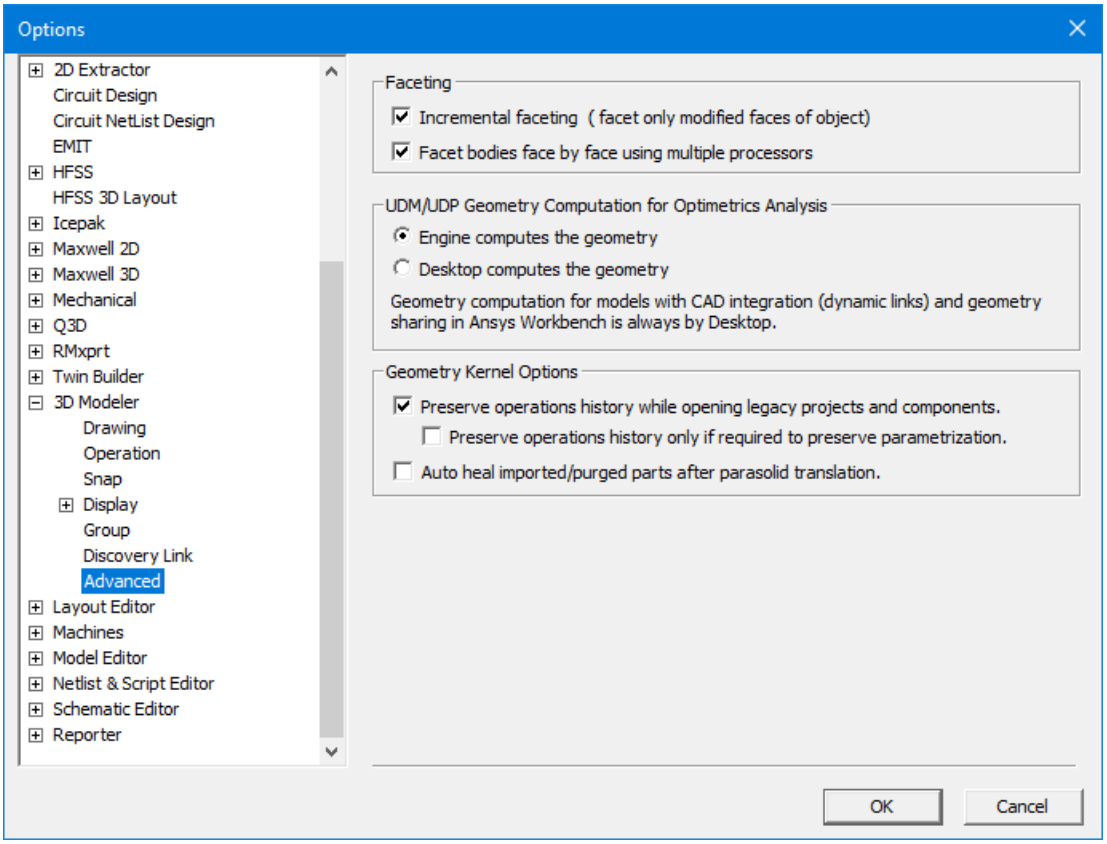
- **Assignments and properties** – Imports material with its material properties. A project material will be created in AEDT with imported properties. Note that if a project material with same name already exists, then it will be used instead of a new material being created.

> **Note:** Discovery currently only supports exporting mechanical and thermal properties, so the **Assignment with properties** option is only supported in Mechanical and Icepak designs. For HFSS, Maxwell, and Q3D designs, this option will be ignored.

- **Import As Lightweight if it is supported** – This feature allows you to import a geometry from Discovery as a lightweight geometry into a design with SBR+ solution type. The Facet Level can be set on a 1-5 range, from coarse to fine, if the geometry in Discovery is not faceted. If it is faceted in Discovery, the facets will be imported directly and the Facet Level setting will be ignored.

3D Modeler Options: Advanced Options

Under 3D Modeler options, **Advanced** options help configure faceting and UDM/UDP.



In the **Faceting** area, select **Incremental faceting** or **Facet bodies face by face using multiple processors**. Face by face will be slower.

In the **UDM/UDP Geometry Computation for Optimetrics Analysis** area, select whether **Engine** or **Desktop** computes the geometry.

In the **Geometry Kernel Options**, enable **Preserve operations history while opening legacy projects** to preserve operations history while opening legacy projects that used the ACIS kernel. When opening legacy projects, the conversion is a one-time event. A renamed copy of the original file is preserved. The default is to preserve operations history. If the option is disabled, the operations history is purged. The option to purge history is available because:

- Some operations cannot be replayed in Parasolid.
- Purging provides backup to translate geometry.
- Parameterization is lost.
- Expect a better conversion success rate with purged models.

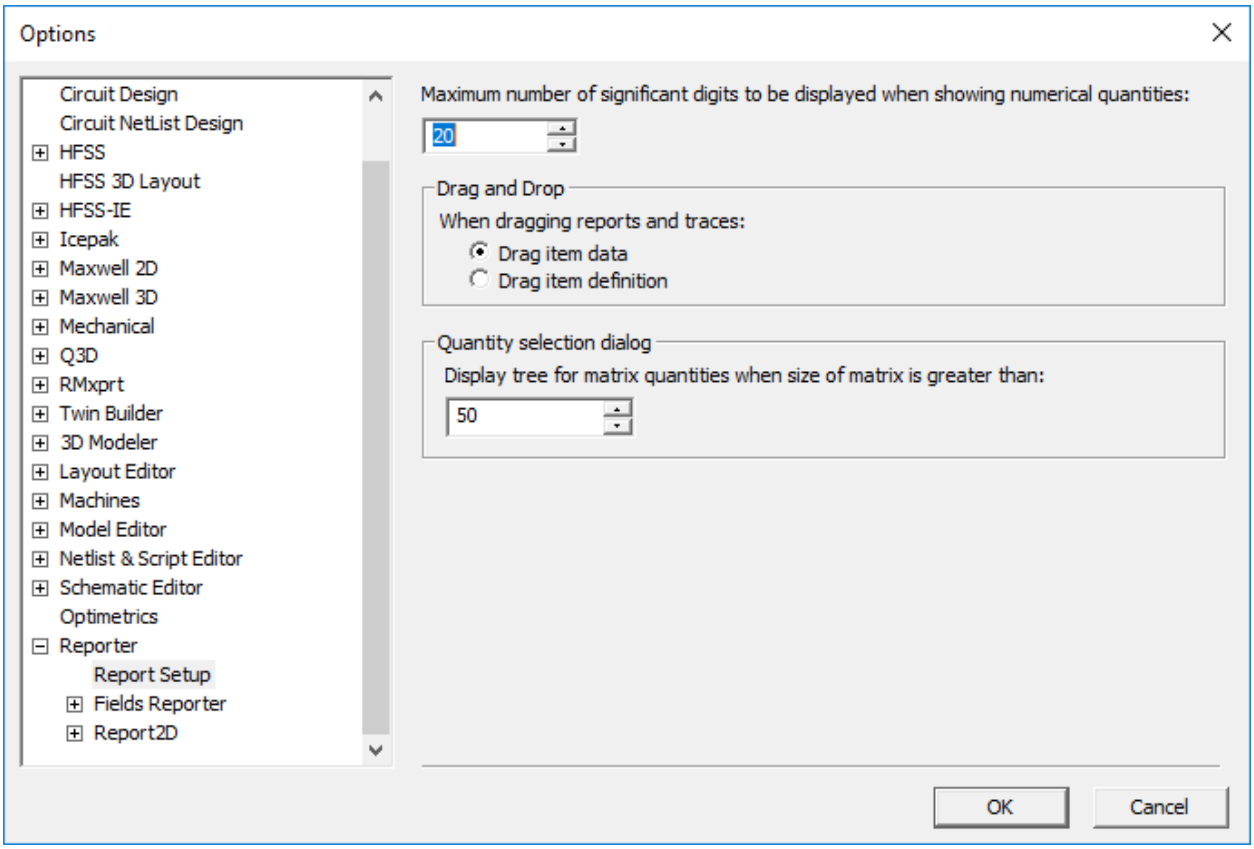
Setting Reporter Options

To set Reporter options in Ansys Electronics Desktop:

1. Click **Tools** > **Options** > **General Options**.
2. In the left pane, select **Reporter**.
3. Select the entries below **Reporter** to display the associated options:
   - Report Setup
   - Fields Reporter
   - Report2D
4. Click each entry and make the desired selections.
5. Click **OK** to apply your preferences.

Reporter Options: Report Setup Options

Under Reporter Options, **Report Setup** options allow you to customize significant digits, drag-and-drop, and quantity selection options for reports.



Set the **Maximum number of significant digits to be displayed when showing numerical quantities**. The value must be between 0 and 20.

In the **Drag and Drop** area, select how Electronics Desktop behaves when you drag reports and traces. You can either **Drag item data** or **Drag item definition**.

In the **Quantity Selection Dialog** area, select a **Display tree for matrix quantities when size of matrix is greater than** value. When the number of matrix elements is larger than the selected 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. This is useful when dealing with large matrices.

Reporter Options: Fields Reporter Options

Under Reporter Options, **Fields Reporter** options fall into three categories:

- Animation Options
- Mesh Plot Options
- Streamline Plot Options

In the left pane, select a category to view those options.

From the **Animation** options, you can set the following:

- **Group Field Overlays by Type** – If selected, field overlays are grouped in the Project Tree. If deselected, they are not.
- **Default Phase Animation Settings** – Select the default start and stop angles as well as the number of steps for Scalar and Vector plots. See: Creating Phase Animations.

From the **Mesh Plot** options, you can set the **Clipping of volume mesh plot**. When dragging a clip plane, the plot can update automatically. Select **Never** to disable this feature, **Always** to enable it at all times, or set a maximum number of mesh elements for automatic update. This option allows you to update plots for smaller meshes but avoid automatically updating larger plots that may consume too much memory. See: Using Clip Planes.

From the **Streamline Plot** options, set **Streamline drawing stopping criteria** and **Streamline marker spacing**.

Reporter Options: Report2D Options

Under Reporter Options, **Report2D** options fall into several categories:

- Curve
- Axis
- Grid
- Header
- Note
- Legend
- Marker
- Marker Table
- X/Y Markers

- Stacked
- Digital
- General
- Table

In the left pane, select a category to view those options.

Report2D Options: Curve

Curve settings change how curves are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Curve**:

- **Line Style** – use the drop-down menu to select from: Solid, Dot, ShortDash, DotShortDash, Dash, DotDash, DotDot, DotDotDash, and Long Dash.
- **Color** – double-click a color box to display the **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** – select the check box to use arrows on curve ends.
- **Show Symbol** – select 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** – use the drop-down menu to select a symbol from: Box, Circle, Vertical Ellipse, Horizontal Ellipse, Vertical Up Triangle, Vertical Down Triangle, Horizontal Left Triangle, and Horizontal Right Triangle.
- **Fill Sym** – select the check box to set the symbol display as solid. Otherwise it will display as hollow.
- **Sym Color** – double-click a color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

Report2D Options: Axis

Axis settings change how axes are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Axis**:

- **Axis Name** – this describes the axis to which a row's settings apply. You cannot change this field.
- **Color** – double-click a color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Font Color** – double-click a color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

- **Edit Font** – click **Edit Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Min Gutter %** – sets the amount of empty space (gutter) around the axis.
- **Font Description** – describes the settings applied from the **Font** window. You cannot change this field. To change the font, use **Edit Font**.

Report2D Options: Grid

Grid settings change how grids are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Grid**:

- **Grid Name** – this describes the grid to which a row's settings apply. You cannot change this field.
- **Line Style** – use the drop-down menu to select from: Solid, Dot, ShortDash, DotShortDash, Dash, DotDash, DotDot, DotDotDash, and Long Dash.
- **Line Color** – double-click a color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

Report2D Options: Header

Header settings change how headers are displayed in reports. You can separately style the Title and Subtitle.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Header**:

- **Color** – click a color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Font** – click **Edit Title Font** or **Edit Subtitle Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Company Name** – Enter a company name to appear on all reports.
- **Show Ansys Logo** – Remove the check from the box to hide the Ansys logo that appears on all reports (checked by default).

Report2D Options: Note

Note settings change how notes are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Note**:

- **Note Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Note Font** – click **Edit Note Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.

- **Background Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Background Visibility** – select the check box to make the background color visible. Deselect to make it transparent.
- **Border Line Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Border Visibility** – select the check box make the note border visible. Deselect to make it transparent.
- **Border Line Width** – set the line width by editing the real value in the text field.

Report2D Options: Legend

Legend settings change how legends are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Legend**:

- **Legend Name** – Default is no name. When non-empty, a header row for the Legend in plot shows up with that string.
- **Show Trace Name** – select the check box to show the trace name; deselect it to hide the trace name.
- **Show Solution Name** – select the check box to show the solution name; deselect it to hide the solution name.
- **Show Variation Key** – select the check box to show the variation key; deselect it to hide the variation key.
- **Highlight Curve on Hover** – select the check box to highlight a curve when you hover the cursor over it; deselect it to leave the curve as-is when you hover the cursor.
- **Text Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Text Font** – click **Edit Text Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Background Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Border Line Color** – click the color box to display the **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** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

You can set different **Text Color** and **Text Font** settings for the Header Row.

Report2D Options: Marker

Marker settings change how markers are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Marker**:

- **Marker Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Marker Font** – click **Edit Marker Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Arrow Direction** – use the drop-down menu to set the arrow direction to Up, Down, Left, or Right.

Report2D Options: Marker Table

Marker Table settings change how marker tables are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Marker Table**:

- **Text Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Text Font** – click **Edit Text Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Background Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Border Line Color** – click the color box to display the **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** – click the color box to display the **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.

You can set a different **Text Color** and **Text Font** for the header row.

Report2D Options: X-Y Markers

X-Y Marker settings change how X-Y markers are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **X-Y Markers**:

Background Colors

- **Marker [#] Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

Properties

- **On-screen intersection** – select the check box to enable on-screen intersection; deselect to disable.
- **Marker Font** – click **Edit Marker Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Text Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Line Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Line Style** – use the drop-down menu to select from: Solid, Dot, ShortDash, DotShortDash, Dash, DotDash, DotDot, DotDotDash, and Long Dash.
- **Line Width** – edit the text field to specify a line width.
- **Show Name** – select the check box to show X-Y marker names; deselect to hide.
- **Snap to Vertex** – select the check box to snap markers to the vertex; deselect to disable snapping.

Inter Marker Deltas

- **Show Delta** – select the check box to show inter-marker deltas; deselect to hide.
- **Delta Font** – click **Edit Delta Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Delta Text Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Line Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Line Style** – use the drop-down menu to select from: Solid, Dot, ShortDash, DotShortDash, Dash, DotDash, DotDot, DotDotDash, and Long Dash.
- **Line Width** – edit the text field to specify a line width.

Report2D Options: Stacked

Stacked settings change how stacked plots are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Stacked**:

- **Auto Fit Mode** – select the check box to enable auto fit; deselect to turn off autofit.
- **Stack Height in Pixels** – set 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.

Report2D Options: Digital

Digital settings change how digital stack heights are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Digital**:

- **Digital Literal Foreground** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Expand Arrays/Records** – select the check box to automatically expand arrays and records; deselect it to disable.
- **Digital Stack Height in Pixels** – enter the stack height for Analog, Digital, Enum, Event, and Literal stacks.

Report2D Options: General

These settings contain general report options.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **General**:

- **Background Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Plot Area Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Highlight Color** – click the color box to display the **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 colorkey (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** – select either **As Shown** or **Full Screen**.
- **Capture Background Color** – select either **As Shown** or **White**.

Format

- **Field Width** – enter a field width value.
- **Precision** – enter a precision value.
- **Use Scientific Notation** – select the check box to use scientific notation; deselect to disable scientific notation.

Report2D Options: Table

Table settings change how tables are displayed in reports.

From the **Options** dialog box, the following options are set in **Reporter** > **Report 2D** > **Table**:

- **Rows Per Page** – enter the number of rows you would like to display per page; the default is 2500.
- **Text Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Text Font** – click **Edit Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Border Width** – enter a value for the border width.
- **Border Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Grid Width** – enter a value for the grid width.
- **Grid Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Background Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Page Link Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.
- **Arrow Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

Header Row

- **Text Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

- **Text Font** – click **Edit Font** to open the **Font** dialog box, where you can select a font, font style (for example, italic), and font size. Then click **OK**.
- **Background Color** – click the color box to display the **Color** dialog box. Select a default or custom color and click **OK**.

Format

- **Field Width** – enter a value for the field width.
- **Precision** – enter a value for the precision.
- **Use Scientific Notation** – select the check box to enable scientific notation; deselect it to disable scientific notation.

Copy to Clipboard

- **With Header** – select to include header when copying the table to a clipboard; deselect to remove it.
- **With Tab Separator** – select to include tab separator when copying the table to a clipboard; deselect to remove it.

Setting HPC and Analysis Options

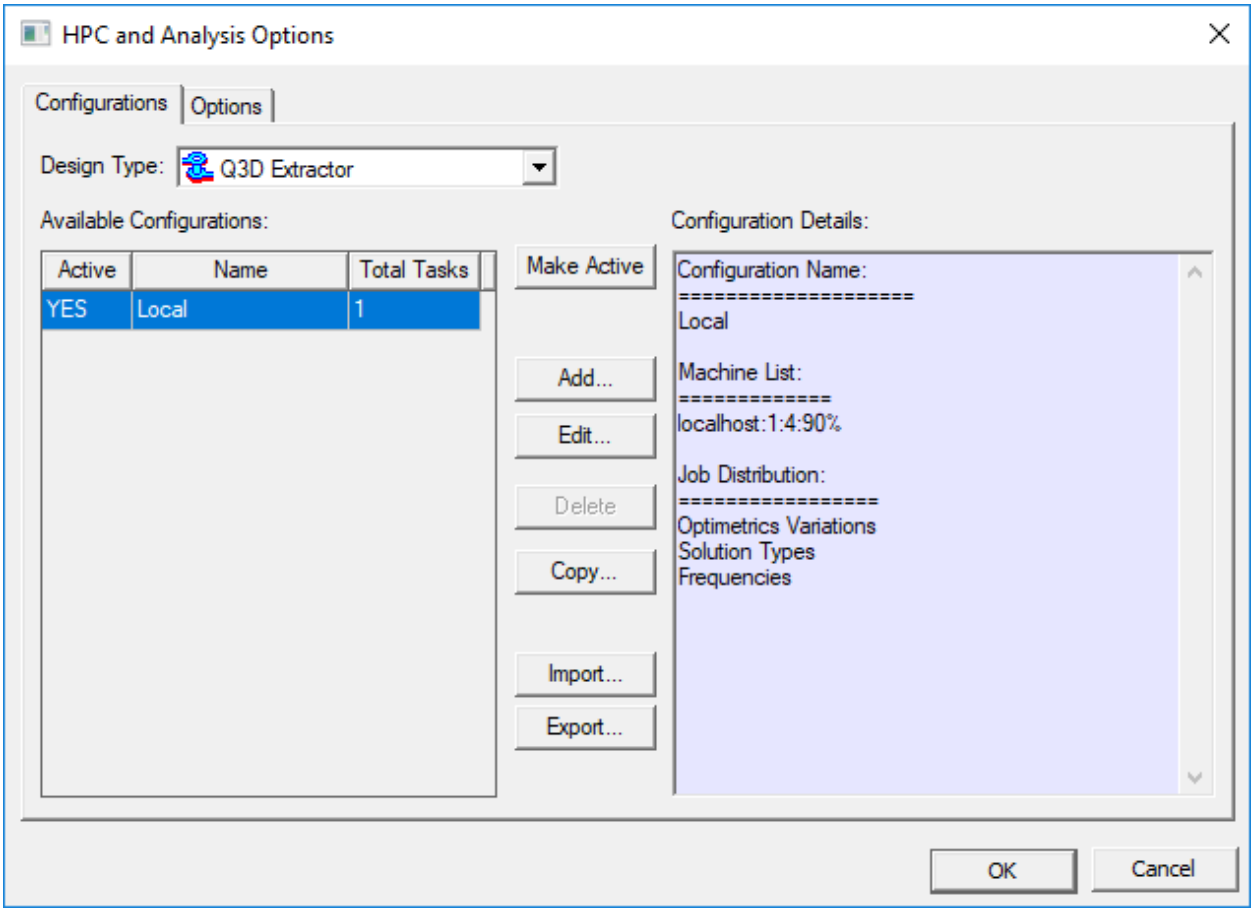
All analysis parameters are accessed via a single window. The machine list and options settings have been integrated into analysis configurations. The default configuration is for solving on a single, local machine. You can create many analysis configurations for remote and distributed solutions, and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

To set HPC and Analysis Options:

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

   You can also access HPC and Analysis Options using the **HPC Options** icon on the **Simulation** ribbon.

   The **HPC and Analysis Options** window appears, displaying the **Configurations** tab.

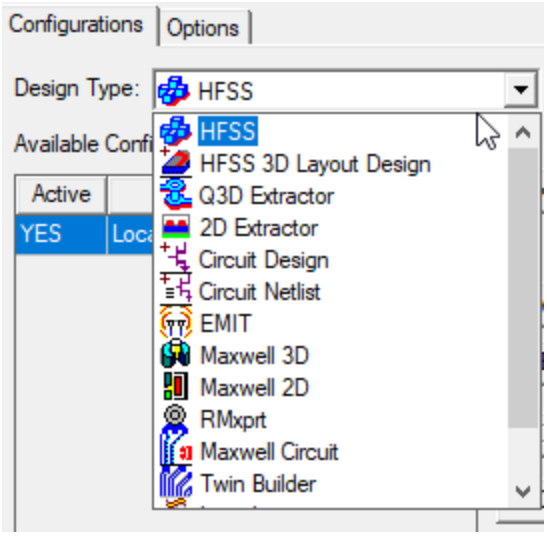


Configurations Tab

Available Configurations are described by **Name**, whether or not they are **Active**, and the **Total Tasks** the configuration can execute.Selecting a configuration from the list displays the details of that configuration in the **Configuration Details** panel.

From the **Configurations** tab, select the **Design Type** to display a list of available configurations for that type.

Configurations must be defined for all design types separately. To use similar analysis parameters for different design types, create separate analysis configurations for each design type. The active configuration is used when solving an analysis for that design type.



Selecting an Available Configuration

To activate a configuration, select it from the **Available Configurations** list and click **Make Active**. The active configuration will be indicated by a **YES** in the **Active** column.

Additional options include:

- **Add** – launches a dialog box to create a new analysis configuration.
- **Edit** – launches a dialog box to edit the currently selected analysis configuration.
- **Delete** – deletes the currently selected analysis configuration(s).

  > **Note:** You cannot delete the Local configuration.

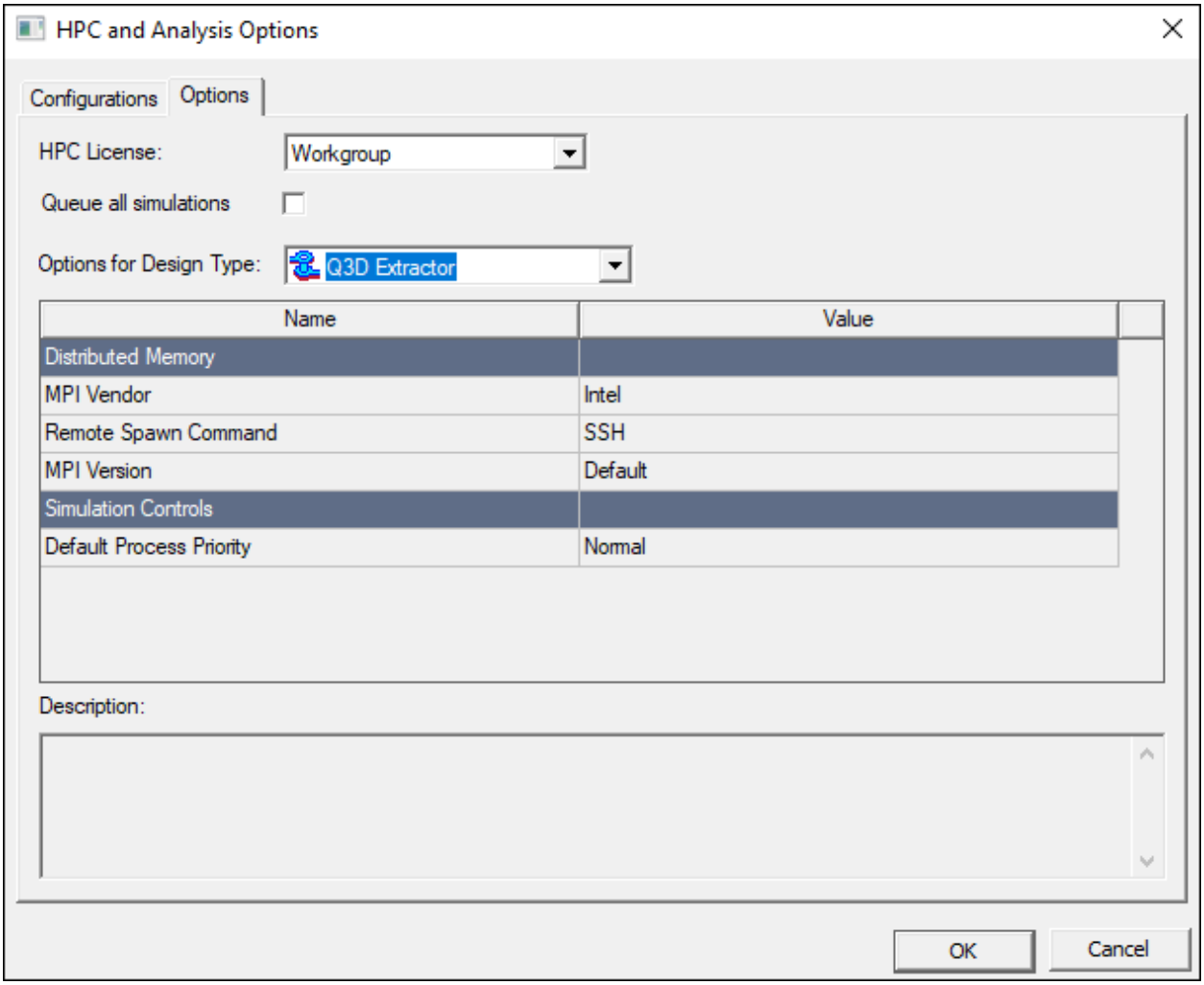
- **Copy** – creates a new analysis configuration, and launches a dialog box to edit it. If the dialog box is canceled, the new analysis configuration is not created.
- **Import** – allows you to import an *.acf file to create an analysis configuration.

  > **Note:** Importing analysis configurations always adds the imported analysis configurations to the current design type. If there is a name conflict between an imported analysis configuration and an existing analysis configuration, the imported configuration is renamed and a notification appears.

- **Export** – allows you to export the selected analysis configurations to an *.acf file. You can then import the configurations into a different design type, or import them on a different machine.

Options Tab

The **Options** tab in the **HPC and Analysis Options** dialog box contains general and product-specific settings.



> **Important:**
>
> Available options vary by design type. Certain items discussed on this page may not be visible, depending on the currently selected design type.

These options are not specified for, or saved as part of, the current analysis configuration. Instead, they are global and are always in effect for the given design type when both of the following conditions are true:

- A design of the matching design type is being solved.
- You have not specified corresponding overriding batch options on the command line.

From the **Options** tab, you can:

- Choose the HPC license type
- Enable queuing
- Specify Distributed Memory settings (for example, MPI for certain solvers)
- Set licensing options
- Enable GPU (for Transient, Matrix and SBR+ solves)
- Set the Default Process Priority

> **Note:**
>
> Solving with Intel MPI on a single Windows machine does not require MPI installation, but registration of the user's password with Intel MPI on the host is required if using password authentication. Solving with Microsoft MPI on a single Windows machine requires Microsoft MPI to be installed and the MS-MPI Launch Service to be started. These steps are not required for Microsoft MPI analysis on a Windows HPC cluster. Solving with Microsoft MPI on multiple Windows hosts is only supported on a Windows HPC cluster.
>
> Users running on Linux do not need to install MPI manually.

For **HPC License**, select **Auto**, **Workgroup** or **Pack**.

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.

- **Workgroup** (formerly "pool") – One HPC workgroup 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,... , in the context of a single simulation.
- **Auto** – delegate the choice of Workgroup or Pack licensing to the Ansys Licensing Settings tool which is a separate application installed along with Electronics Desktop.

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 begins 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 workgroup licenses, or two HPC pack licenses.

HPC licenses enable all parallel and distributed simulations, including distributed variations. Distributed variations require a single set of solver licenses, plus HPC to enable the variations.

For HPC Workgroup, distributing *N* variations requires 8*(*N*–1) workgroup licenses and, together with the solver licenses, enables up to four HPC units per variation. Each additional set of *N* workgroup licenses will enable one additional HPC unit per variation. For HPC Pack, distributing *N* variations requires *N*–1 pack licenses and, together with the solver licenses, enables up to four HPC units per variation. Each additional set of *N* pack licenses will enable 8, 32, 128,... additional HPC units per variation.

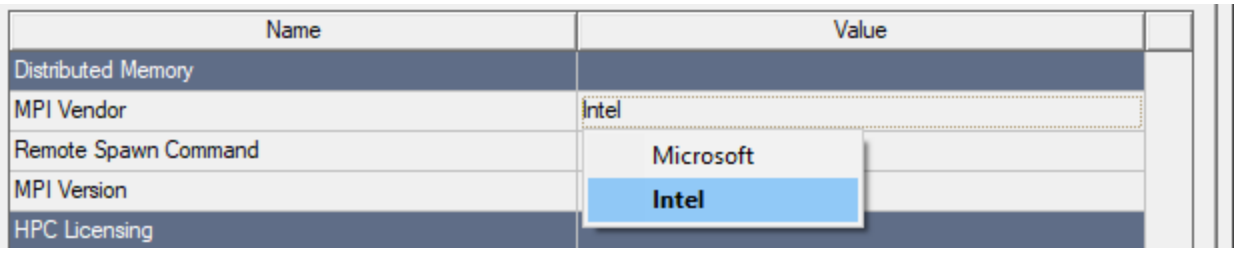
Ansys licensing supports distributed simulations when Electronics Desktop is called from other Ansys tools, such as optiSLang and Workbench. In such cases, distributed design points (variations) generally use HPC counts as described above.

> **Note:**
>
> Licensing for some calling products may include some distributed design points, in which case the total required HPC will be reduced.

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 the **General Options** and then processes them in order. You can view and change the queue by using Show Queued Simulations.

To configure options:

1. Under **Distributed Memory**, for solvers that use MPI (HFSS, HFSS 3D Layout, Icepack, Maxwell and Q3D) use the drop-down menu to select the MPI Vendor for the selected design type.



   The solvers use the industry 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 is 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

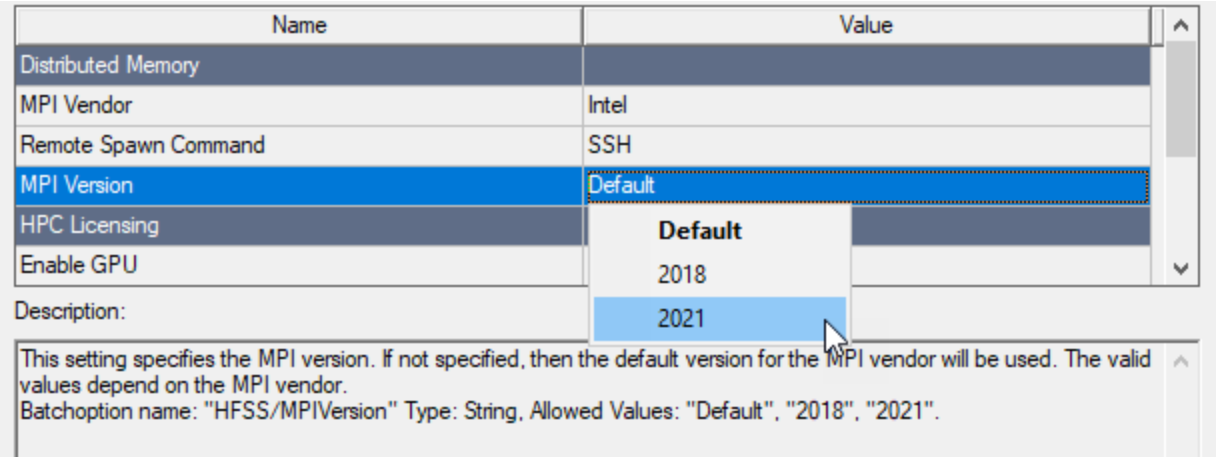
problem at hand. For solving on a single machine, MPI is not required, nor does it provide an advantage.

To use the distributed memory solution you will need to install MPI software from a supported third party vendor on all the machines you intend to use.

Depending on the MPI vendor, you may need to set passwords 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 specify the MPI version. If not specified, 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 2021 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.



Also see Setting Intel MPI Interconnect for more details.

InfiniBand Support for Windows

By default, MPI vendors use the fastest interconnect by default (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. Also see Setting Intel MPI Interconnect for more details

2. For Linux authentication, specify the Remote Spawn command as RSH or SSH (the default).

3. Optionally, you can select one of the following from the **Default Process Priority** drop-down menu:
    - **Critical** (highest) (Not recommended)
    - **Above Normal** (Not recommended)
    - **Normal** (Default)
    - **Below Normal**
    - **Idle** (lowest)

You can also set these values using VB or Python scripts.

For information on editing configurations, see Editing Distributed Machine Configurations.

Specifying the Remote Spawn Command (Linux)

An important step in using a high performance cluster is setting up authentication across machines so that the machines can be accessed without a password. By default, Q3D Extractor uses 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**.

SSH

You will need to set up passwordless access to use Q3D Extractor on a Linux cluster with SSH or RSH. In general, for SSH, this is accomplished by:

1. Verifying that you have working SSH servers and clients on your machines.
2. Verifying 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. Copying 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. Testing the connection. Log in 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 log in without requesting a password.

Consult the documentation for your machines and network for detailed instructions.

RSH

If you choose to use RSH, you will need to make sure RSH is installed on all the machines and set up the machines so that you are not prompted for a password. There are different ways to set up passwordless RSH, so be sure to consult 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 /etc/hosts.equiv file you should have something similar to the following:

Contents of the /etc/hosts.equiv file:

```
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. Consult your local documentation for detailed instructions and troubleshooting suggestions.

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, SGE or SLURM scheduler, and only if the MPI Vendor is "Intel".

When submitting a job using the AnsysEM 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, SGE or SLURM scheduler, and only if the MPI Vendor is "Intel". To specify the value 'Scheduler' for this option for a 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.

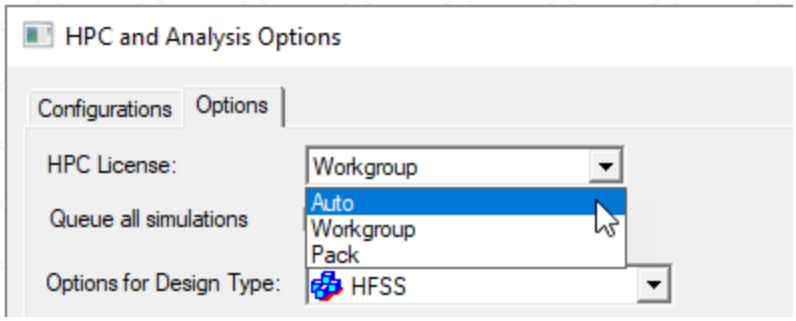
Setting MPI Version

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.

Licensing Settings Tool

HPC licensing includes a choice called “Auto”, along with Workgroup and Pack. The “Auto” choice is available in the HPC License combobox found on the **Options** tab of the **HPC and Analysis Options** dialog box. If you make changes in the License Settings Tool, we recommend that you restart Electronics Desktop.

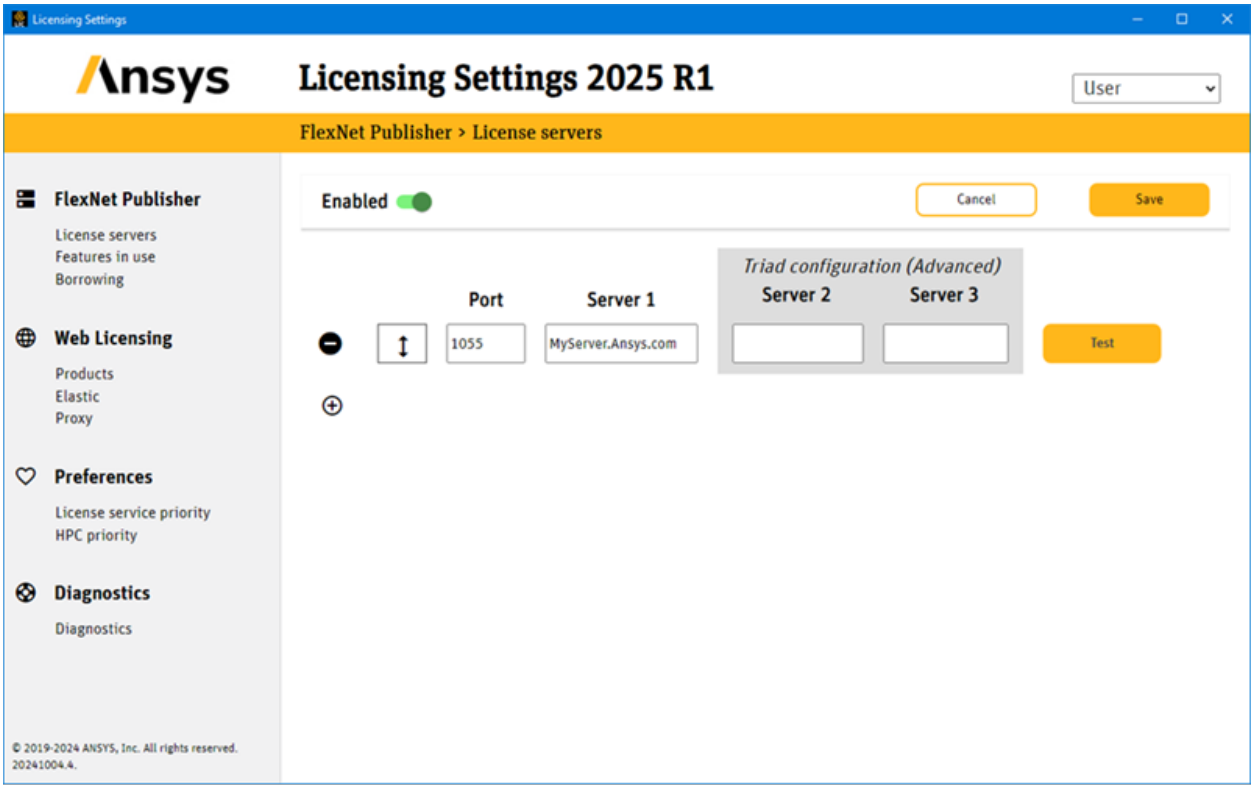


"Auto" means delegate the choice of Workgroup or Pack licensing to the Ansys Licensing Settings tool which is a separate application installed along with Electronics Desktop, accessible from the **Tools** menu. The HPC settings specified by the Licensing Settings tool are shared by other Ansys products. This provides you a single place where you can set the HPC preferences.

> **Note:** Important: HPC licensing preferences in the Licensing Settings tool are used only when Auto is selected in Electronics Desktop.

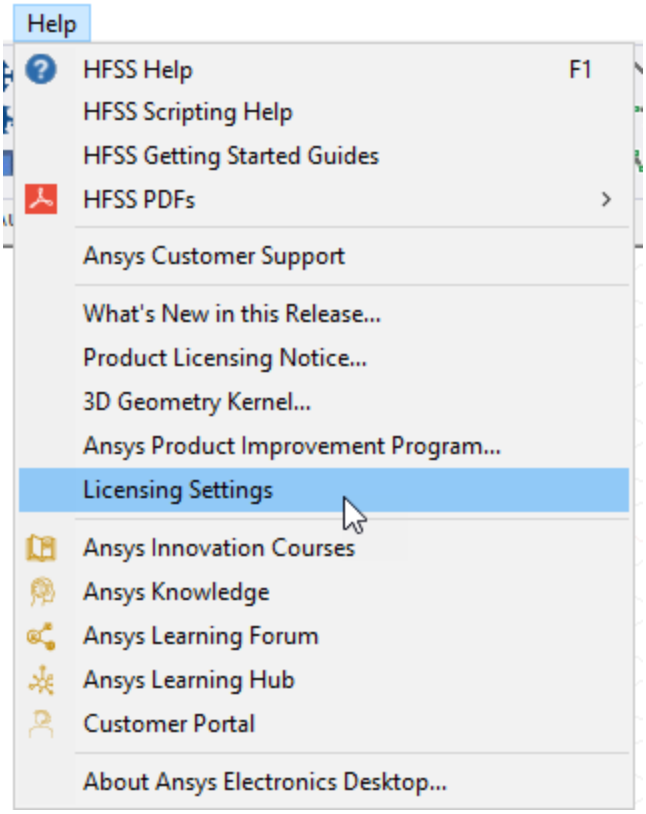
Launching the Licensing Settings Tool from the Main Menu

Click **Tools** > **Licensing Settings** to launch the Licensing Settings tool.



Licensing Settings Help

You can obtain help with the Licensing Settings tool from the **Licensing Settings** menu item in the **Help** menu.



License Fallbacks

When using Auto HPC licensing, in some cases if there aren't enough available licenses of the highest priority HPC license type, the lower priority HPC license type will be used. For instance, assume the user has set the priority order to "Ansys HPC, Ansys HPC Pack" in the Licensing Settings tool. If the user requests a nominal solve using 14 cores (10 Ansys HPC), but 10 Ansys HPC licenses aren't available, 2 Ansys HPC Pack licenses will be used if available.

Setting Options via Configuration Files

In addition to setting options from the Electronics Desktop user interface (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 they may apply to all hosts or only to specific hosts. There are four levels of settings, 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

In the list above, settings at any level override settings at lower levels . If there is no setting in any file, the application default value is used. See UpdateRegistry for instructions on selecting these levels.

> **Important:**
>
> Options set from the Desktop UI override and update user settings in configuration files. Otherwise, the existing configuration file settings are used.

Behavior Examples

Consider running an application as user **jsmith** on host **host123**. If 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 there is a setting for the "Expand Project Tree on Insert" option in the host-independent user options config file for user **jsmith**, then the latter setting will be 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**. If there is no setting for the *Expand Project Tree on Insert* option in the host-dependent user options config file for **jdoe** on **host123** or in the host-independent user options config file for user **jdoe** or in the host-dependent default options config file for host **host123**, then the value from the installation default config file are 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 Options dialog boxes in the Desktop UI. If the dialog box is closed with the **Cancel** button, changes made on any of the tabs are discarded. If the dialog box is closed with the **OK** button, any settings that have been changed by the user are written to the host-dependent user options config file. The changed values written to this file are then used the next time that the application is run 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*>\ANSYS Inc\v251\AnsysEM\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 will have a separate user-specific config directory, with a different value for the <*ApplicationName&Version*> directory name.

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	*"<Installation_Directory>*\ANSYS Inc\v251\AnsysEM\config"	*"<Installation_Directory>*/ansys_inc/v251/AnsysEM/config"
installation default	install	default.XML		

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 ElectronicsDesktop2025.1)
- **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
- **<*Installation_Directory*>** is the root folder where the Ansys software is installed (typically, C:\Program Files, on Windows, or /opt, 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 251)

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 **2022 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\ ElectronicsDesktop2025.1\config	/home/jsmith/Ansoft/ ElectronicsDesktop2025.1/config
host independent user options	user	user.XML		
host dependent default options	install_machine	host123.XML	"C:\Program Files\ANSYS Inc\v251\ AnsysEM\config"	/opt/ansys_inc/v251/AnsysEM/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, ElectronicsDesktop2025.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v251\AnsysEM\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, ElectronicsDesktop2025.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v251\AnsysEM\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, ElectronicsDesktop2025.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v251\AnsysEM\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, ElectronicsDesktop2025.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v251\AnsysEM\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.

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 ElectronicsDesktop2025.1
```

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 ANSYSElectronicsDesktop2025.1
 -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 ElectronicsDesktop2025.1
 -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 ElectronicsDesktop2025.1
 -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 ElectronicsDesktop2025.1
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryLevel user
```

User Options and the UpdateRegistry Tool

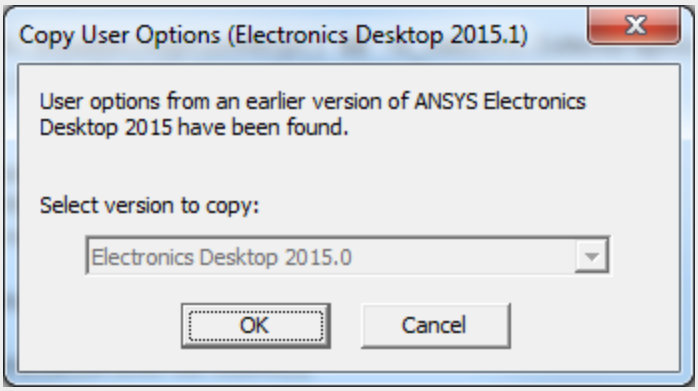
> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

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 use the UpdateRegistry tool to add or modify settings in the host-dependent user options config file; however, you cannot use the Desktop UI to remove settings from the host-dependent user options config file. *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, when they first run 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.

Note:

If a current registry does not exist, the 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 from which you can select an earlier version from which the registry will be ported.



Copy over registry entries (both Windows and Ansys .xml files).

Example: Removing a Host-Dependent User Option Setting

For this example, user jsmith always uses host jshost to run Ansys Electronics Desktop. At some point, jsmith set the Autosave interval in **General Options** > **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 ElectronicsDesktop2025.1
-RegistryKey Desktop/Settings/ProjectOptions/AutoSaveInterval
-RegistryLevel user_machine
```

Example: 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 ElectronicsDesktop2025.1 -
RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort -
RegistryLevel user
```

Example: 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 ElectronicsDesktop2025.1 -
RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
```

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 for Ansys Electronics Desktop. 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 ElectronicsDesktop2025.1
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryValue 250 -RegistryLevel user
```

Example: Setting a RegistryKey not defined in ElectronicsDesktopRegistrySyntax.xml

```
"C:\Program Files\ANSYS Inc\v251\AnsysEM">UpdateRegistry -Set -
ProductName ElectronicsDesktop2025.1 -RegistryKey "3D
Editors/Preferences/Geometry3D/q" -RegistryValue "m"

registry key is not defined in this product

Register value <3D Editors/Preferences/Geometry3D/q> is created with
value <m>
```

Example: Change a Newly Set Registry Key

```
"C:\Program Files\ANSYS Inc\v251\AnsysEM\">UpdateRegistry -Set -
ProductName ElectronicsDesktop2025.1 -RegistryKey "3D
Editors/Preferences/Geometry3D/q" -RegistryValue "n"
```

```
Register value <3D Editors/Preferences/Geometry3D/q> is set to <n>
```

If you attempt to set a registry key to the wrong data type, a message like the following appears.

```
Cannot set the registry value to different data type
The work around is delete it, then create it again
error on to set registry value at <3D
Editors/Preferences/Geometry3D/q>
```

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*>\ANSYS Inc\v251\AnsysEM\config"	<*InstallationDirectory*>/ansys_inc/v251/AnsysEM/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 ElectronicsDesktop2025.1)
- **%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 Ansys software is installed (typically, C:\Program Files, on Windows, or /opt, 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 251)

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.

Temporary Directory Configuration File Format

This section describes the format of 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 them.

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.

Important:

- 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 backslash (\) may be used as a directory separator on Windows only.
- The backslash (\) 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 backslash may be used to remove this special meaning, and include a single quote in the string.
- To use a backslash as a directory separator on Windows, it must be escaped. That is, a double backslash "\\" is used to denote a single directory separator.
- On Windows, a UNC path normally begins with two backslash characters. In a tempdirectory string, each of these backslash characters must be doubled, so four consecutive backslashes "\\\\" are used in the config file.

Example: Config File with UNC

```
$begin 'Config'
tempdirectory='\\\\hostxyz\\TEMP\\abc'
$end 'Config'
```

Here, hostxyz is a host with a sharename TEMP having subdirectory abc used as the temporarydirectory. This shows that four backslashes are required for UNC names and that backslashes used as directory separators must be doubled.

Example: Config File with Single Quote

```
$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.

UpdateRegistry: Setting or Removing Temporary Directory Values in Configuration Files

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, adding, changing, or deleting a temporary directory setting.

The -GetKeys option does not list the temporary directory key.

Batchoptions Command Line Examples

The `-batchoptions` entries command line argument may be used to specify one or more batchoptions settings from 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.

> **Important:**
>
> 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

> **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 HFSS analysis of project file project1.aedt. We want all temporary files and directories to be created in C:\temp\HFSS instead of using the default 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 four processors for of 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/HFSS'
'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\\HFSS'
 '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.

Example -Batchsolve with -Machinelist (Windows)

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

Suppose that we want to run a batch HFSS 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 (HFSS/NumCoresPerDistributedTask) and non-distributed (Desktop/Settings/ProjectOptions/NumberOfProcessors) parts of the job. The analysis contains a sweep that will be distributed across three hosts: ahmed, bill, and catherine. The hosts ahmed and bill have four cores each, so we run two distributed COM engines on each of these hosts, each using two threads. Host catherine 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 RAMLimitPercent of 75 for this analysis.

Here is a sample command line for this analysis, where the project file \\dennis\projects\project1.aedt is located in a shared directory specified using a UNC path:

```
ansysedt -BatchSolve -Distributed
-Machinelist list=ahmed,ahmed,bill,bill,catherine
-batchoptions "'Hfss/RAMLimitPercent'=75
'Desktop/Settings/ProjectOptions/NumberOfProcessors'
'Hfss/NumCoresPerDistributedTask'=2"
\\dennis\projects\project1.aedt
```

Example -Batchsolve with -Machinelist (Linux)

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

In this example, we run a batch HFSS analysis of project file project2.aedt. We have four identical host computers—host1, host2, host3, and host4 for analysis, and each host has 4 cores. We do not use multiprocessing for the distributed analysis, so NumCoresPerDistributedTask=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. We will specify a RAM limit of 75% for this analysis.

Here is a sample command line for this analysis, where the project file /home/jsmith/projects/project2.aedt is located in a shared directory:

```
hfss -BatchSolve -Distributed
-Machinelist file=/home/jsmith/hosts/list2
-batchoptions "HFSS/'RAMLimitPercent'=75
'Desktop/Setings/ProjectOptions/NumberOfProcessors'=4
'HFSS/NumCoresPerDistributedTask'=1"
/home/jsmith/projects/project2.aedt
```

For this example, the hostnames are in the text file /home/jsmith/hosts/list2. The file contents are as follows:

```
host1
host1
host1
host1
host2
host2
host2
host2
host3
host3
host3
host3
host4
host4
host4
host4
```

Example -Batchsolve for Local (Windows)

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

In this example, we run a batch analysis of project file testproject.adsn on the local host. We want all temporary files and directories created in directory C:\temp\ansysedt instead of using the installation default 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.adsn is located in a shared directory specified using a UNC path:

```
ansysedt -BatchSolve -Local -batchoptions
"TempDirectory='C:/temp/ansysedt'
'Planar EM/SolverOptions/NumProcessors'=4
'Planar EM/SolverOptions/NumProcessorsDistrib'=4"
\\host123\projects\testproject.adsn
```

Note that the batchoptions pathnames 'Planar EM/SolverOptions/NumProcessors' and 'Planar EM/SolverOptions/NumProcessorsDistrib' must be in single quotes because they both contain embedded spaces.

Batch Options 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:

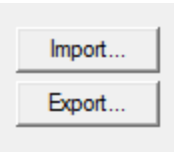
  Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

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 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 using 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***>\ANSYS Inc\v251\AnsysEM"**

2. Type and enter the following command:

   UpdateRegistry -GetKeys -ProductName ElectronicsDesktop20*xx.y* **>** <*text_file_path*>**\Batchoptions.txt**

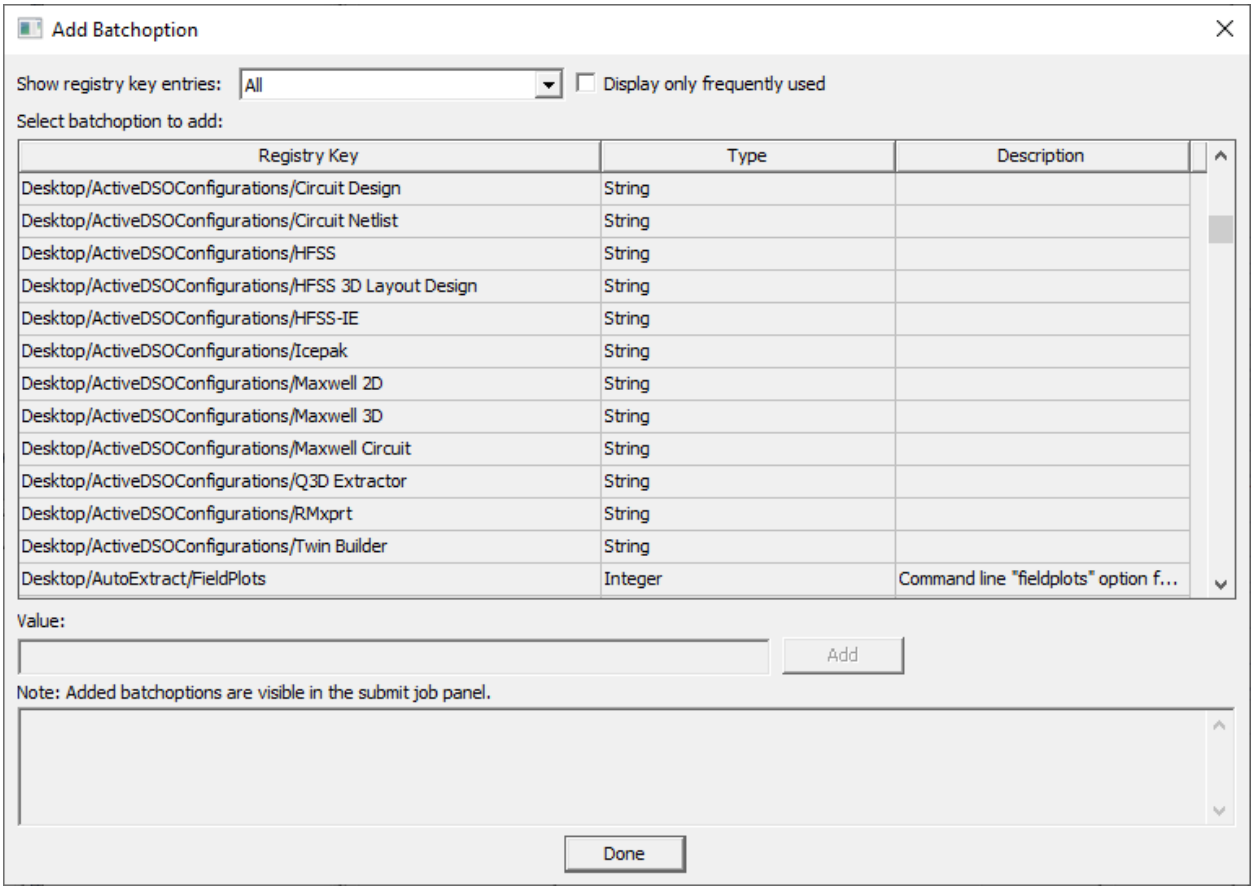
   Substitute the last two digits of the installed product version year and the minor release number for *xx.y* (such as **25.1**). Also, substitute the desired *text_file_path* (such as **%UserProfile%\Desktop**).

   The resulting **Batchoptions.txt** file will contain a nearly complete list of available options.

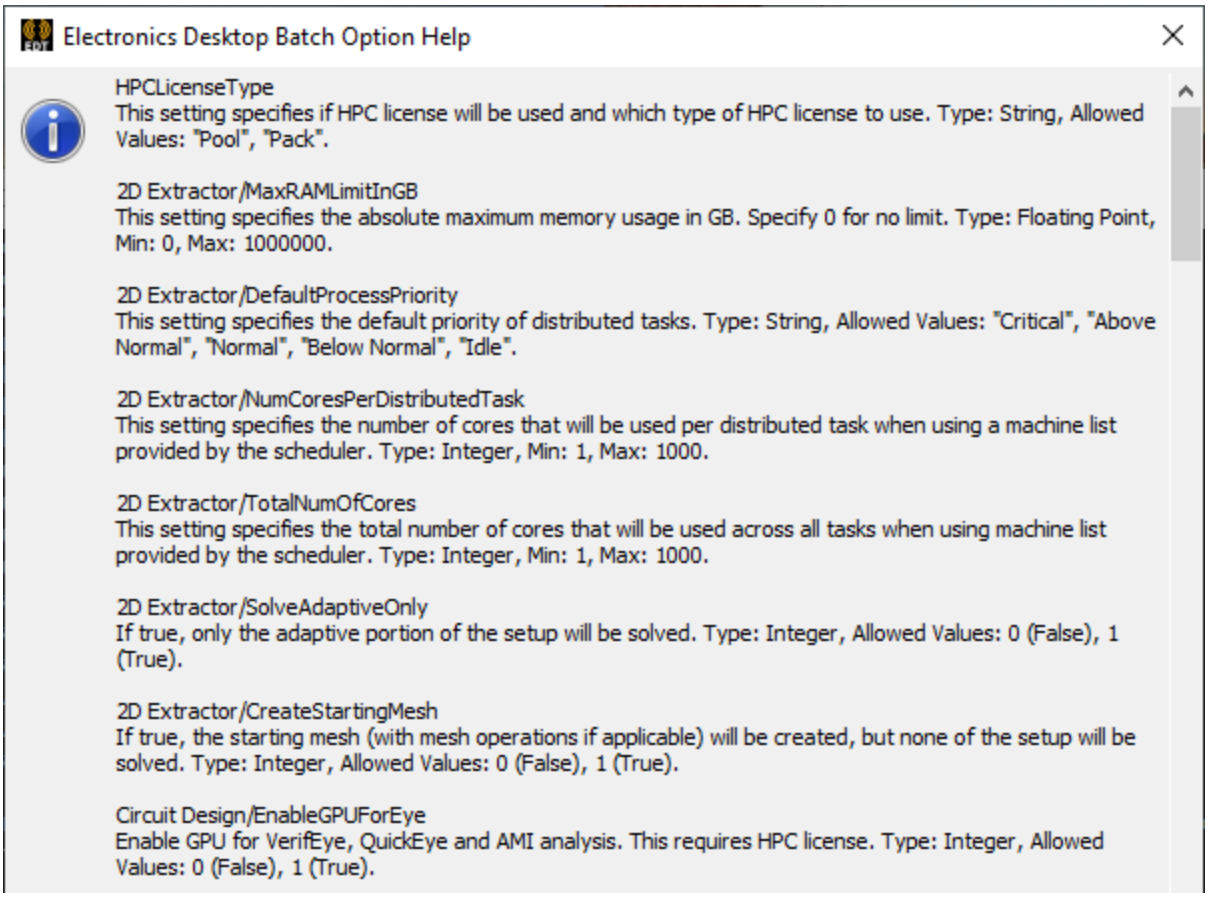
This procedure also works from the Linux <*installation_directory*>, which has the same downstream path (for example, under "opt/") as a Windows installation (typically under "C:\Program Files\").

> **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:



Managing Projects and Designs

An Ansys Electronics Desktop *project* is essentially a folder that includes one or more models, or *designs*.

Some basic tasks that can be performed on projects and designs include:

- Opening Projects
- Creating Projects
- Saving Projects
- Importing and Exporting Projects and Data
- Setting Project Options
- Validating Projects
- Copying and Pasting Projects and Designs
- Renaming Projects
- Deleting Projects or Designs
- Setting Read Only Designs

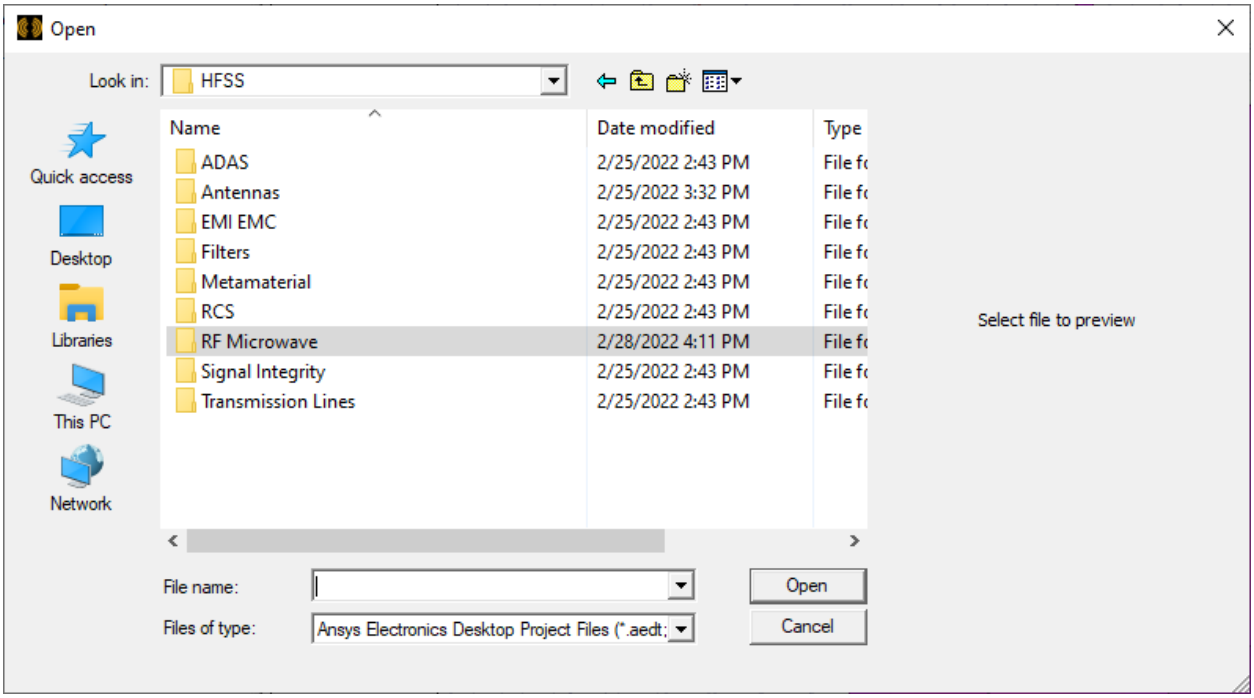
- Updating Design Components
- Undoing and Redoing Commands
- Inserting a Documentation File
- Saving Project Notes
- Printing

Opening Projects

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

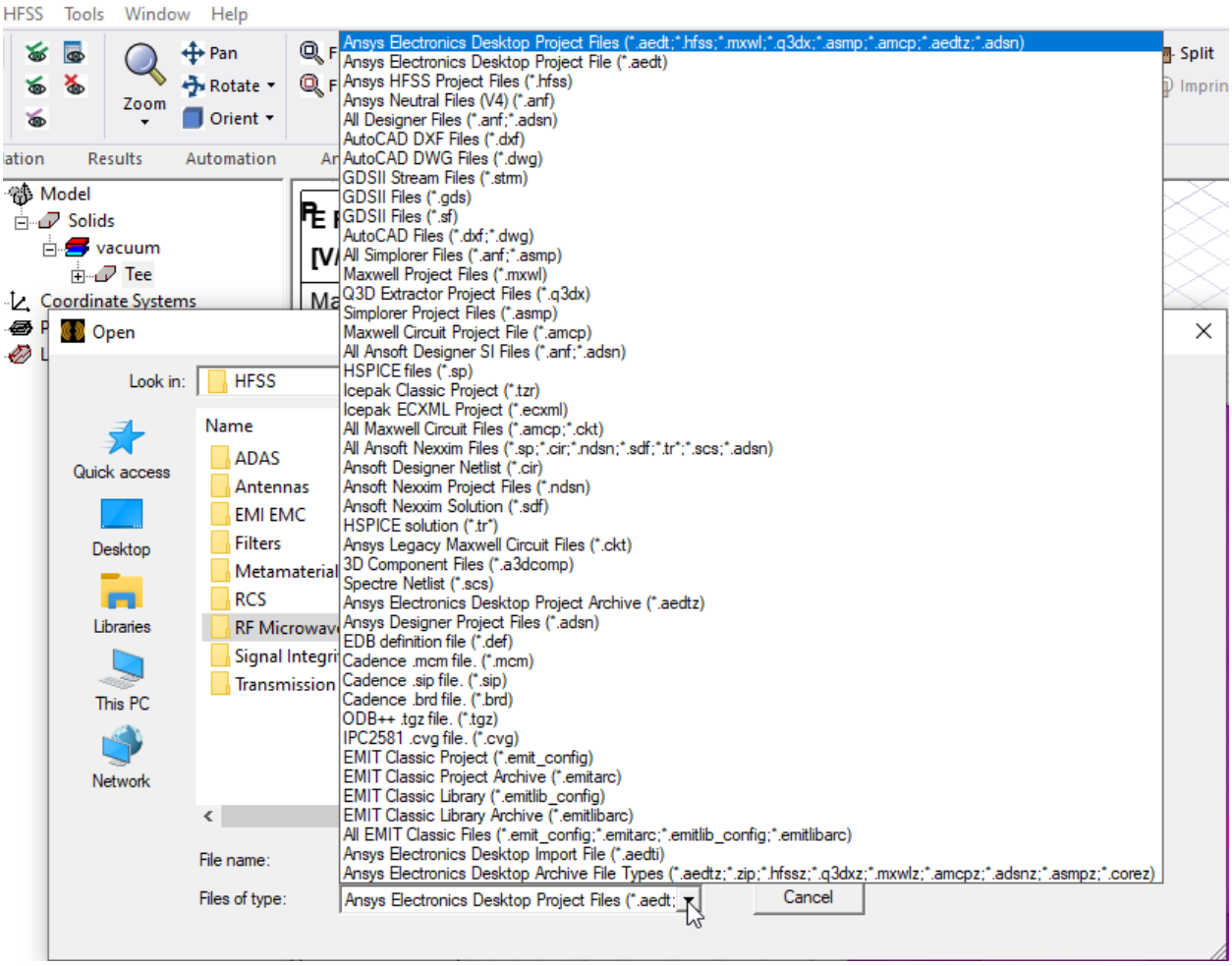
1. Click **File** > **Open** or click the **Open** icon on the **Desktop** tab of the ribbon.

2. Use the file browser to find the project file.

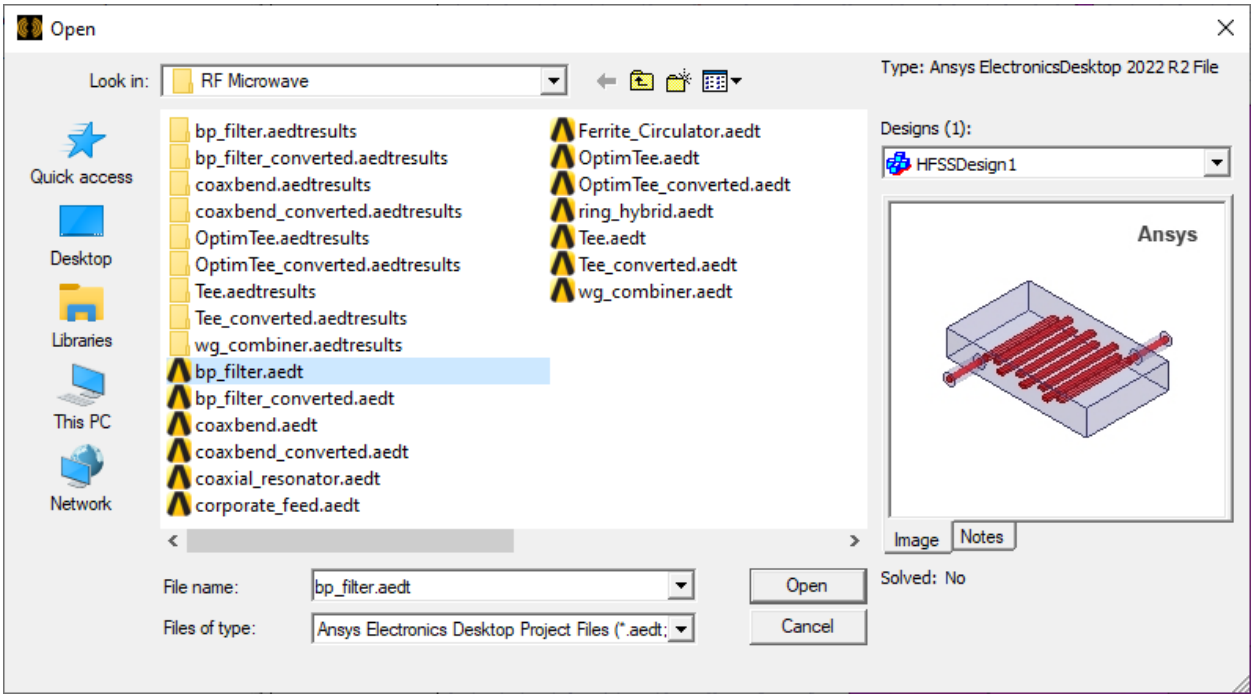


By default, only files that can be opened or translated display. *.aedt is the main default, although, you can also view several types of legacy files. The dropdown for Files of type

lists the kinds file and archive formats for which you can select filters.



3. Most of these will be familiar to those seeking them. The Ansys Electronics Desktop Import file (*.aedti) format is for importing exported projects from ANSYS Discovery, an ANSYS Workbench tool. This format allows import of side by side *.aedti and the correponding geometry definition *.sat into the Ansys Electronics Desktop, and creates a ready-to-solve HFSS project.
4. Select the file you want to open.



5. Click **OK**.

   The project information appears in the Project tree.

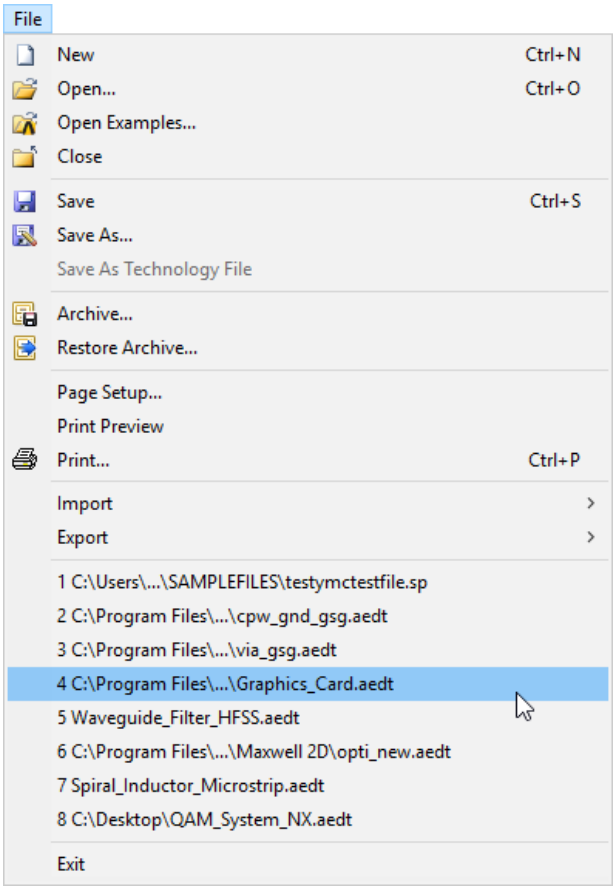
If you open another project without editing the automatically created project, the automatically created project is removed.

You can also open a saved project by:

- Dragging a project file icon to the Open icon
- Dragging a project file icon to the desktop
- Double-clicking a project file icon

Opening Recent Projects

Recently opened projects appear at the bottom of the **File** menu. To open one of these files, click its file name.



Opening Example Projects

You can directly access and open example projects included with product installation using **File** > **Open Examples**, or by clicking **Open Examples** on the **Desktop** tab of the ribbon. See Example Projects for additional description of these files.

Opening Legacy Projects

Q3D Extractor cannot open projects created in Q3D Extractor version 5.x or earlier. Q3D Extractor 7 files can be opened, but saving them in the current format renders them unable to be reopened in version 7.

To open a legacy project:

1. Click **File** > **Open**.
2. From the **Look in** drop-down menu, select the location of the project. In the folder list, double-click folders to find the one that contains the project.
3. Double-click the project you want to open.

Legacy Project Translation

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.

In the previous version of Q3D, objects were considered non-model if the **Model Object** check box was cleared in the **Object Attributes** window. If the object was excluded in the legacy project, it is considered a non-model object in this version.

The following table contains 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. • 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. • View visualization settings apply to the saved design. If these have been changed from the default (15 deg), this affects the memory and CPU required to open the project.
Excitations and Boundaries	• If the legacy project contained both port impedance and calibration lines, the impedance lines are translated to integration lines, and calibration lines are ignored. If the project contained both impedance and terminal lines, both are translated to integration lines. The impedance lines are ignored for Driven Terminal solutions and terminal lines are ignored if the project is changed to a Driven Modal solution. • Boundaries assigned to named interface selections or rectangle selections are not translated. • For a boundary assigned to the intersection of two faces, a new 2D sheet object is created from the intersecting area and assign the boundary to that object.
Hybrid Regions	• Radiation Boundaries with FEBI become Hybrid Regions assigned as FEBI.
Materials	• Functions defined in legacy projects become project variables; therefore, functional material properties are translated. • Perfect conductors become regular materials with conductivity values of 1E30. • Nonlinear materials from legacy projects that have magnetic saturation values greater than zero are treated as ferrite materials. Their properties are not modified.

Mesh Operations	• Mesh refinement operations performed on arbitrary boxes in legacy projects are ignored. • Area 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 Types	• Driven solver projects that contained terminal lines are translated to Driven Terminal solution types.
Solution Setup	• Impedance-only and emissions-only solutions are not supported; therefore, these selections in legacy projects are ignored. • The design's initial mesh is used for the solution. Current meshes are not translated. • Saving dominant-only or higher-order-only mode S-matrix entries are not supported; therefore, these mode selections in legacy projects are ignored. • For frequency sweeps, the **Number of Steps** value specified in the legacy project is converted to the corresponding **Step Size** value. • The total number of requested adaptive passes in the legacy project becomes the **Maximum Number of Passes** value. For example, if you request 3 adaptive passes, solve them, and then request 2 adaptive passes, 5 will be the value specified for the **Maximum Number of Passes**.
Solutions	• Solution data is not translated; therefore, you must solve legacy projects again.

Note:

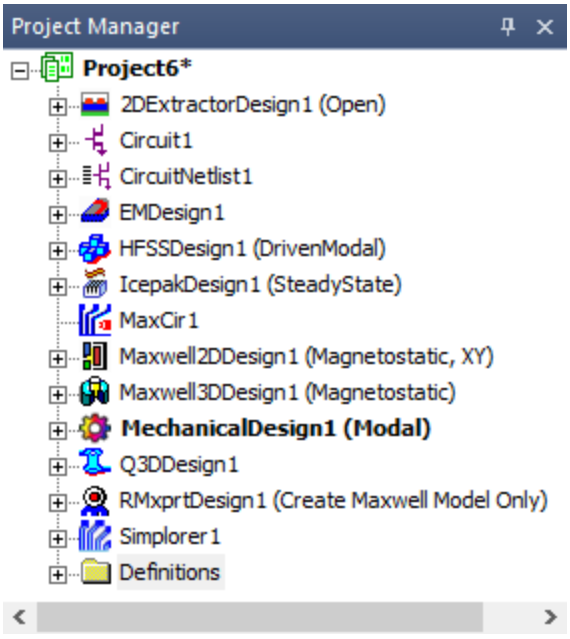
Legacy Circuit (Serenade) projects (**.ssp** extension) do not open correctly from the **File Open** dialog box. Contact Ansys Technical Support for assistance in converting legacy projects. Opening an Ansys Neutral File project (**.anf** suffix) begins with a conversion dialog box. See Importing ANF Design Data. The Planar EM simulator can use ANF data.

Creating Projects

To create a project, click **File** > **New**, or click the **New** icon on the **Desktop** tab of the ribbon.

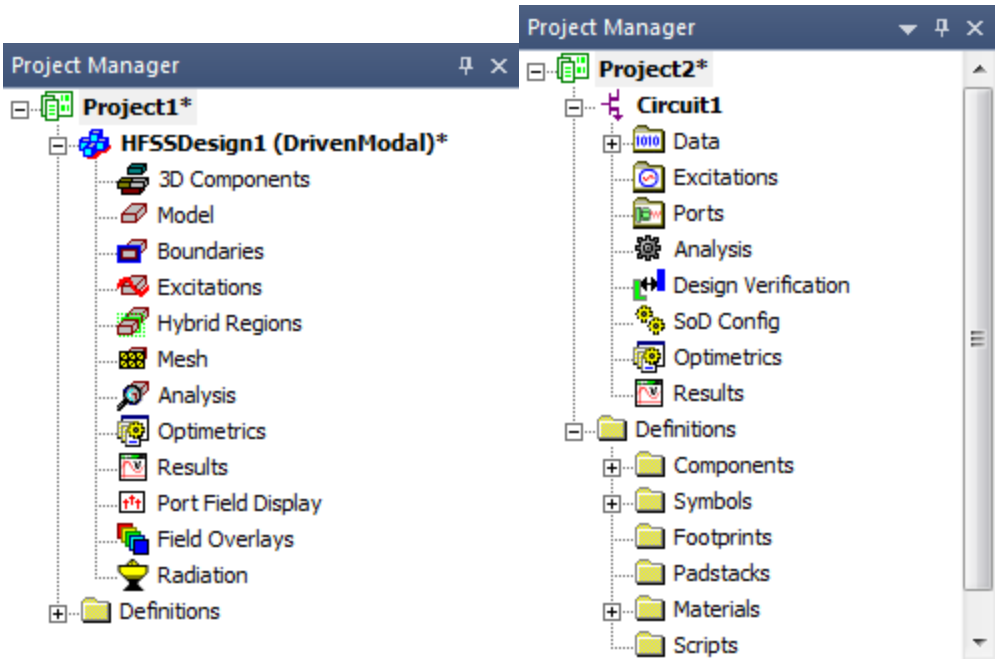
A new project is listed in the project tree. It is named Project*n* by default, where *n* is the order in which the project was added to the current project folder.

You can insert designs of any type into the project, where they are stored in the project tree.



The default name for each inserted project is <designType>*n*. You can also specify the name of the project when you save it using the **File** > **Save** or **File** > **Save As** commands.

You can view the contents of a project by clicking the plus sign (+) for each level of the hierarchy in the project or design.



Creating New 2D Designs from 3D Designs

Q3D Extractor supports both 2D and 3D projects. Often, you may want to take a cross-section of a 3D model to perform a 2D analysis. You can take a Q3D model and automatically generate a cross-section for 2D Extractor by specifying a coordinate system and section plane.

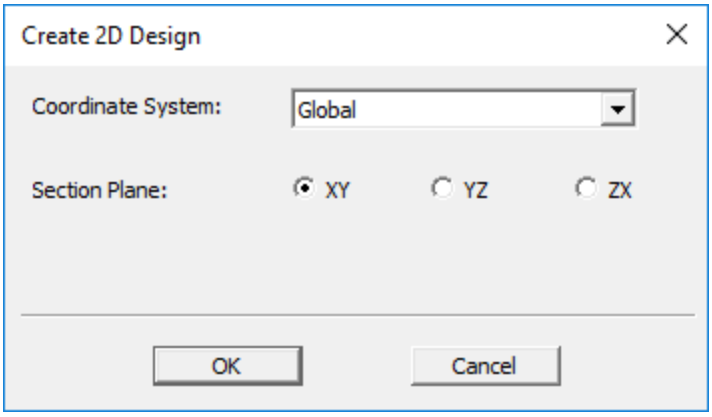
Keep the following points in mind before converting any project:

- 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.

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

1. Click **Q3D Extractor** > **Create 2D Design**.

   The **Create 2D Design** window 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. 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.

Creating New 3D Designs from 2D Designs

Q3D Extractor supports both 2D and 3D projects. Often, you may want to sweep a 2D model to create a full 3D design. You can take a 2D Extractor model and automatically generate a Q3D model by specifying the z-length.

You can automatically convert a 2D XY model to a 3D model.

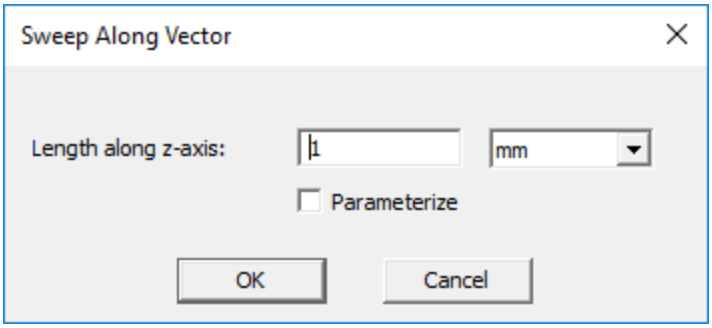
Keep the following points in mind before you convert any project:

- 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.

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

1. Click **2D Extractor** > **Create 3D Design**.

   The **Sweep Along Vector** window appears.



2. Enter a value in the **Length along z-axis** box to obtain a swept length of the model.
3. Choose the desired units of the sweep distance entered from the drop-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.

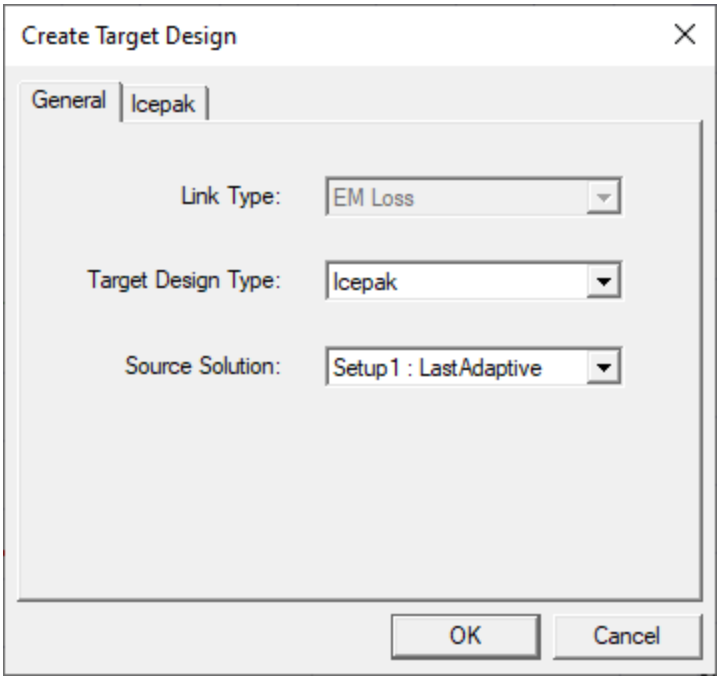
Creating a Target Design from 3D Designs

In an HFSS, Q3D, or Maxwell 3D design, **Create Target Design** automatically generates and inserts a thermal design (Icepak or Mechanical Thermal) into your project.

To create a target design from an existing **Q3D Extractor** design:

1. Click **Q3D Extractor** > **Create Target Design**.

   The **Create Target Design** window appears.



2. Select the **Target Design Type** to be created.
3. Select the **Source Solution** to specify the solution data to be linked to the target design.
4. For Icepak designs, specify **Forced Convection** or **Natural Convection** on the **Icepak** tab.
    - For **Forced Convection**, specify **Flow Speed** and **Flow Direction**.
    - For **Natural Convection**, specify the **Gravity Vector Direction**.
5. Click **OK**.

Icepak Designs

The Icepak design is generated based on the source design and selections in the **Create Target Design** dialog box:

- Model geometry is created with corresponding material assignments. Vacuum objects are not created in target Icepak designs.
- Any assigned materials that do not contain thermal properties are assigned default thermal properties in Icepak.
- The Region (computational domain) is created with the appropriate dimensions.

- An EM loss excitation is assigned to the geometry, and variable mapping and intrinsic variables are linked to the source design.
- Design settings are specified.
    - Solution type: **Steady State**
    - Problem type: **Temperature and Flow**
- For forced convection simulations, the following are defined:
    - A free opening boundary condition with a velocity specified is assigned to the Region face to create forced flow.
    - A free opening boundary condition is assigned to the face of the Region opposite to the opening with velocity specified.
    - In the **Icepak Solve Setup** dialog box, forced convection **Solve Setup Defaults** are defined.
- For natural convection simulations, the following are defined:
    - **Gravity Vector**
    - Free opening boundary conditions are assigned to each face of the air Region.
    - In the **Icepak Solve Setup** dialog box, natural convection **Solve Setup Defaults** are defined.

Mechanical Designs

The Mechanical Thermal design is generated based on the source design:

- Model geometry is created with corresponding material assignments. For any vacuum objects, **Solve Inside** is disabled.
- Any assigned materials that do not contain thermal properties are assigned default thermal properties in Mechanical.
- A convection boundary is assigned to the faces of all 3D objects (not 2D sheet objects). **A Film Coefficient** is specified as **Uniform** at 10 w_per_m2kel. Objects with **Solve Inside** disabled and those with air assigned as the material are not assigned to the convection boundary.
- An EM loss excitation is assigned to the geometry, and variable mapping and intrinsic variables are linked to the source design.

Saving Projects

Use the **File** > **Save As** command or select the **Desktop** tab of the Ribbon and click the **Save As** icon to:

- 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 or select the **Desktop** tab of the Ribbon and click the **Save** icon to save the active project.

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

Each solver has a Save Before Solving setting located in the **Tools** > **Options** window. This setting is enabled by default.

A prompt appears when you attempt to save a previously versioned file. If you agree to the prompt, the file is upgraded to the Ansys Electronics Desktop 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 its previous compatibility.

If you have a simulation running, you see a warning that if you continue, Ansys Electronics Desktop will abort the simulation. If you OK the warning, Ansys Electronics Desktop 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. Below 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, typically stored in the `ProjectName.aedb` folder
- The temporary directory and most temporary files

  Because some temporary files and directories have automatically generated names that are long, it is best to use a short pathname for the temporary directory.

  > **Important:**
  >
  > Although most temporary files are created within the temporary directory, there may be some temporary files within the project directory. As a result, the project file directory pathname should be well below the 259 character limit.

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

- Long paths must be enabled on the machine

  Use either of two ways to do this:

  - **Via 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`

Changing Auto-Save Settings

Recent actions performed on the active project are stored in an 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. 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 10 edits. An edit is any action performed that changes data in the project or the design, including actions associated with project management, model creation, and solution analysis.

After a problem occurs, you may be able to choose to re-open the original project file in an effort to recover the solution data, or to open the auto-save file. If the original file is not available, attempting to open the file provides a message that the auto-save is being used. If neither file is available, an error message displays.

To modify auto-save settings:

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

   The **Options** window appears.

2. Under **Desktop Configuration**, 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 to 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 will automatically be saved as Project1.aedt.auto.

> **Warning:**
>
> When you close or rename a project, Ansys Electronics Desktop deletes the auto-save file.

Save Before Solving Option

The **Save before solving** option forces a full save before running the solve and is enabled by default. For efficiency reasons, the project is saved only if it has been modified since its last save.

To change this setting, click **Tools** > **Options** to open the Options window. Select the **General** tab and use the **Save before solving** check box to toggle the option.

You can start a new solve while running another without having to abort the running solve. If you start a solve while another solve is running and the Save before solving option is set, Ansys Electronics Desktop asks if you want solve without saving first. This lets you perform multiple solves, and if you have not edited the project in between solves, crash recovery will work.

Recovering Project Data from an Auto-Save File

Following a sudden workstation crash or other unexpected problem, you can recover project data from its 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.

Importing and Exporting Projects and Data

Ansys Electronics Desktop can import and export a variety of file types.

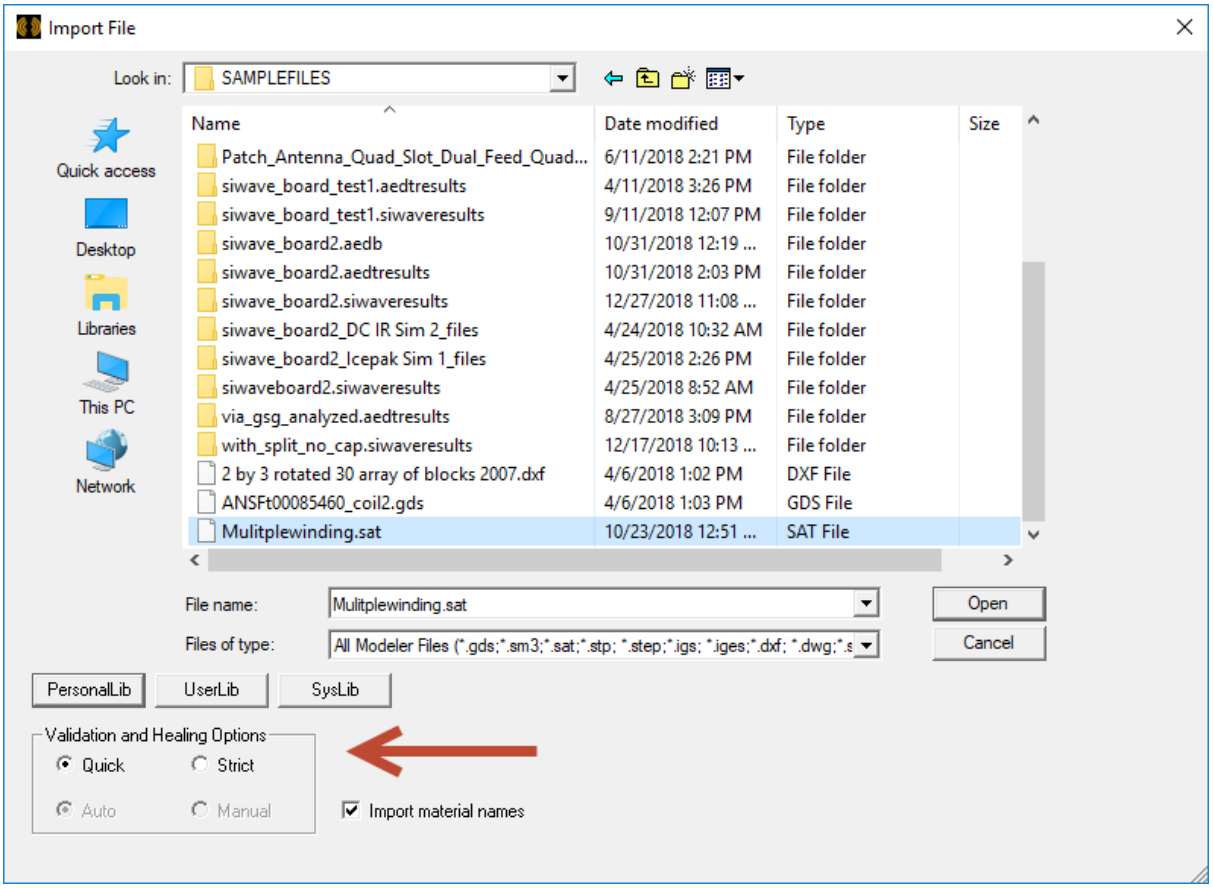
Importing Files to Q3D Extractor

You can import the following types of files to Q3D Extractor:

- 2D model (GDSII) files
- 3D model files
- DXF and DWG format files

Healing Options on Import

When you click **Modeler** > **Import**, the **Import File** window appears.



By default, **Validation and Healing** options are selected for certain types of files. These types are:

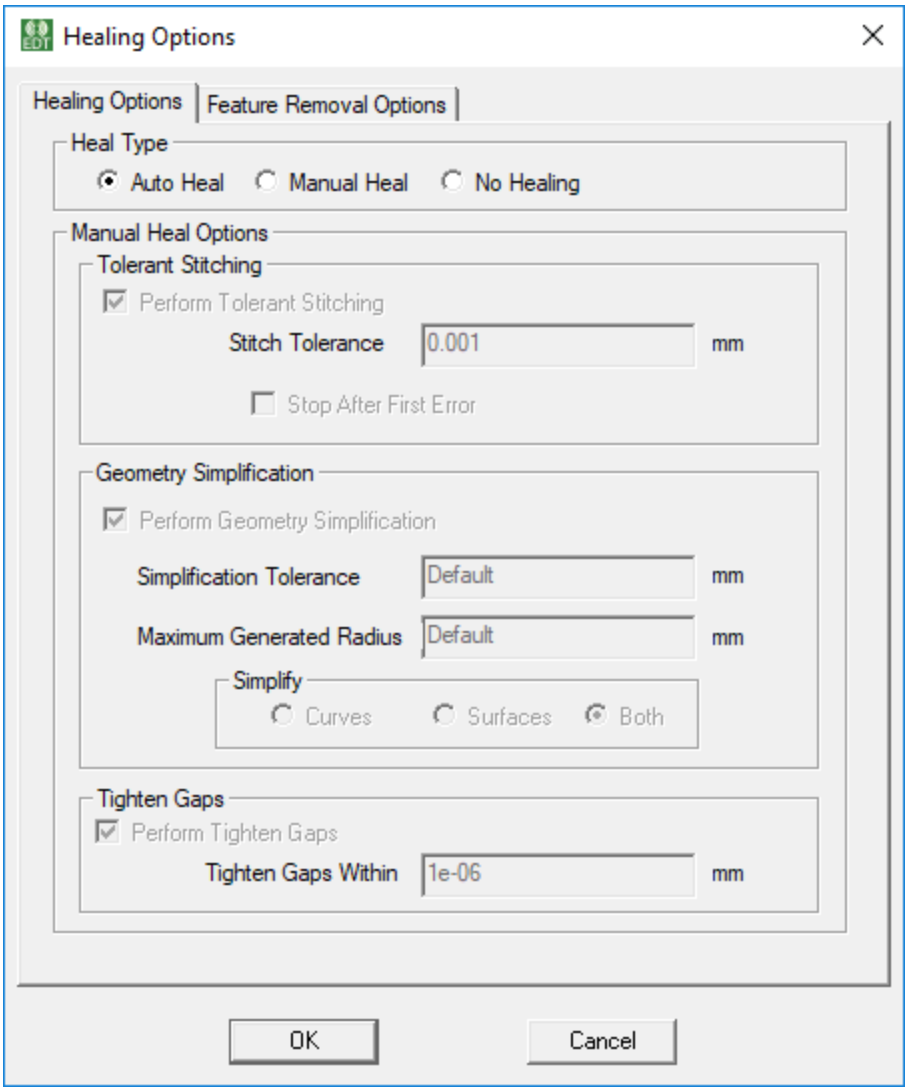
- ACIS SAB files (*.sab)
- ACIS SAT files (*.sat)
- Ansys 3D Modeler files (*.sm3)
- Autodesk Inventor files (*.ipt, *.iam)
- CATIA V4/V5 files (*.exp, *.model, *.CATpart, *.CATproduct)
- Creo Parametric files (*.prt, *.asm)
- IGES files (*.iges, *.igs)
- NX files (*.prt)
- Parasolid files (*.x_t, and *.x_b)
- SOLIDWORKS Files (*.SLDPRT, *.SLDASM)
- STEP files (*.step, *.stp)

Selecting **Quick** validation disables the healing options. Selecting **Strict** validation allows you to choose between **Auto** and **Manual** healing. Imported objects with only one operation on the history tree can be healed.

If you select **Manual** healing:

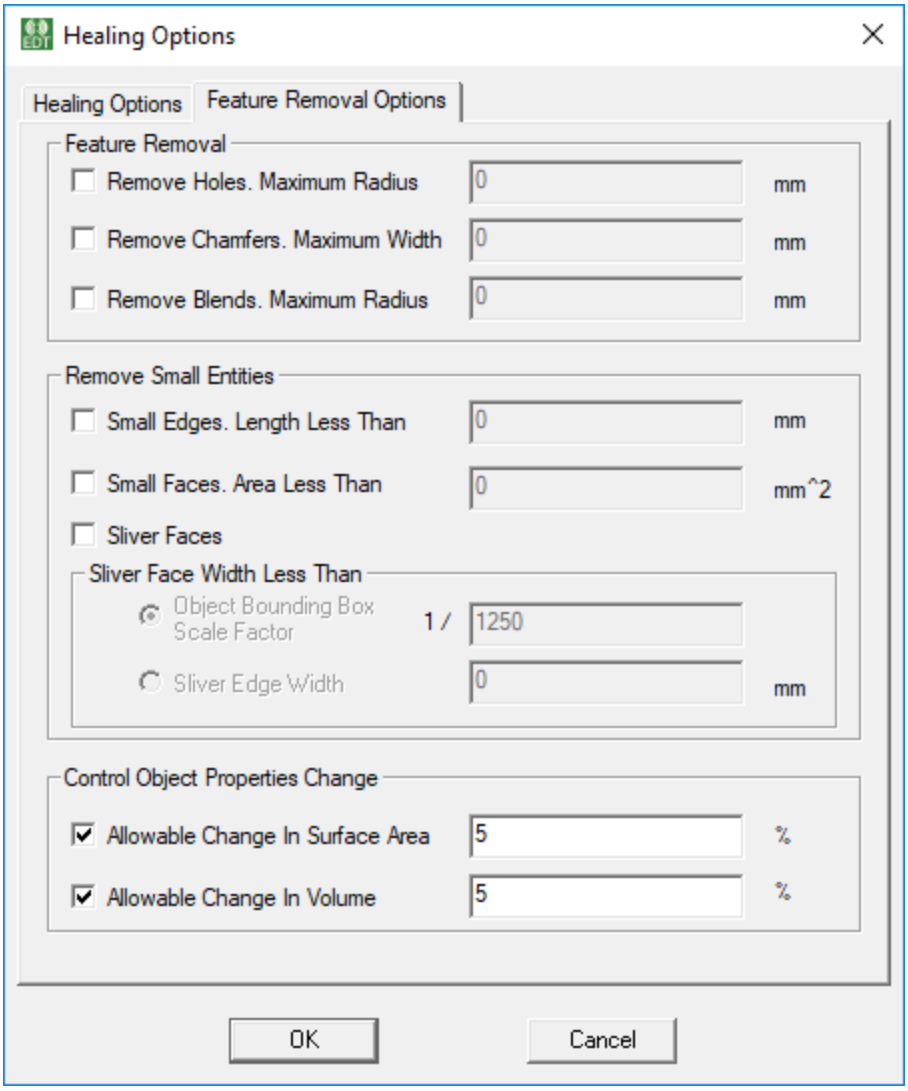
1. Click **Open**.

   The **Healing Options** window appears, on the **Healing Options** tab.



2. Set parameters for **Tolerant Stitching**, **Geometry Simplification**, and **Tighten Gaps**.

3. Click the **Feature Removal Options** tab.



You can set parameters to remove:

- Holes smaller than a maximum radius
- Chamfers smaller than a maximum width
- Blends smaller than a maximum radius
- Small Edges less than a specified value
- Small Faces less than a specified value
- Sliver Faces less than either:
    - Bounding box, less than a specified scale factor
    - Sliver Edge width, less than a specified value

You can also set an allowable change in Surface Area and Volume.

4. Click **OK** to apply the specified healing options and to analyze the model.

You can reopen the **Healing Options** window at any time by selecting **3D Model** > **Model Analysis** > **Heal**.

Importing 2D Model (GDSII) Files

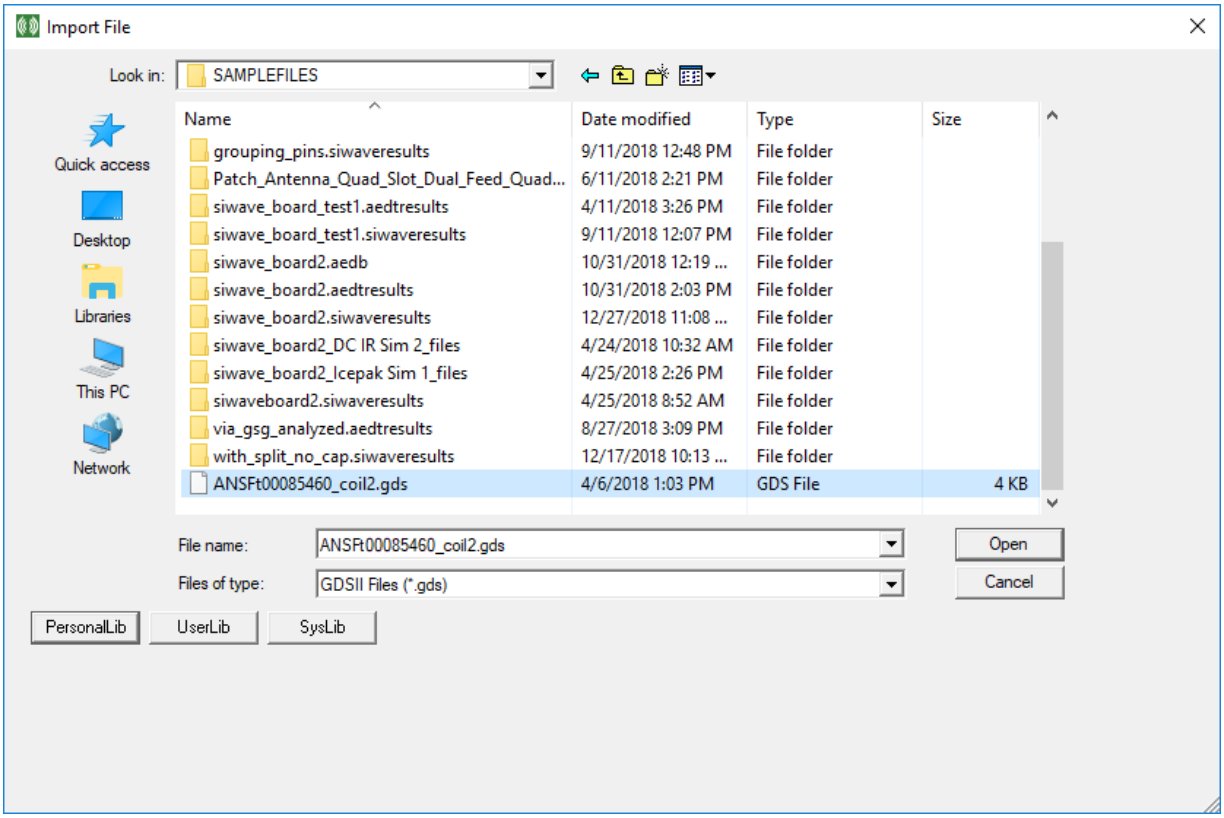
2D model files can be imported directly into the active **Modeler** window. GDSII is the industry standard file format for 2D graphical design layout data.

> **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 2D model file:

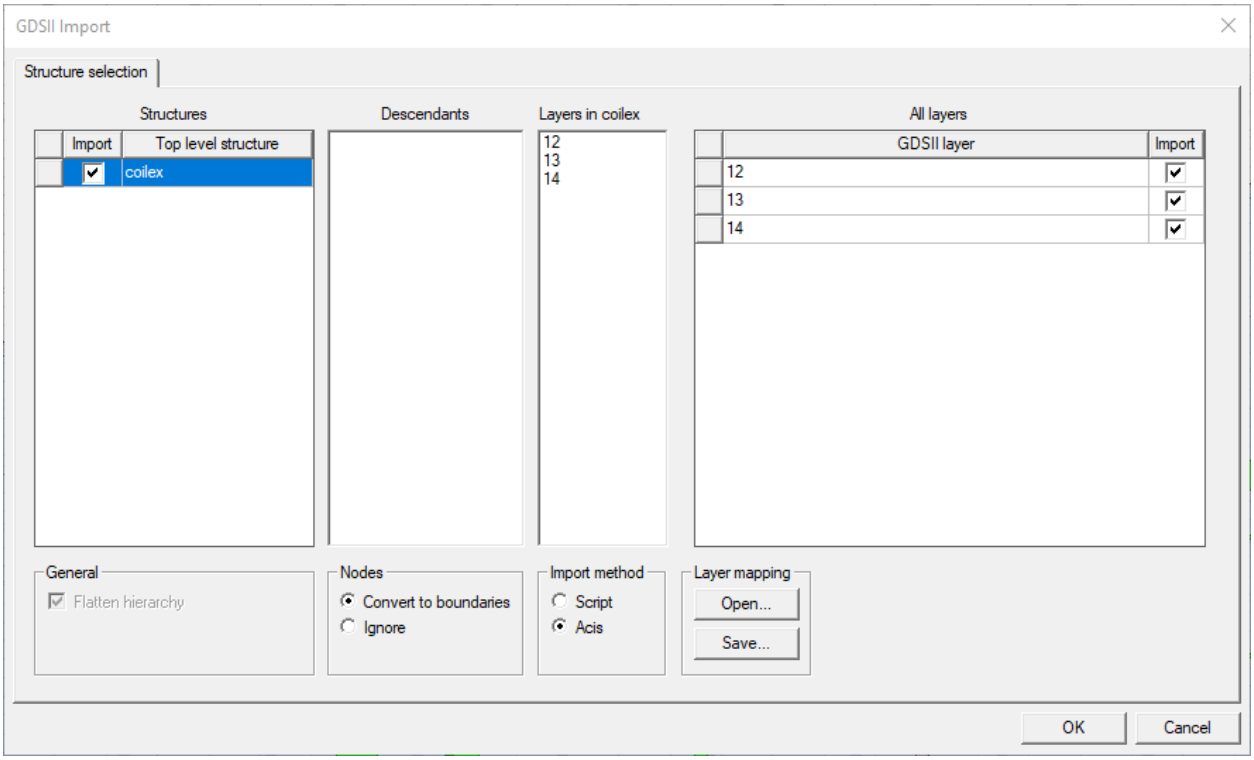
1. Click **Modeler** > **Import**.

   The **Import File** dialog box appears.



2. Select a file type from the **Files of type** drop-down menu. For 2D model files, this is **GDSII Files (*.gds)**.
3. Use the file browser to find and select the file you want to import.
4. Click **Open**.

   The **GDSII Import** window appears.



5. Use the panes to select your desired options:
    - **Structure Selection** – a GDSII file may contain several top-level structures. Use this pane to select structure(s) for import. When multiple structures are imported, Q3D Extractor creates multiple designs under the current project, one for each of the GDSII structures.
    - **Descendants** – GDSII files are hierarchical and may contain many sub-layouts. This panel shows the sub-layouts in the selected top-level designs.
    - **Layers in [Structure Name]** – this panel's name changes based on the name of the selected structure, and displays that structure's layers, by layer number.
    - **All Layers** – lists all layers from all structures in the file. Use the **Import** check boxes to select or deselect layers for import.
    - **General** – the **Flatten hierarchy** check box is selected by default. Q3D Extractor always flattens any hierarchical geometry in the GDSII.
    - **Nodes** – GDSII supports nodes and boundaries as separate data types. Normally, boundaries represent polygons. Q3D Extractor can either convert objects that use the nodes data type to boundary types, or ignore them. Use the radio buttons to select either **Convert to boundaries** (default) or **Ignore**.
    - **Import Method** – select either Script or Acis.
    - **Layer Mapping** – if desired, create a mapping of GDSII layer numbers to layer names in the design stackup.

6. Click **OK**.

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

Creating GDSII Layer Mapping Files

While importing a 2D model (GDSII) file, you can create a mapping of the GDSII layer numbers to layer names in the design stackup.

To create and use the mapping file:

1. Use a text editor to create a text file that maps GDSII layer numbers to layer names in the stackup. The layer mapping file must have a ***.tech** suffix.

   In *.tech format:

   - / is the comment character
   - Units may be specified with a line UNITS <string> before the lines of layer information. <string> is any of the allowed desktop length units. The default unit is nm.
   - Each layer is specified by a line that contains <*import layer*> <*product layer*> <*layer color*> <*layer elevation*> <*layer thickness*>

   where:

   - <*import layer*> is the name of the DXF layer
   - <*destination layer*> is the name to map the DXF layer to
   - <*layer color*> is a string from the choices listed here
   - <*layer elevation*> is specified in nms
   - <*layer height*> is specified in nms

   For example:

```
/ ------------------------------------------------------------
/ import#  destination     Color    Elevation  Thickness
/ ------------------------------------------------------------
S12         signal12         blue      1100       530
TR          trace            red       6620       530
S3          signal3          yellow    8150       2000
```

2. Save the *.tech file locally.
3. During import, click **Open** in the **Layer mapping** panel to locate and open the existing layer mapping file.

4. Click **OK**.

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

Tech File Colors with RGB Values

Consult the table below for color string choices and their corresponding RGB values.

> **Note:**
>
> - Colors listed as two words can also be entered in Pascal case (e.g., GhostWhite is a valid substitute for ghost white).
> - Gray can be spelled using either the American or British spelling (Grey).

Color String(s)	RGB
snow1, snow	{ 255, 250, 250 }
snow2	{ 238, 233, 233 }
snow3	{ 205, 201, 201 }
snow4	{ 139, 137, 137 }
ghost white	{ 248, 248, 255 }
white smoke	{ 245, 245, 245 }
gainsboro	{ 220, 220, 220 }
floral white	{ 255, 250, 240 }
old lace	{ 253, 245, 230 }
linen	{ 250, 240, 230 }
antique white	{ 250, 235, 215 }
AntiqueWhite1	{ 255, 239, 219 }
AntiqueWhite2	{ 238, 223, 204 }
AntiqueWhite3	{ 205, 192, 176 }
AntiqueWhite4	{ 139, 131, 120 }
papaya whip	{ 255, 239, 213 }
blanched almond	{ 255, 235, 205 }
bisque1, bisque	{ 255, 228, 196 }
bisque2	{ 238, 213, 183 }
bisque3	{ 205, 183, 158 }
bisque4	{ 139, 125, 107 }
PeachPuff1, peach puff	{ 255, 218, 185 }
PeachPuff2	{ 238, 203, 173 }
PeachPuff3	{ 205, 175, 149 }
PeachPuff4	{ 139, 119, 101 }

Color String(s)	RGB
NavajoWhite1, navajo white	{ 255, 222, 173 }
NavajoWhite2	{ 238, 207, 161 }
NavajoWhite3	{ 205, 179, 139 }
NavajoWhite4	{ 139, 121, 94 }
moccasin	{ 255, 228, 181 }
cornsilk1, cornsilk	{ 255, 248, 220 }
cornsilk2	{ 238, 232, 205 }
cornsilk3	{ 205, 200, 177 }
cornsilk4	{ 139, 136, 120 }
ivory1, ivory	{ 255, 255, 240 }
ivory2	{ 238, 238, 224 }
ivory3	{ 205, 205, 193 }
ivory4	{ 139, 139, 131 }
LemonChiffon1, lemon chiffon	{ 255, 250, 205 }
LemonChiffon2	{ 238, 233, 191 }
LemonChiffon3	{ 205, 201, 165 }
LemonChiffon4	{ 139, 137, 112 }
seashell1, seashell	{ 255, 245, 238 }
seashell2	{ 238, 229, 222 }
seashell3	{ 205, 197, 191 }
seashell4	{ 139, 134, 130 }
honeydew	{ 240, 255, 240 }
honeydew2	{ 224, 238, 224 }
honeydew3	{ 193, 205, 193 }
honeydew4	{ 131, 139, 131 }
mint cream	{ 245, 255, 250 }
azure1, azure	{ 240, 255, 255 }
azure2	{ 224, 238, 238 }
azure3	{ 193, 205, 205 }
azure4	{ 131, 139, 139 }
alice blue	{ 240, 248, 255 }
lavender	{ 230, 230, 250 }
LavenderBlush1, lavender blush	{ 255, 240, 245 }
LavenderBlush2	{ 238, 224, 229 }
LavenderBlush3	{ 205, 193, 197 }
LavenderBlush4	{ 139, 131, 134 }
MistyRose1, misty rose	{ 255, 228, 225 }

Color String(s)	RGB
MistyRose2	{ 238, 213, 210 }
MistyRose3	{ 205, 183, 181 }
MistyRose4	{ 139, 125, 123 }
DarkSlateGray1	{ 151, 255, 255 }
DarkSlateGray2	{ 141, 238, 238 }
DarkSlateGray3	{ 121, 205, 205 }
DarkSlateGray4	{ 82, 139, 139 }
dark slate gray	{ 47, 79, 79 }
SlateGray1	{ 198, 226, 255 }
SlateGray2	{ 185, 211, 238 }
SlateGray3	{ 159, 182, 205 }
SlateGray4	{ 108, 123, 139 }
slate gray	{ 112, 128, 144 }
light slate gray	{ 119, 136, 153 }
midnight blue	{ 25, 25, 112 }
navy, navy blue	{ 0, 0, 128 }
cornflower blue	{ 100, 149, 237 }
slate blue	{ 106, 90, 205 }
medium slate blue	{ 123, 104, 238 }
light slate blue	{ 132, 112, 255 }
SlateBlue1	{ 131, 111, 255 }
SlateBlue2	{ 122, 103, 238 }
SlateBlue3	{ 105, 89, 205 }
SlateBlue4	{ 71, 60, 139 }
dark slate blue	{ 72, 61, 139 }
medium blue	{ 0, 0, 205 }
royal blue	{ 65, 105, 225 }
RoyalBlue1	{ 72, 118, 255 }
RoyalBlue2	{ 67, 110, 238 }
RoyalBlue3	{ 58, 95, 205 }
RoyalBlue4	{ 39, 64, 139 }
blue1, blue	{ 0, 0, 255 }
blue2	{ 0, 0, 238 }
blue3	{ 0, 0, 205 }
blue4, dark blue	{ 0, 0, 139 }
DodgerBlue1, dodger blue	{ 30, 144, 255 }
DodgerBlue2	{ 28, 134, 238 }

Color String(s)	RGB
DodgerBlue3	{ 24, 116, 205 }
DodgerBlue4	{ 16, 78, 139 }
DeepSkyBlue1, deep sky blue	{ 0, 191, 255 }
DeepSkyBlue2	{ 0, 178, 238 }
DeepSkyBlue3	{ 0, 154, 205 }
DeepSkyBlue4	{ 0, 104, 139 }
sky blue	{ 135, 206, 235 }
SkyBlue1	{ 135, 206, 255 }
SkyBlue2	{ 126, 192, 238 }
SkyBlue3	{ 108, 166, 205 }
SkyBlue4	{ 74, 112, 139 }
light sky blue	{ 135, 206, 250 }
LightSkyBlue1	{ 176, 226, 255 }
LightSkyBlue2	{ 164, 211, 238 }
LightSkyBlue3	{ 141, 182, 205 }
LightSkyBlue4	{ 96, 123, 139 }
steel blue	{ 70, 130, 180 }
SteelBlue1	{ 99, 184, 255 }
SteelBlue2	{ 92, 172, 238 }
SteelBlue3	{ 79, 148, 205 }
SteelBlue4	{ 54, 100, 139 }
light steel blue	{ 176, 196, 222 }
LightSteelBlue1	{ 202, 225, 255 }
LightSteelBlue2	{ 188, 210, 238 }
LightSteelBlue3	{ 162, 181, 205 }
LightSteelBlue4	{ 110, 123, 139 }
light blue	{ 173, 216, 230 }
LightBlue1	{ 191, 239, 255 }
LightBlue2	{ 178, 223, 238 }
LightBlue3	{ 154, 192, 205 }
LightBlue4	{ 104, 131, 139 }
powder blue	{ 176, 224, 230 }
pale turquoise	{ 175, 238, 238 }
PaleTurquoise1	{ 187, 255, 255 }
PaleTurquoise2	{ 174, 238, 238 }
PaleTurquoise3	{ 150, 205, 205 }
PaleTurquoise4	{ 102, 139, 139 }

Color String(s)	RGB
dark turquoise	{ 0, 206, 209 }
medium turquoise	{ 72, 209, 204 }
turquoise	{ 64, 224, 208 }
Turquoise1	{ 0, 245, 255 }
Turquoise2	{ 0, 229, 238 }
Turquoise3	{ 0, 197, 205 }
Turquoise4	{ 0, 134, 139 }
cyan1, cyan	{ 0, 255, 255 }
cyan2	{ 0, 238, 238 }
cyan3	{ 0, 205, 205 }
cyan4, dark cyan	{ 0, 139, 139 }
LightCyan1, light cyan	{ 224, 255, 255 }
LightCyan2	{ 209, 238, 238 }
LightCyan3	{ 180, 205, 205 }
LightCyan4	{ 122, 139, 139 }
cadet blue	{ 95, 158, 160 }
CadetBlue1	{ 152, 245, 255 }
CadetBlue2	{ 142, 229, 238 }
CadetBlue3	{ 122, 197, 205 }
CadetBlue4	{ 83, 134, 139 }
medium aquamarine	{ 102, 205, 170 }
aquamarine1, aquamarine	{ 127, 255, 212 }
aquamarine2	{ 118, 238, 198 }
aquamarine3	{ 102, 205, 170 }
aquamarine4	{ 69, 139, 116 }
dark green	{ 0, 100, 0 }
dark olive green	{ 85, 107, 47 }
DarkOliveGreen1	{ 202, 255, 112 }
DarkOliveGreen2	{ 188, 238, 104 }
DarkOliveGreen3	{ 162, 205, 90 }
DarkOliveGreen4	{ 110, 139, 61 }
dark sea green	{ 143, 188, 143 }
DarkSeaGreen1	{ 193, 255, 193 }
DarkSeaGreen2	{ 180, 238, 180 }
DarkSeaGreen3	{ 155, 205, 155 }
DarkSeaGreen4	{ 105, 139, 105 }
SeaGreen1	{ 84, 255, 159 }

Color String(s)	RGB
SeaGreen2	{ 78, 238, 148 }
SeaGreen3	{ 67, 205, 128 }
SeaGreen4, sea green	{ 46, 139, 87 }
medium sea green	{ 60, 179, 113 }
light sea green	{ 32, 178, 170 }
pale green	{ 152, 251, 152 }
PaleGreen1	{ 154, 255, 154 }
PaleGreen2	{ 144, 238, 144 }
PaleGreen3	{ 124, 205, 124 }
PaleGreen4	{ 84, 139, 84 }
SpringGreen1, spring green	{ 0, 255, 127 }
SpringGreen2	{ 0, 238, 118 }
SpringGreen3	{ 0, 205, 102 }
SpringGreen4	{ 0, 139, 69 }
medium spring green	{ 0, 250, 154 }
lawn green	{ 124, 252, 0 }
light green	{ 144, 238, 144 }
green1, green	{ 0, 255, 0 }
green2	{ 0, 238, 0 }
green3	{ 0, 205, 0 }
green4	{ 0, 139, 0 }
chartreuse1, chartreuse	{ 127, 255, 0 }
chartreuse2	{ 118, 238, 0 }
chartreuse3	{ 102, 205, 0 }
chartreuse4	{ 69, 139, 0 }
green yellow	{ 173, 255, 47 }
lime green	{ 50, 205, 50 }
yellow green	{ 154, 205, 50 }
forest green	{ 34, 139, 34 }
olive drab	{ 107, 142, 35 }
OliveDrab1	{ 192, 255, 62 }
OliveDrab2	{ 179, 238, 58 }
OliveDrab3	{ 154, 205, 50 }
OliveDrab4	{ 105, 139, 34 }
dark khaki	{ 189, 183, 107 }
khaki	{ 240, 230, 140 }
khaki1	{ 255, 246, 143 }

Color String(s)	RGB
khaki2	{ 238, 230, 133 }
khaki3	{ 205, 198, 115 }
khaki4	{ 139, 134, 78 }
LightYellow1, light yellow	{ 255, 255, 224 }
LightYellow2	{ 238, 238, 209 }
LightYellow3	{ 205, 205, 180 }
LightYellow4	{ 139, 139, 122 }
yellow1, yellow	{ 255, 255, 0 }
yellow2	{ 238, 238, 0 }
yellow3	{ 205, 205, 0 }
yellow4	{ 139, 139, 0 }
gold1, gold	{ 255, 215, 0 }
gold2	{ 238, 201, 0 }
gold3	{ 205, 173, 0 }
gold4	{ 139, 117, 0 }
pale goldenrod	{ 238, 232, 170 }
light goldenrod yellow	{ 250, 250, 210 }
light goldenrod	{ 238, 221, 130 }
LightGoldenrod1	{ 255, 236, 139 }
LightGoldenrod2	{ 238, 220, 130 }
LightGoldenrod3	{ 205, 190, 112 }
LightGoldenrod4	{ 139, 129, 76 }
goldenrod	{ 218, 165, 32 }
goldenrod1	{ 255, 193, 37 }
goldenrod2	{ 238, 180, 34 }
goldenrod3	{ 205, 155, 29 }
goldenrod4	{ 139, 105, 20 }
dark goldenrod	{ 184, 134, 11 }
DarkGoldenrod1	{ 255, 185, 15 }
DarkGoldenrod2	{ 238, 173, 14 }
DarkGoldenrod3	{ 205, 149, 12 }
DarkGoldenrod4	{ 139, 101, 8 }
rosy brown	{ 188, 143, 143 }
RosyBrown1	{ 255, 193, 193 }
RosyBrown2	{ 238, 180, 180 }
RosyBrown3	{ 205, 155, 155 }
RosyBrown4	{ 139, 105, 105 }

Color String(s)	RGB
indian red	{ 205, 92, 92 }
IndianRed1	{ 255, 106, 106 }
IndianRed2	{ 238, 99, 99 }
IndianRed3	{ 205, 85, 85 }
IndianRed4	{ 139, 58, 58 }
saddle brown	{ 139, 69, 19 }
sienna	{ 160, 82, 45 }
sienna1	{ 255, 130, 71 }
sienna2	{ 238, 121, 66 }
sienna3	{ 205, 104, 57 }
sienna4	{ 139, 71, 38 }
peru	{ 205, 133, 63 }
burlywood	{ 222, 184, 135 }
burlywood1	{ 255, 211, 155 }
burlywood2	{ 238, 197, 145 }
burlywood3	{ 205, 170, 125 }
burlywood4	{ 139, 115, 85 }
beige	{ 245, 245, 220 }
wheat	{ 245, 222, 179 }
wheat1	{ 255, 231, 186 }
wheat2	{ 238, 216, 174 }
wheat3	{ 205, 186, 150 }
wheat4	{ 139, 126, 102 }
sandy brown	{ 244, 164, 96 }
tan	{ 210, 180, 140 }
tan1	{ 255, 165, 79 }
tan2	{ 238, 154, 73 }
tan3	{ 205, 133, 63 }
tan4	{ 139, 90, 43 }
chocolate	{ 210, 105, 30 }
chocolate1	{ 255, 127, 36 }
chocolate2	{ 238, 118, 33 }
chocolate3	{ 205, 102, 29 }
chocolate4	{ 139, 69, 19 }
firebrick	{ 178, 34, 34 }
firebrick1	{ 255, 48, 48 }
firebrick2	{ 238, 44, 44 }

Color String(s)	RGB
firebrick3	{ 205, 38, 38 }
firebrick4	{ 139, 26, 26 }
brown	{ 165, 42, 42 }
brown1	{ 255, 64, 64 }
brown2	{ 238, 59, 59 }
brown3	{ 205, 51, 51 }
brown4	{ 139, 35, 35 }
dark salmon	{ 233, 150, 122 }
salmon	{ 250, 128, 114 }
salmon1	{ 255, 140, 105 }
salmon2	{ 238, 130, 98 }
salmon3	{ 205, 112, 84 }
salmon4	{ 139, 76, 57 }
LightSalmon1, light salmon	{ 255, 160, 122 }
LightSalmon2	{ 238, 149, 114 }
LightSalmon3	{ 205, 129, 98 }
LightSalmon4	{ 139, 87, 66 }
orange1, orange	{ 255, 165, 0 }
orange2	{ 238, 154, 0 }
orange3	{ 205, 133, 0 }
orange4	{ 139, 90, 0 }
dark orange	{ 255, 140, 0 }
DarkOrange1	{ 255, 127, 0 }
DarkOrange2	{ 238, 118, 0 }
DarkOrange3	{ 205, 102, 0 }
DarkOrange4	{ 139, 69, 0 }
coral	{ 255, 127, 80 }
coral1	{ 255, 114, 86 }
coral2	{ 238, 106, 80 }
coral3	{ 205, 91, 69 }
coral4	{ 139, 62, 47 }
light coral	{ 240, 128, 128 }
tomato1, tomato	{ 255, 99, 71 }
tomato2	{ 238, 92, 66 }
tomato3	{ 205, 79, 57 }
tomato4	{ 139, 54, 38 }
OrangeRed1, orange red	{ 255, 69, 0 }

Color String(s)	RGB
OrangeRed2	{ 238, 64, 0 }
OrangeRed3	{ 205, 55, 0 }
OrangeRed4	{ 139, 37, 0 }
red1, red	{ 255, 0, 0 }
red2	{ 238, 0, 0 }
red3	{ 205, 0, 0 }
red4, dark red	{ 139, 0, 0 }
hot pink	{ 255, 105, 180 }
HotPink1	{ 255, 110, 180 }
HotPink2	{ 238, 106, 167 }
HotPink3	{ 205, 96, 144 }
HotPink4	{ 139, 58, 98 }
DeepPink1, deep pink	{ 255, 20, 147 }
DeepPink2	{ 238, 18, 137 }
DeepPink3	{ 205, 16, 118 }
DeepPink4	{ 139, 10, 80 }
pink	{ 255, 192, 203 }
pink1	{ 255, 181, 197 }
pink2	{ 238, 169, 184 }
pink3	{ 205, 145, 158 }
pink4	{ 139, 99, 108 }
light pink	{ 255, 182, 193 }
LightPink1	{ 255, 174, 185 }
LightPink2	{ 238, 162, 173 }
LightPink3	{ 205, 140, 149 }
LightPink4	{ 139, 95, 101 }
maroon	{ 176, 48, 96 }
maroon1	{ 255, 52, 179 }
maroon2	{ 238, 48, 167 }
maroon3	{ 205, 41, 144 }
maroon4	{ 139, 28, 98 }
pale violet red	{ 219, 112, 147 }
PaleVioletRed1	{ 255, 130, 171 }
PaleVioletRed2	{ 238, 121, 159 }
PaleVioletRed3	{ 205, 104, 137 }
PaleVioletRed4	{ 139, 71, 93 }
medium violet red	{ 199, 21, 133 }

Color String(s)	RGB
violet red	{ 208, 32, 144 }
VioletRed1	{ 255, 62, 150 }
VioletRed2	{ 238, 58, 140 }
VioletRed3	{ 205, 50, 120 }
VioletRed4	{ 139, 34, 82 }
magenta1, magenta	{ 255, 0, 255 }
magenta2	{ 238, 0, 238 }
magenta3	{ 205, 0, 205 }
magenta4, dark magenta	{ 139, 0, 139 }
violet	{ 238, 130, 238 }
plum	{ 221, 160, 221 }
plum1	{ 255, 187, 255 }
plum2	{ 238, 174, 238 }
plum3	{ 205, 150, 205 }
plum4	{ 139, 102, 139 }
orchid	{ 218, 112, 214 }
orchid1	{ 255, 131, 250 }
orchid2	{ 238, 122, 233 }
orchid3	{ 205, 105, 201 }
orchid4	{ 139, 71, 137 }
medium orchid	{ 186, 85, 211 }
MediumOrchid1	{ 224, 102, 255 }
MediumOrchid2	{ 209, 95, 238 }
MediumOrchid3	{ 180, 82, 205 }
MediumOrchid4	{ 122, 55, 139 }
dark orchid	{ 153, 50, 204 }
DarkOrchid1	{ 191, 62, 255 }
DarkOrchid2	{ 178, 58, 238 }
DarkOrchid3	{ 154, 50, 205 }
DarkOrchid4	{ 104, 34, 139 }
dark violet	{ 148, 0, 211 }
blue violet	{ 138, 43, 226 }
purple	{ 160, 32, 240 }
purple1	{ 155, 48, 255 }
purple2	{ 145, 44, 238 }
purple3	{ 125, 38, 205 }
purple4	{ 85, 26, 139 }

Color String(s)	RGB
medium purple	{ 147, 112, 219 }
MediumPurple1	{ 171, 130, 255 }
MediumPurple2	{ 159, 121, 238 }
MediumPurple3	{ 137, 104, 205 }
MediumPurple4	{ 93, 71, 139 }
thistle	{ 216, 191, 216 }
thistle1	{ 255, 225, 255 }
thistle2	{ 238, 210, 238 }
thistle3	{ 205, 181, 205 }
thistle4	{ 139, 123, 139 }
gray0, black	{ 0, 0, 0 }
gray1	{ 3, 3, 3 }
gray2	{ 5, 5, 5 }
gray3	{ 8, 8, 8 }
gray4	{ 10, 10, 10 }
gray5	{ 13, 13, 13 }
gray6	{ 15, 15, 15 }
gray7	{ 18, 18, 18 }
gray8	{ 20, 20, 20 }
gray9	{ 23, 23, 23 }
gray10	{ 26, 26, 26 }
gray11	{ 28, 28, 28 }
gray12	{ 31, 31, 31 }
gray13	{ 33, 33, 33 }
gray14	{ 36, 36, 36 }
gray15	{ 38, 38, 38 }
gray16	{ 41, 41, 41 }
gray17	{ 43, 43, 43 }
gray18	{ 46, 46, 46 }
gray19	{ 48, 48, 48 }
gray20	{ 51, 51, 51 }
gray21	{ 54, 54, 54 }
gray22	{ 56, 56, 56 }
gray23	{ 59, 59, 59 }
gray24	{ 61, 61, 61 }
gray25	{ 64, 64, 64 }
gray26	{ 66, 66, 66 }

Color String(s)	RGB
gray27	{ 69, 69, 69 }
gray28	{ 71, 71, 71 }
gray29	{ 74, 74, 74 }
gray30	{ 77, 77, 77 }
gray31	{ 79, 79, 79 }
gray32	{ 82, 82, 82 }
gray33	{ 84, 84, 84 }
gray34	{ 87, 87, 87 }
gray35	{ 89, 89, 89 }
gray36	{ 92, 92, 92 }
gray37	{ 94, 94, 94 }
gray38	{ 97, 97, 97 }
gray39	{ 99, 99, 99 }
gray40	{ 102, 102, 102 }
gray41, dim gray	{ 105, 105, 105 }
gray42	{ 107, 107, 107 }
gray43	{ 110, 110, 110 }
gray44	{ 112, 112, 112 }
gray45	{ 115, 115, 115 }
gray46	{ 117, 117, 117 }
gray47	{ 120, 120, 120 }
gray48	{ 122, 122, 122 }
gray49	{ 125, 125, 125 }
gray50	{ 127, 127, 127 }
gray51	{ 130, 130, 130 }
gray52	{ 133, 133, 133 }
gray53	{ 135, 135, 135 }
gray54	{ 138, 138, 138 }
gray55	{ 140, 140, 140 }
gray56	{ 143, 143, 143 }
gray57	{ 145, 145, 145 }
gray58	{ 148, 148, 148 }
gray59	{ 150, 150, 150 }
gray60	{ 153, 153, 153 }
gray61	{ 156, 156, 156 }
gray62	{ 158, 158, 158 }
gray63	{ 161, 161, 161 }

Color String(s)	RGB
gray64	{ 163, 163, 163 }
gray65	{ 166, 166, 166 }
gray66	{ 168, 168, 168 }
dark gray	{ 169, 169, 169 }
gray67	{ 171, 171, 171 }
gray68	{ 173, 173, 173 }
gray69	{ 176, 176, 176 }
gray70	{ 179, 179, 179 }
gray71	{ 181, 181, 181 }
gray72	{ 184, 184, 184 }
gray73	{ 186, 186, 186 }
gray74	{ 189, 189, 189 }
gray	{ 190, 190, 190 }
gray75	{ 191, 191, 191 }
gray76	{ 194, 194, 194 }
gray77	{ 196, 196, 196 }
gray78	{ 199, 199, 199 }
gray79	{ 201, 201, 201 }
gray80	{ 204, 204, 204 }
gray81	{ 207, 207, 207 }
gray82	{ 209, 209, 209 }
light gray	{ 211, 211, 211 }
gray83	{ 212, 212, 212 }
gray84	{ 214, 214, 214 }
gray85	{ 217, 217, 217 }
gray86	{ 219, 219, 219 }
gray87	{ 222, 222, 222 }
gray88	{ 224, 224, 224 }
gray89	{ 227, 227, 227 }
gray90	{ 229, 229, 229 }
gray91	{ 232, 232, 232 }
gray92	{ 235, 235, 235 }
gray93	{ 237, 237, 237 }
gray94	{ 240, 240, 240 }
gray95	{ 242, 242, 242 }
gray96	{ 245, 245, 245 }
gray97	{ 247, 247, 247 }

Color String(s)	RGB
gray98	{ 250, 250, 250 }
gray99	{ 252, 252, 252 }
gray100, white	{ 255, 255, 255 }

Importing 3D Model Files

3D model files can be imported directly into the active **Modeler** window. Supported 3D model file formats are listed in the table that follows.

Type	Extension(s)	Supported Version(s)
ACIS SAB	*.sab	Up to 2024
ACIS SAT	*.sat	Up to 2024
Ansoft Geometry	*.AnstGeom	All
Ansys 3D Modeler	*.sm3	Up to 2022R2 See UDMs and CAD integration with Workbench.
AutoCAD	*.dxf, *.dwg	**Linux**: Versions 2.5 (AC1002) through 2012 (AC1024) **Windows**: Versions 2.5 (AC1002) through 2017 (AC1027) See Importing DXF and DWG Format Files.
Autodesk Inventor	*.ipt, *.iam	Up to 2025
CATIA	*.exp, *.model, *.CATPart, *.CATproduct	CATIA V4 – Up to 4.2.5 CATIA V5 – Up to V5_6R2024
Creo Parametric	*.prt, *.asm	Pro/Engineer 19.0 to Creo 11.0
IGES	*.iges, *.igs	5.1, 5.2, 5.3
JT	*.jt	Up to v10.9
NASTRAN	*.nas	CROD, CBEAM, CTRIA3, and CQUAD4 element types
Siemens NX	*.prt	UG11 to UG18, UG NX, NX5 to NX12, NX1847 to NX2406
Parasolid	*.x_t, *.x_b	9.0 through 36.1.185
SOLIDWORKS	*.SLDPRT, *.SLDASM	From 97 up to 2024
STEP	*.step, *.stp	AP 203 E1/E2, AP 214, AP 242 (geometry only)

Type	Extension(s)	Supported Version(s)
STL	*.stl	2.0 (see Lightweight Geometry in SBR+)

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. Set any import options available for the selected file type. Options 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 IGES files, specify 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.

   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 and select a file for import.
5. Click **Open**.

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

> **Note:**
>
> While objects created in Q3D Extractor can always be classed in the history tree as either a solid, sheet, or wire, some imported objects may have mixture of these. Q3D Extractor places such objects in an **Unclassified** folder in the history tree.

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

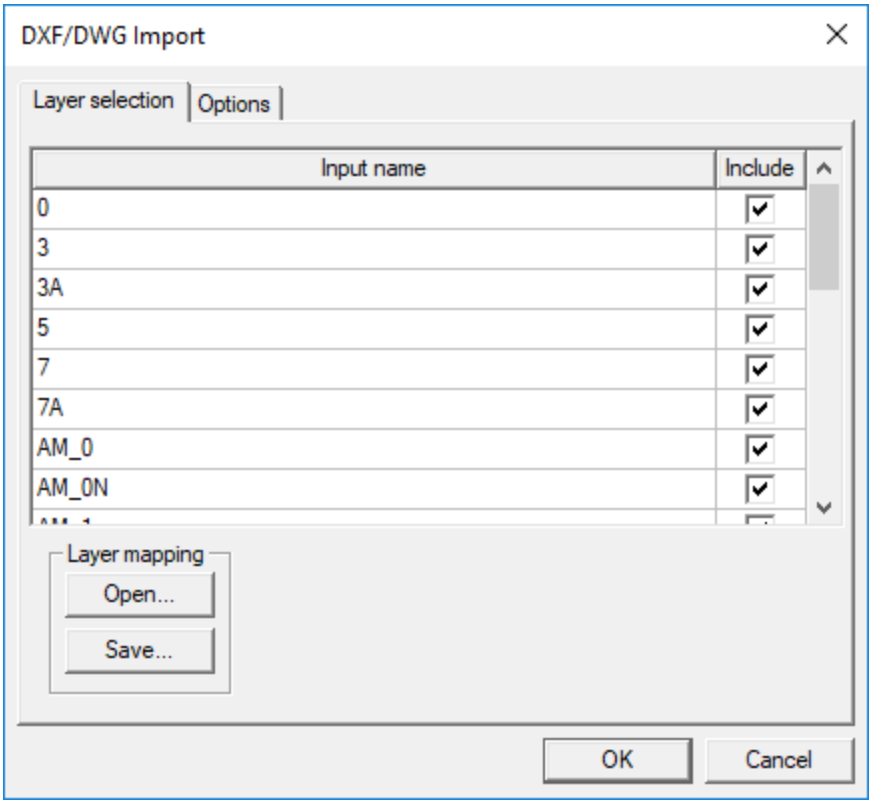
To import a ***.dxf** or ***.dwg** model file (which may use an associated *.tech file):

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

   The **Import File** dialog box appears.

2. Use the file browser to find and select an AutoCAD file for import.
3. Click **Open**.

   The **DXF/DWG Import** dialog box opens with the **Layer Selection** tab selected.



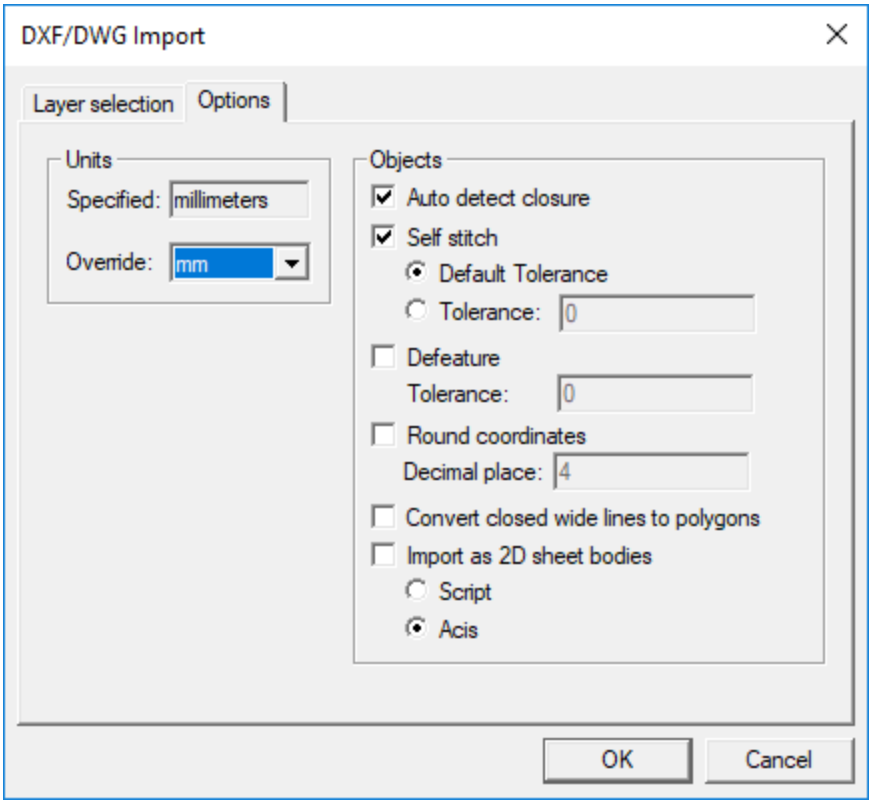
The **Input Name** field shows the name of the layer in the DXF/DWG file. This field is not editable.

4. Use the **Include** check boxes to specify which layers to import from the selected file.
5. If there is an associated *.tech file, click **Open** to find and select it. A *.tech file is a plain text file that includes units, layer names, color, elevation, and thickness information. For example:

```
units um

//Layer_Name Color Elevation Thickness

BOTTOMLAYER purple 0 200

MIDLAYER green 500 200

TOPLAYER blue 1000 200
```

6. Click **Options**.

   The **Options** tab displays.



7. In the **Units** area, the **Specified** field displays the layout units in the file for import. Use the **Override** drop-down menu to select a different unit of measure.
8. In the **Objects** area, use the 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** joins multiple straight line segments to form polylines. If the resulting polyline is closed, a polygon is created.
        - **Default Tolerance** uses default values to whether two coordinates should be considered for joining.
        - **Tolerance** allows you to specify a specific value for tolerance. This is useful if particular features in a model are outside of a normal tolerance allowance.
    - **De-feature 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 Decimal Place** rounds all imported data to the specified number of decimal points.
    - **Convert closed wide lines to polygons** imports wide polylines as polygons. This affords more flexibility in changing the shape of such an object.

- **Import as 2D sheet bodies** causes imported objects to be organized in terms of 2D sheets.

9. Select either **Script** or **Acis** for the import method.
10. Click **OK**.

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

Exporting Files

You can export the following types of files from Ansys Electronics Desktop projects:

- 2D model files
- 3D model files
- Graphics files
- Reports as data or graphics files

Exporting 2D Model Files

Q3D Extractor exports 2D models in AutoCAD *.dxf format. In this format, the geometry located within the XY plane is exported.

> **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.

To export a file to ***.dxf** 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 **AutoCAD DXF Files (*.dxf)**.
5. Click **Save**.

   The file is exported to the specified location with the appropriate file format.

Exporting 3D Model Files

You can export 3D models to 3D model file formats.

Supported 3D model file formats include the following:

Type	Extension(s)
ACIS SAB (exported with version 5.0)	*.sab
ACIS SAT (exported with version 5.0)	*.sat
AutoCAD	*.dxf
GDSII	*.gds
IGES	*.iges, *.igs
OBJ Files (for importing models to Ensight.)	*.obj
Parasolid	*.x_t, *.x_b
STEP	*.step, *.stp
STL	*.stl
GLTF (only available on Windows)	.gltf or .glb

To export a file to a 3D model format:

1. Click **Modeler** > **Export**.

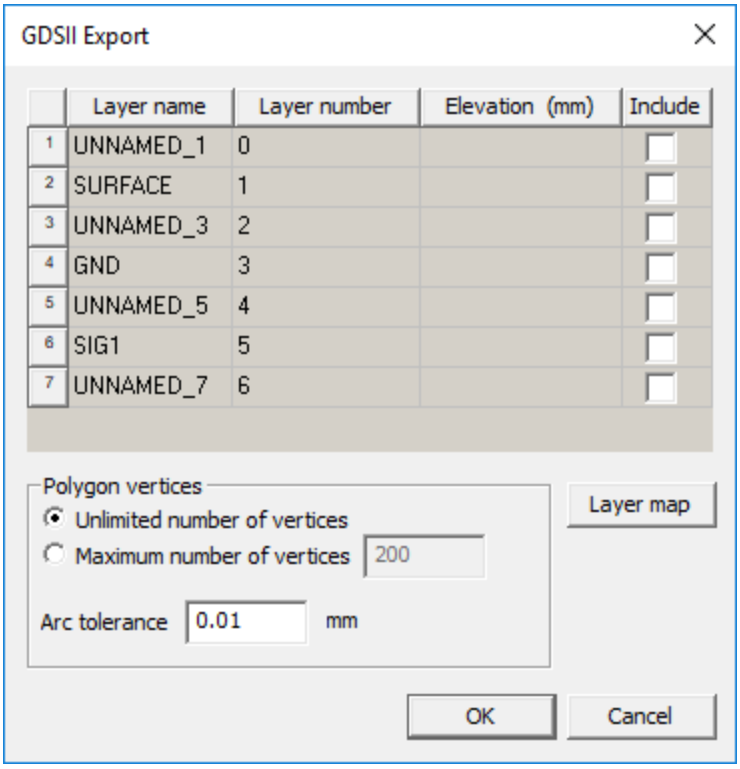
   The **Export File** window appears.

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 file type.
5. Click **Save**.

   The file is exported to the specified location with the appropriate file format. Some file types launch an additional dialog box for specifying settings prior to export. These are described below.

GDSII Files

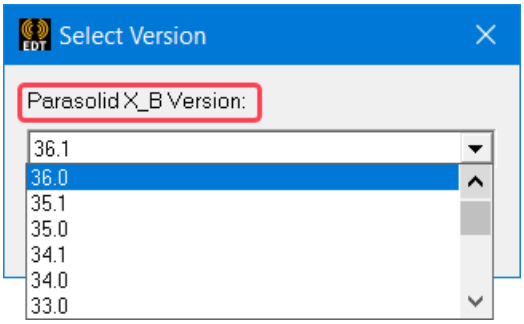
There are additional options for *.gds files:

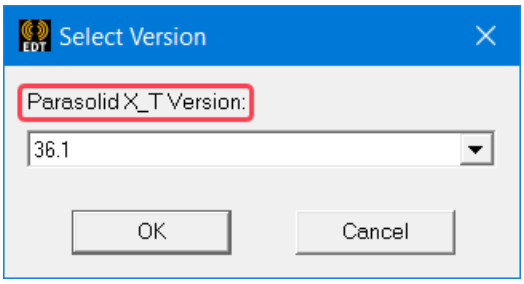


- Use the **Include** check boxes to select layers for export.
- Click **Layer map** to define a *.layermap file, if desired. This is a text file that maps GDSII layer numbers to layer names in the stackup. The *.layermap file can have the same format as the .tech file used in GDSII import, but it only 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.

Parasolid Files

There is one additional option for Parasolid files:





Use the drop-down menu to select a Parasolid version.

Exporting Graphics Files

You can export to the following graphics formats:

Extension	Contents
.bmp	Bitmap files
.gif	Graphics Interchange Format files
.jpg	Joint Photographics Experts Group files
.png	Portable Network Graphics format files
.tiff	Tagged Image File Format files.
.wrl	Virtual Reality Modeling Language (VRML) files.
.gltf or **.glb**	Graphics Language Transmission Format (GLTF); only available on Windows

To export a Modeler image to a graphics format:

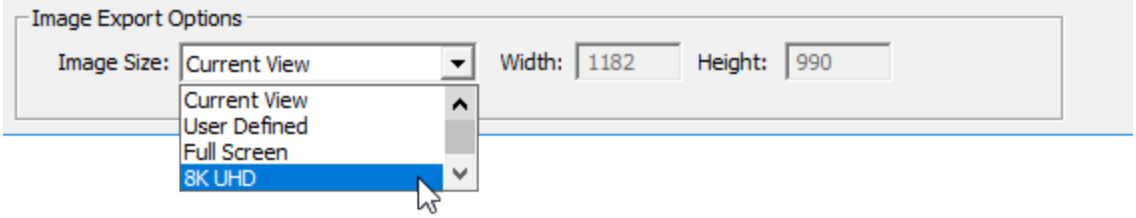
1. Click **Modeler** > **Export**.

   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 image format.

   When you select an image format, **Image Export Options** appear. Note that this window does not appear for .gltf, .glb, or .wrl files, which are not image formats; if you are saving to one of these formats, skip to Step 6.



5. Choose your **Image Size**.

   Options include: Current View, User Defined (which allows you to specify **Width** and **Height**), Full Screen, 8K UHD, 4K UHD, 1080p HD, 720p HD, and 480p SD.

6. Click **Save**.

   The file is exported to the specified location with the appropriate file format.

You can also export an image file to a specified resolution using scripting commands. Fonts and line thickness are not scaled, only the image. You will have to iteratively increase font sizes until you find a suitable output.

Example Script:

```
oEditor.ExportModelImageToFile("C:/Users/Documents/highresexample_
image.jpg", 7680, 4320,

[

"NAME:SaveImageParams",

"ShowAxis:="        , "True",

"ShowGrid:="        , "True",

"ShowRuler:="       , "True",

"ShowRegion:="      , "Default",

"Selections:="      , ""

])
```

Exporting Options Files

Ansys Electronics Desktop can export user and host options files in XML format.

Exporting Options Files Using the Desktop UI

Export options files by selecting **Tools** > **Options** > **Export Options Files**. This brings up a browser dialog box to select the destination directory.

Click **Open** to copy all 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.

Exporting Options Files Using a Script

A script command has been added that exports options config files:

ExportOptionsFiles

Syntax	`ExportOptionsFiles <DestinationDirectory>`
Return Value	None.
Parameters	BSTR <DestinationDirectory>

VBScript Example	`Dim oAnsoftApp`
	`Dim oDesktop`
	`Set oAnsoftApp = CreateObject ("Ansoft.ElectronicsDesktop")`
	`Set oDesktop = oAnsoftApp.GetAppDesktop()`
	`oDesktop.ExportOptionsFiles "D:/test/export/"`

IPY Example	`oDesktop.ExportOptionsFiles ('D:/test/export')`

Validating Projects

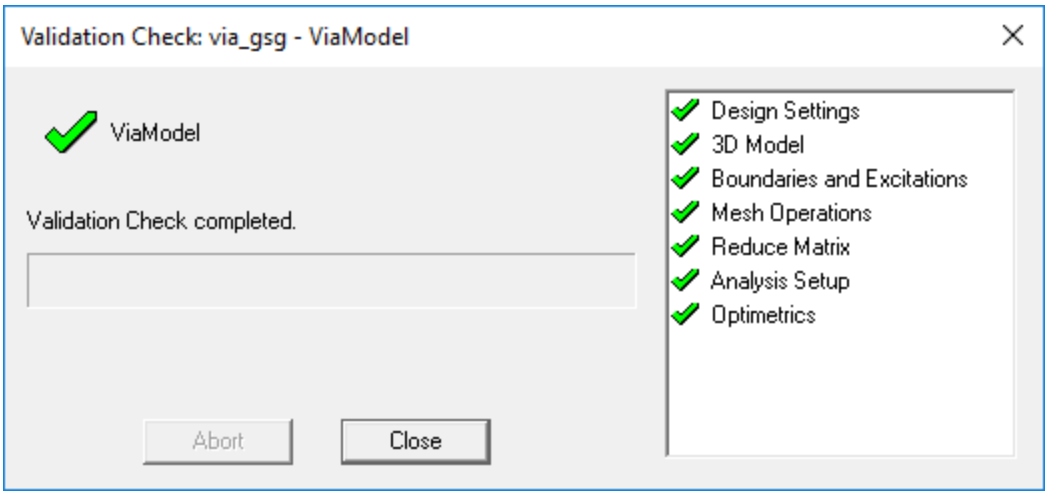
Before running an analysis on a model, it is important to first perform a validation check on the project. During a validation check, the solver examines the setup details of the active project to verify that all the necessary steps have been completed and that their parameters are reasonable.

To perform a validation check on the active project:

1. Click **Q3D** > **Validation Check**. You can also click the **Validate** icon on the **Simulation** ribbon tab.

   Electronics Desktop checks the project setup, and the **Validation Check** window appears.

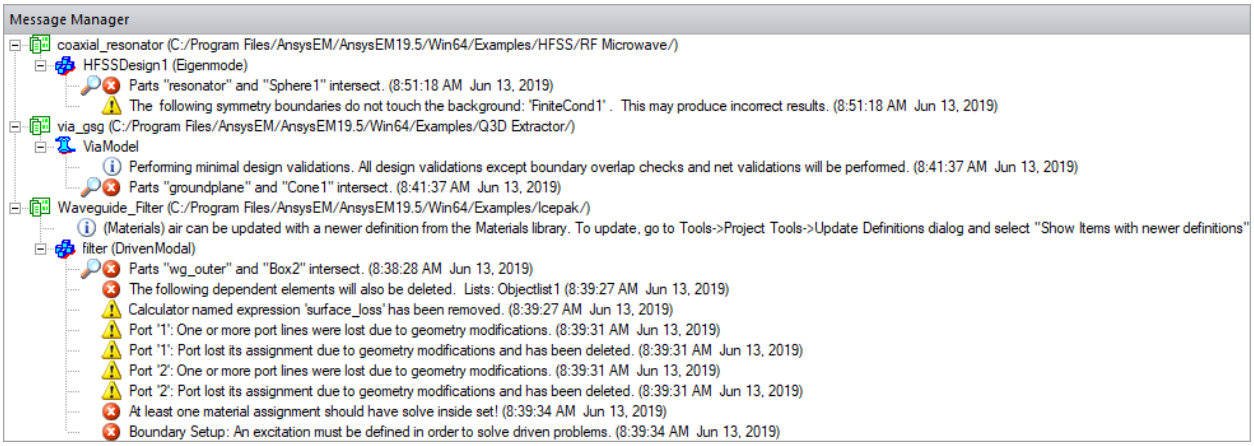
2. View the results of the validation check in the **Validation Check** window.

   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. After you have revised any setup details, click **Q3D** > **Validation Check** to run another validation check. Repeat until the Validation Check completes with no issues.

> **Note:**
>
> You can change the strictness with which 3D Model faults are validated.

6. Click **Close**.

Modeler Validation Settings

You can adjust the degree to which the software checks a model for faults that could jeopardize mesh accuracy.

There are three levels of model validation for a given design: **Warning Only**, **Basic**, and **Strict**.

> **Note:**
>
> This setting only affects the "3D Model" stage of a design validation.

- **Warning Only** – allows all models to pass 3D Model validation regardless of any faults that are found (acis_entity check errors). These faults are posted in the message window as warnings.
- **Basic** – allows most models to pass 3D Model validation. This excuses non-manifold errors and most acis_entity_check errors. Some faults are flagged as model errors (basic entity check errors), thereby prohibiting a design from proceeding to the meshing stage of an analysis. You must either correct such errors before attempting to analyze the design under the **Basic** setting, or change the **Model Validation** level to **Warning Only**.
- **Strict** – enforces a tighter tolerance for model faults than the **Warning Only** and **Basic** settings. All model faults 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, otherwise you must change the **Model Validation** level to **Basic** or **Warning Only**.

To set the Model Validation level:

1. Select **Q3D Extractor** > **Design Settings**.

   The **Design Settings** dialog box appears. The **Validations** tab lets you set the validation as **Basic**, **Strict**, or **Warning Only** and includes solver-specific selections.

2. Choose the desired level of validation from the **Entity Check Level** drop-down menu.

   You can also click the **Save as Default** button to make the current selection the default, or select **Restore Default**.

3. Click **OK** to accept the selection and close the dialog box.

> **Note:** For more information on Maxwell-specific settings, see Maxwell Validation Tab.

Copying and Pasting a Project or Design

To **copy** a project or design:

1. In the Project Tree, select a project or design 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. In the Project Tree, select a project or design to enable the menu command **Edit** > **Paste**.
2. Click **Edit** > **Paste**.

   The project or design is pasted under the selected project, and an icon is added to the project tree.

> **Note:**
>
> You can also use keyboard shortcuts.

Renaming Projects

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 rename an existing, active project:

1. Select the project in the Project Tree.
2. Right-click and select **Rename**.

   This activates the text field for the project name.

3. Type the new project name and press **Enter**.

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

Archiving Projects

Use the **File** > **Archive** command to place a project and any other files related to the project that you want to include in a *.<product>z file or *.zip format archive. You can make notes about the contents of the archive and specify whether to include results and solutions files. The **Archive** command attempts to automatically detect the necessary files for linked projects and automatically include them in the archive. You can also add additional files to the archive, including results files, external files and projects. For example, if a project linked to the main project also has linked or associated files, you can add them.

Archive File Types

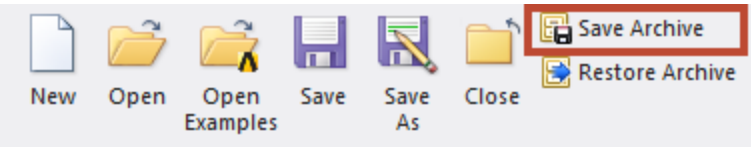
Internally, project archive files are .zip files, and are compatible with any program that can read .zip files. Project archive files have an extension that is unique for each product, and is generated by adding a 'z' to the project file extension (for example, *.aedtz, *.adsnz). This extension displays as the default when saving and restoring archive files.

Archive Preview

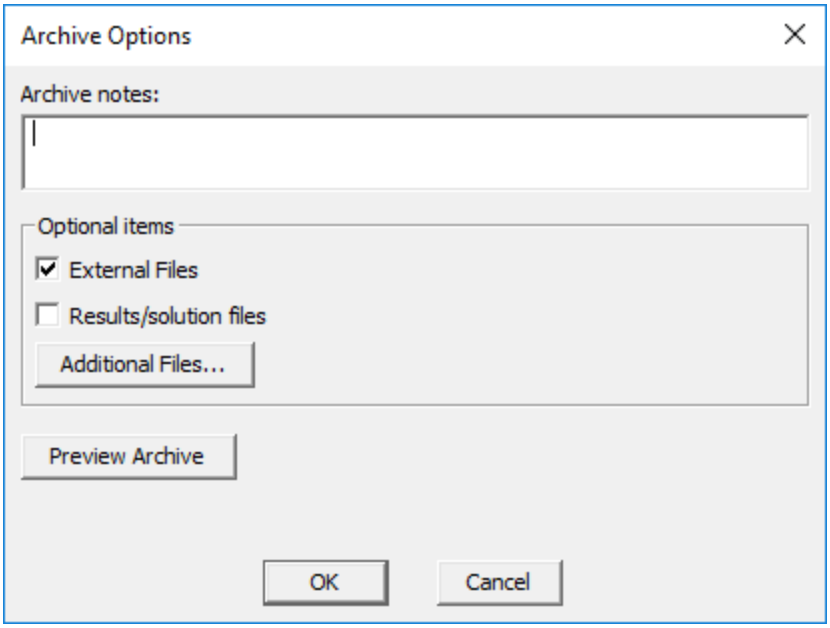
The **Archive** command includes a preview feature that lets you review the contents of a planned archive.

To archive the current project:

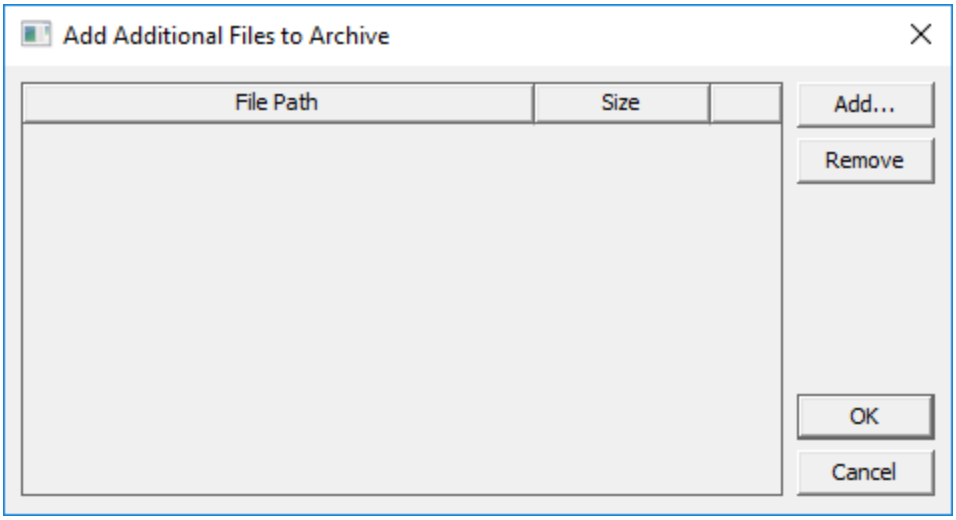
1. Click **File** > **Archive** or select the **Desktop** tab of the ribbon and click **Save Archive**.



The **Archive Options** dialog box opens.



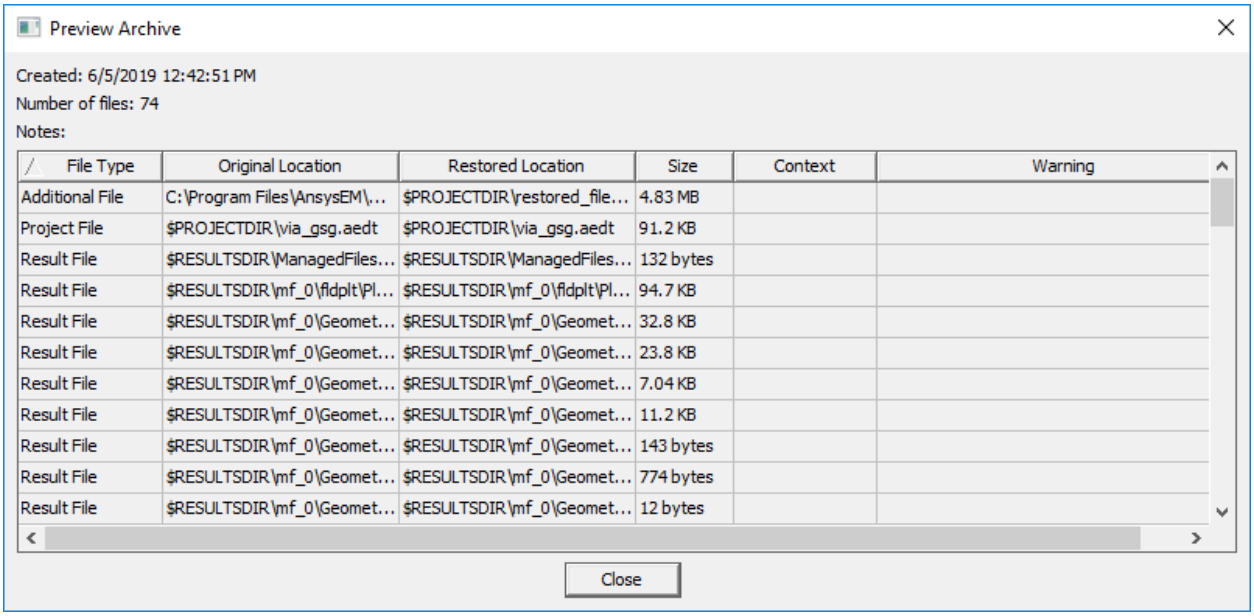
- **Archive Notes** – in this field, you can specify notes that will be visible when previewing the archive. These notes can be viewed from the preview dialog box without restoring the archive.
- **External Files** – selecting this check box causes all external files to be included in the archive. These include 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 causes the entire results directory to be included in the archive. This may greatly increase the size of the archive file.
- **Additional Files** – clicking this opens the **Add Additional Files to Archive** dialog box.



You can click **Add** to browse and locate additional files you want to include in the archive. You can select and the **Remove** any files listed.

You can click **OK** to accept changes or **Cancel** any proposed changes.

2. Select any optional items, and make any desired notes in the text field.
3. When you have made your selections, click **Preview Archive** to view archive contents, file types, the locations of the archive files, and the locations where restoring from archive would place them. Additionally, there may be context or warning information.



To read longer locations, you can drag the column header to expand them.

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.

4. When you are ready to create the archive, **Close** the preview and click **OK**.

   A browsing window appears.

5. Specify a name for your file and select the format you want to use from the **Save as type** drop-down menu.
6. Click **OK** to create the archive.

File Relocation

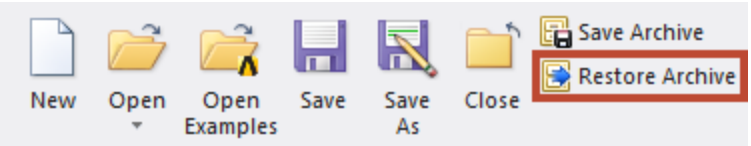
In a project to be archived, external files can be located anywhere on the user's system. One of the goals is for the restored project to be relatively self contained, and to NOT allow the restoring of an archived project to haphazardly write files anywhere on the restoring user's system.

To achieve this, it is sometimes necessary to change the location of files in the archived project so that the external files are 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.

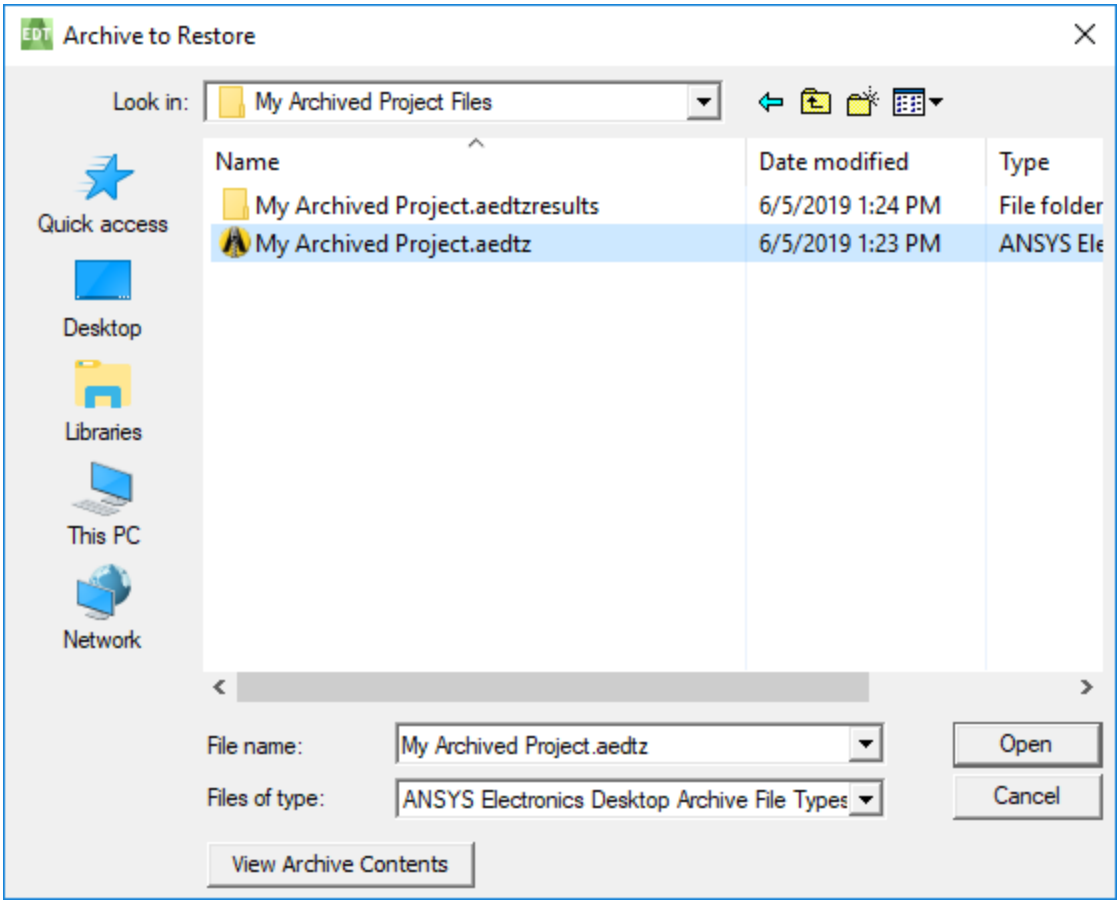
Restoring an Archived Project

To restore an existing archive created with **File** > **Archive...**, use the **File** > **Restore Archive** command.

1. Click **File**> **Restore Archive** or select the **Desktop** tab of the ribbon and click **Restore Archive**.

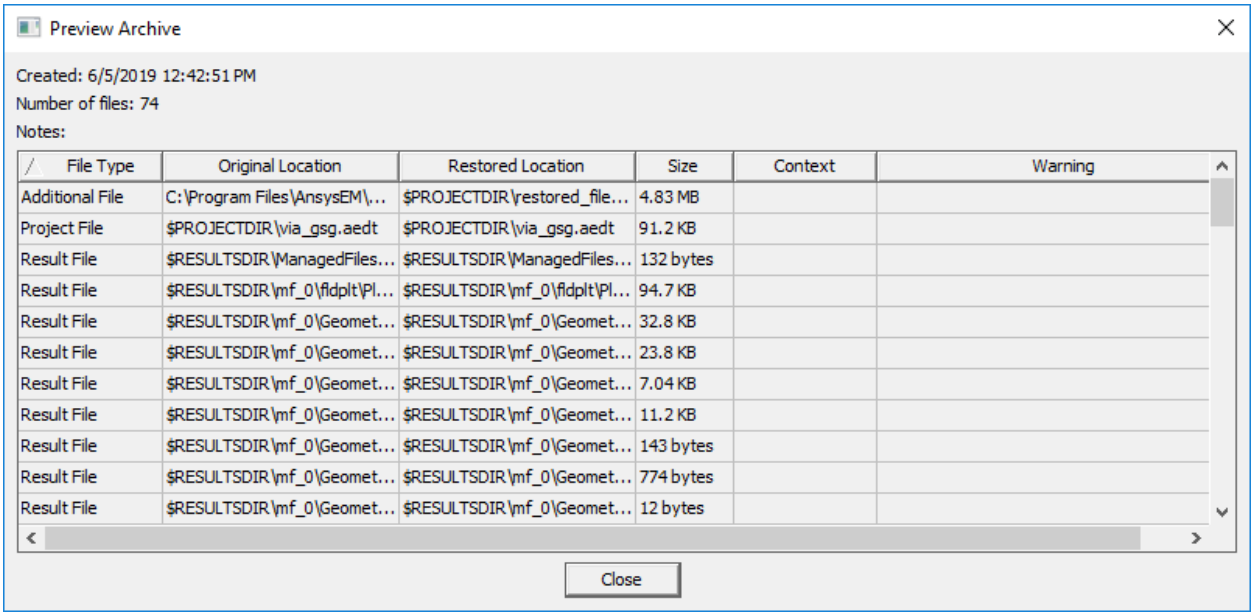


   This displays an **Archive to Restore** browser window that lets you navigate your file system for *.aedtz, legacy *.*<solversuffix>z* or *.zip archive files.

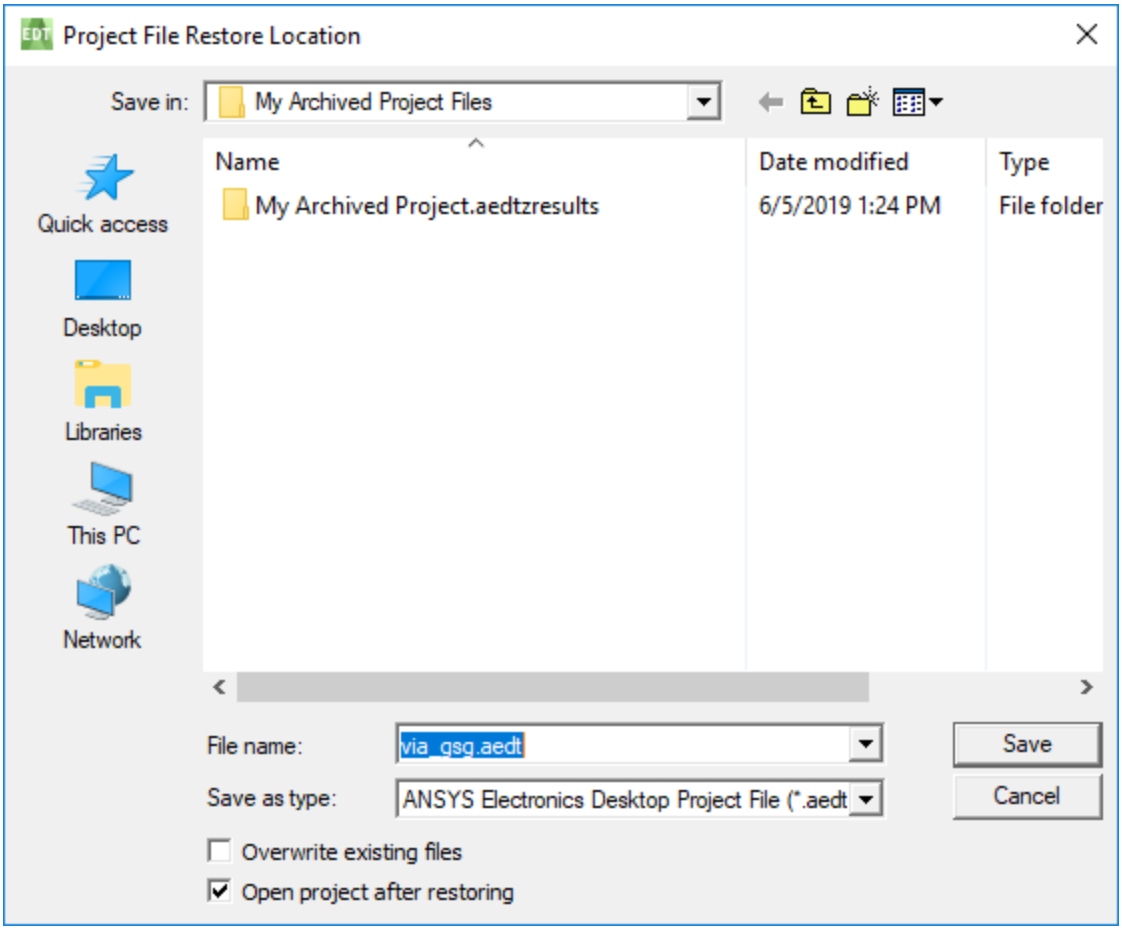


2. If you select a valid archive file, you can click **View Archive Contents** to preview the contents.

   The **Preview Archive** dialog box lists all files in the archive, notes, the original and restored locations, and any 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 **Close** to exit the preview.
4. In the browsing window, click **Open** to change to a **Project File Restore Location** browser.

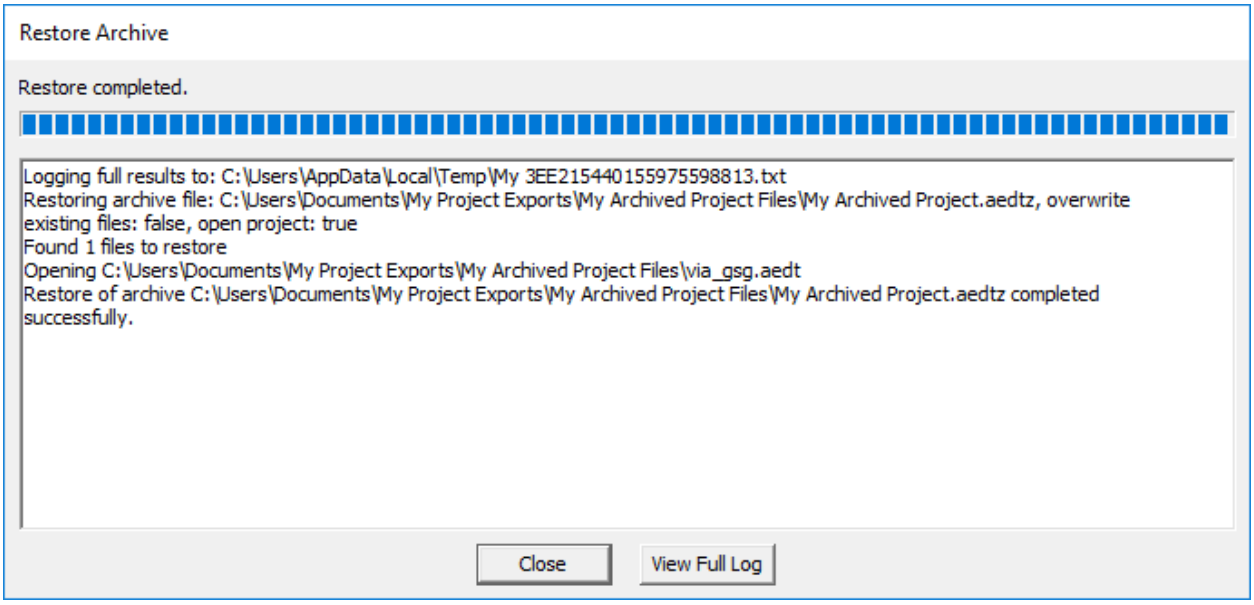


You can edit the file name, and select options:

- **Overwrite existing files** – If this check box is selected, restored files will automatically overwrite existing files during the restore process. If this button is deselected, existing files will not be modified.
- **Open project after restoring** – If this check box is selected, the project will be opened in this instance of the application after all files have been restored.

5. Click **Save** to restore the archived file.

   While restoring an archive, a dialog box displays showing restore results. A progress bar shows the relative progress, and the text window displays important information and warnings.



Electronics Desktop also generates a full log file that 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 open this log file, or use a text editor and open the file at the specified location.

Deleting a Project or Design

To delete a project or design:

1. In the Project Tree, select a project or design.
2. Click **Edit** > **Delete**, or press the **Delete** key.
3. Confirm the warning box to complete the delete operation.

   The project or design 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.

Viewing the Design List

You can view and edit details about a Q3D or 2D Extractor model from the Design List.

To see the Design List:

1. Click [**Q3D Extractor** / **2D Extractor**] > **List**.

   The **Design List** window appears, with the appropriate tabs:

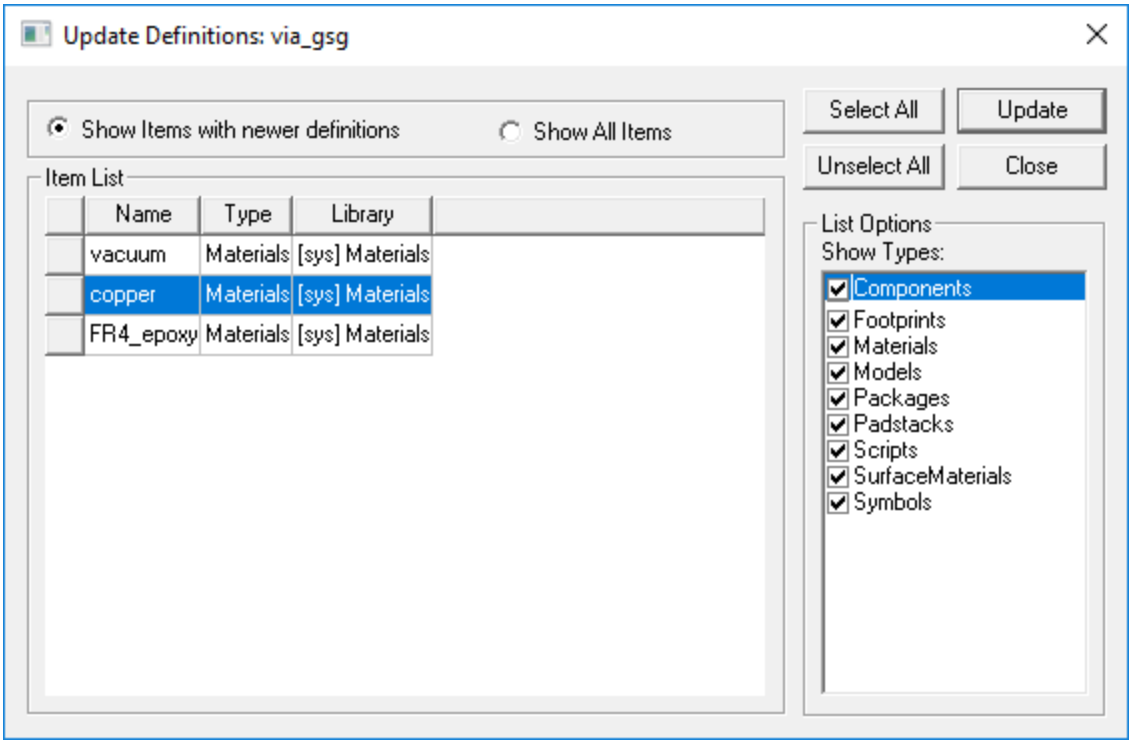
   - **Model** – displays a list of model components.
   - **Boundaries** – displays a list of boundaries.
   - **Conductors** – displays a list of conductors. *2D Extractor only.*
   - **Nets and Terminals** – displays a list of nets and terminals. *Q3D Extractor only*.
   - **Mesh Operations** – displays a list of mesh operations.
   - **Analysis Setup** – displays a list of analysis setups.
2. Select the appropriate tab for the information you want to select.
3. Sort the columns by clicking the column headers.
4. Select an item from the list. You can also select an item by entering its name in the search field and clicking **Select**.
5. If you want to edit an item, select it and click **Properties**.
6. If you want to delete an item, select it and click **Delete**.
7. Click **Done** to exit the **Design List**.

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 either **Show Items with newer definitions** or **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**.
4. Select the item(s) you want to update from the **Item List**, or use **Select All.**
5. Click **Update**.

   A message appears telling you the update was successful.

6. Click **OK** to close the message.

When you are finished updating definitions, click **Close**.

Undoing and Redoing Commands

The **Edit** menu contains **Undo and** Redo commands.

Use the **Undo** command to cancel the last action you performed on the active project or design.

Use the **Redo** command to reperform an action you previously undid.

Both commands are useful for dealing with unintended actions related to project management, model creation, and post-processing.

To use **Undo** or **Redo**:

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

   > **Note:**
   >
   > You cannot undo an analysis that you've performed on a model—that is, the **Q3D Extractor** > **Analyze** command.

2. Click **Edit** > [**Undo/Redo**], or click the **Undo** or **Redo** button (Undo Redo) on the **Desktop** tab.

   Your action is undone or redone.

> **Important:**
>
> When you save a project, Electronics Desktop clears the entire undo/redo history for the project and its designs.

Inserting a Documentation File

You may want to add a documentation file to the project tree.

1. Click **Project** > **Insert Documentation File**.

   This opens a file browser that lets you navigate your file system.

2. Select the documentation file and click **OK**.

   The documentation file is placed in the Project Tree.

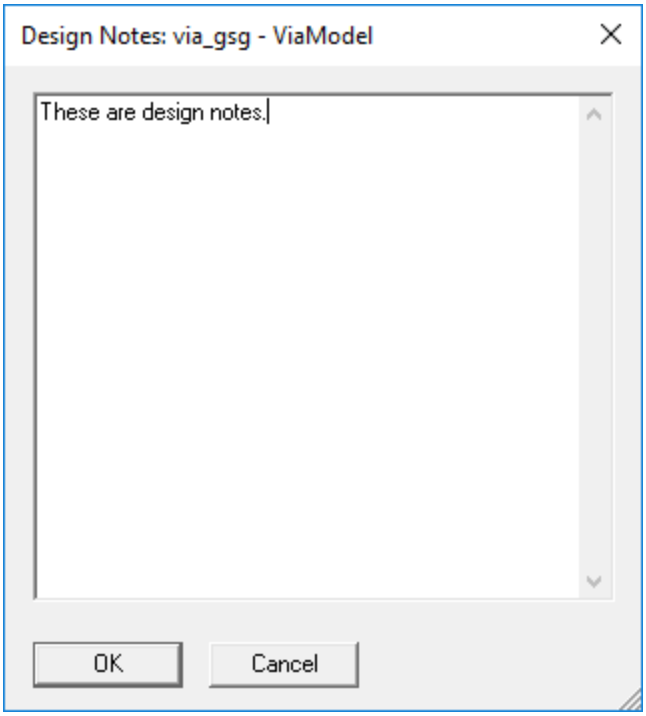
Working with Design Notes

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

To add notes:

1. Click **Q3D Extractor** or **2D Extractor** > **Edit Notes**.

   The **Design Notes** window appears.



2. Click in the window and type your notes.
3. Click **OK** to save the notes with the current project.

To edit notes, use the same window or double-click the **Notes** icon (Notes) in the Project Tree.

To delete the existing notes for a design:

1. Select the **Notes** icon in the Project Tree.
2. Click **Edit** > **Delete**, or right-click and select **Delete**.

   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 File** command.

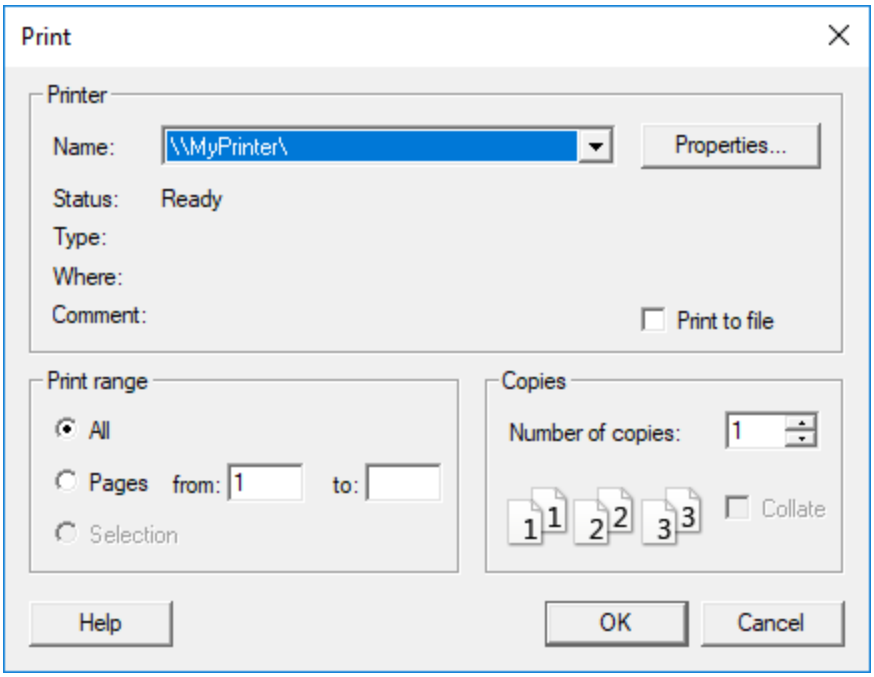
Printing

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

To print the project:

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

   A dialog box similar to the following one appears:



2. Print options will vary based on your printer. Specify your desired options.
3. Click **OK** to print.

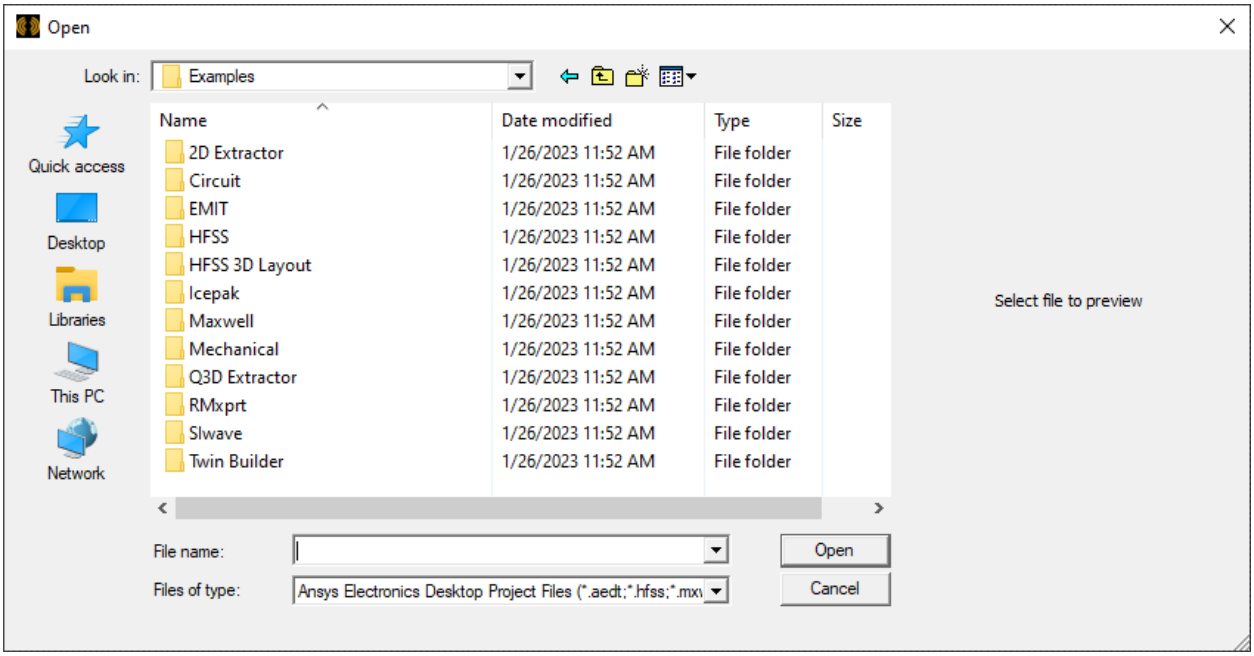
To change additional print settings, click **File** > **Page Setup**.

To preview what your printout will look like, click **File** > **Print Preview**.

> **Important:**
>
> To print from Ansys software on Linux, you must first configure a printer from the MainWin control panel.
>
> For more information, consult documentation for your specific instance of Linux.

Example Projects

Your Ansys Electronics Desktop installation includes an example directory containing projects folders for several kinds of designs.



Example projects are organized by the design type.

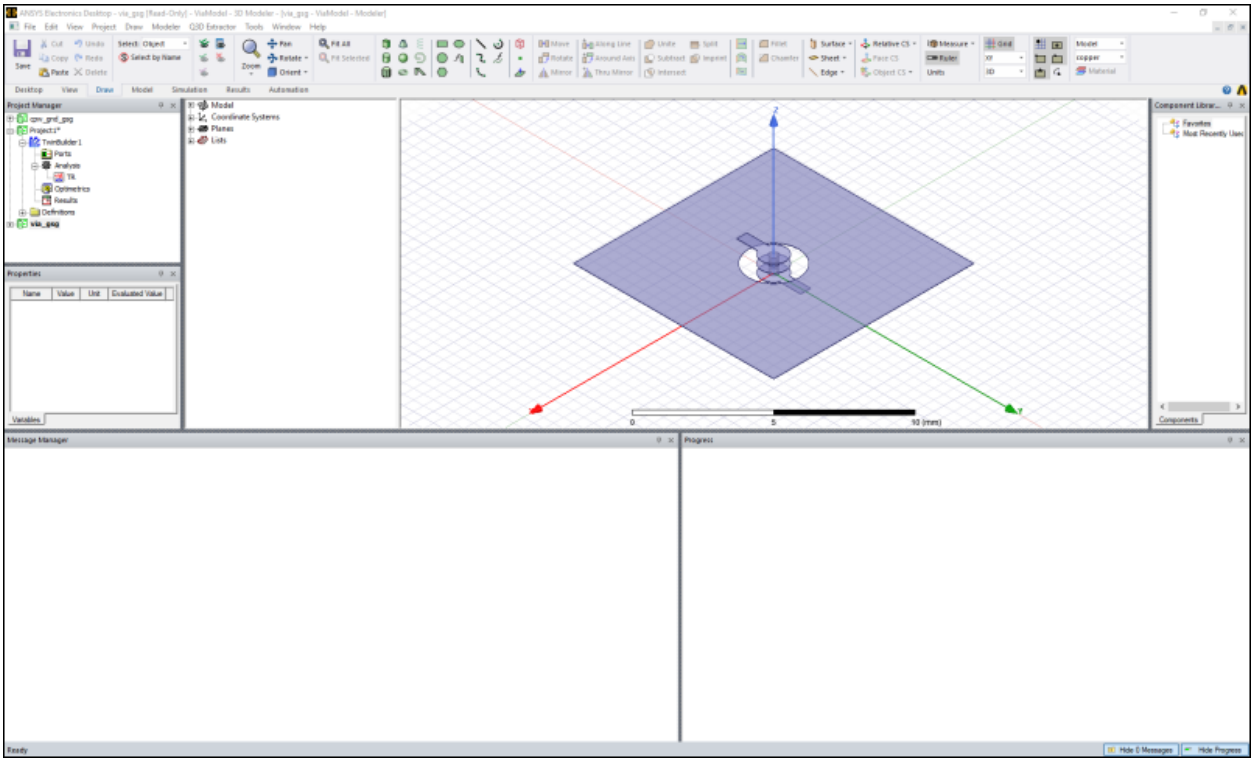
Several of these projects are associated with detailed getting started guides.

Q3D Extractor Example Project

The Examples\Q3D Extractor directory contains a ViaModel project (via_gsg.aedt), described in its associated Getting Started Guide.

To open the example, click **File** > **Open Examples**, or click **Open Examples** on the **Desktop** tab.

The Getting Started Guide is located in the "...\ANSYS Inc\v251\AnsysEM\Help\Q3DExtractor\GSG" directory.

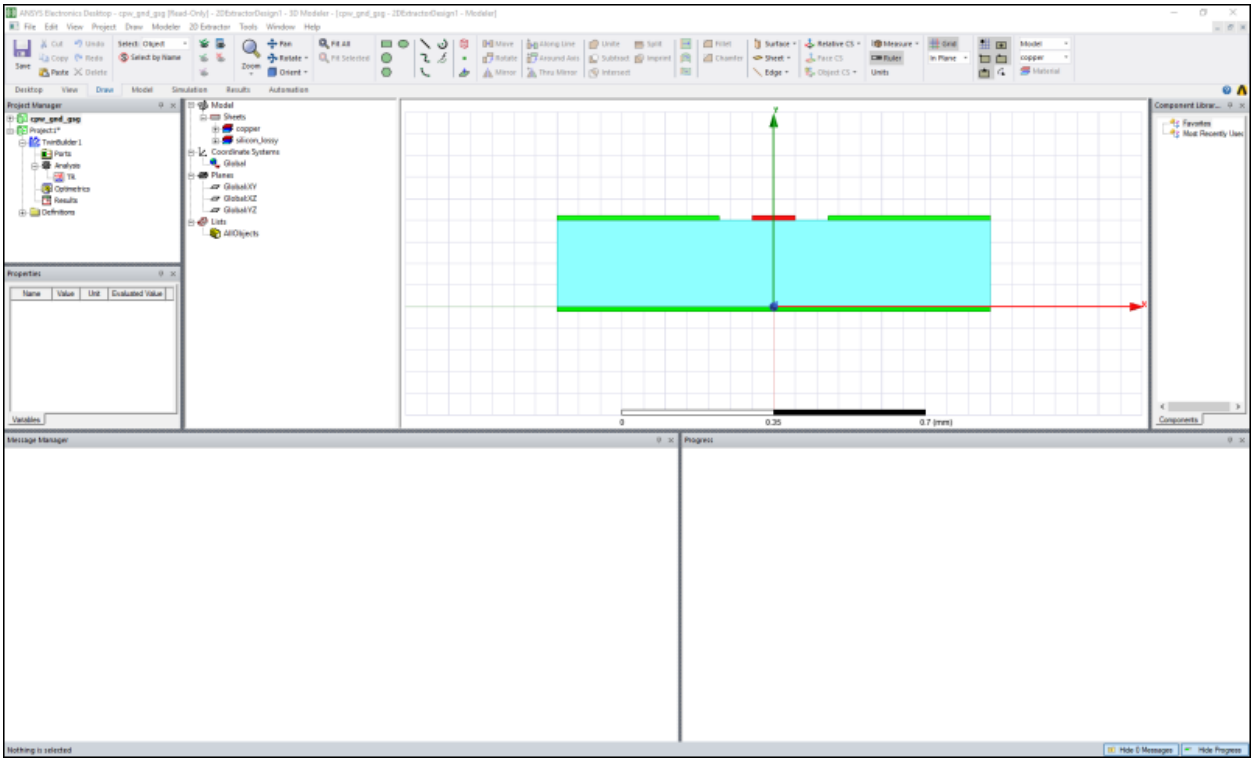


2D Extractor Example Project

The Examples\2D Extractor directory contains the cpw_gnd_gsg.aedt project, described in its associated Getting Started Guide.

To open the example, click **File** > **Open Examples**, or click **Open Examples** on the **Desktop** tab.

The Getting Started Guide is located in the "...\ANSYS Inc\v251\AnsysEM\Help\Q3DExtractor\GSG" directory.



Working with 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 piecewise linear Intrinsic Functions. 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.

Project-level datasets are typically used for defining material properties at the project level (applicable to all designs in the project).

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 also 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.

Datasets are provided for Q3D Extractor designs.

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** characteristic data; and to launch the **SheetScan** tool to extract data from graphics such as data sheets.

To access the **Datasets** window:

- For project-level datasets, including 3D datasets, click **Project** > **Datasets**.
- For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

Adding Datasets

The following procedure describes how to add a dataset manually.

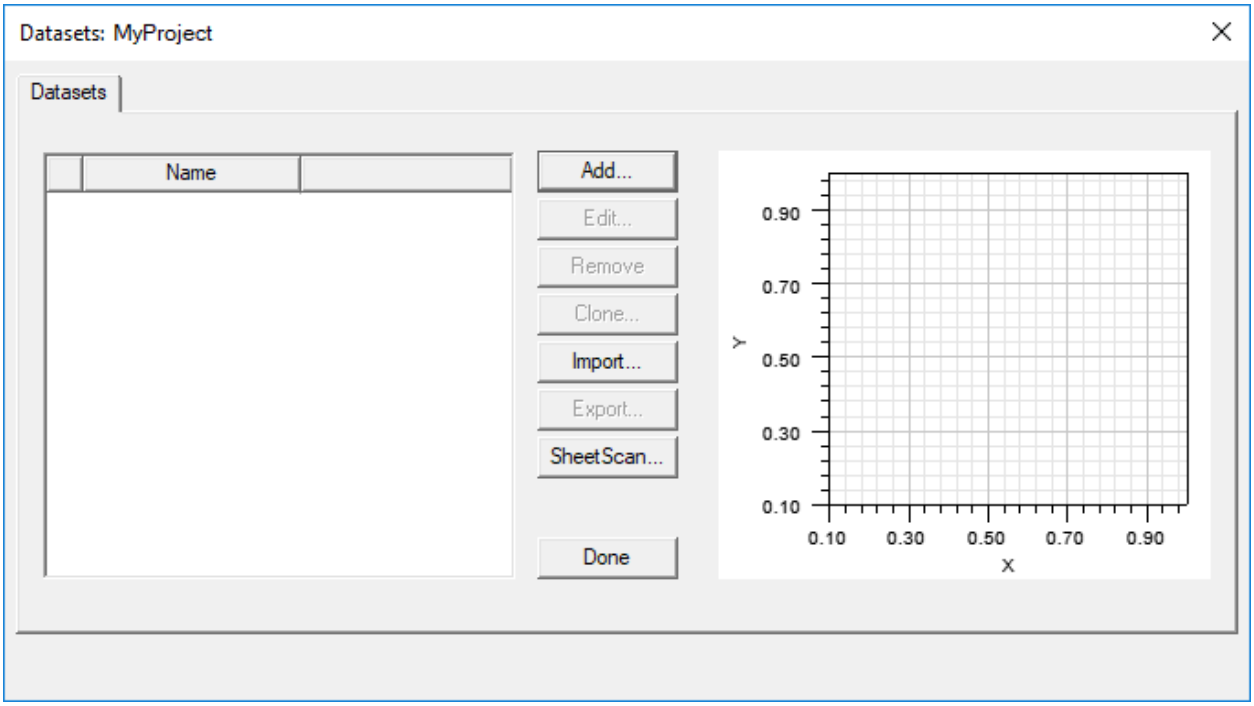
You can also add datasets:

- By importing data from an external tab delimited file using the **Import...** button.
- Using the Sheetscan tool.

To add a dataset manually:

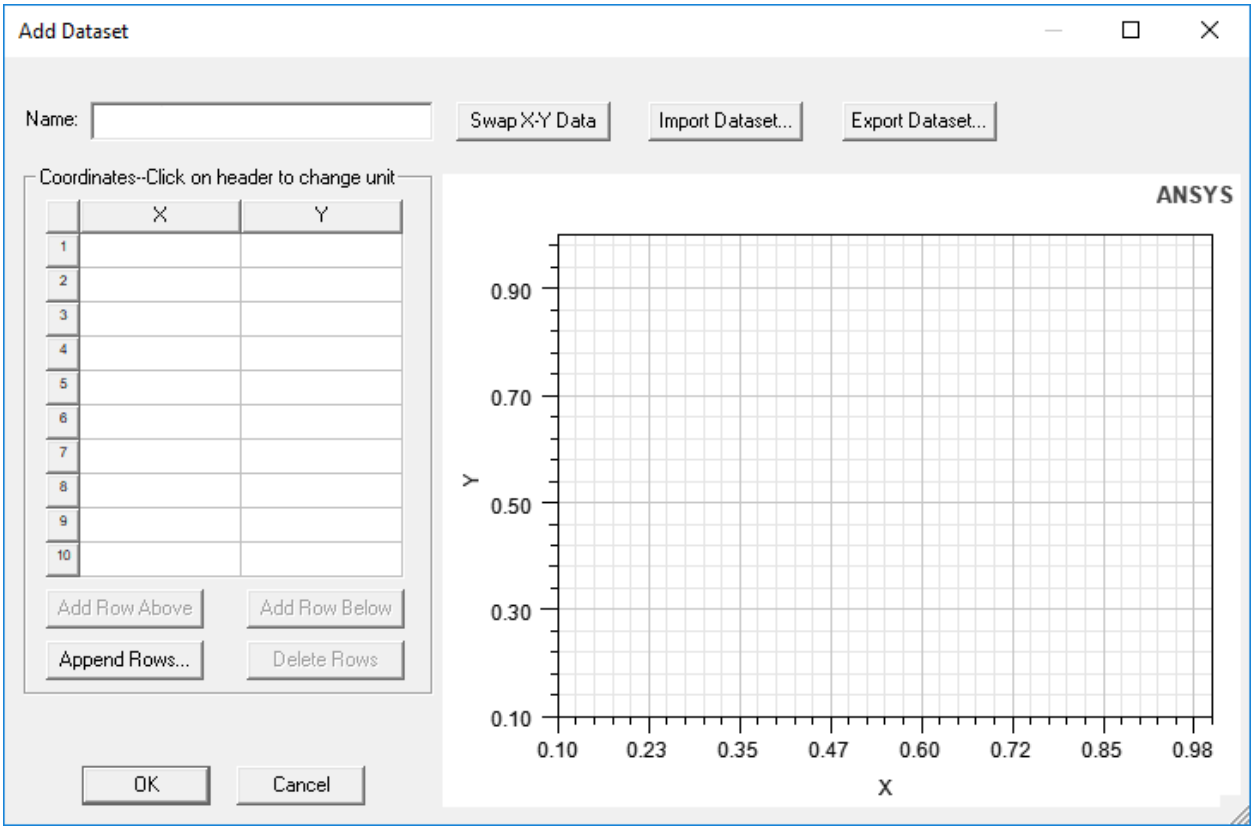
1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets** from the menu bar, or click  **Datasets** on the **Desktop** ribbon tab.
   - For design-level datasets, click **Q3D Extractor** > **Design Datasets** from the menu bar.

   The **Datasets** window appears, listing any project or design-level datasets.



2. Click **Add**.

   The **Add Dataset** window appears.



The left side of the **Add Dataset** window contains fields for the dataset name and a table for X- and Y- coordinates. The right side 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 box that lets you specify a number of rows to add to the table.
- **Delete Row** – deletes the selected row or rows.

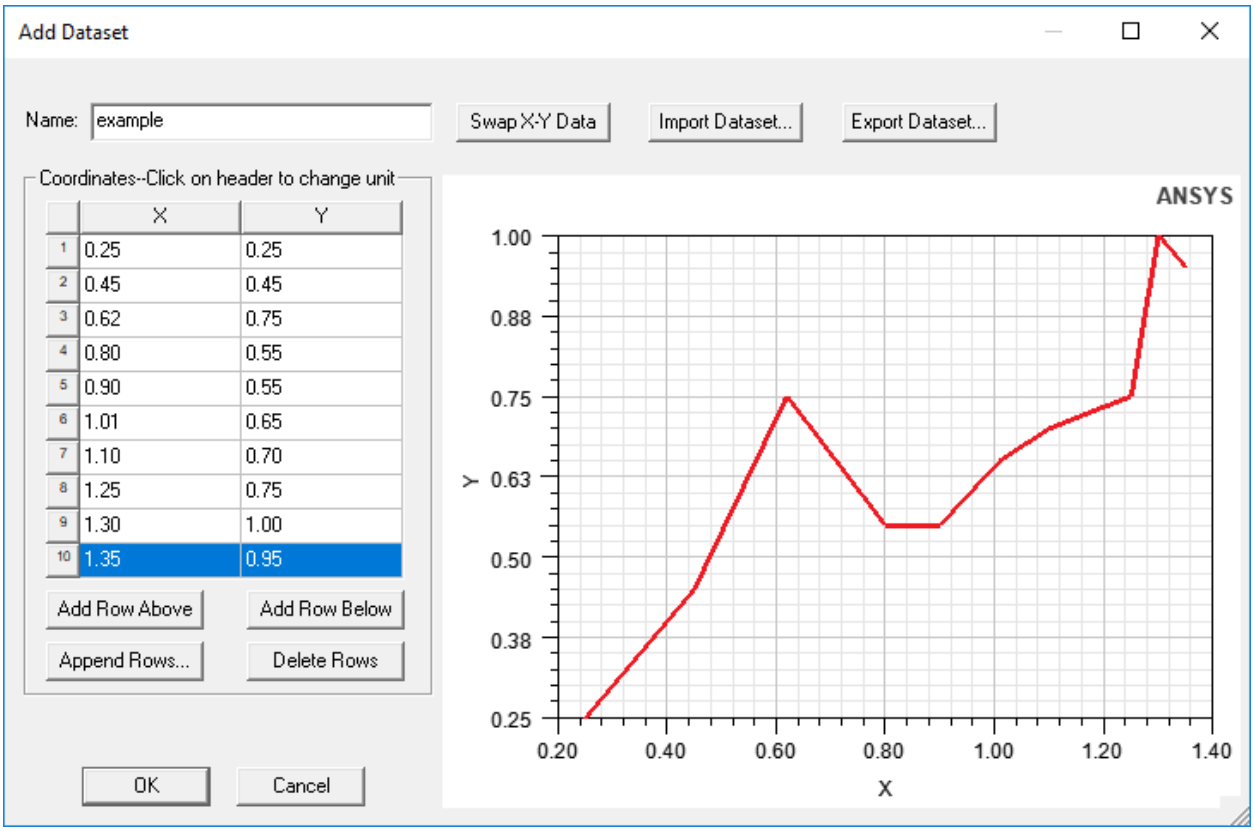
3. Enter a **Name** for the dataset.

> **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 X/Y coordinates, either by manually typing or by clicking **Import Dataset**.

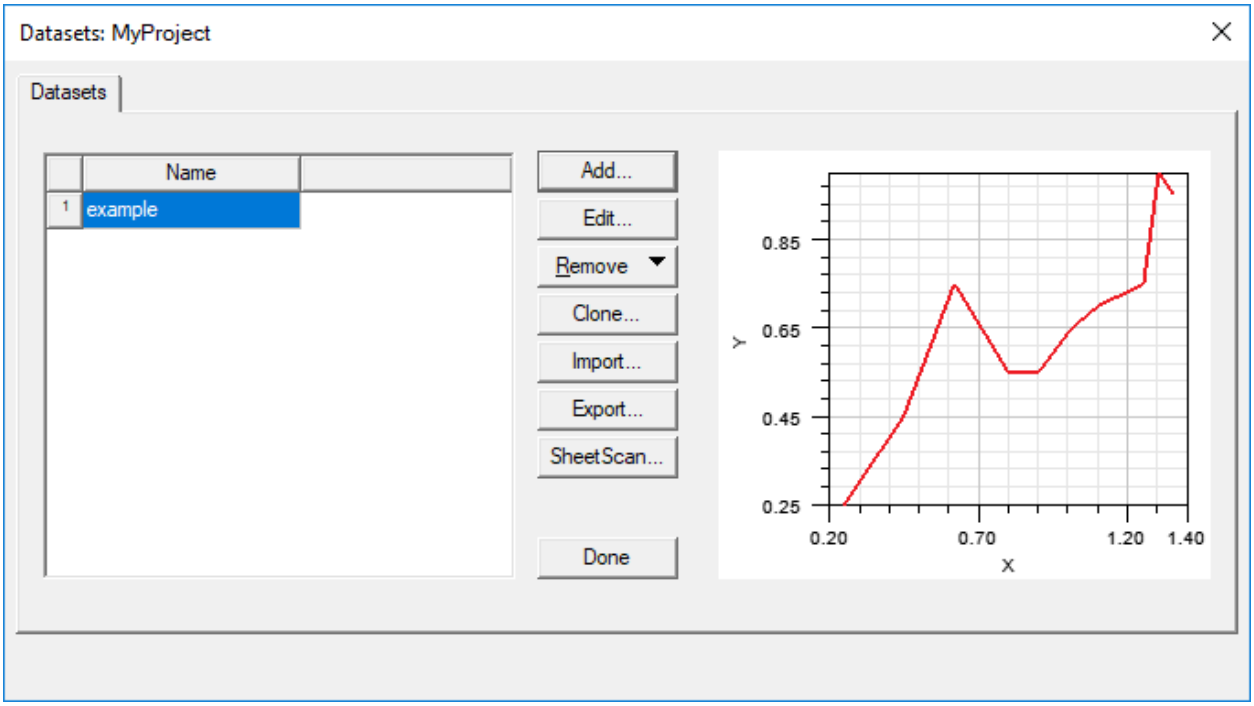
   If desired, click the **X** and **Y** column headers to change the unit.

   The display updates to show your coordinates.



5. When you are finished, click **OK** to exit the **Add Datasets** window.

   The new dataset appears in the **Datasets** list.



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.

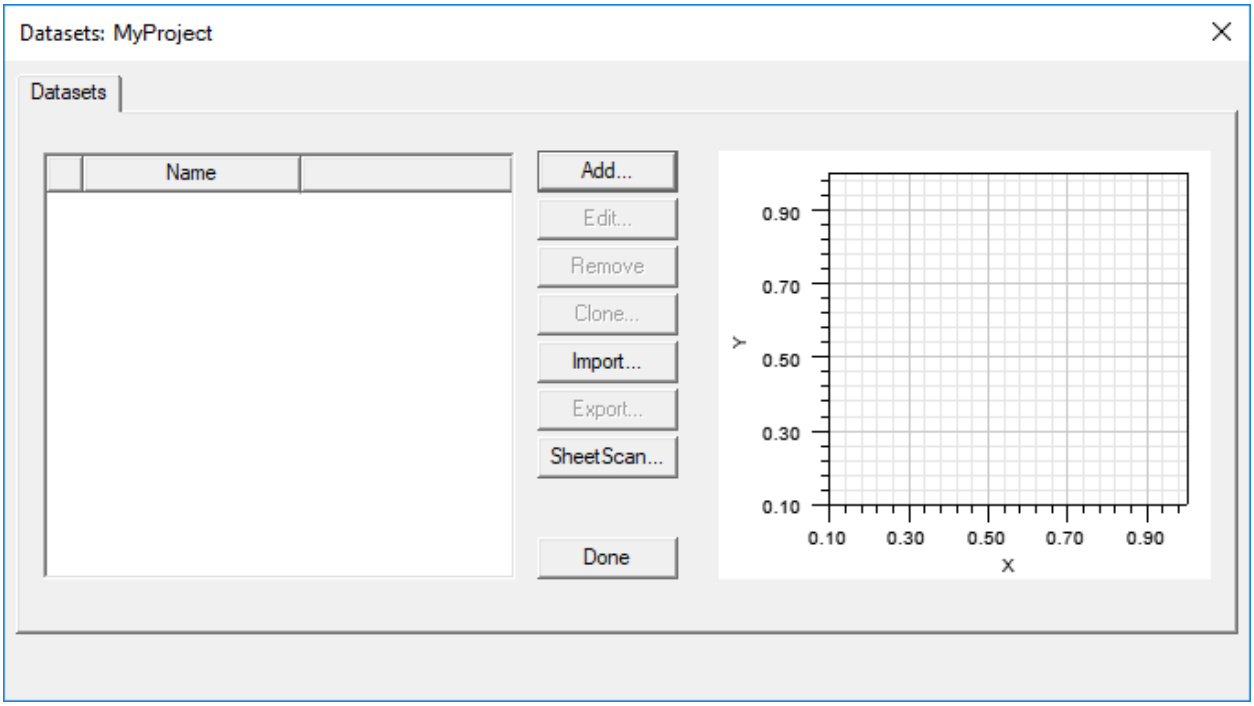
6. Click **Done** to exit the **Datasets** window.

Importing Datasets

To import a dataset from a file:

1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

   The **Datasets** window appears.



2. Click **Import**.
3. In the file browser that appears, use the drop-down menu to select the file format of the file you wish to import.

> **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

   a. If you select an **.mdx**, **.mda**, **.out**, **.cfg**, or **.tab** file, the data is imported immediately into **Datasets** window.

b. If you select an **.xls** or **.xlsx** file containing multiple sheets, a **Table Properties** dialog box appears, where you can choose the desired sheet from a drop-down menu. Otherwise, selecting an .xls or .xlsx file imports the data immediately.

> **Note:**
>
> Microsoft Excel is required to import .xls or .xlsx files.
>
> 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.

c. Selecting a .**txt** or .**csv** file opens an **Import** dialog box where you can specify 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.

4. When satisfied with the import settings, click **OK** to import the data.

   Imported datasets take the name of the imported file. If a dataset of the same name already exists, a number will be appended to the end of the new dataset (for example, a duplicate 'dataset' becomes 'dataset1').

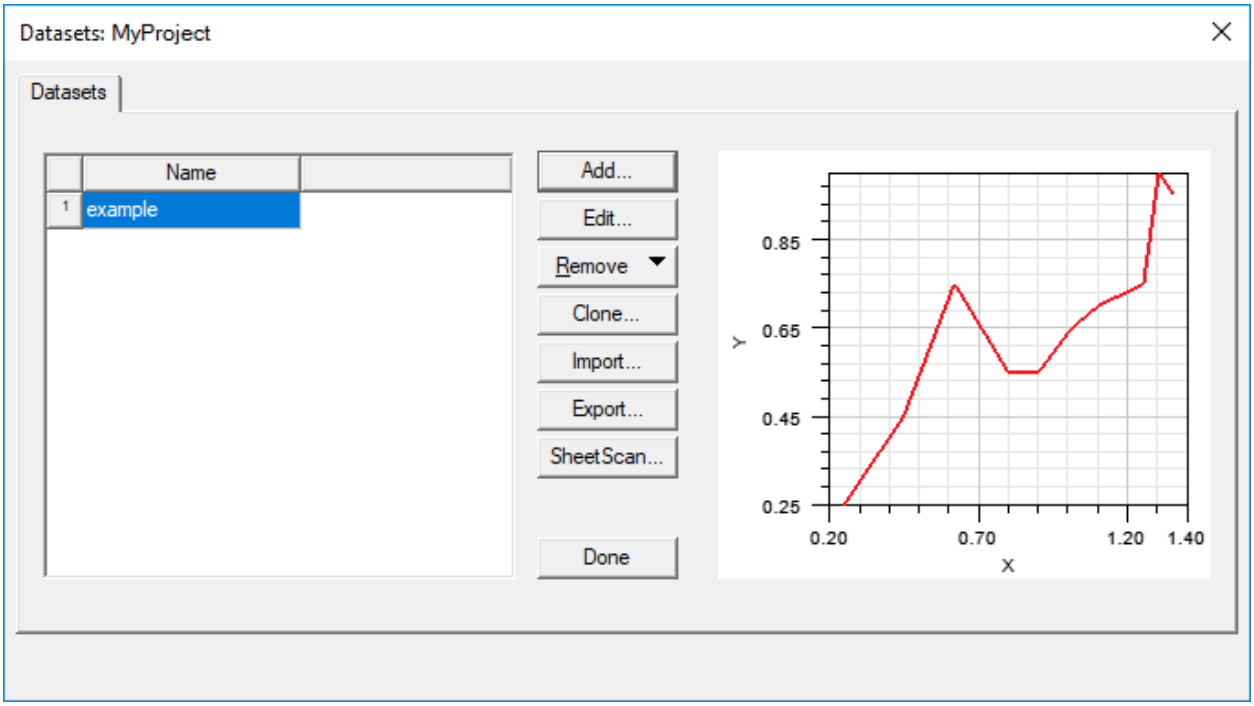
   Once the data has been imported, you can edit, remove, clone, or export the dataset. You can also change plot display properties.

Editing Datasets

To modify an existing dataset:

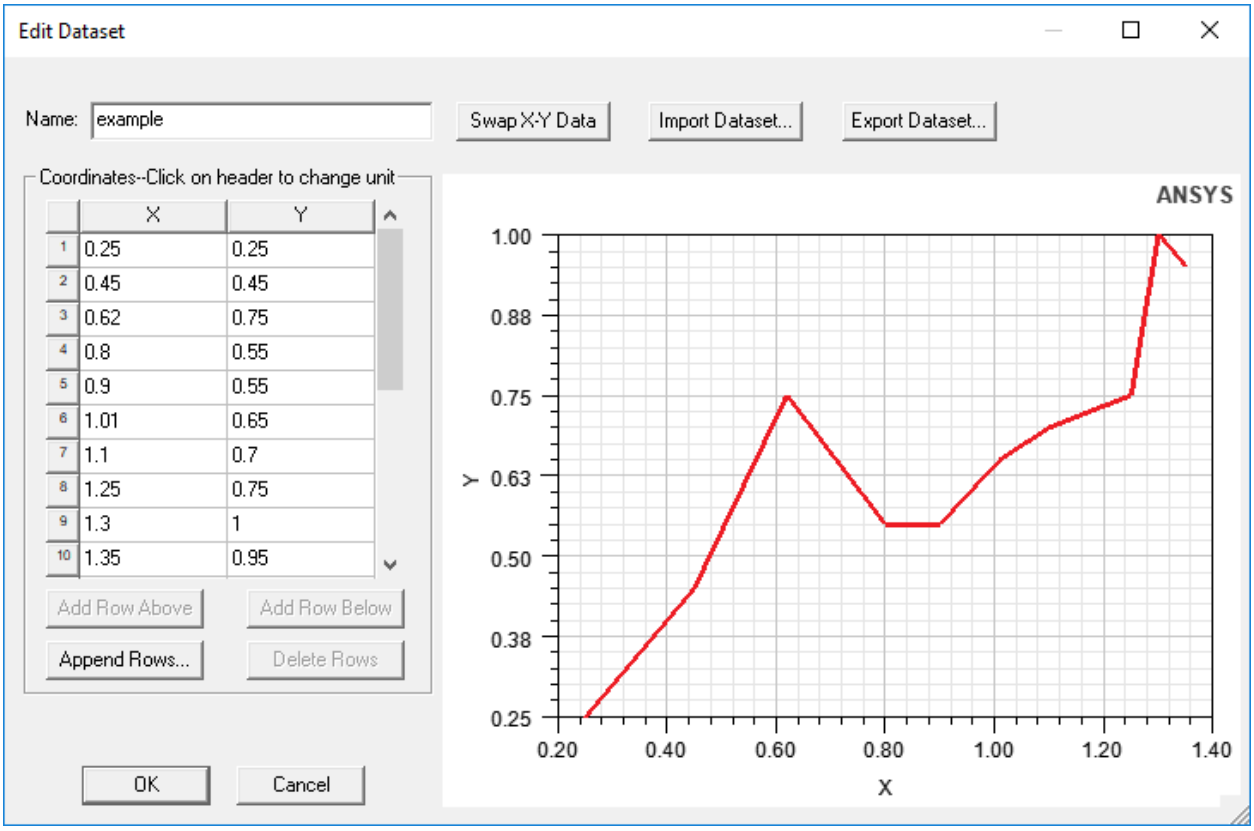
1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

   The **Datasets** window appears, listing any project or design-level datasets.



2. Click **Edit**.

   The **Edit Dataset** window appears.



3. Change the **Name** and **X/Y** coordinates as desired.
4. When you are finished, click **OK** to exit the **Edit Dataset** window.
5. Click **Done** to exit the **Datasets** window.

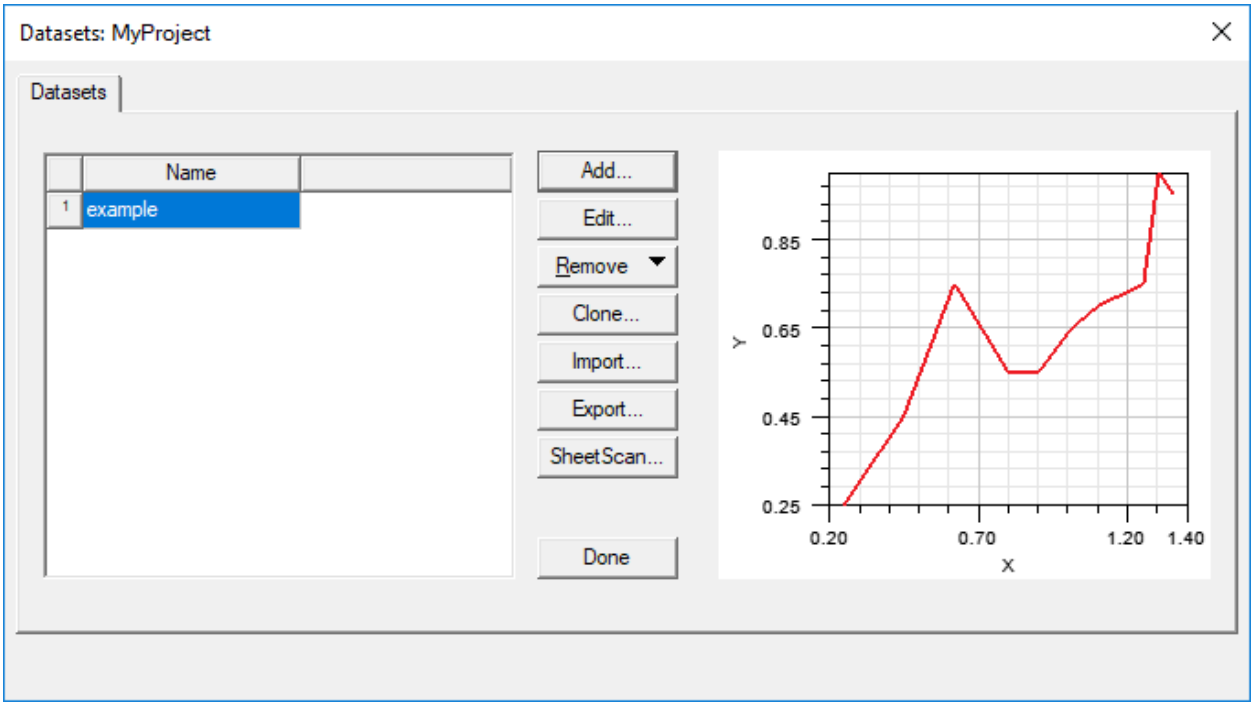
Cloning Datasets

Cloning a dataset creates an identical copy of an existing dataset. The clone can then be modified as necessary.

To clone a dataset:

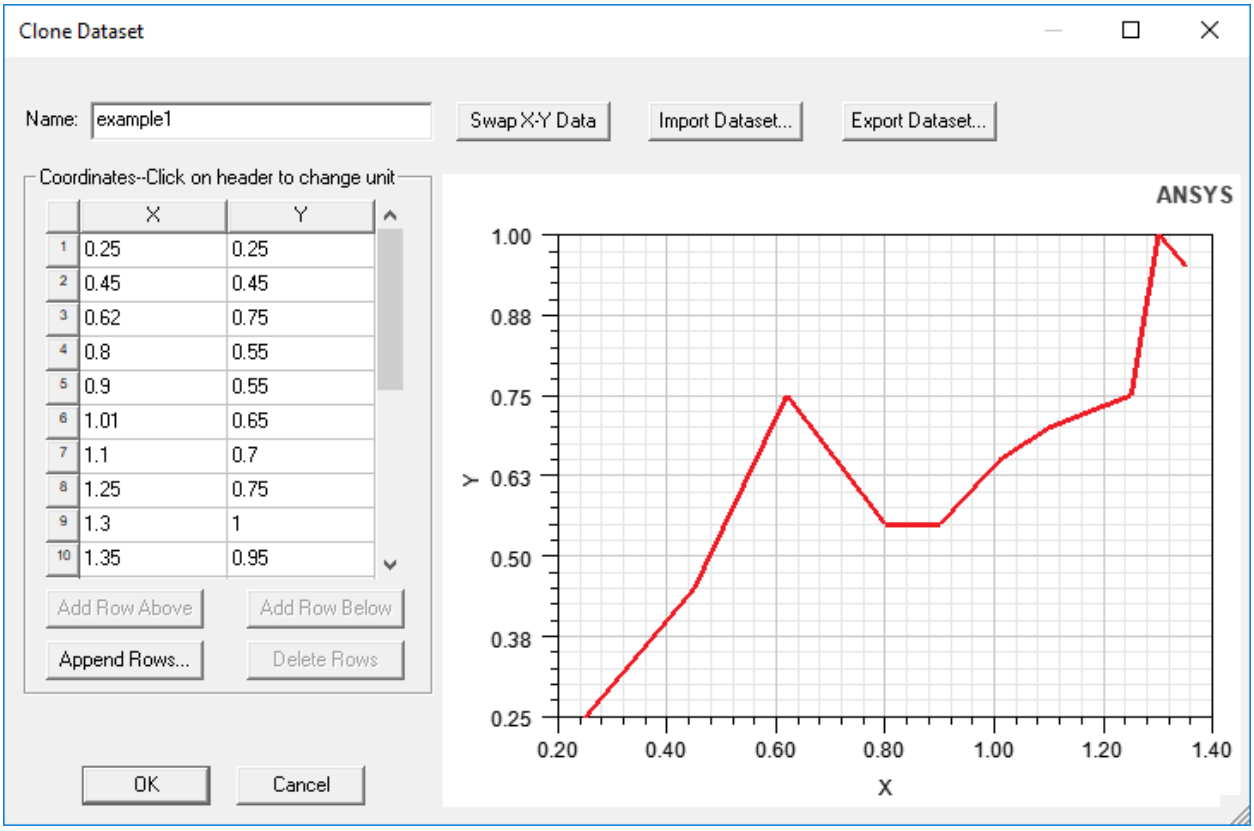
1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

   The **Datasets** window appears, listing any project or design-level datasets.



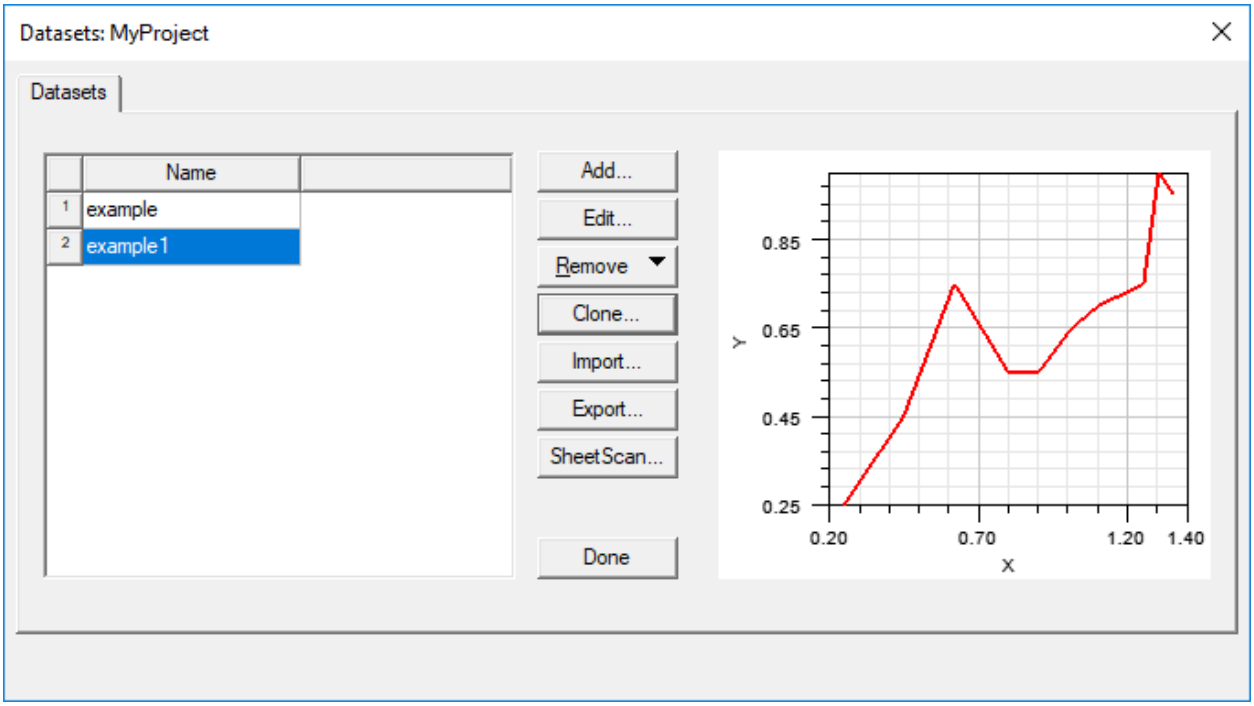
2. Click **Clone**.

   The **Clone Dataset** window appears. The default name is the original dataset name with a number appended to the end.



3. Change the **Name** and **X/Y** coordinates as desired.
4. When you are finished, click **OK** to exit the **Clone Dataset** window.

   The cloned dataset appears in the list.



5. Click **Done** to exit the **Datasets** window.

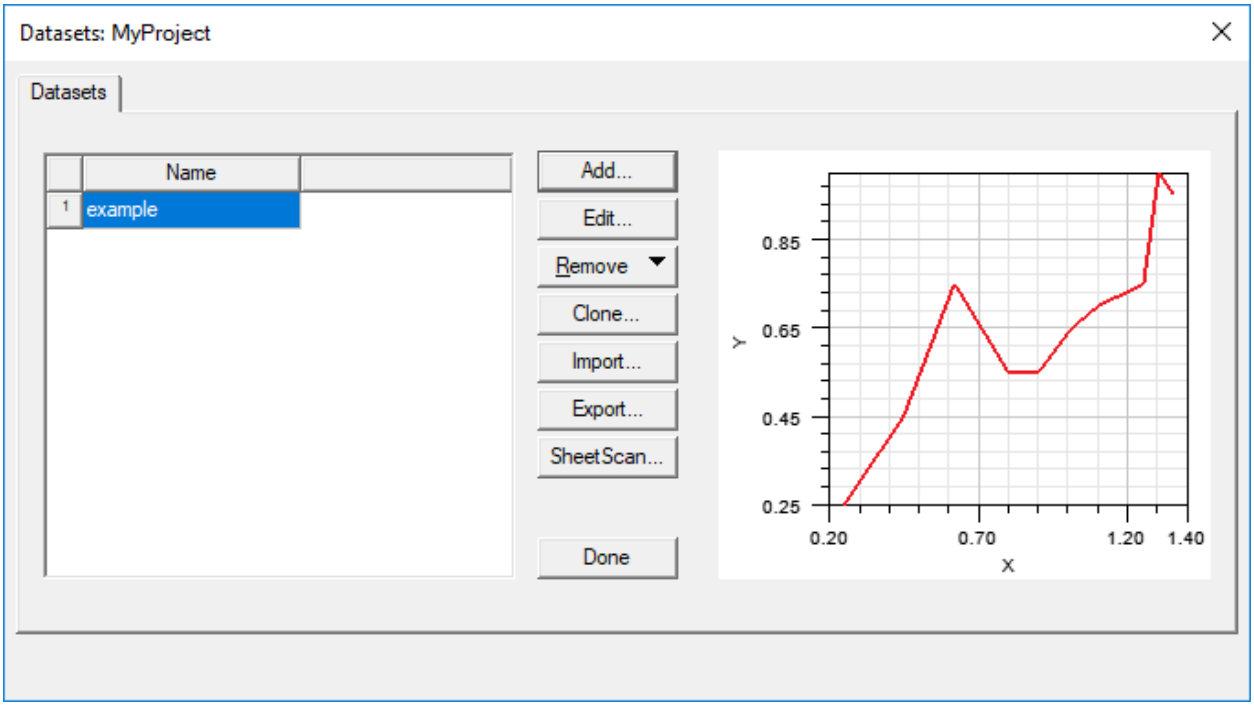
Exporting Datasets

You can export datasets in **.tab** format.

To export a dataset:

1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

   The **Datasets** window appears, listing any existing datasets.



2. Select a dataset and click **Export**.
3. Browse to the location where you would like to save the *.tab file.
4. Enter a name for the file.
5. Click **Save**.

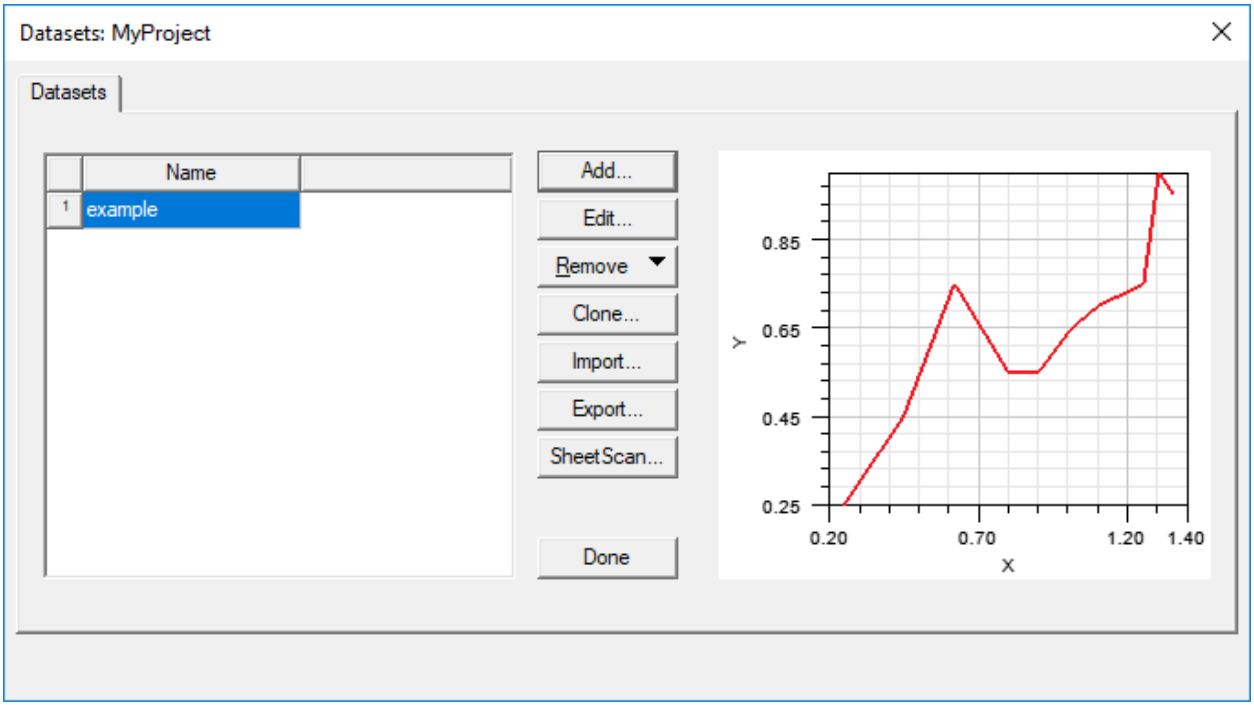
   The dataset file is saved to the specified location for later import.

Removing Datasets

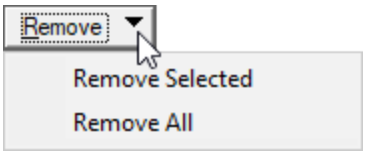
To remove a previously added dataset:

1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

   The **Datasets** window appears, listing any existing datasets.



2. Select a dataset and click **Remove**.
3. Select either **Remove Selected** to remove the selected dataset or **Remove All** to remove all project or design-level datasets.



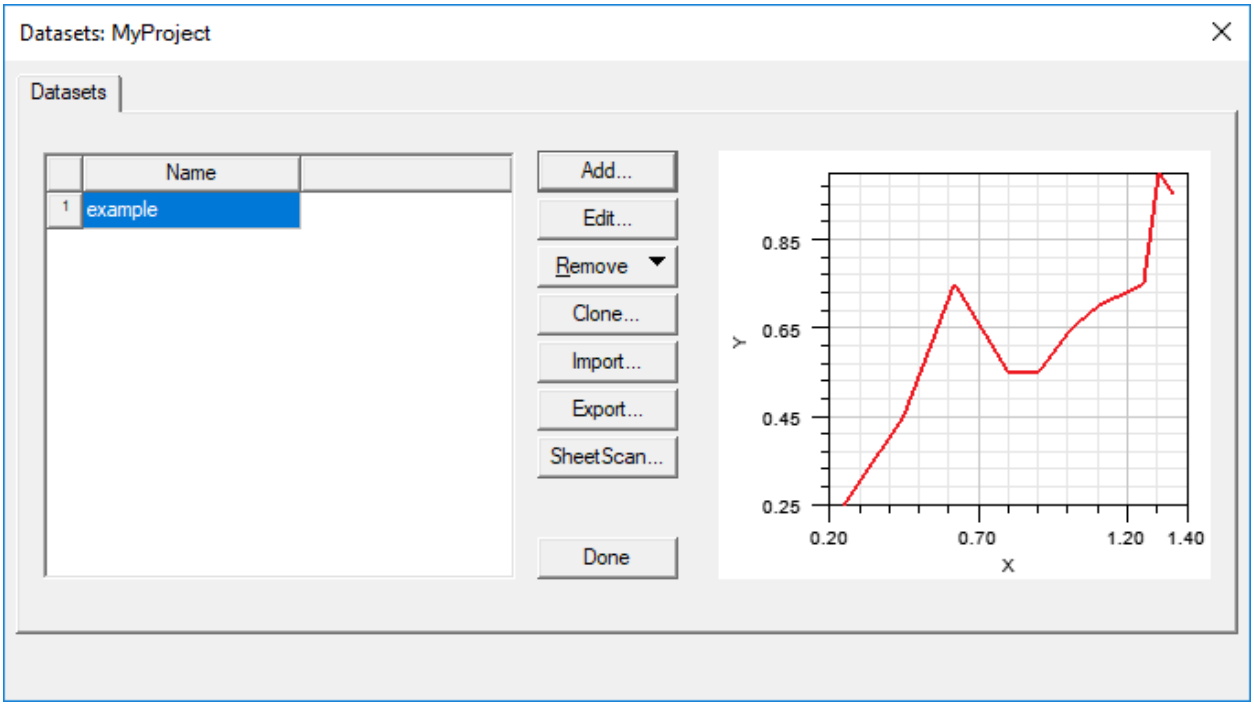
The selected dataset(s) are deleted from the list.

> **Note:**
>
> The **Remove All** option does not work across dataset types. That is, selecting **Remove All** from the **Project Datasets** window does not remove design-level datasets, and vice versa.

4. Click **Done** to exit the **Datasets** window.

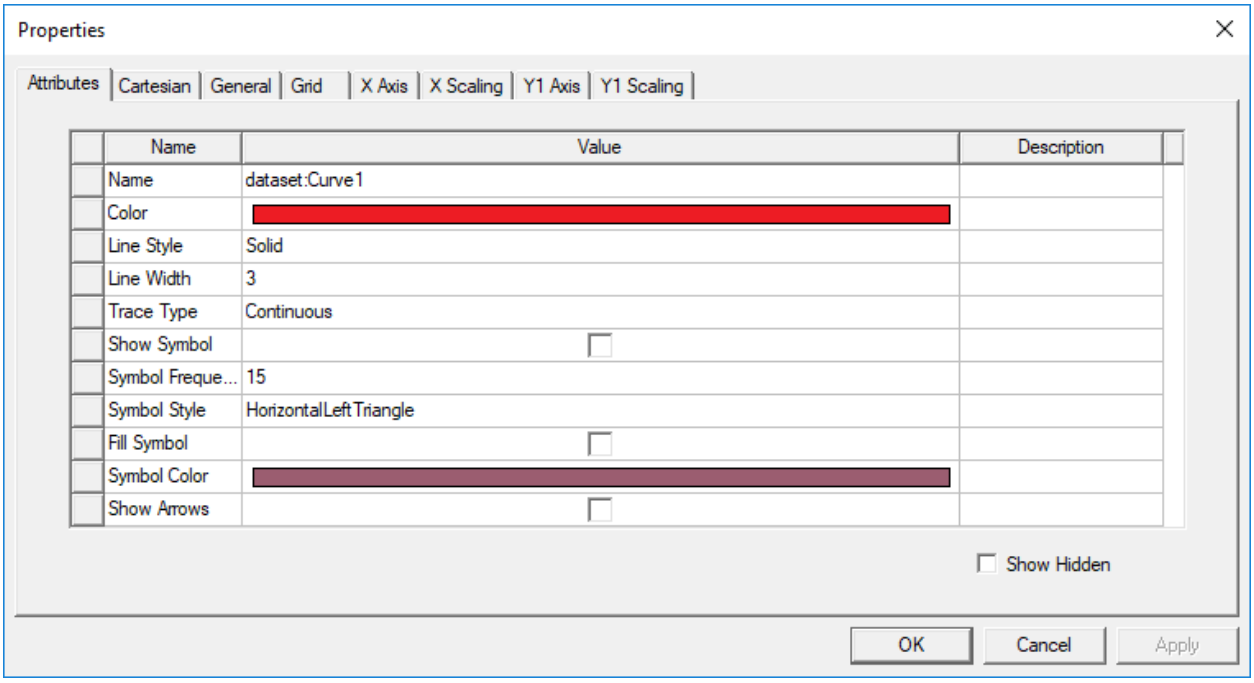
Changing Dataset Plot Properties

From the **Datasets** window, you can change the appearance of a dataset's plot.



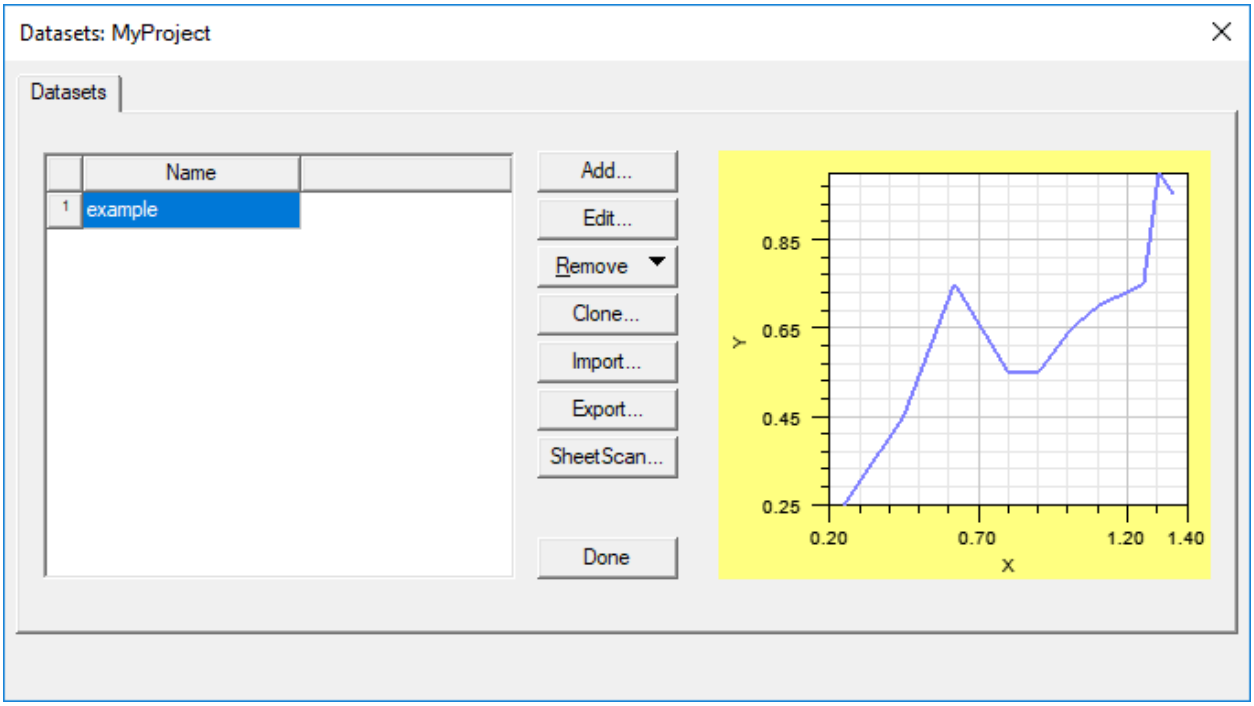
To change a plot's appearance:

1. Double-click a plot element (for example, a line, axis, or title) to open that element's **Properties** window.



2. Set the colors, fonts, and styles as applicable and as desired.
3. Click **Apply**.
4. Click **OK** to exit the **Properties** window.

   The plot now displays as specified.



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.

Access SheetScan from the **Datasets** window:

- For project-level datasets, click **Project** > **Datasets**.
- For design-level datasets, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.

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 Ansys Electronics Desktop format, and the extracted data exported to an Ansys Electronics Desktop 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.

SheetScan Menus and Settings

In the SheetScan window, various menus and toolbars allow you to access SheetScan functionality.

The top menu bar contains the following submenus:

- **File** – allows you to open, close, save, and print files.
- **Edit** – provides access to cut, copy, and paste functions.
- **View** – allows you to select which toolbars appear in the SheetScan window (see below).
- **Options** – provides access to document settings.
- **Picture** – provides options for loading a picture or HTML file into SheetScan, and for deleting a picture.

> **Note:**
>
> The **Delete** action cannot be undone. If you delete a picture from the SheetScan editor, you must reload it from the source file or Internet web page.

- **Coordinate System** – provides options for creating and editing a coordinate system.
- **Curve** – provides options that allow you to define a curve.
- **Window** – provides options for rearranging the currently open SheetScan windows (for example, overlapping and tiling).
- **Help** – accesses the Electronics Desktop online help.

SheetScan View Options

You can toggle the following toolbars via **View** > **Toolbar**:

- **Standard** – provides access to basic functions such as file Open and Save, Cut, Copy, Paste, Print, and Help.

- **Curve** – contains a drop-down menu allowing you to select a curve and tools for working with curve values.

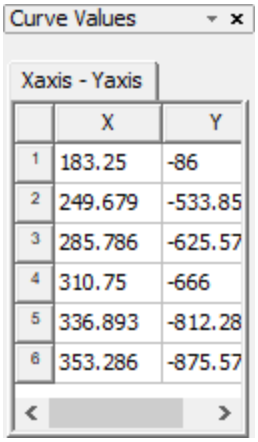
- **Zoom** – provides tools for scaling the current view.

From the **View** menu, you can also toggle the following:

- **Status Bar** – displays, at the bottom of the screen, the current cursor coordinates.

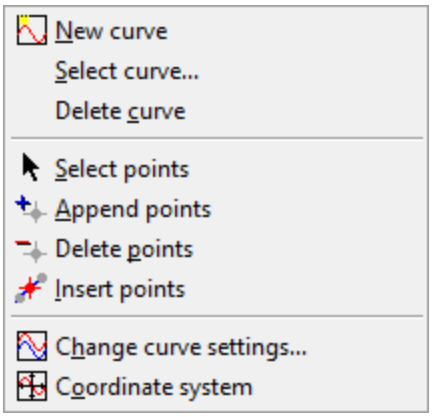
- **Curve Values** – dockable window that displays the data points used while creating a characteristic curve.

- **Grid** – displays the coordinate grid you created over the sheet.

SheetScan Right-click Menu

Once you have loaded a sheet, you can right-click in the SheetScan window to access curve and coordinate system options:



SheetScan Settings

Default settings can be changed from **Options** > **Settings**.

The **Settings** dialog box contains three tabs:

- **Document** – allows you to set the width and height of the sheet created when a picture is imported. You can either enter the dimensions manually, or allow SheetScan to adapt the dimensions to the picture being loaded.
- **Axis** – allows you to set the name, units, scaling, and offset for the X and Y axis. You can also select **Monotonicity in X** to prevent adding consecutive data points whose X-values are not increasing.
- **Representation** – allows you to choose whether to connect points on the characteristic curve and to choose the connecting line color. You can also decide whether to display markers and choose marker colors.

> **Note:**
>
> You can also override the default settings on the **Axis** and **Representation** tabs for individual curves (See: 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: directly, or by using the HTML Viewer.

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 HTML Viewer

1. Choose **Picture** > **Internet** to open the HTML Viewer.
2. Browse the Internet for the desired datasheet.
3. Resize the HTML Viewer 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 into the SheetScan editor window.

> **Note:**
>
> To hide the datasheet picture, select **View** > **Picture**.

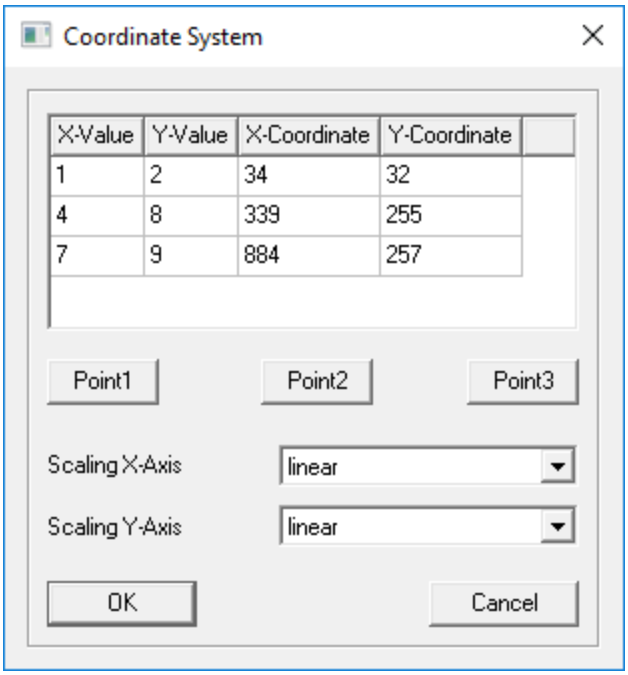
After loading a datasheet picture into the SheetScan editor, the next step is defining a coordinate system for the imported picture.

Defining a SheetScan Coordinate System

You can define a coordinate system 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.



   You can enter points manually or select them using a crosshair.

2. To activate the crosshair cursor, click **Point1**.

   The cursor transforms into a crosshair.

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 the above steps for **Point2** and **Point3**.

7. Click **OK**.

   The grid is placed over the graphic.

   Note:

   - You can edit the grid after placement by selecting **Coordinate System** > **Properties**, by clicking the coordinate system icon on the Curve toolbar ( ), or by right-clicking in the SheetScan editing window and selecting **Coordinate System**.
   - You can hide the grid by selecting **View** > **Grid**.

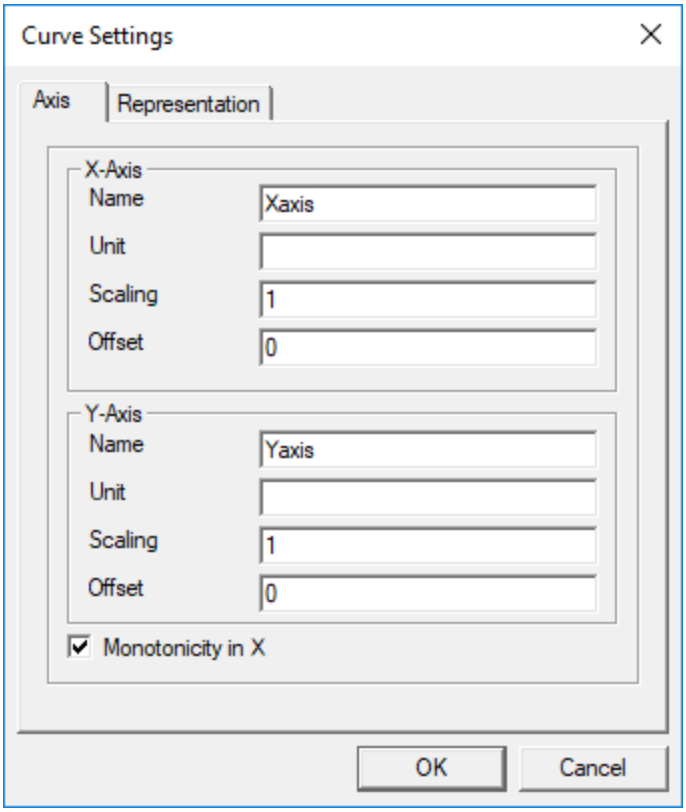
8. You can now define a Characteristic Curve.

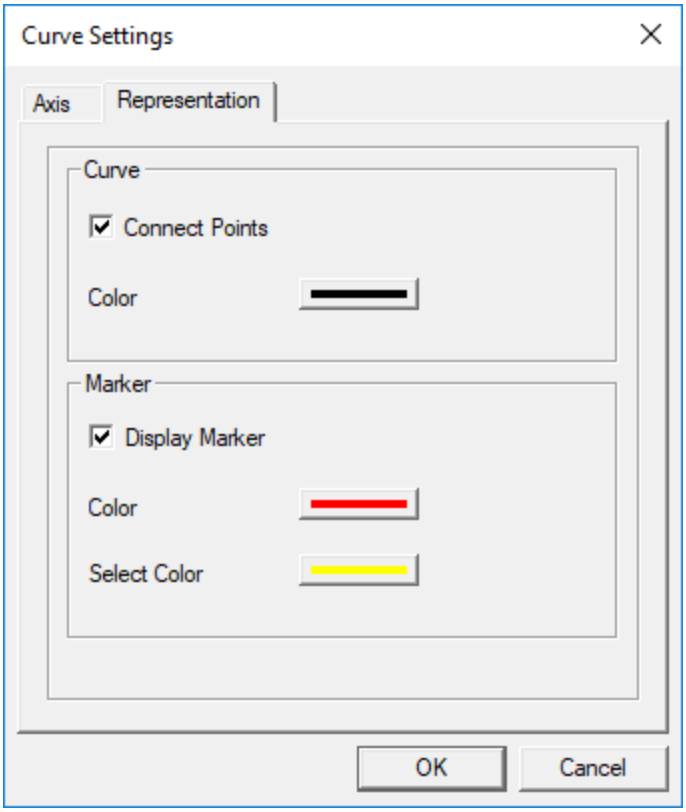
Defining a Characteristic Curve in SheetScan

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, containing the **Axis** and **Representation** tabs.





2. Use the fields on the **Axis** tab to define the curve's properties. Use the buttons on the **Representation** tab to change the curve's appearance.
3. Click **OK**.

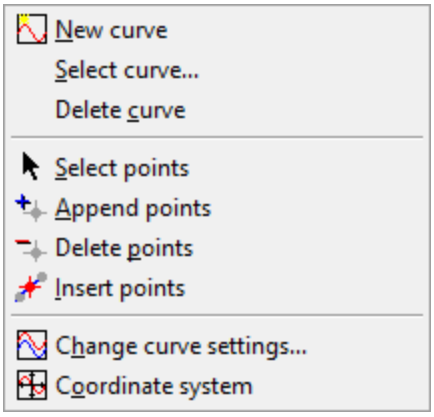
   The cursor changes to a crosshair.

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. See: *Performing Operations on SheetScan Curves.*

Performing Operations on SheetScan Curves

Use the **Curve** menu (from the toolbar or by right-clicking in the sheet) to perform actions on SheetScan curves.



You can perform the following actions:

- **New Curve** – allows you to define a new characteristic curve.
- **Select Curve** – opens the **Select Curve** dialog box, listing all active curves. Select the desired curve and click **OK**.
- **Delete Curve** – deletes the selected curve.

  > **Warning:**
  >
  > You cannot undo this action. If you delete a curve and its data points from the SheetScan editor, you must reconstruct it manually.

- **Select Points** – select this option, then click a point to select it. **Ctrl+click** allows you to select multiple points.
- **Append Points** – select this option, then click to add data points to the end of a curve.
- **Delete Points** – select this option, then click a data point to remove it from the curve.
- **Insert Points** – select this option, then click to insert new data points between existing data points.
- **Change Curve Settings** – opens the **Curve Settings** dialog box, where you can edit the initial curve settings.
- **Coordinate System** – opens the Coordinate System dialog box, where you can edit the coordinate system settings.

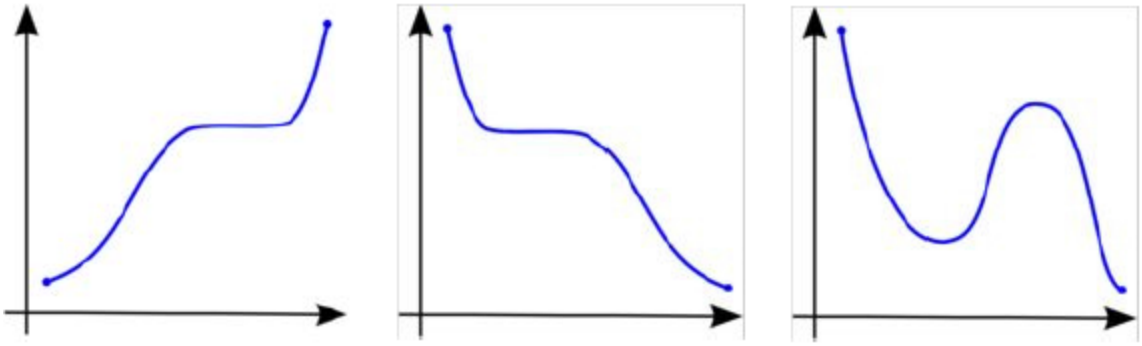
Checking for Monotonicity in X

Q3D Extractor 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.

To check for monotonicity in X from SheetScan:

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.

Importing Characteristic Data into SheetScan

SheetScan supports data import from the following file types:

- Twin Builder Characteristic (*.mdx, *.mda)
- Microsoft Excel (*.xls, *.xlsx)
- Text (*.txt)
- Comma Separated Value (*.csv)
- Spice (*.out)
- Comtrade (*.cfg)
- TEK Oscilloscope (*.dat)

> **Note:**
>
> Before importing characteristic data, you must define a coordinate system.

To import characteristic curve data into SheetScan:

1. Click **File** > **Import**.
2. In the file **Open** dialog box, select the desired data file and click **Open**.
   a. 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 menu.

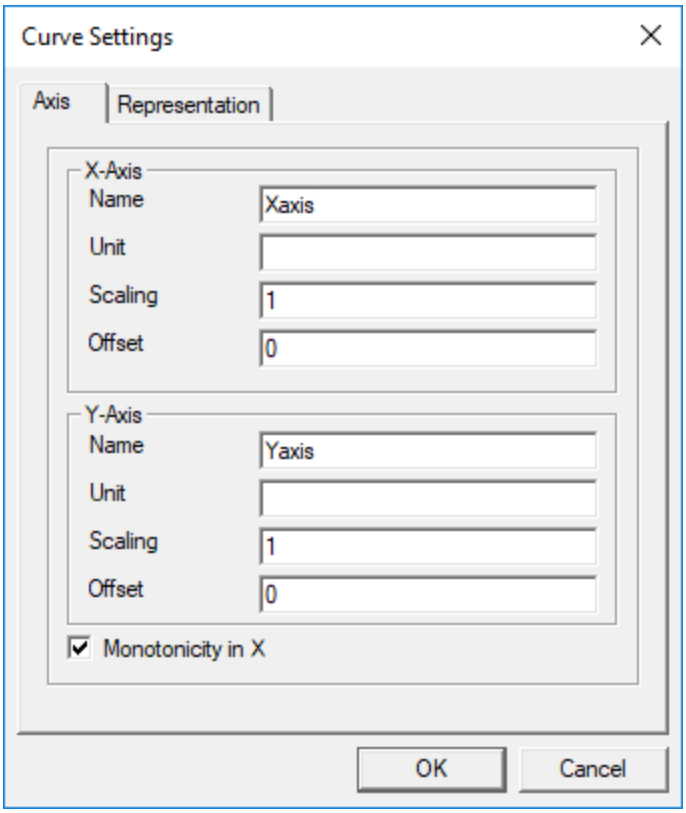
Otherwise, selecting an **.xls** or **.xlsx** file imports the data immediately into the **Add Dataset** dialog box.

> **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.

b. 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.

The **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

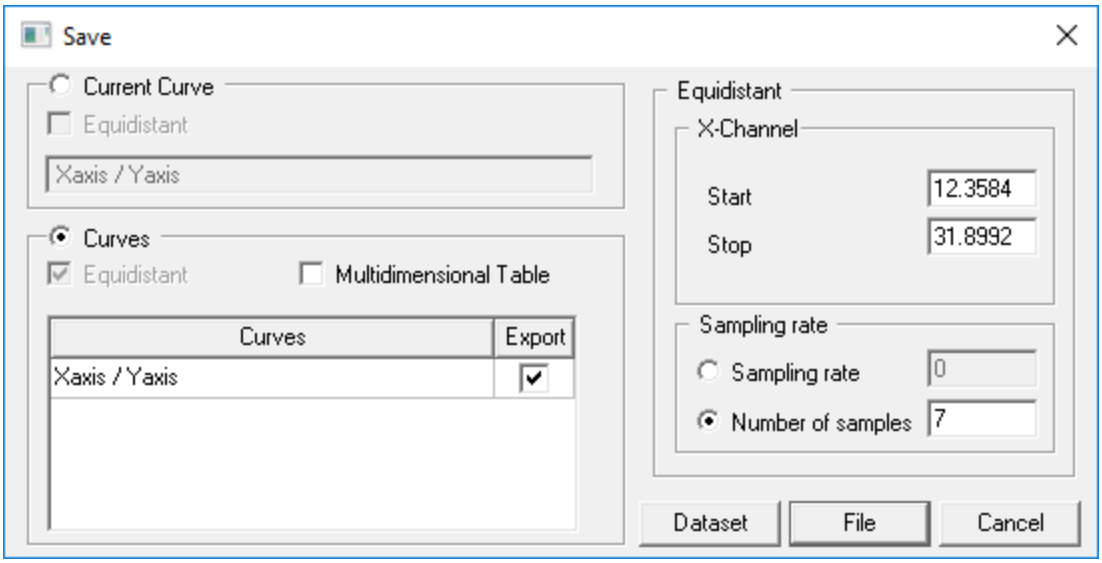
After defining a coordinate system and adding a curve, you can export SheetScan curve data directly to:

- Electronics Desktop Dataset
- Twin Builder Characteristic (*.mdx) file
- Comma Separated Value (*.csv) file
- Comtrade (*.cfg) file

To export curve data:

1. Click **File** > **Export**.

   The **Save** dialog box appears.



2. In the **Save** dialog box, choose **Current Curve** (default) to export current curve data, or **Curves** 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.

> **Note:**
>
> The **Multidimensional Table** option is not currently supported.

3. Choose **Equidistant** if you want to set the **Start** and **Stop X-Channel** values and a **Sample Rate** or **Number of samples** for the exported dataset(s).
4. Click **Dataset** to export curve data directly to the project's dataset file, or click **File** to export as one of the supported file types.

Working with Variables

A variable is a numerical value, mathematical expression, or mathematical function that can be assigned to a design parameter in **Q3D Extractor** or **2D Extractor**. You can assign a variable to any dimension, material property, or output value. Variables are useful in the following situations:

- You expect to change a parameter often.
- You expect to use the same parameter value often.
- You intend to run a parametric analysis, in which you specify a series of variable values within a range to solve.
- You intend to optimize a parameter value by running an optimization analysis or using Design of Experiments to generate a response surface.
- You intend to run a convergence on an output variable.
- You intend to calculate derivatives for variables.
- You intend to animate a plot against a variable.

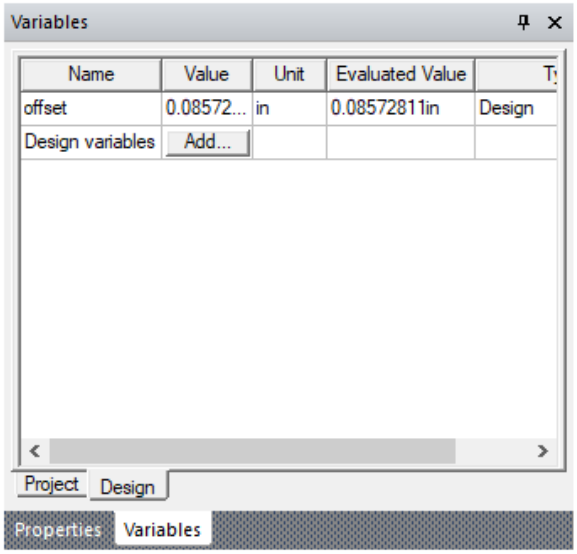
Variables can be regular variables, array index variables, or arrays. A project variable's value can reference other project variables or intrinsics. Values can be a number, variable, or mathematical expression.

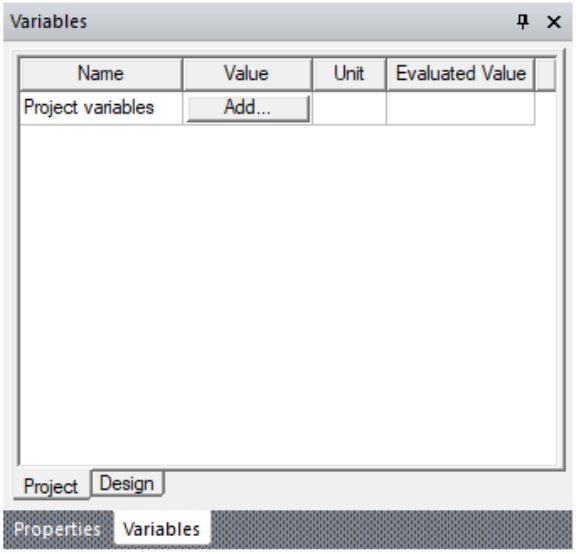
There are two types of variables in Q3D Extractor:

Project Variables	A project variable can be assigned to any parameter value in the Q3D Extractor project in which it was created. Q3D Extractor differentiates project variables from other types of variables by prefixing the variable name with dollar sign symbol ($). You can manually include the $ in the project variable's name, or Q3D Extractor will automatically append the project variable's name after you define the variable. Project variables can be designated as Design, ArrayIndex or Separator variables but not as Post-Processing Variables.
Design (Local) Variables	A design variable can be assigned to any parameter value in the Q3D Extractor design in which it was created. From the Design Variables **Properties** dialog box, you can Add, Add Array, Edit, or Remove design variables. Design Variables can be designated as Design, ArrayIndex or Separator variables and as Post-Processing Variables.

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

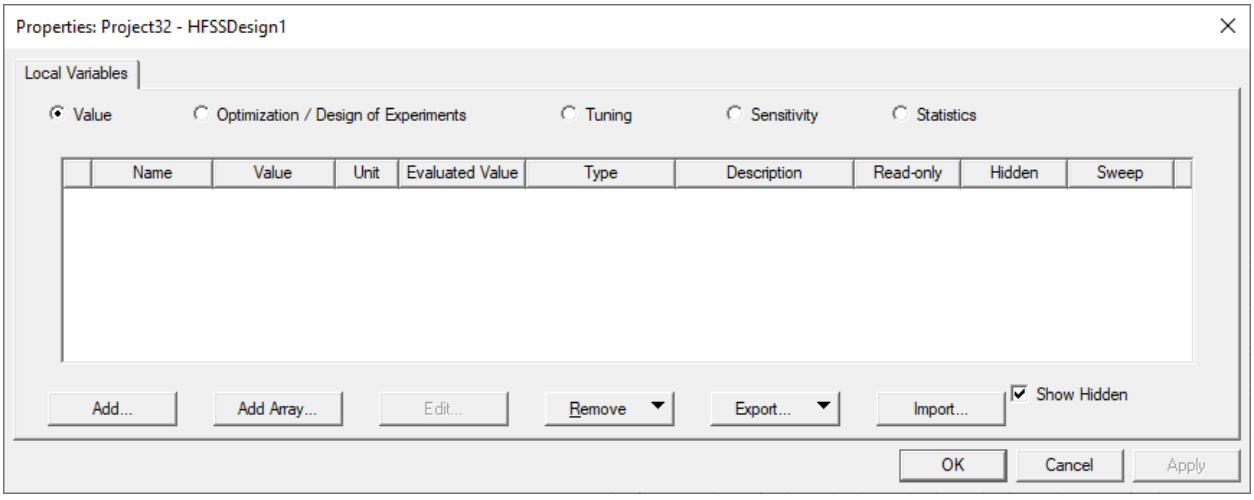
project variable tab. When there is an active design, there will be a corresponding design variable tab. Each tab contains an **Add...** button allowing creation of new variable of this type. If variables exist for the Project or Design, they are shown in the corresponding tab.





Importing and Exporting Variables

Once you have defined Design and/or Project Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The **Properties** dialogs for Design Variables and Project variables include **Import...** and **Export...** buttons.



Working with Project Variables

Project variables are available across all hierarchical levels of a project. A project variable can be assigned to a parameter value in the Q3D Extractor project in which it was created. Q3D Extractor differentiates project variables from other types of variables by prefixing the variable name with the dollar sign symbol ($). You can manually include the $ in the project variable's name when you create it, or Q3D Extractor will automatically append it after you define the variable.

The **Project Variables** menu item allows you to access three tabs:

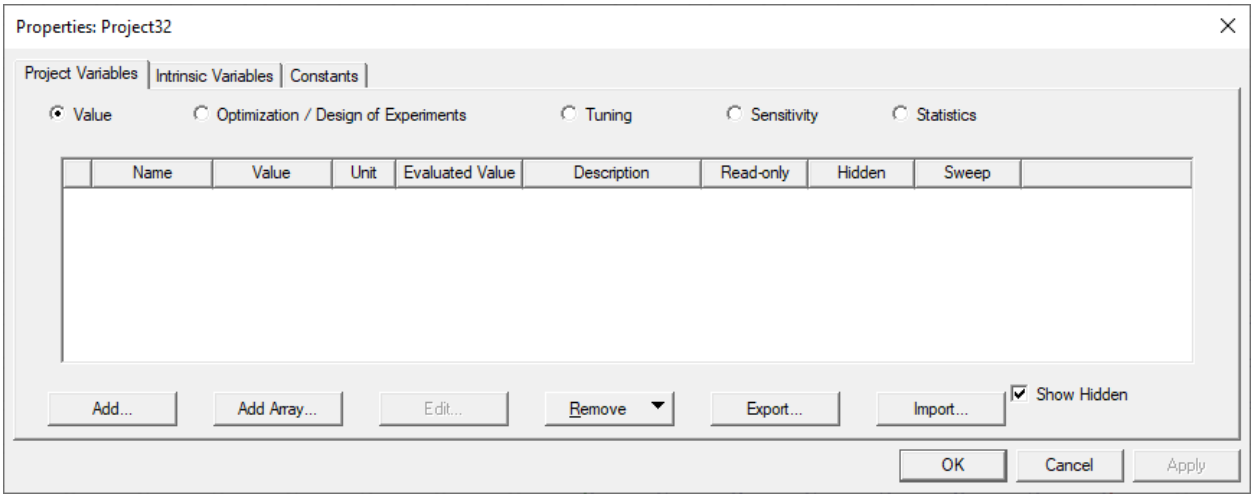
- **Project Variables** – variables, which you define, that apply to the entire project.
- **Intrinsic Variables** – pre-defined variables that cannot be changed. See: List of Intrinsic Variables.
- **Constants** – pre-defined constants that cannot be changed. See: List of Constants.

Adding a Project Variable

To define a Project Variable:

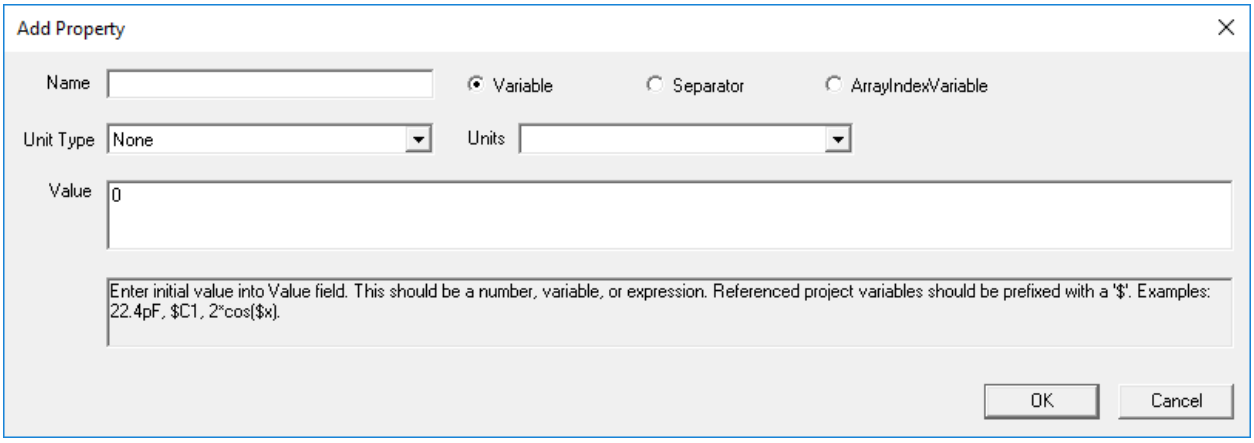
1. Access **Project Variables** one of three ways:
   - Click **Project** > **Project Variables**.
   - Right-click the project name in the Project Tree and select **Project Variables**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

   The **Properties** window appears, on the **Project Variables** tab.



2. From the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.



3. Enter information for the variable, as applicable:
   - **Name** – project variable names must start with the dollar sign symbol ($), followed by a letter. The name can contain alphanumeric characters and underscores (_). You cannot use the names of Intrinsic Variables or pre-defined Constants.
   - **Variable Type** – use the **Variable**, **Separator**, and **ArrayIndexVariable** radio buttons to select the variable type.

     Your selection impacts which properties you can edit:

Variable Type	Editable Properties
Variable	Unit Type, Units, Value.

Variable Type	Editable Properties
Separator	Name. A separator variable provides a bolded name for a blank line to facilitate grouping variables in variable lists.
Array Index Variable	Associate Array Variable, Value

- **Unit Type** – for Variables, use the drop-down menu to select a type from the list (for example, Charge, Density, Energy, ...). “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.

- **Units** – for Variables, use the drop-down menu to select a unit of measure.
- **Value** – for Variables and ArrayIndexVariables, enter a number, variable, or mathematical expression. 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.

  For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

  > **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.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

The new variable appears in the list. You can sort project variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing

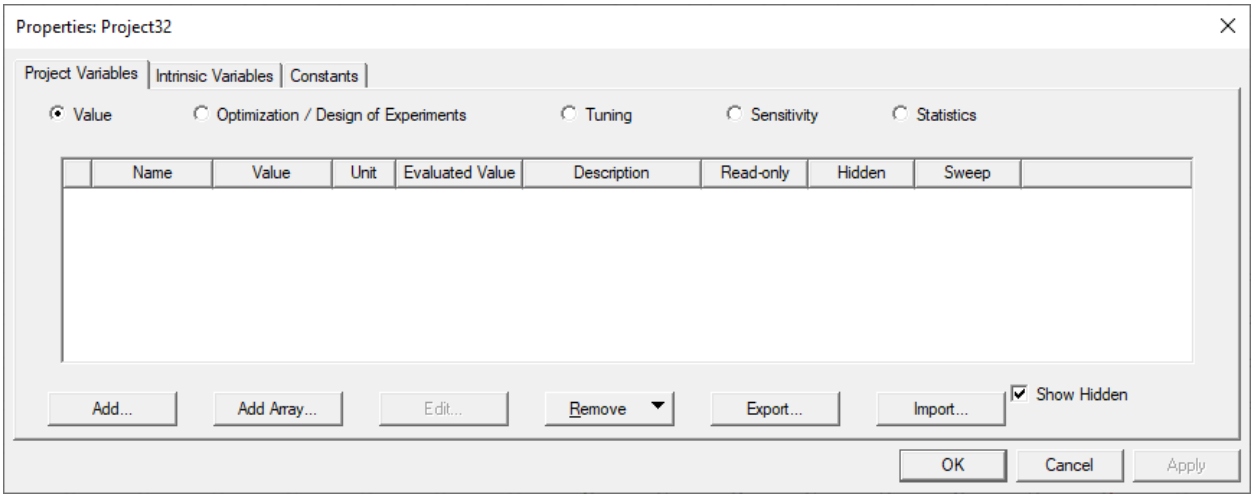
up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

5. If desired, use the check boxes to designate a variable as **Read-only**, **Hidden**, or **Sweep**.
   - **Read-only** – when selected, the variable's name, value, unit, and description cannot be modified.
   - **Hidden** – hidden variables do not appear in the **Properties** window unless **Show Hidden** is selected.
   - **Sweep** – allows you to designate variables to include in solution indexing as a way to permit faster post-processing. Variables with the Sweep check box cleared are not used in solution indexing. If a solution exists, selecting or clearing a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click OK to dismiss the warning.
6. Click **Apply** to apply changes.
7. Click **OK** to exit the window.

The new variable can be assigned to a parameter value in the project in which it was created. You can enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity, or Statistics. See: Optimetrics.

Importing and Exporting Project Variables

Once you have defined Project Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The **Properties** dialogs for and Project variables include **Import...** and **Export...** buttons.



When you click **Export....** a dialog lets you name the file and select a location. When you **Save**, if the file exists, an overwrite prompt should pop up.

Deleting a Project Variable

To remove a Project Variable:

1. Access **Project Variables** as described above.

   The **Properties** window appears, on the **Project Variables** tab.

2. Remove one or more variables:
   - To remove a specific variable, select it and click **Remove** > **Remove Selected**.
   - To remove all unused variables, click **Remove** > **Remove All Unused**.
   - To force the removal of all unused variables, including those in the project's undo/redo history, click **Remove** > **Force Remove All Unused**.

Editing a Project Variable

To edit a Project Variable:

1. Access **Project Variables** as described above.

   The **Properties** window appears, on the **Project Variables** tab.

2. Select the variable you want to edit and click **Edit**.
3. Change the properties as desired and click **OK**.

Adding a Project Variable Array

You can define array variables that contain numbers or strings. Number array variables can be used in component property expressions, while string array variables can be used in certain component property values.

Note:

Text array variables can be used in certain component property values, but not all, and cannot be combined with operators to form more complex expressions.

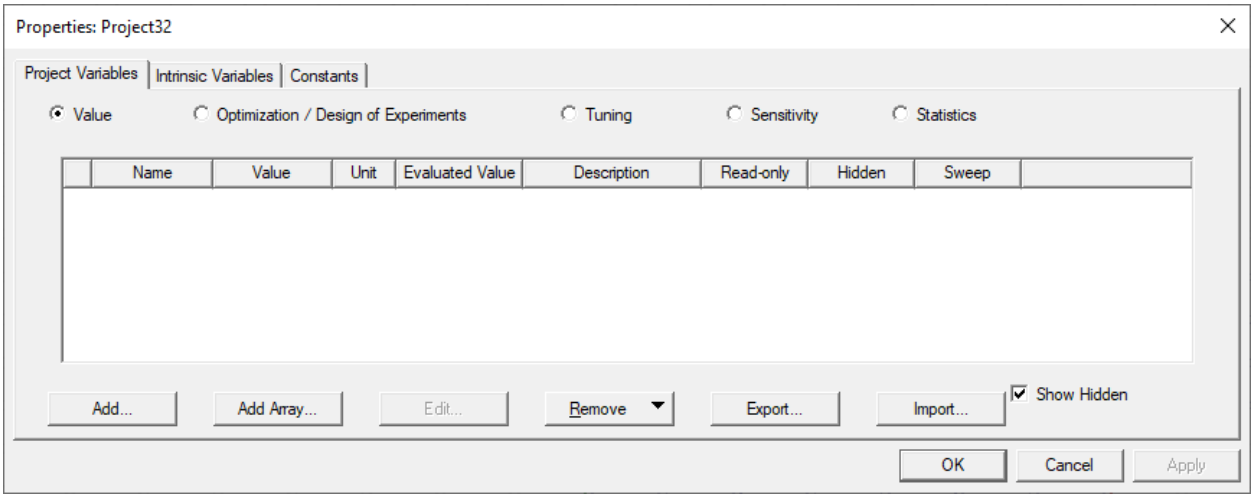
The index for a text array variable reference can either be a constant (1) or can be an index (ii) or even an expression (ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept. The following are properties that currently allow text array variables:

- V_PRBS
- V_PRBS_JITTER
- V_PRBSD
- V_PRBSG_JITTER
- V_PSK
- V_QAM
- V_CPM

To define an array variable, you must first define a Project Variable Array:

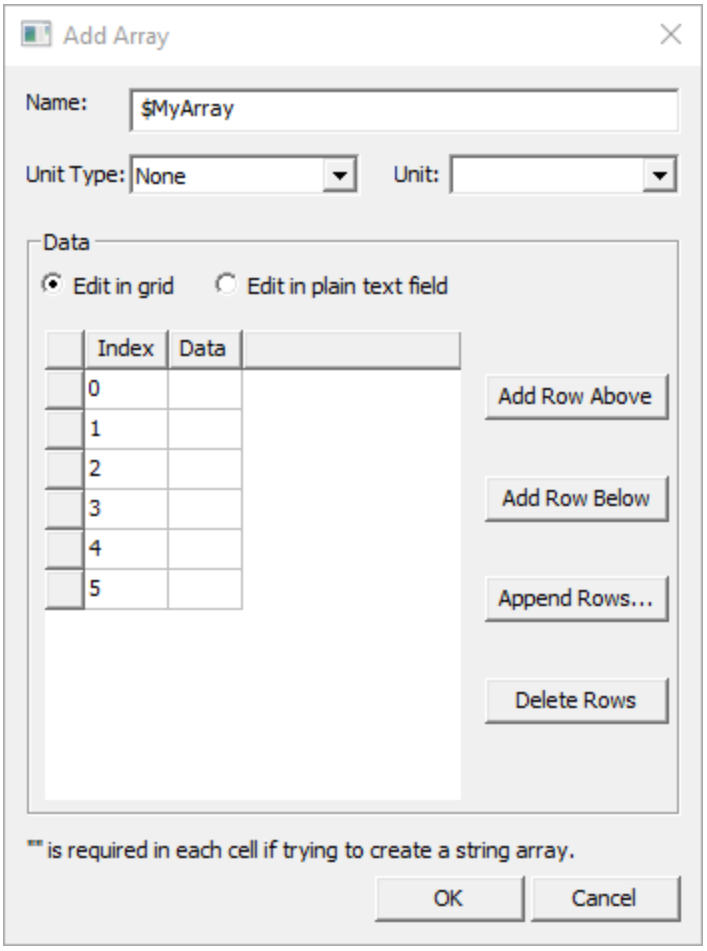
1. Access **Project Variables** one of three ways:
    - Click **Project** > **Project Variables**.
    - Right-click the project name in the Project Tree and select **Project Variables**.
    - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**

    The **Properties** window appears, on the **Project Variables** tab.



2. Click **Add Array**.

   The **Add Array** window appears.



The value grid displays each array item's **Index** number, and the **Data** associated with that index number.

3. Give the variable array a **Name**, and select a **Unit Type** and **Unit** from the drop-down menus.
4. Use the control buttons to add, delete, and reposition rows in the value grid at left. The default is **Edit in grid**, but you may select **Edit in plain text field**.

> **Note:**
>
> Quotation marks (“ ”) are *required* as delimeters when array values are entered in either the grid or text field.

5. When you have finished entering values, click **OK**.

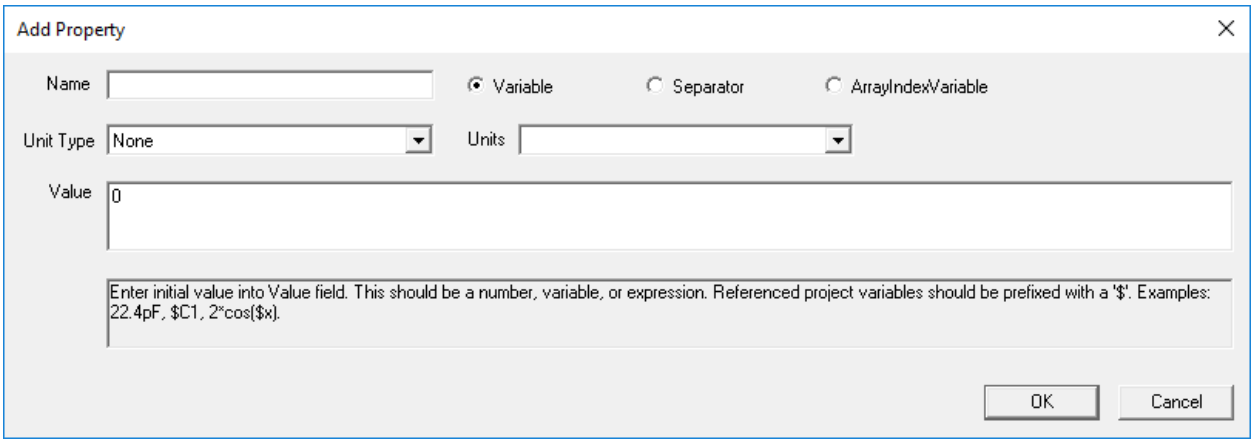
   The array appears in the **Project Variables** list.

You can add a variable to the array by adding an ArrayIndexVariable on the **Project Variables** tab.

1. Access **Project Variables** one of three ways:
   - Click **Project** > **Project Variables**.
   - Right-click the project name in the Project Tree and select **Project Variables**.
   - From the **Setup Optimization** dialog box, click **Edit Variables** > **Edit Project Variables**.

   The **Properties** window appears, on the **Project Variables** tab.
2. From the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.



3. Enter information for the variable, as applicable:
   - **Name** – project variable names must start with the dollar sign symbol ($), followed by a letter. The name can contain alphanumeric characters and underscores (_). You cannot use the names of Intrinsic Variables or pre-defined Constants.
   - **Variable Type** – select ArrayIndexVariable.
   - **Value** – enter a number, variable, or mathematical expression. 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.

     For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array

variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

> **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.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

List of Intrinsic Variables

AEDT 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.

To view the list of Intrinsic Variables, click **Project** > **Project Variables** and select the **Intrinsic Variables** tab.

Name	Unit	Description
_Empty		Empty value, taken to be model default by simulator (Twin Builder)
_I1	mA	Armature current; **this variable is only used in RMxprt**.
_I2	mA	Exciting current; **this variable is only used in RMxprt**.
_I3	mA	Current; **this variable is only used in RMxprt**.
_I4 through _I9		Not used
_t,**_u**,**_v**		Variables to define parametric equation-based curve.
_V1 to **_V9**	mV	Port Voltage in user-defined model (V).
Ang	Ang	Angle. Postprocessing variable
Budget Index		Postprocessing variables; cannot be set by the user
Distance	mm	
Electrical Degree	deg	Electric degree of the rotating machine; cannot be set by the user.
F	GHz	Frequency of the circuit/system analysis
F1, **F2**, **F3**	GHz	Frequency tones 1, 2, 3 in harmonic balance analysis. (Hz).
FNoi	GHz	Offset noise frequency in harmonic balance noise analysis.

Name	Unit	Description
Freq	Hz	Frequency for analysis and postprocessing; cannot be set by the user
Hmax	ns	
Hmin	ns	
Ia, **Ib**	mA	Postprocessing variables; cannot be set by the user
Index		Identifier for a data point; cannot be set by the user
IWave Phi, **IWave Theta**	deg	Incident wave spherical coordinate variables. Phi is the angle from the origin in the z direction and Theta the angle from the x-axis.
Normalized Deformation, **Normalized Distance**		Postprocessing variables; cannot be set by the user
OP	mW	Postprocessing variables; cannot be set by the user
Pass		Post processing variables, not settable by the user.
Phase	deg	Angle of a complex number.
Phi	deg	The angle measured from the x-axis and can be from 0 to 360 degrees.
R	mm	R is the cylindrical coordinate system variable.
Rho		Rho is the spherical coordinate system variable.
Rspeed	rpm	Speed of the machine.
Spectrum	GHz	Postprocessing variable.
Temp	cel/deg	Analysis temperature (deg)
Tend	ns	
Theta	deg	The angle measured from the z-axis, which is the axis perpendicular to the plane of the work space, and must be from 0 to 180 degrees.
Time	ns	Time point in transient analysis
Time0	ns	Time 0 point in a transient analysis
Vac, **Vbe**, **Vce**, **Vds**, **Vgs**	mV	Post processing variables; cannot be set by the user
X,**Y**,**Z**	mm	Point coordinate variables in the modeler.
ZAng, **XRho**	deg	A spherical coordinate system; cannot be set by the user

List of Constants

To view the list of constants, click **Project** > **Project Variables** and select the **Constants** tab.

Name	Value	Description
abs0	-273.15	Absolute zero (C)

Name	Value	Description
boltz	1.3806503e-23	Boltzmann constant (J/K)
c0	299792458	Speed of light in a vacuum (m/s)
e0	8.854187817e-12	Permittivity of vacuum (F/m)
elecq	1.602176462e-19	Electron charge (C)
eta	376.730313461	Impedance of vacuum (Ohm)
false	0	Boolean False
g0	9.80665	Acceleration due to gravity of Earth (m/s^2)
mathE	2.718281828	Euler's number (Napier's constant)
pi	3.1415926535898	Ratio of circumference to diameter
planck	6.6260755e-34	Planck's constant (m^2*kg/s)
true	1	Boolean True
u0	1.2566370614359e-06	Permeability of vacuum (H/m)

Working with Design Variables

Design Variables (Local Variables) are only available within the Q3D Extractor design for which they are defined. A design variable can be assigned to a parameter value in the design. You can also add a variable defined with an array of values.

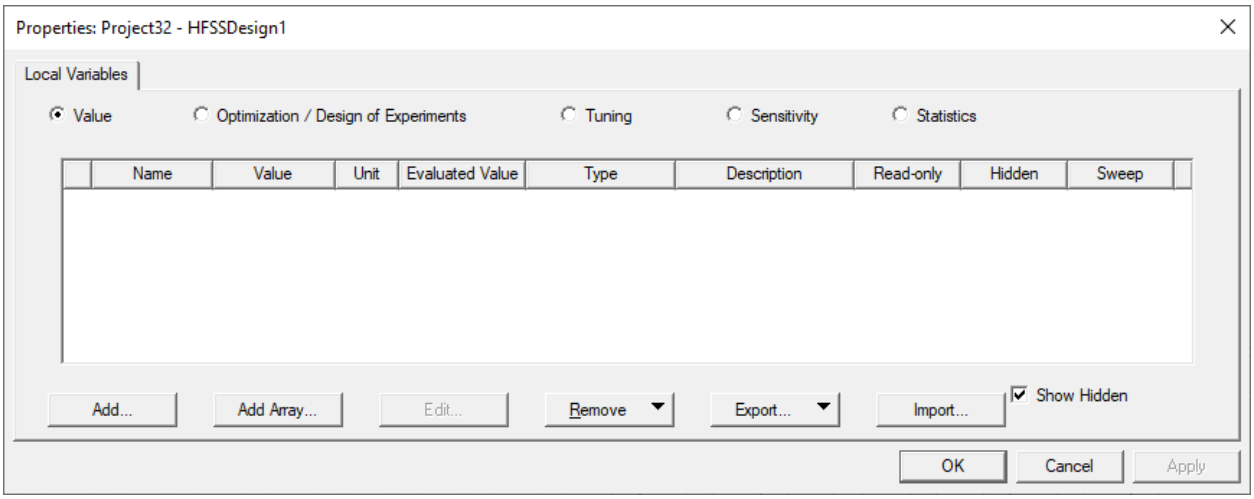
The **Design Properties** menu item allows you to access up to three tabs, depending on your design:

- **Parameter Defaults** – local variables with default values that can be overridden in instances of a design. For example, if three subcircuit instances contain a parameter default C1 that is defined as equal to 11.3pF, C1 may be overridden as 11.8pF in the first instance, overridden as 10.9pF in the second, and left at its default value of 11.3pF in the third. A property value that has been set by means of a parameter default is called a *passed parameter*. Parameter defaults can be defined at the design level.
- **Local Variables** – variables, which you define, that apply to the current design only. Local variables can be defined at the design level.
- **General** – pre-defined design variables that cannot be changed. You cannot use the names of these variables when creating a new variable.

> **Note:**
>
> Not all tabs appear for all designs.

Importing and Exporting Variables

Once you have defined Design Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The Properties dialogs for Design Variables include **Import...** and **Export...** buttons.



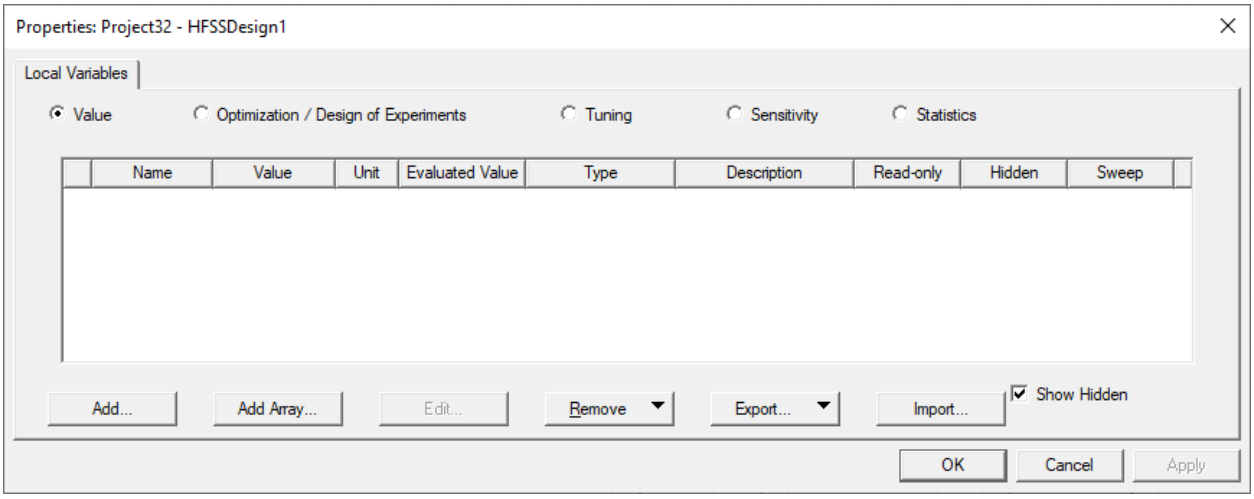
Defining Local Variables at the Design Level

Adding a Local Variable

To define a Local Variable:

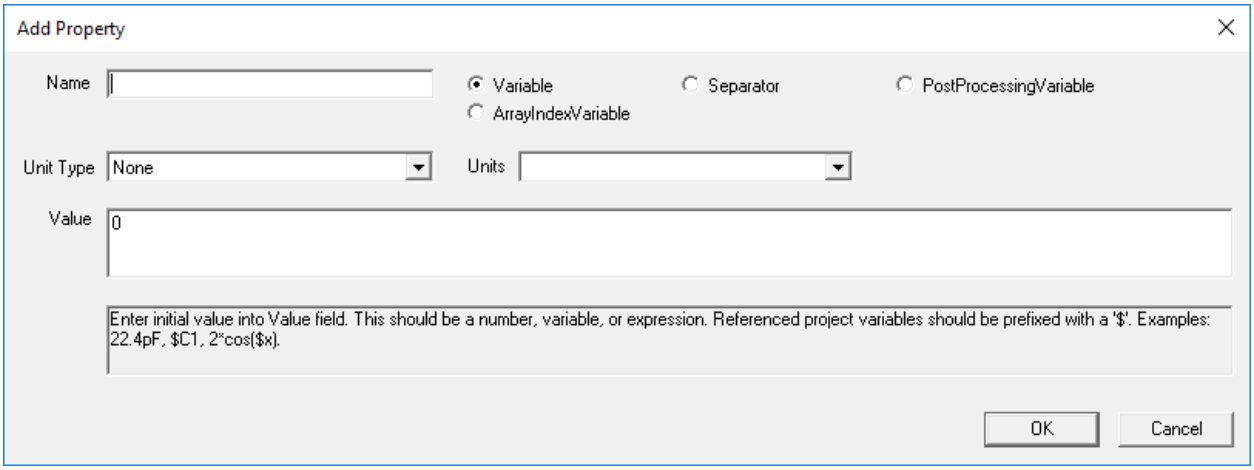
1. Access **Design Properties** one of three ways:
    - Click **Q3D Extractor or 2D Extractor** > **Design Properties**.
    - Right-click the design name in the Project Tree and select **Design Properties**.
    - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**

    The **Properties** window appears with the **Local Variables** tab.



2. Click **Add**.

   The **Add Property** window appears.



3. Enter information for the variable, as applicable:
   - **Name** – the name must start with a letter and can contain alphanumeric characters and underscores (_). You cannot use any names already defined on the **General** tab.
   - **Variable Type** – use the **Variable**, **Separator**, **ArrayIndexVariable**, and **PostProcessingVariable** radio buttons to select the variable type.

     > **Note:**
     >
     > Not all variable types are available for all designs.

Your selection impacts which properties you can edit:

Variable Type	Editable Properties
Variable	Unit Type, Units, Value.
Separator	Name. A separator variable provides a bolded name for a blank line to facilitate grouping variables in variable lists.
Array Index Variable	Associate Array Variable, Value
PostProcessingVariable	Unit Type, Units, Value.

- **Unit Type** – for Variables and PostProcessingVariables, use the drop-down menu to select a type from the list (for example, Charge, Density, Energy, ...).
- **Units** – for Variables and PostProcessingVariables, use the drop-down menu to select a unit of measure.
- **Value** – for Variables, ArrayIndexVariables, and PostProcessingVariables, enter a number, variable, or mathematical expression. 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.

  For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

  > **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.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

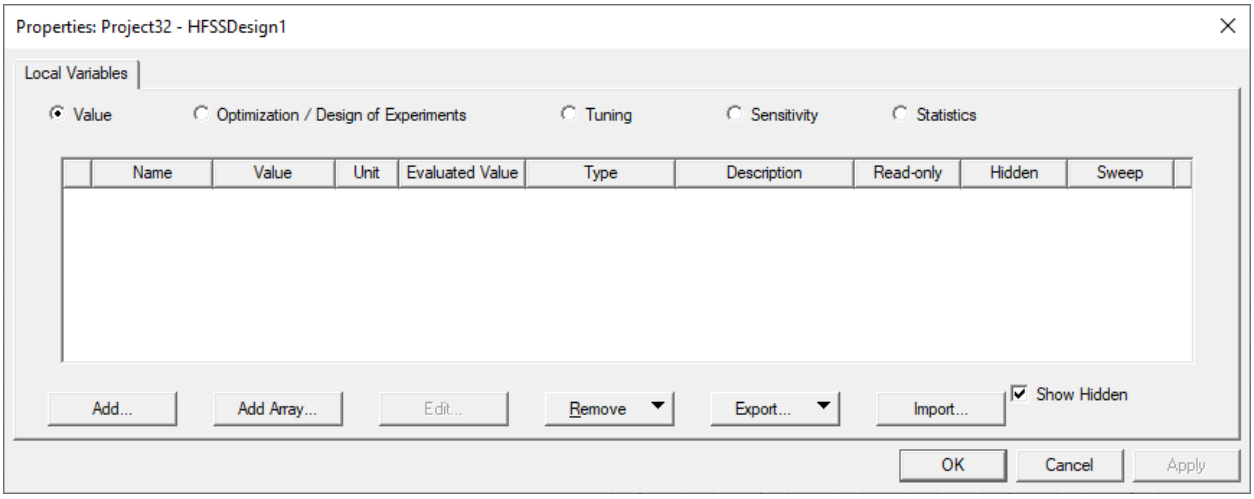
The new variable appears in the list. You can sort local variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

5. If desired, use the check boxes to designate a variable as **Read-only**, **Hidden**, or **Sweep**.
    - **Read-only** – when selected, the variable's name, value, unit, and description cannot be modified.
    - **Hidden** – hidden variables do not appear in the **Properties** window unless **Show Hidden** is selected.
    - **Sweep** – allows you to designate variables to include in solution indexing as a way to permit faster post-processing. Variables with the Sweep check box cleared are not used in solution indexing. If a solution exists, selecting or clearing a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click OK to dismiss the warning.
6. Click **Apply** to apply changes.
7. Click **OK** to exit the window.

The new variable can be assigned to a parameter value in the project in which it was created. You can enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity, or Statistics. See: Optimetrics.

Importing and Exporting Variables

Once you have defined Design Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The Properties dialogs for Design Variables include **Import...** and **Export...** buttons.



Deleting a Local Variable

To remove a Local Variable:

1. Access **Local Variables** as described above.
2. Remove one or more variables:
    - To remove a specific variable, select it and click **Remove** > **Remove Selected**.
    - To remove all unused variables, click **Remove** > **Remove All Unused**.
    - To force the removal of all unused variables, including those in the project's undo/redo history, click **Remove** > **Force Remove All Unused**.

Editing a Local Variable

To edit a Local Variable:

1. Access **Local Variables** as described above.
2. Select the variable you want to edit and click **Edit**.
3. Change the properties as desired and click **OK**.

Overriding a Local Variable

You can override a variable value from an analysis setup's **General** tab.

Select the **Override** check box next to the value you wish to override, and enter a new value in the **Value** field.

Design Variable	Override	Value	Units
padrad	☐	0.6	mm
viarad	☐	0.2	mm

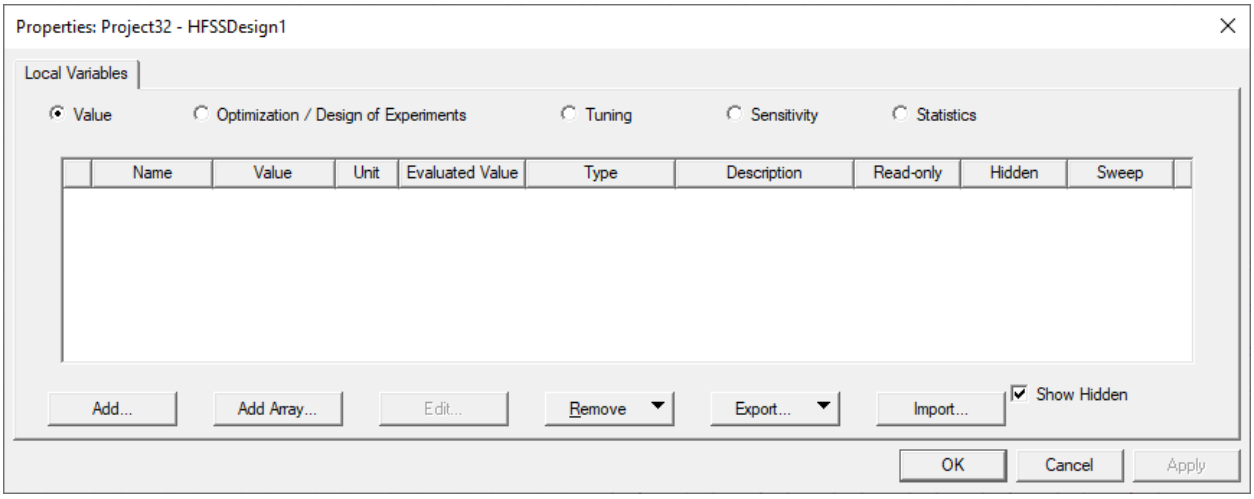
Defining Parameter Defaults at the Design Level

Adding a Parameter Default

To define a new parameter default at the design level:

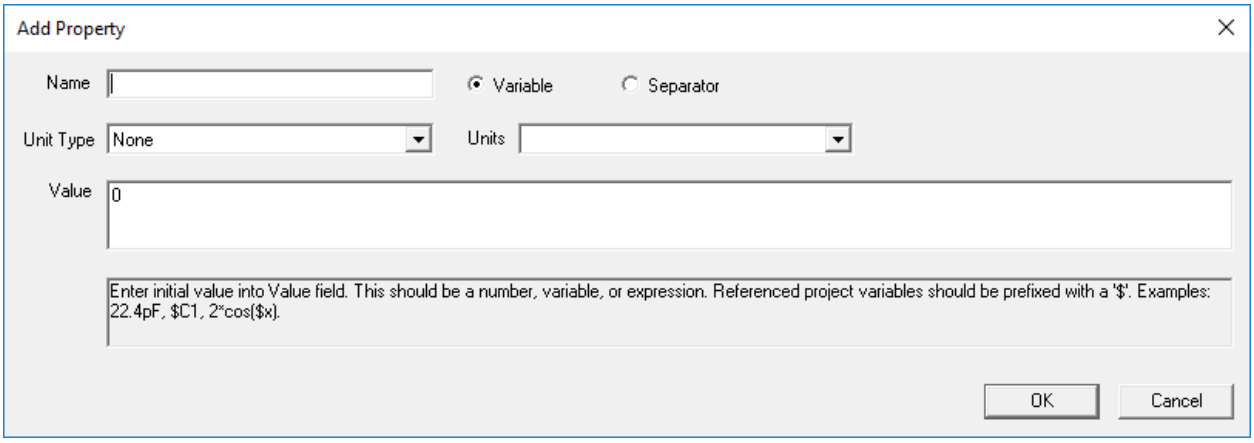
1. Access **Design Properties** one of three ways:
    - Click **Q3D Extractor or 2D Extractor** > **Design Properties**.
    - Right-click the design name in the Project Tree and select **Design Properties**.
    - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**

    The **Properties** window for design variables appears.



2. Click **Add...**.

   The **Add Property** window appears.



3. Enter information for the parameter default, as applicable:
   - **Name** – the name can contain alphanumeric characters and underscores (_). You cannot use any names already defined on the **General** tab.
   - **Variable Type** – select **Variable** to enter a value, or select **Separator** to provide a bolded name for a blank line.

     Your selection impacts which properties you can edit:

Variable Type	Editable Properties
Variable	Unit Type, Units, Value.

Variable Type	Editable Properties
Separator	Name. A separator variable provides a bolded name for a blank line to facilitate grouping variables in variable lists.

- **Unit Type** – for Variables, use the drop-down menu to select a type from the list (for example, Charge, Density, Energy, ...).
- **Units** – for Variables, use the drop-down menu to select a unit of measure.
- **Value** – for Variables, enter a valid numeric quantity. Alternately, you can assign a variable to a parameter by typing a variable name or mathematical expression in place of a parameter value.

4. Click **OK**.

   The new variable appears in the list. You can sort local variables by clicking the Name column header. 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.

5. If desired, use the check boxes to designate a variable as **Read-only**, **Hidden**, or **Sweep**.
   - **Read-only** – when selected, the variable's name, value, unit, and description cannot be modified.
   - **Hidden** – hidden variables do not appear in the **Properties** window unless **Show Hidden** is selected.
6. Click **Apply** to apply changes.
7. Click **OK** to exit the window.

Deleting a Parameter Default

To remove a Parameter Default:

1. Access **Parameter Defaults** as described above.
2. Select a parameter from the list and click **Remove**.

Editing a Parameter Default

To edit a Parameter Default:

1. Access **Parameter Defaults** as described above.
2. Select the parameter you want to edit and click **Edit**.
3. Change the properties as desired and click **OK**.

Overriding a Parameter Default

You can override a parameter default from an analysis setup's **General** tab.

Select the **Override** check box next to the value you wish to override, and enter a new value in the **Value** field.

Design Variable	Override	Value	Units
padrad	☐	0.6	mm
viarad	☐	0.2	mm

Adding a Design Variable Array

You can define array variables that contain numbers or strings. Number array variables can be used in component property expressions, while string array variables can be used in certain component property values. Design variable arrays are only available within the Q3D Extractor design for which they are defined.

> **Note:**
>
> Text array variables can be used in certain component property values, but not all, and cannot be combined with operators to form more complex expressions.
>
> The index for a text array variable reference can either be a constant (1) or can be an index (ii) or even an expression (ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept. The following are properties that currently allow text array variables:
>
> - V_PRBS
> - V_PRBS_JITTER
> - V_PRBSD
> - V_PRBSG_JITTER
> - V_PSK
> - V_QAM
> - V_CPM

To define an array variable, you must first define a Local Variable Array:

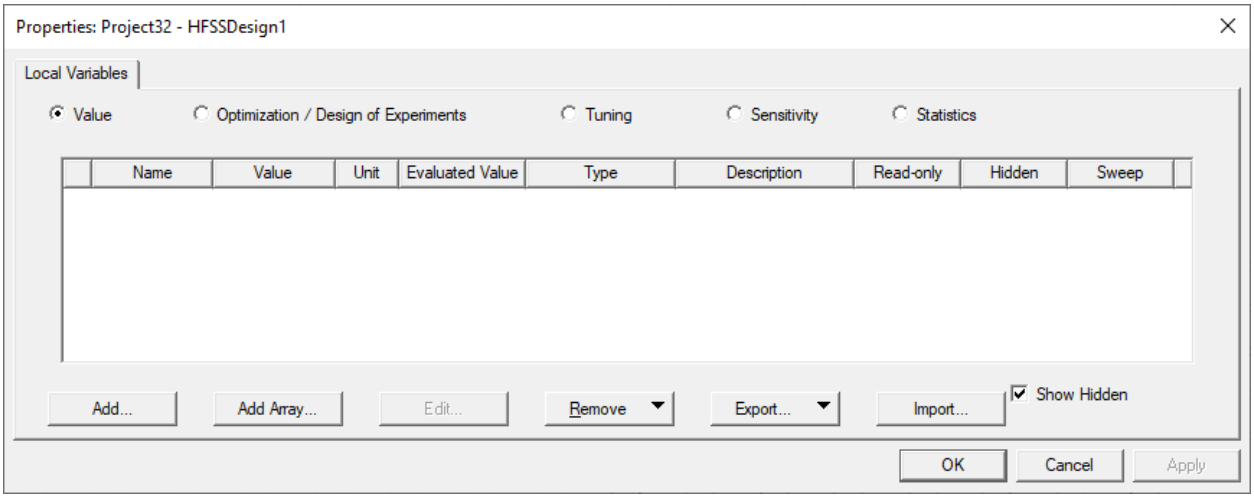
1. Access **Local Variables** one of three ways:
   - Click **Q3D Extractor or 2D Extractor** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.

- 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** window appears.

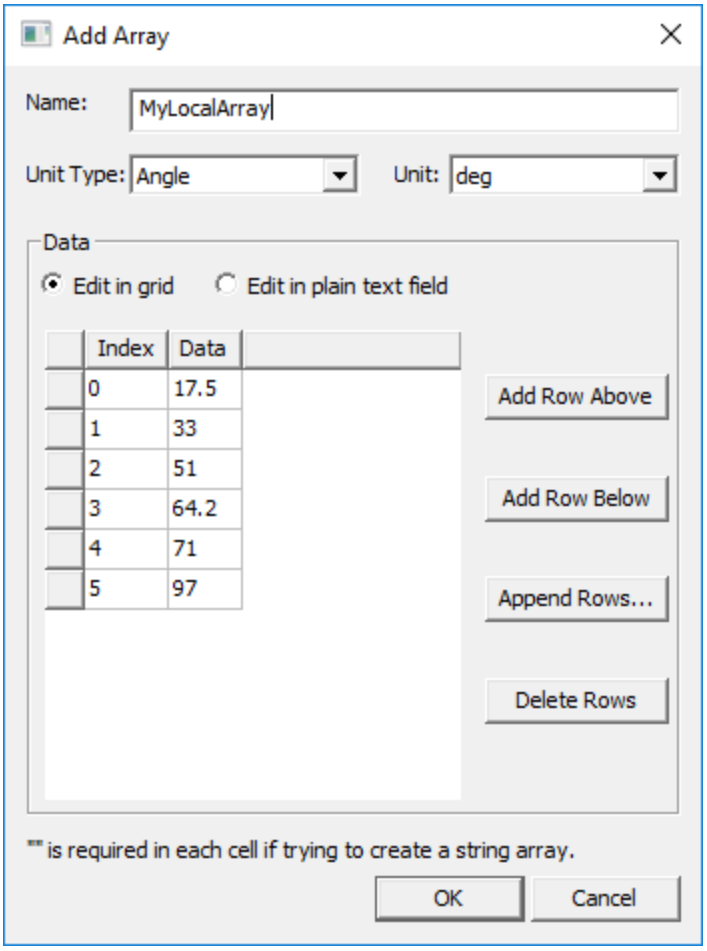
2. Select the **Local Variables** tab.

   The **Local Variables** tab displays.



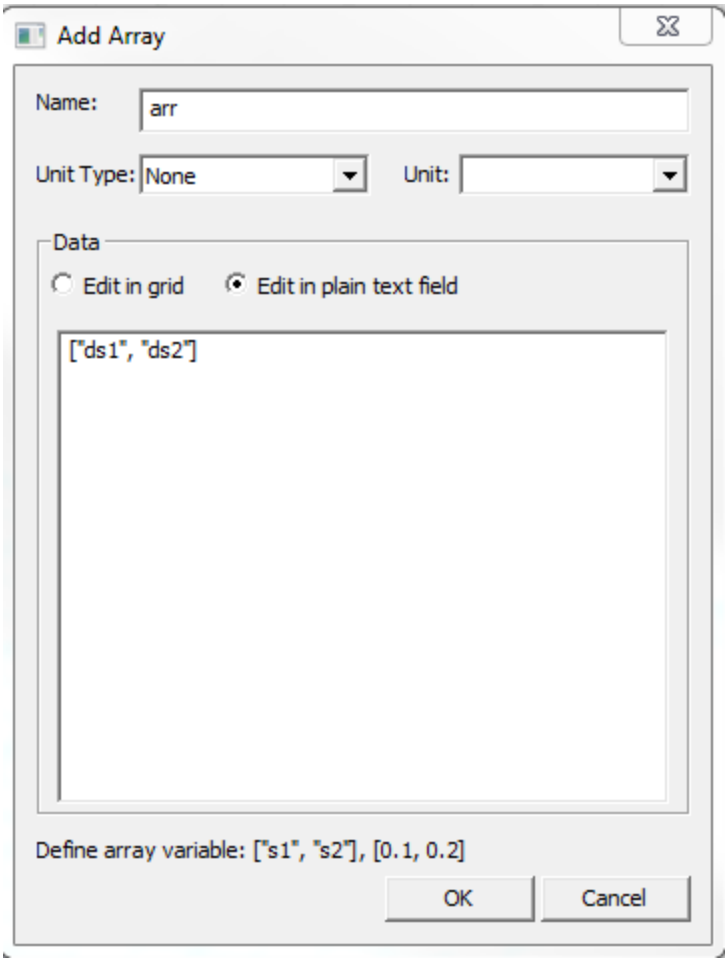
3. Click **Add Array**.

   The **Add Array** window appears.



The value grid displays each array item's **Index** number, and the **Data** associated with that index number.

If you elected to **Edit in plain text field** in the Add Array dialog box, the bracketed and comma delimited format is used.



4. Give the variable array a **Name**, and select a **Unit Type** and **Unit** from the drop-down menus.
5. Use the control buttons to add, delete, and reposition rows in the value grid at left. The default is **Edit in grid**, but you may select **Edit in plain text field**.

   > **Note:**
   >
   > Quotation marks ("") are *required* as delimeters when array values are entered in either the grid or text field.

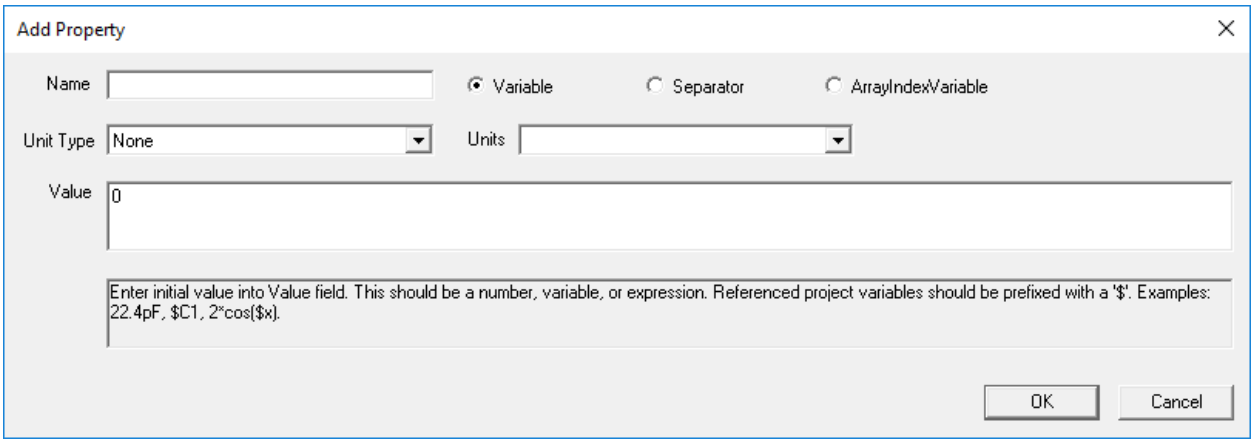
6. When you have finished entering values, click **OK**.

   The array appears in the **Local Variables** list.

You can add a variable to the array by adding an ArrayIndexVariable on the **Local Variables** tab.

1. Access **Local Variables** one of three ways:
   - Click **Q3D Extractor or 2D Extractor** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From the **Setup Optimization** dialog box, click **Edit Variables** > **Edit Design Variables**.
2. From the **Local Variables** tab, click **Add**.

   The **Add Property** dialog box appears.



3. Enter information for the variable, as applicable:
   - **Name** – the name can contain alphanumeric characters and underscores (_). You cannot use the names of Intrinsic Variables or pre-defined Constants.
   - **Variable Type** – select ArrayIndexVariable.
   - **Value** – enter a number, variable, or mathematical expression. 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.

     For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

> **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.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

Converting Variables and Parameter Defaults

Design (Local) variables can be converted to parameter defaults, and vice versa.

To convert a parameter default or local variable:

1. Access **Design Properties** one of three ways:
   - Click **Q3D Extractor or 2D Extractor** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From the **Setup Optimization** dialog box, click **Edit Variables** > **Edit Design Variables**.

   The **Properties** window appears.
2. Select either the **Parameter Defaults** tab or the **Local Variables** tab, depending on which item you want to convert.
3. Select the variable/parameter and click **Convert to Parameter** or **Convert to Variable**.

   The item will move to the list on the appropriate tab.

Some special cases to consider:

- If a local variable references another local variable, it cannot be converted to a parameter default. You will see a pop-up message if you attempt to convert. This is because local variables can reference parameter defaults, but parameter defaults cannot reference local variables.
- Parameter defaults can be overridden and can have different values in each subcircuit, while a Local Variable cannot. If a parameter default has been overridden and you convert it to a variable, it will lose all overrides. Attempting to do so will display a pop-up message asking if you really want to do the conversion and lose the override values.

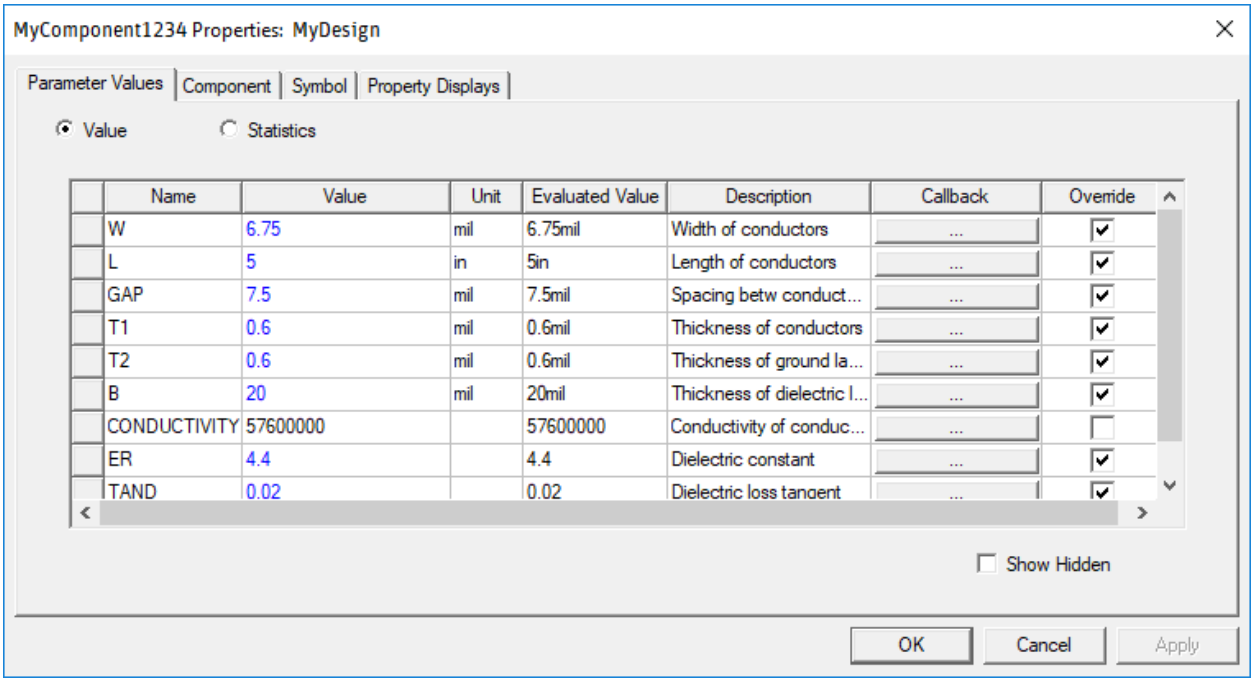
Adding Variables and Parameter Defaults from Components

To define a new variable or parameter default from a component:

1. Display a component's **Properties** window one of three ways:
   - Double-click the component.
   - Select the component and click **Edit** > **Properties**.
   - Select the component, right-click and select **Properties**.

   The **Properties** window appears.
2. Select the **Parameter Values** tab.

   The **Parameter Values** tab displays.



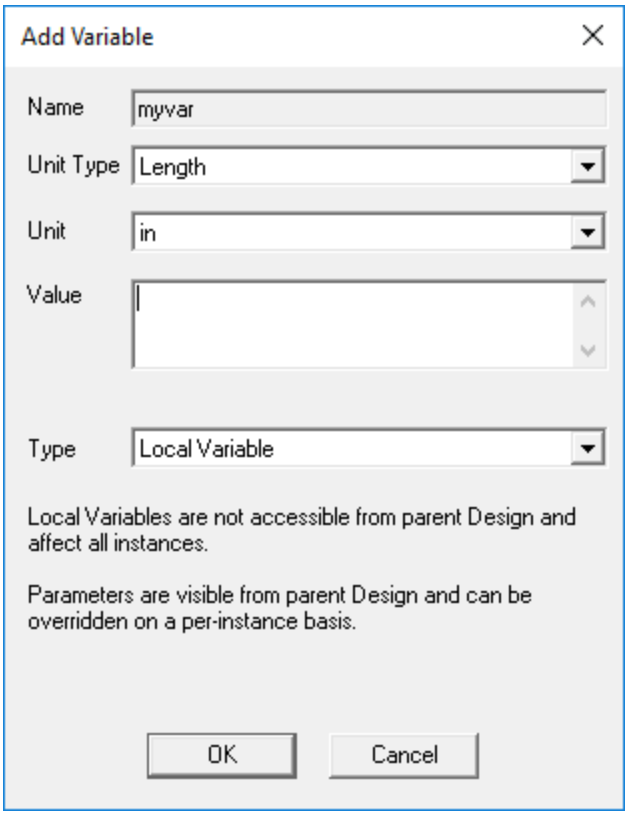
3. Select the **Value** radio button.
4. Select the **Value** field for the parameter you want to set equal to a local variable.
5. Type the variable name. The name may not be the same as an intrinsic variable or constant.

   Variable names may include alphanumeric characters and underscores (_).

> **Important:**
>
> If you wish to define a Project Variable, you must begin the name with a dollar sign ($).
>
> When you enter a name beginning with $, you will only be allowed to define it as a Project Variable.

6. Press **Enter**.

   The **Add Variable** window appears.



7. Select a **Unit Type** and **Unit** using the drop-down menus.
8. Enter a **Value**.

   The Value can be a number, variable, or mathematical expression. 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.

> **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.

9. From the **Type** drop-down menu, select **Local Variable** or **Parameter Default**.

> **Note:**
>
> If you wish to define a Project Variable, your variable name must begin with a dollar sign ($).

10. Click **OK** to exit the **Add Variable** window.
11. Click **Apply** to save your changes.
12. Click **OK** to exit the **Properties** window.

Fixed vs. Non-Fixed Variables

By default, both Project variables and Design variables are included in the list of variables that index solution data. As a result, they are visible when selecting data to be displayed in plots.

However, if you define a variable as *fixed*, it is not swept and will not index solution data. Consequently, defining variables as fixed can speed up simulation—particularly if there are many variables to simulate.

To define a variable as fixed:

1. Open the **Properties** window for either Project or Design variables.
2. Clear the **Sweep** check box next to the variable you want to be fixed.

	Name	Value	Unit	Evaluated Value	Type	Description	Read-only	Hidden	Sweep
	wg_width	7.1	mm	7.1mm	Design		☐	☐	☑
	fillet_radius	0	mm	0mm	Design		☐	☐	☐

- If a **Sweep** check box is selected (default) the corresponding variable is NOT fixed.
- If a **Sweep** check box is cleared, the variable is fixed.

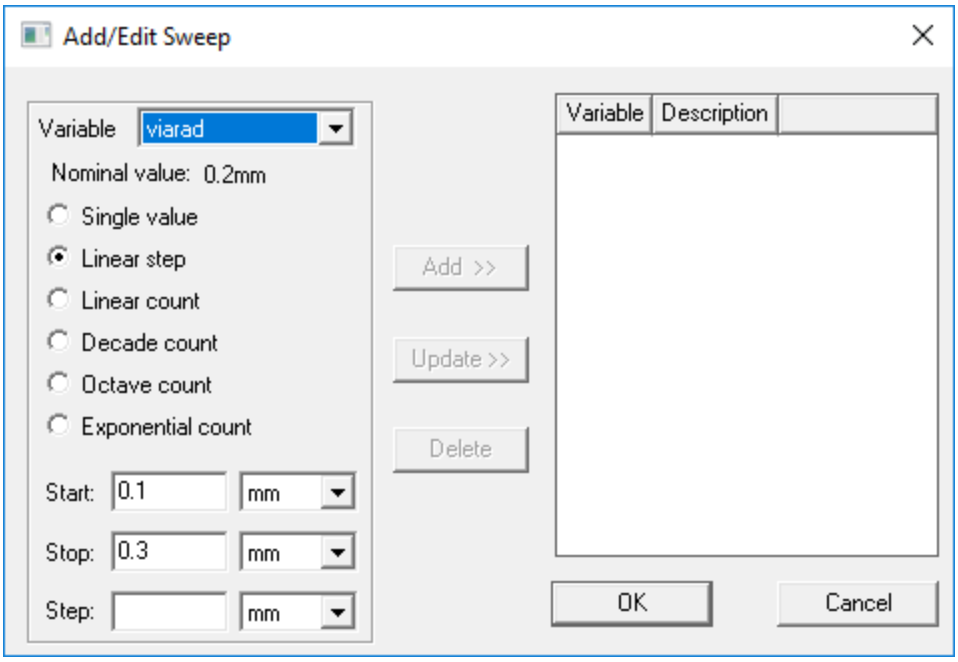
Non-Fixed Variable Sweeps

When setting up a sweep, the **Setup** window contains a list of variables being swept.

To add a non-fixed variable sweep:

1. Click **Add**.

   The **Add/Edit Sweep** window appears.



2. Use the **Variable** drop-down menu to select a non-fixed variable.
3. Select the desired settings and click **Add**.
4. Click **OK** to close the **Add/Edit Sweep** window.
5. Click **OK** to exit the **Setup**.

Orphaned Sweeps

If you create a simulation setup that contains a sweep of a variable, and then subsequently clear the variable's **Sweep** check box, the variable becomes fixed and, as a result, its sweep is "orphaned." The sweep is removed from the setup and the simulation runs as if the sweep did not exist.

- When you close the **Properties** window, a warning message is added to the **Message Window** for each orphaned sweep.

- If you edit an orphaned variable in the **Properties** window and reselect its **Sweep** box, the orphaned sweep will be restored to the setup list.
- If you double-click a setup that contains orphaned sweeps, a pop-up dialog box asks if you want to delete the orphaned sweeps. If you respond **Yes**, all orphaned sweeps are deleted when the **Properties** window is closed. Even if you reselect a **Sweep** check box, its orphaned sweep will NOT be restored. This action is undoable.
- Orphaned sweeps are not written to disk when the project is saved, so once you save a project and close it, any orphaned sweeps are permanently lost.

Choosing a Variable to Optimize

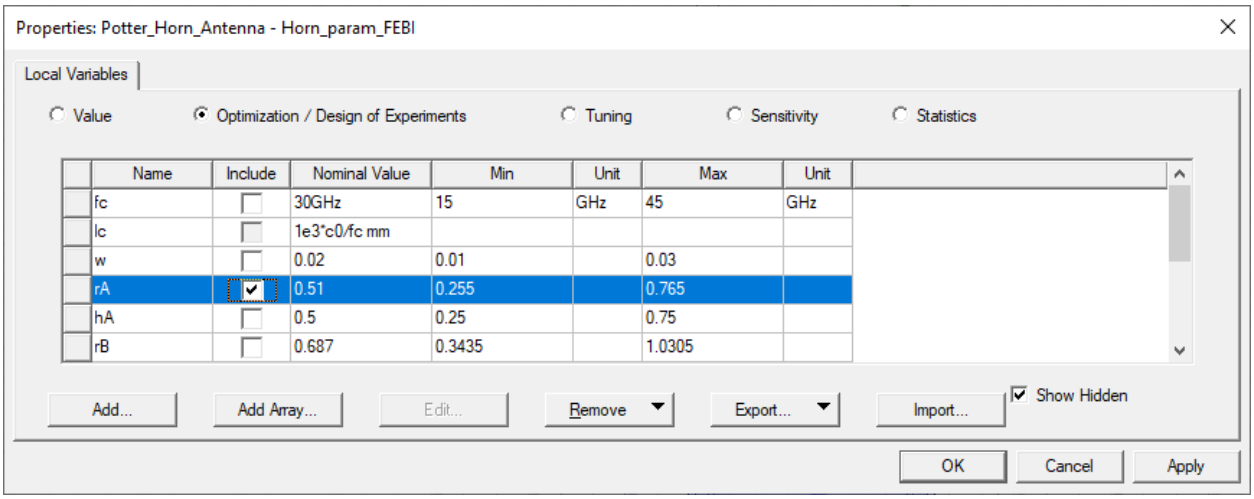
Before a variable can be optimized, you must specify that you intend for it to be used during an optimization analysis in the **Properties** window.

To specify that a variable is to be used during optimization:

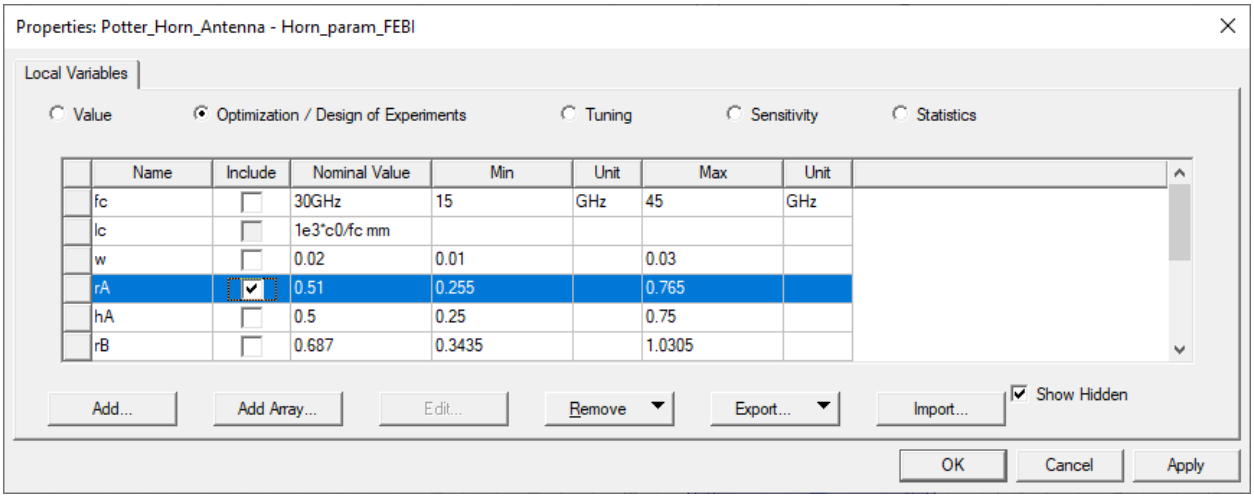
1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Q3D Extractor** or **2D Extractor** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to optimize.
3. Select the **Optimization / Design of Experiments** radio button.
4. Use the **Include** check boxes to select variables for optimization.



> **Important:**
>
> Dependent variables cannot be optimized.
>
> Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.



The selected variable(s) will be available for optimization in an Optimetrics setup defined in the current design or project.

5. If desired, use the **Min** and **Max** fields to override the default minimum and maximum values that Optimetrics will use for the variable in every optimization analysis. During optimization, the optimizer will not consider variable values that lie outside of this range.

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** window.

To specify that a variable is to be tuned:

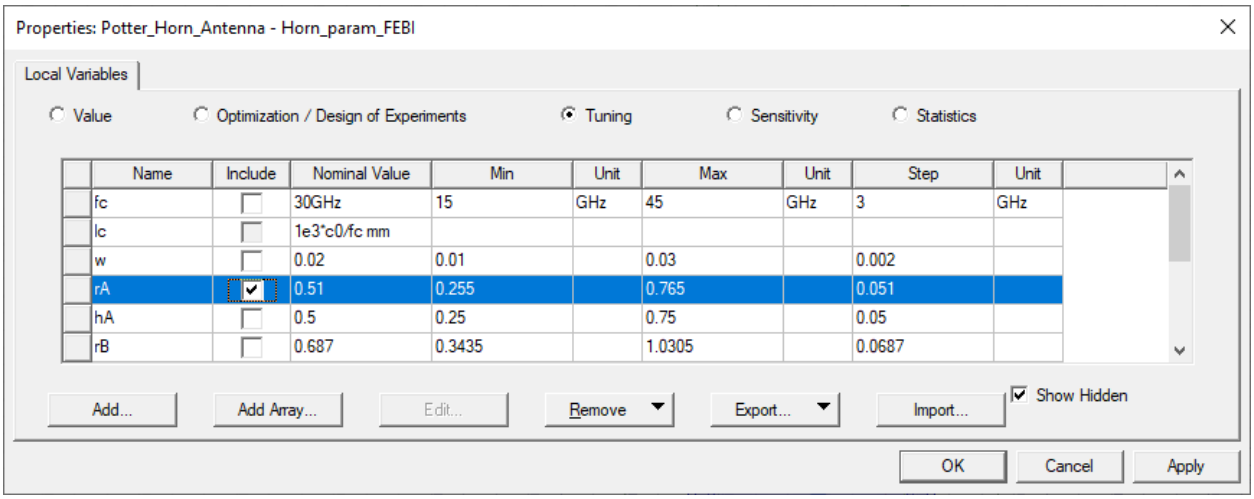
1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Q3D Extractor** or **2D Extractor** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to optimize.
3. Select the **Tuning** radio button.

4. Use the **Include** check boxes to select variables for tuning.

> **Important:**
>
> Dependent variables cannot be tuned.
>
> Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.



The selected variable(s) will be available for tuning.

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** window.

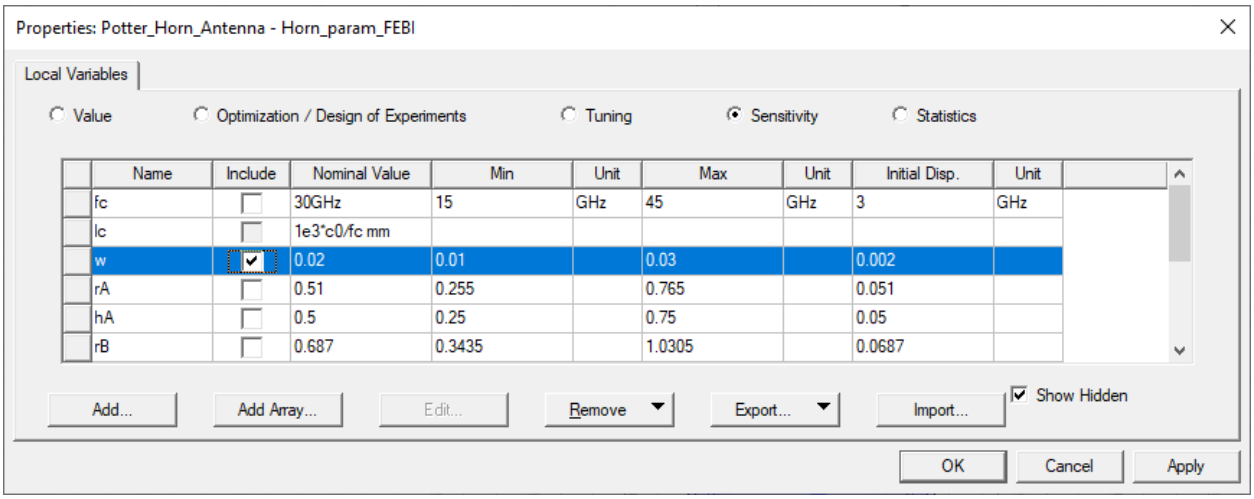
To specify that a variable is to be used during sensitivity analysis:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Q3D Extractor** or **2D Extractor** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to use.
3. Select the **Sensitivity** radio button.

4. Use the **Include** check boxes to select variables for sensitivity analysis.



> **Important:**
>
> Dependent variables cannot be included in a sensitivity analysis.
>
> Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.

The selected variable(s) will be available for sensitivity analysis in the current design or project.

5. If desired, use the **Min** and **Max** fields to override the default minimum and maximum values that Optimetrics will use for the variable in every sensitivity analysis. During analysis, the optimizer will not consider variable values that lie outside of this range.
6. If desired, use the **Initial Disp.** field to override the default initial displacement value that Optimetrics will use for the variable in every sensitivity analysis. During 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.

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** window.

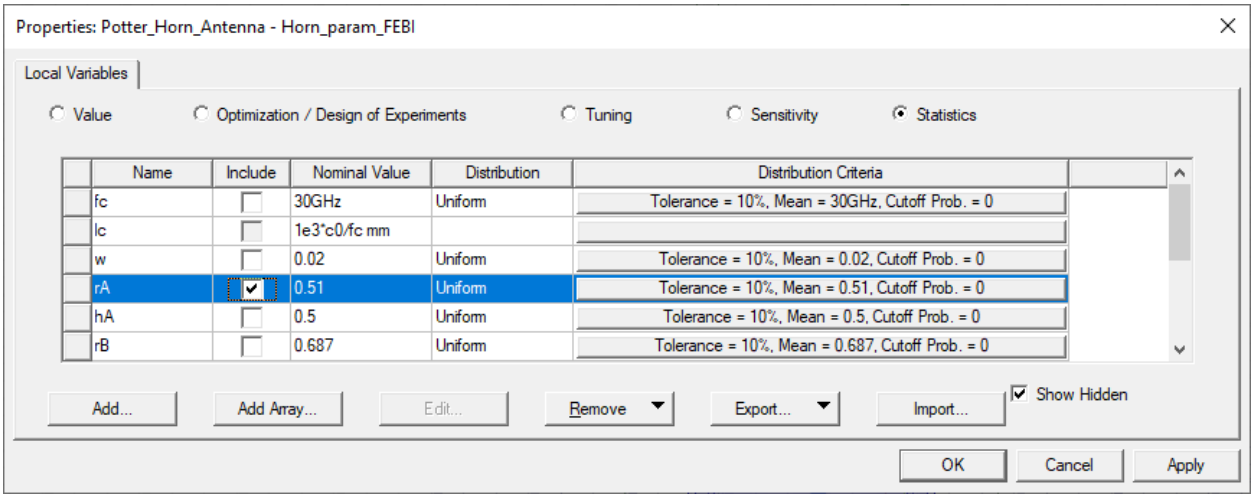
To specify that a variable is to be used during statistical analysis:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Q3D Extractor** or **2D Extractor** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to use.
3. Select the **Statistics** radio button.
4. Use the **Include** check boxes to select variables for statistical analysis.

> **Important:**
>
> Dependent variables cannot be included in a statistical analysis.
>
> Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.



   The selected variable(s) will be available for statistical analysis in the current design or project.

5. If desired, click the **Distribution Criteria** button to override the distribution criteria that Optimetrics will use for the variable in every statistical analysis.

Copying and Pasting a Variables List

From either the Project or Design variables window, you can copy a list of variables and their values to the clipboard. You can then paste them in a plain text, tab-separated format. This is useful for creating documentation.

To copy and paste the variables list:

1. Navigate to the Project Variables or Design Variables **Properties** window.
2. Right-click and select **Copy to Clipboard**.
3. Open a text editor and **paste**.

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

Defining an Expression

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). Defining one variable in terms of another makes it 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** window lists all available pre-defined constants. These 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**.

Valid Operators for Expressions

The operators that can be used to define an expression or function are performed in a sequence.

The following table lists 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
&&	Logical and	8
||	Logical or	8

Using Intrinsic Functions in Expressions

Q3D Extractor 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.

> **Note:**
>
> If units are not specified, 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. To have values interpreted in degrees, supply the argument with the unit name **deg**.

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	Arctangent (in range of -90 to 90 degrees)	atan(x)
atan2	Arctangent (in range of -180 to 180 degrees)	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. (**pwl** can be used with datasets for Design Variables but not for Project variables).	pwl(dataset_exp,variable)
pwl_periodic	Piecewise Linear for 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	Implements smooth interpolation employing weighted impact of all points of the dataset (not just the closest one). See formula below. **Note:** If used with a large 3D dataset, clp function will degrade.	clp (*datasetName* , X,Y,Z)

clp Formula

$$clp\left(Dataset,X,Y,Z\right)=\begin{cases} DatasetValue_i \text{ if } r_i\left(X,Y,Z\right) < r_{rounding} \\ \Sigma_i \frac{DatasetValue_i}{r_i^4} \Big/ \Sigma_i \frac{1}{r_i^4} \quad otherwise \end{cases}$$

$$r_i\left(X,Y,Z\right)=\sqrt{\left(X-DatasetX_i\right)^2+\left(Y-DatasetY_i\right)^2+\left(Z-DatasetZ_i\right)^2}$$

Using Piecewise Linear Functions in Expressions

The following piecewise linear intrinsic functions are accepted in expressions:

- **`pwl (dataset_expression, variable)`** – interpolates along the X-axis and returns a corresponding Y value.
- **`pwlx (dataset_expression, variable)`** – interpolates along the X-axis and returns a corresponding Y value.
- **`pwl_periodic (dataset_expression, variable)`** – also interpolates along the X-axis, but periodically.

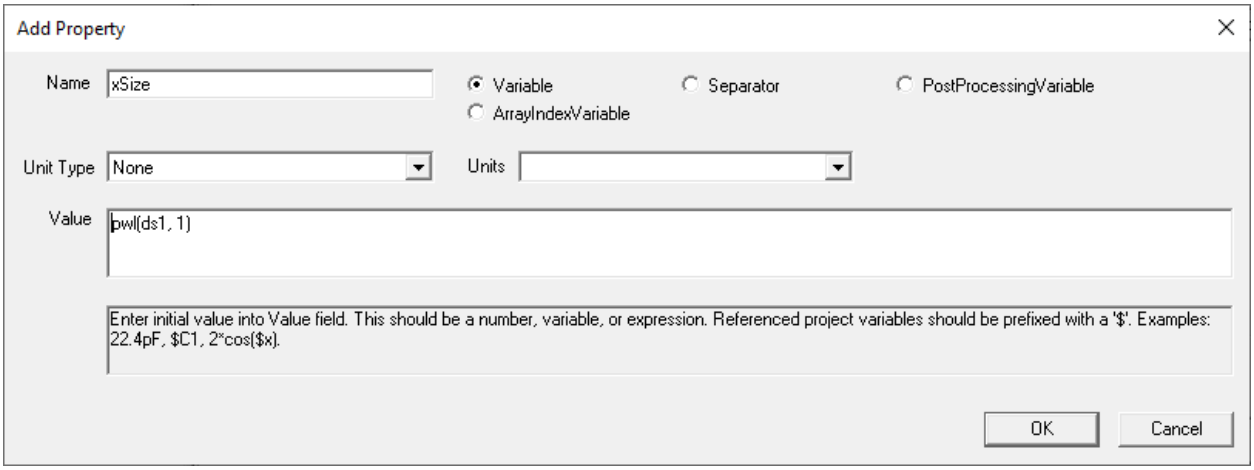
You can use **pwl** in an expression that uses datasets for such things as a frequency dependent material property (See: Adding Datasets).

For example, you can specify BulkConductivity as:

```
pwlx($ds1, Freq)
```

You can create a design variable representing a dimension `xSize` as `pwl(ds1, 1)` where `ds1` is a dataset.

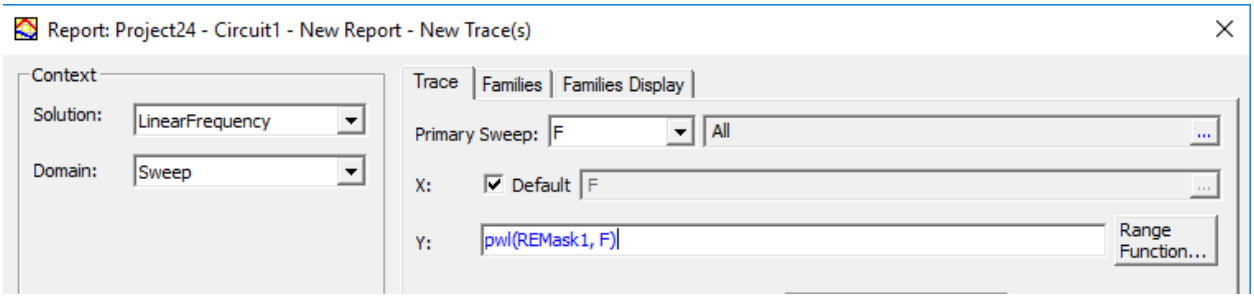
Doing so looks like this:



After the dataset is configured, the **Properties** window shows the new variable.

To use datasets in reports, create a report using **pwl(dataset_expression,**`variable`**)** where `variable` is the primary sweep in the report.

For example:



> **Note:**
>
> pwl can be used with datasets for Design Variables but not for Project variables.

Using Dataset Expressions

A dataset is a collection of data. It can take the following form:

$$\$ds1((x_0,y_0), \ldots, (x_n,y_n))$$

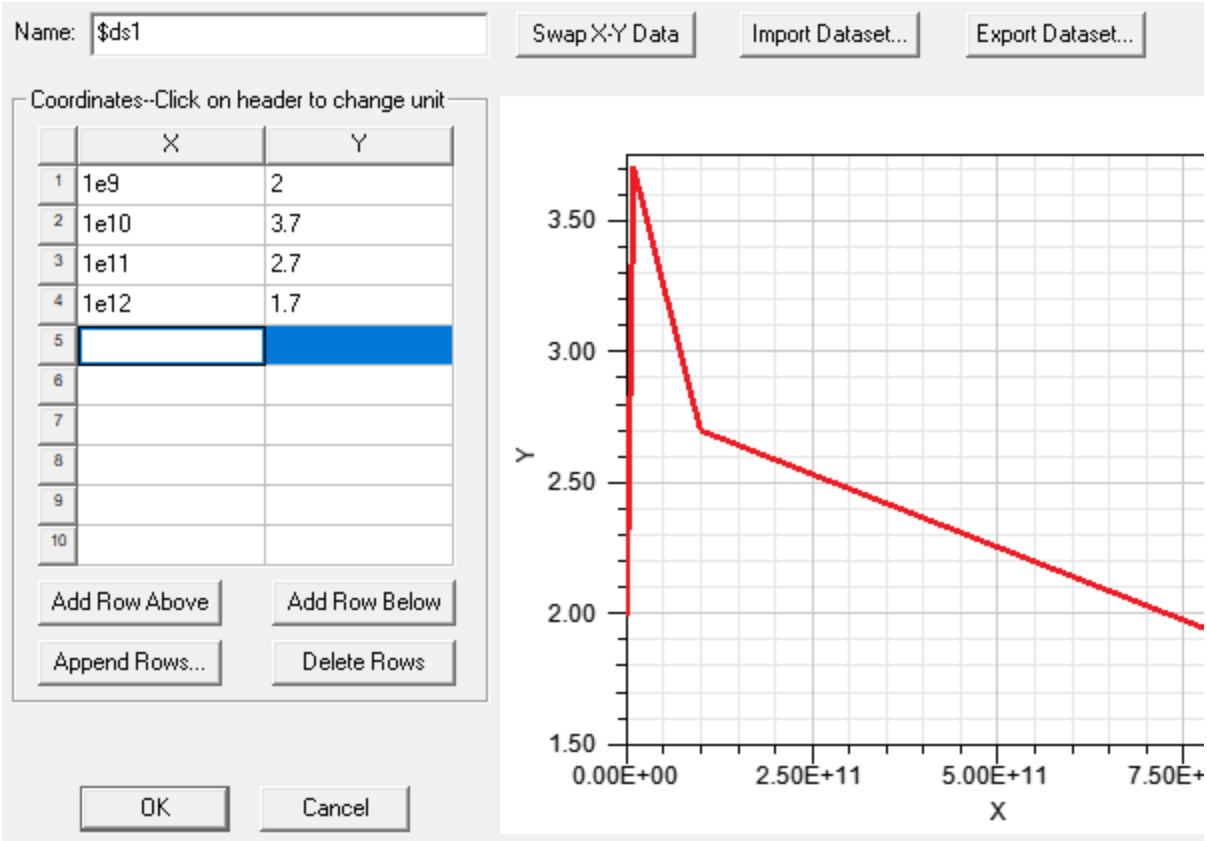
Once created, a dataset (such as $ds1) may be used as the first parameter to piecewise linear (**pwl**, **pwlx**, and **pwl_periodic**) functions, and may also be assigned to variables, in which case the variable may be used as the second parameter to **pwl**, **pwlx**, and **pwl_periodic** functions.

You can generate a dataset using a series of points in a plot on the **Datasets** dialog box (See: Adding Datasets). Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot. This curve, which best fits the segmented plot, consists of the co-ordinates used in the creation of the dataset. The dataset may then be used in the piecewise linear intrinsic functions.

> **Note:**
>
> The following is an example showing how to make a material property frequency dependent using a dataset as the first parameter to a **pwl** function. The values used are arbitrary.

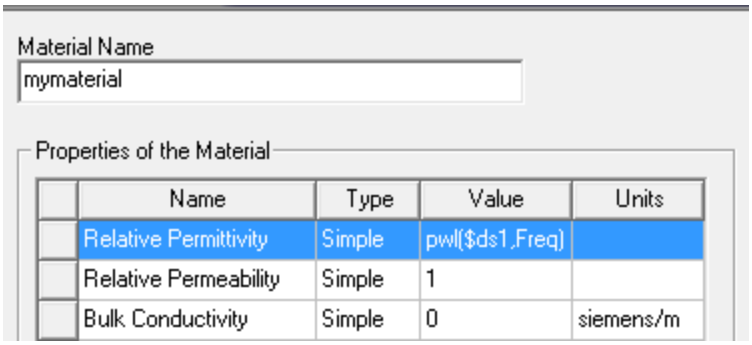
1. For a project-level dataset, click **Project** > **Datasets**. For a design-level dataset, click **Q3D Extractor** or **2D Extractor** > **Design Datasets**.
2. Click **Add**.
3. Set the **Name** field and **Coordinates** as desired and click **OK**.

   The dataset is created.



> **Note:**
>
> By default, the dollar sign ($) is assigned to a project dataset even if you do not use one while naming it.

4. Go to **Tools** > **Edit Libraries** > **Materials** to open the **Edit Libraries** dialog box.
5. Click **Add Material**.
6. Type in the piecewise linear function and use the dataset $ds1, as shown in the following figure.



7. Click **OK**.

Handling Delta Temperature Units in Expressions

When Temperature and Delta Temperature quantities are used as operands to plus or minus operations in an expression, they are handled specially. The biggest difference is an automatic unit change of the resulting values based on the units of the operands.

- When two Temperature quantities are being subtracted (either the first or both operands is/are temperature quantities), the result value has Delta Temperature units.
- When a Delta Temperature quantity is added or subtracted from a Temperature quantity, the result value has Temperature units.
- Adding or subtracting two Delta Temperature quantities results in a quantity with Delta Temperature units.
- Subtracting two temperature quantities is an observable behavior change when compared to earlier releases of Ansys Electronics Desktop.

Temperature Units

Celsius Family

- cel, delta_cel (Legacy name for delta_cel, celdiff continues to be supported)

Kelvin Family

- mkel, delta_mkel (milli Kelvin) (Legacy name for delta_mkel, mkeldiff continues to be supported)
- ckel, delta_ckel (centi Kelvin)
- dkel, delta_dkel (deci Kelvin)
- kel, delta_kel (Legacy name for delta_kel, keldiff continues to be supported)

Farenheit Family

- fah, delta_fah

Temperature-related Use Cases

The high-level legal use cases (in unit types) are:

- Temperature - Temperature = DeltaTemperature
- Temperature +/- DeltaTemperature = Temperature
- DeltaTemperature +/- DeltaTemperature = DeltaTemperature

The minor variations in these use cases specify how the actual units (Celsius is a unit of the Temperature unit type) are handled.

Temperature - Temperature

If they have the same unit, then the Delta Temperature unit will be a matching one.

For example:

- 10 cel - 5 cel = 5 delta_cel
- 100 kel - 90kel = 10 delta_kel
- 10 fah - 1 fah = 9 delta_fah

If they have different units, they are converted to the default units for temperature difference.

For example:

- 10cel - 1 kel = 9 delta_kel

Temperature +/- DeltaTemperature

The resulting temperature quantity will retain the units of the Temperature quantity (first operand)

DeltaTemperature +/- DeltaTemperature

- If they have the same unit, the resulting Delta Temperature quantity will retain that.
- If they have different units, the resulting Delta Temperature quantity will have the default Delta Temperature units
- 5 delta_cel + 10 delta_cel = 15 delta_cel
- 10 delta_cel + 1 delta_kel = 11 delta_kel

All other use cases in a plus or minus arithmetic operation are physically meaningless and revert to previous bahavior, where Electronics Desktop converts quantities to their SI values and then operates on the plain numbers.

- Temperature + Temperature
- Temperature +/- non-Temperature/non-Delta Temperature
- Delta Temperature +/- non-Temperature/non-Delta Temperature
- non-Temperature/non-Delta Temperature +/- Delta Temperature
- non-Temperature/non-Delta Temperature +/- Temperature

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** (log to the base 10), **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 (Cartesian) 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.

4 - Assigning Materials

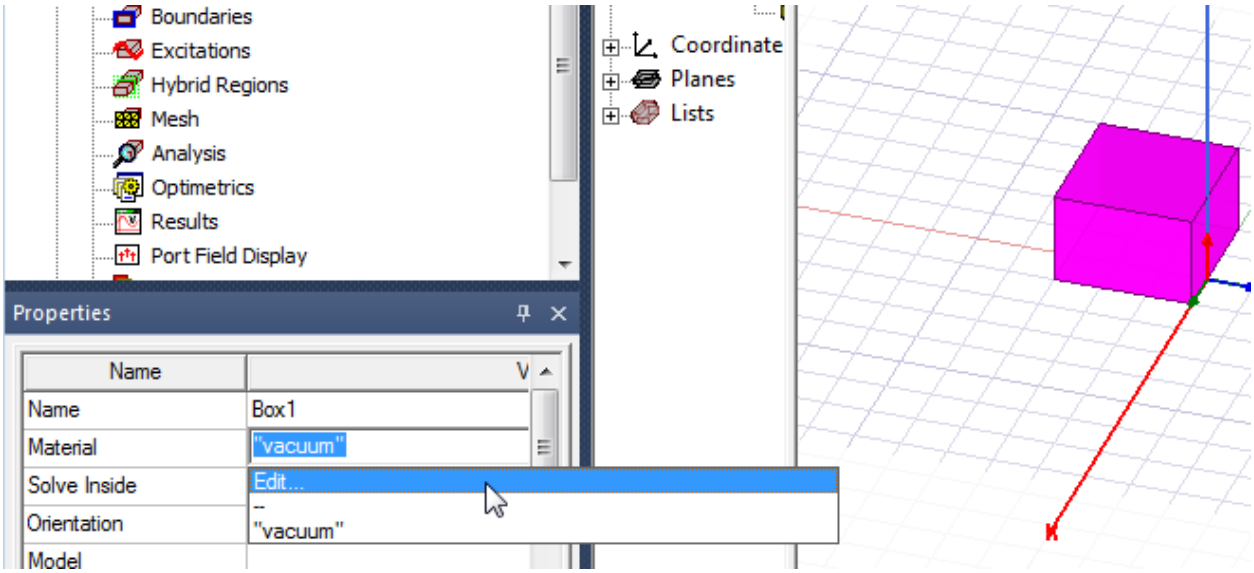
You can add, remove, and edit materials in two main ways:

- Using the **Tools** > **Edit Libraries** > **Materials** menu command.
- Under **Definitions** in the Project Manager, right-click **Materials** and select **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 this general procedure:

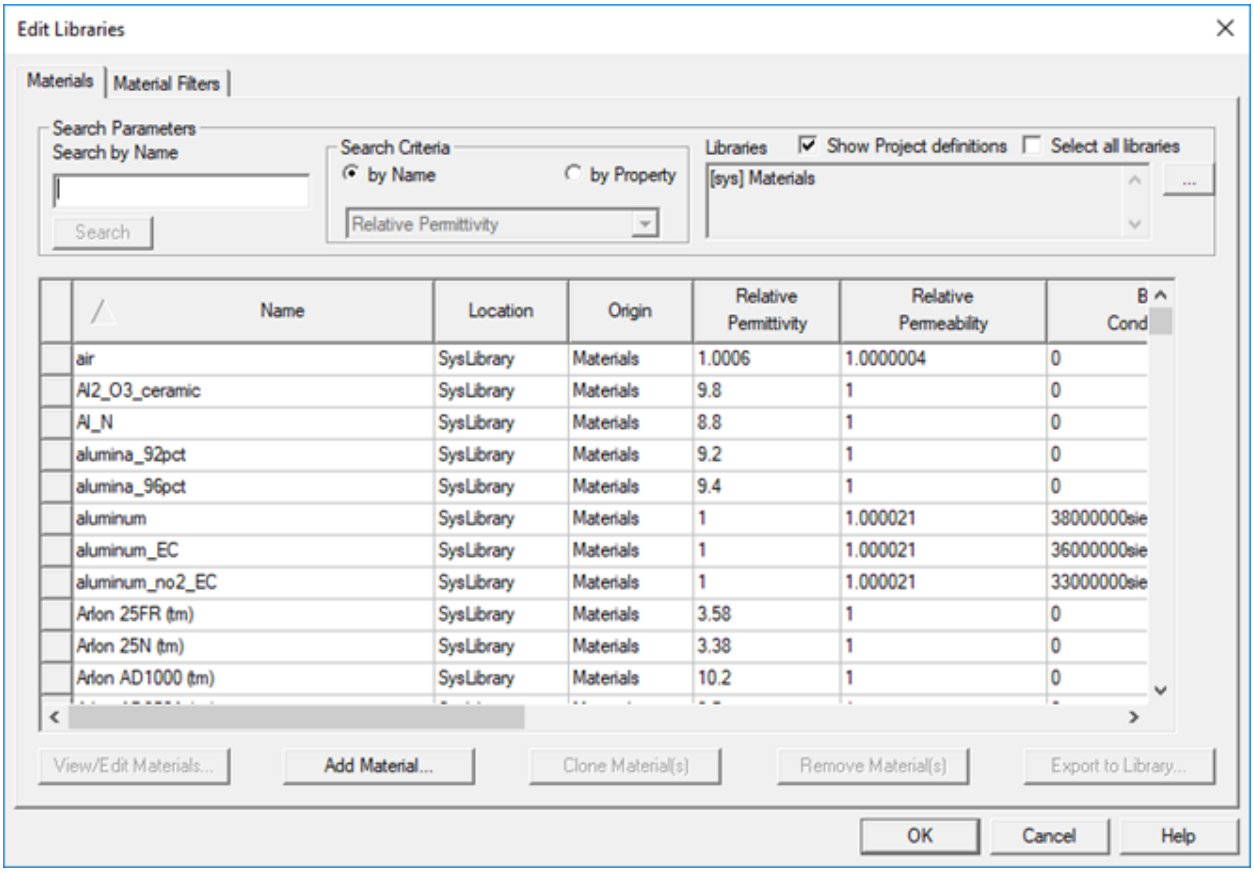
1. Select the object to which you want to assign a material.
2. Click **Modeler** > **Assign Material** or select the Material field in the **Properties** window for the selected object, and select **Edit...** from the drop-down menu:



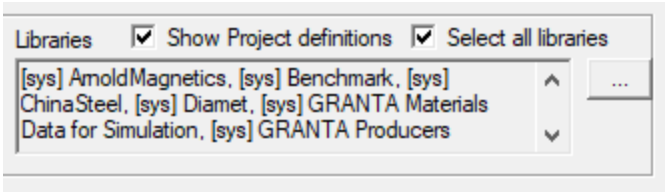
The **Select Definition** window appears. The current material is highlighted, with the Name, Location, Origin library, and parameter values shown.

> **Note:** You can resize the dialog box to show more of the materials and/or material properties without having to use the scroll bars.

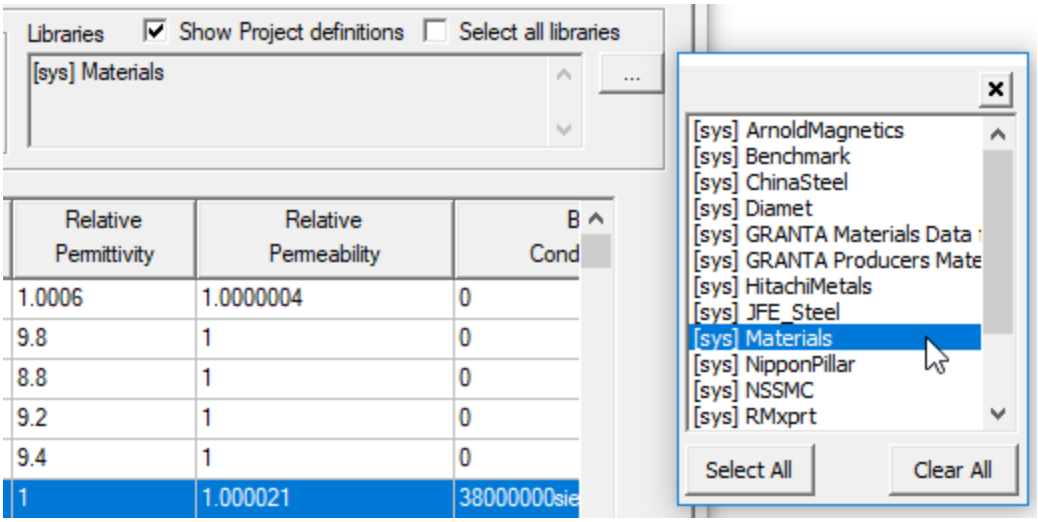
Select Definition Dialog Box - Materials



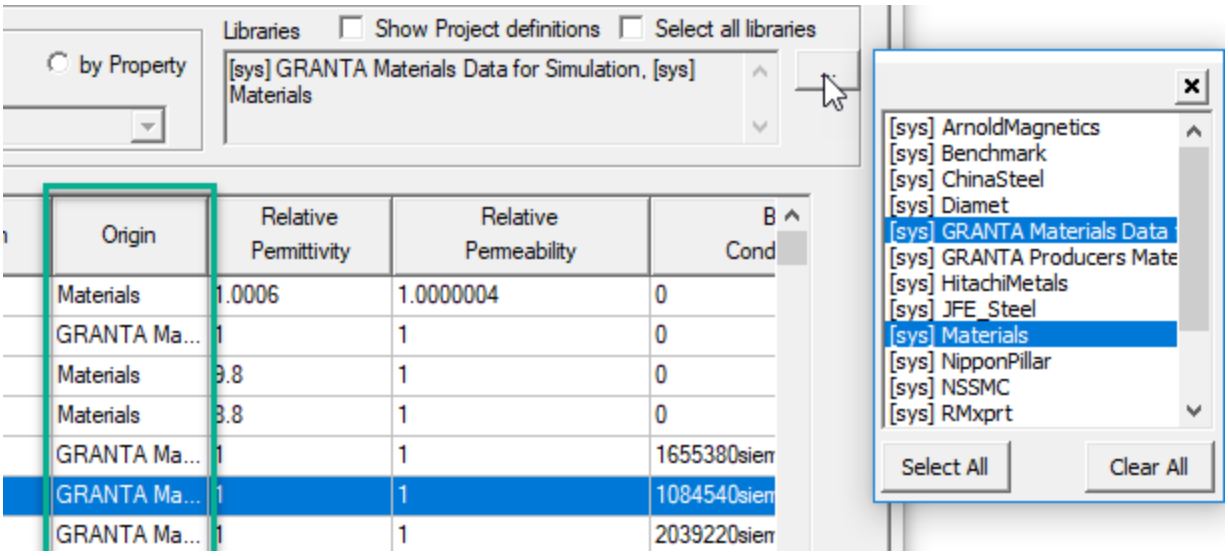
When the **Select all libraries** check box is selected, the window lists all of the materials in Ansys Electronics Desktop's global material library as well as the project's local material library



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.



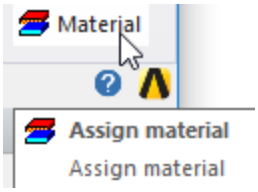
For further information on the materials and their intended uses, you can refer to the published information on materials from those libraries. For example, the Schott materials are described in detail in the Wily Series in Materials for Electronic & Optoelectronic Applications, *Microwave Dielectric Materials and Applications*, edited by M. T. Sebastian, Rick Ubic, and Heli Jantunen, volumes 1 and 2.

GRANTA Materials Data for Simulation (MDS) is an optional, licensed feature, including more than 700 generic materials and over 2100 producer-specific magnetic materials and PCB 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 box 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:

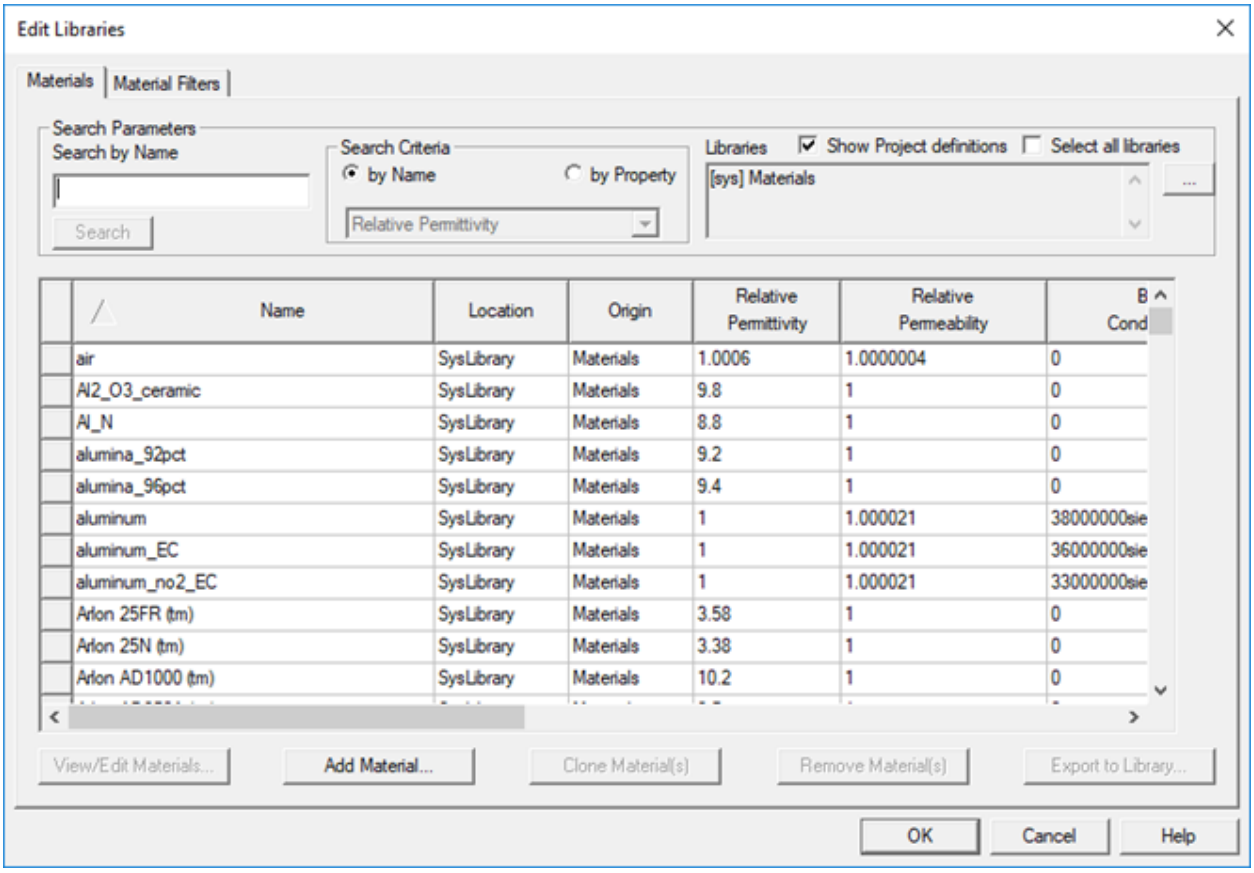
- In the **Properties** dialog box for the object, click the material name under the **Attributes** tab. A drop-down menu shows an **Edit...** button that opens the **Select Definition** window.
- With an object selected in the **Modeler** window, on the **Draw** ribbon, select the **Material** icon.



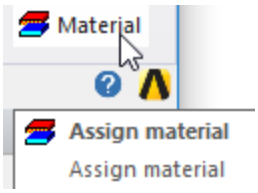
The menu also lists materials included in the current project. Selecting one of these materials provides another way to assign materials to an object.

- 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.

Select Definition Dialog Box - Materials



3. With an object selected in the **Modeler** window, on the **Draw** ribbon, select the **Material** icon.



4. Select a material from the list.

> **Note:**
>
> You can search the listed materials by name or property value.

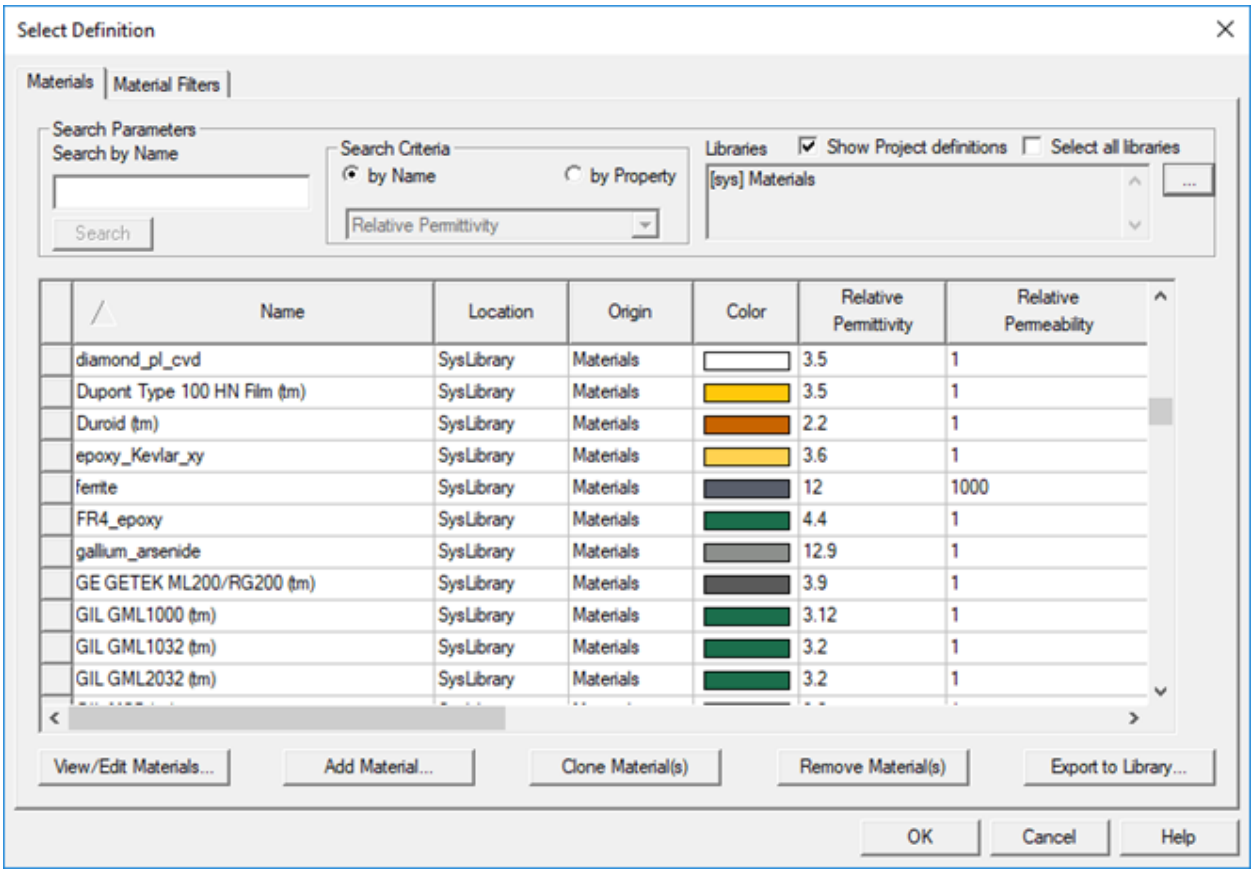
If the material you want to assign is not listed, add a new material to the global or local material library, and then select it.

5. Click **OK**.

   The material you chose is assigned to the object.

> **Note:**
>
> In the History Tree, by default, objects are grouped by material. You can easily change the default grouping. Using the menu bar, click **Modeler** > **History Tree Layout** and clear **Group Objects by Material** along with any other listed option you may wish to disable. You can also perform this action from the **Tree View** drop-down menu on the **View** ribbon tab or from the shortcut menu that appears when you right-click most of the items in the History Tree.

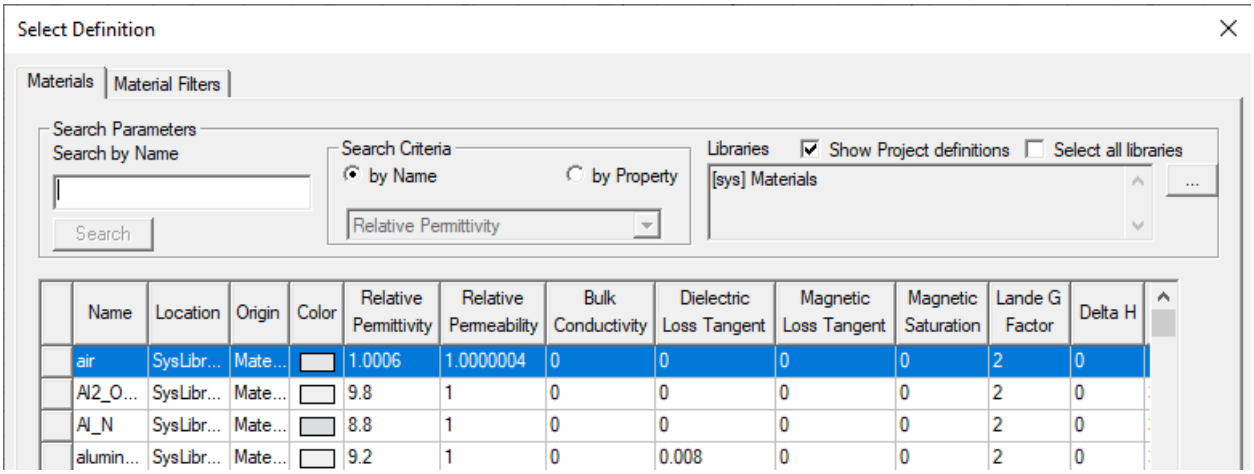
If you check **Show Material Colors** on the Material Filters tab of the **Select Definition** or **Edit Library** dialog box, the **Materials** tab will include a **Color** column showing a color swatch for each listed material.



You can edit the color and transparency values for materials in the View/Edit Material dialog box.

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.

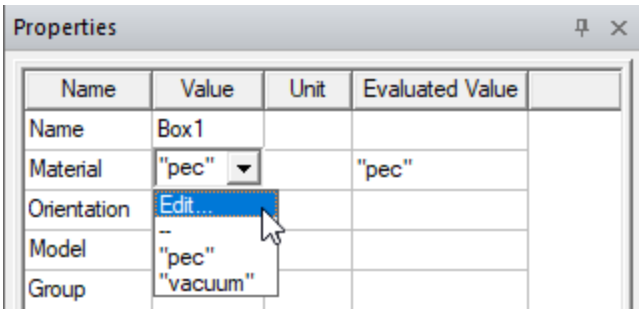


To search for a material **by Name**:

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):

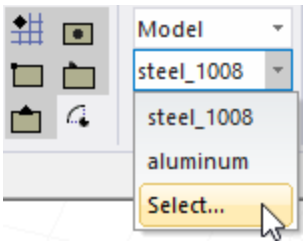
   - Click **Modeler** > **Assign Material**
   - Click the **Material** *value* in the docked **Properties** window, and select **Edit...** from the drop-down menu.

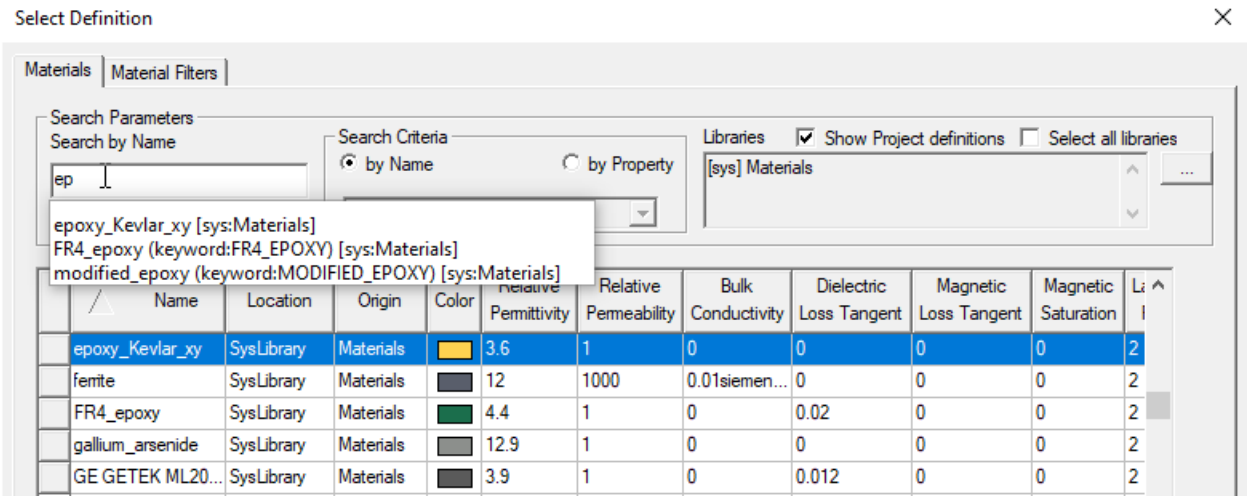
   - On the **Draw** ribbon tab, click **Assign material**.

   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.

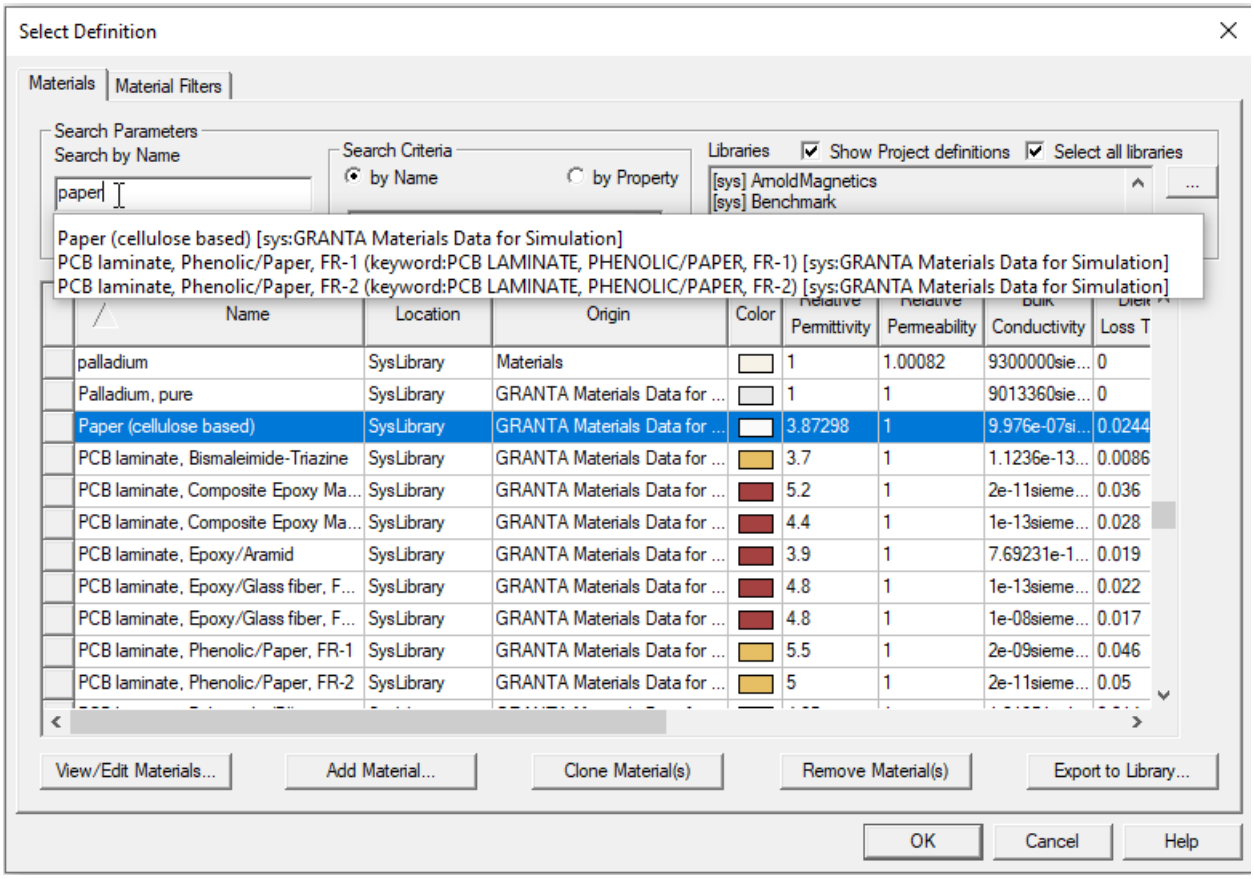
If the selected material is not the one you are searching for, do one of the following:

- 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

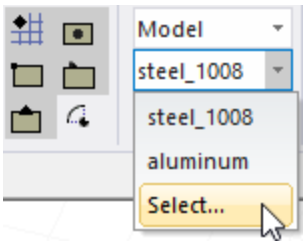
1. Access the **Select Definition** dialog box using one of the following methods:

   With one or more objects selected (that is, to assign a material to selected objects):

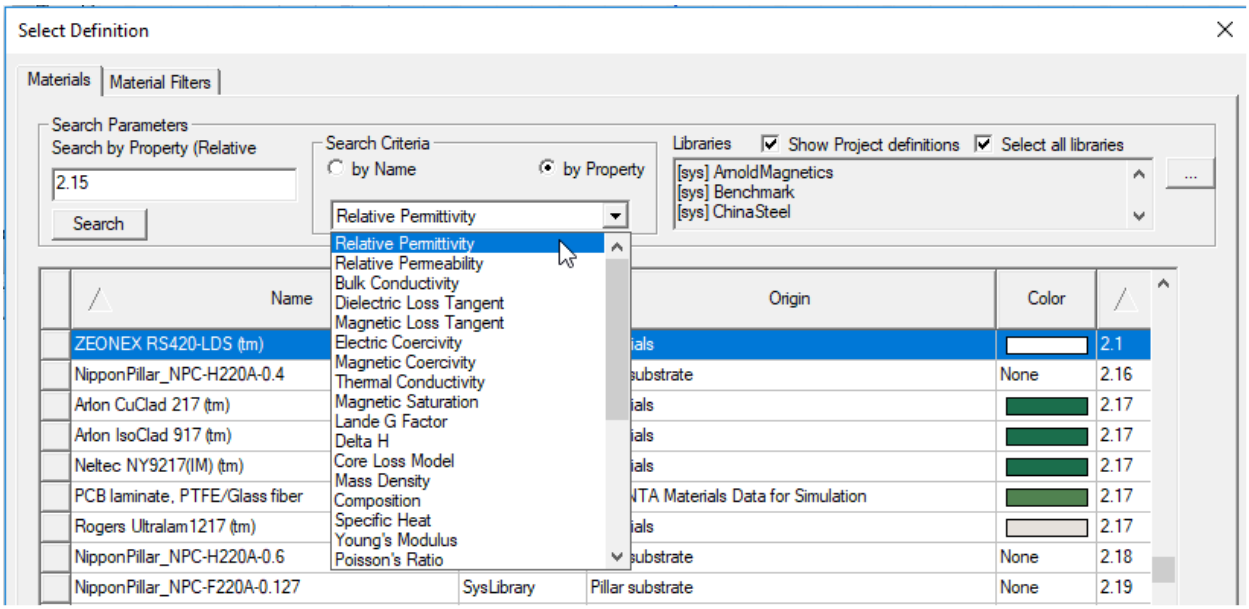
   - Click **Modeler** > **Assign Material**
   - Click the **Material** *Value* in the docked *Properties* window, and select **Edit...** from the drop-down menu.
   - On the **Draw** ribbon tab, click **Assign material**.

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, 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 in the table, though all properties are available in this drop-down menu. To view the complete table of properties, see Filtering Materials.

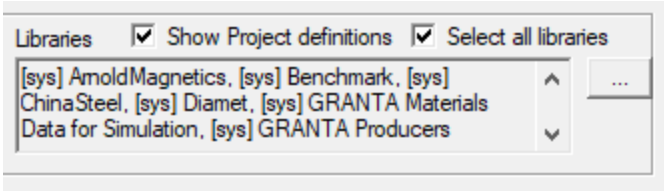
4. In the **Search Parameters** area, type a numerical value in the **Search by Property** text box and then click **Search**.

   The materials are sorted according to the property you selected. Additionally, the material with the property value closest to the one you typed, but without exceeding it, 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 **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.



Adding New Materials

You can add a new material to a project or 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** window or the **Edit Libraries** window.

To open the **Select Definition** window:

- Click **Modeler** > **Assign Material**.

  The **Select Definition** window appears.

To open the **Edit Libraries** window:

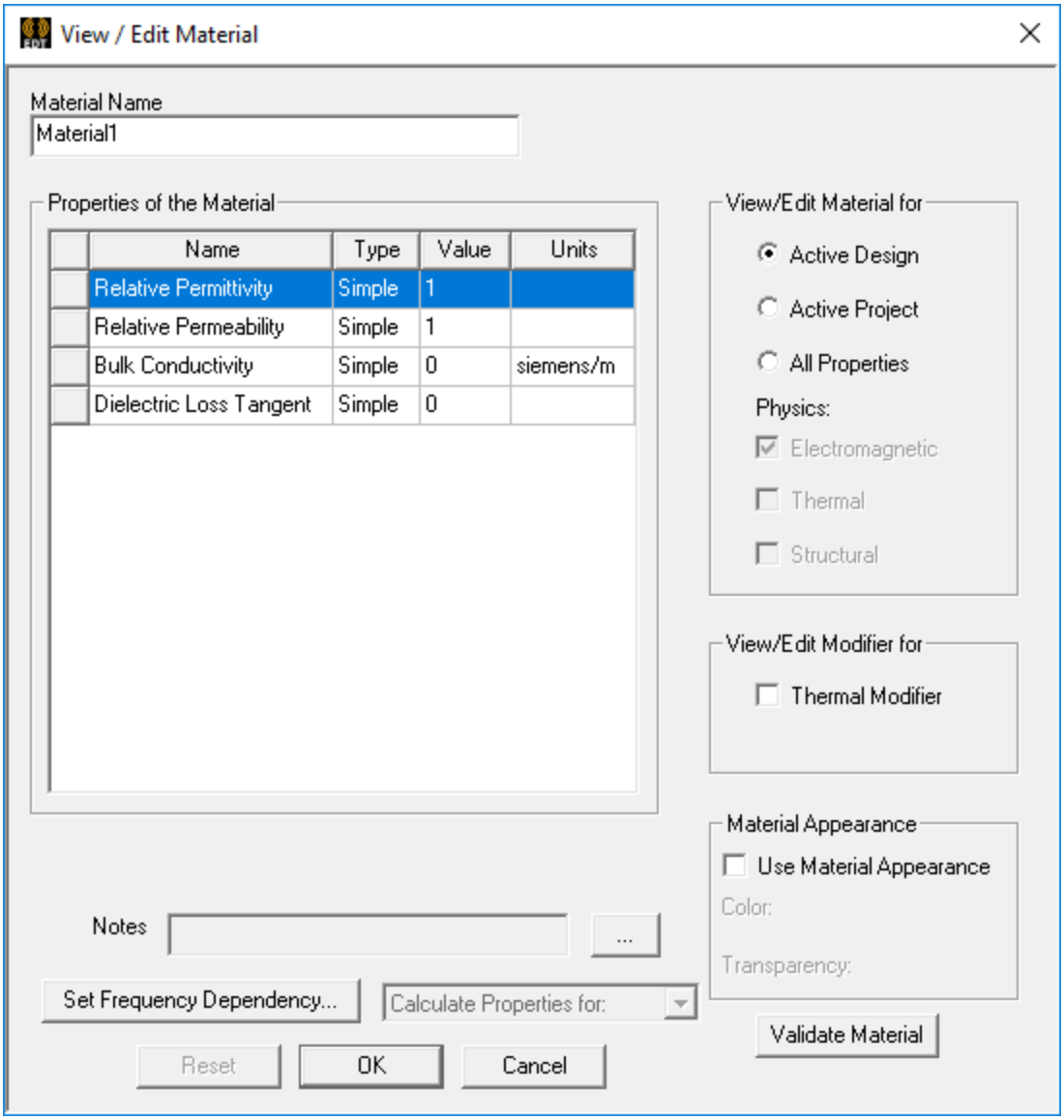
- Click **Tools** > **Edit Libraries** > **Materials**. In the project tree, you can also right-click **Materials**, and select **Edit Library**.

  The **Edit Libraries** window appears.

To add a new material:

1. From either the **Select Definition** window or the **Edit Libraries** window, click **Add Material**.

   The **View/Edit Material** dialog box appears.



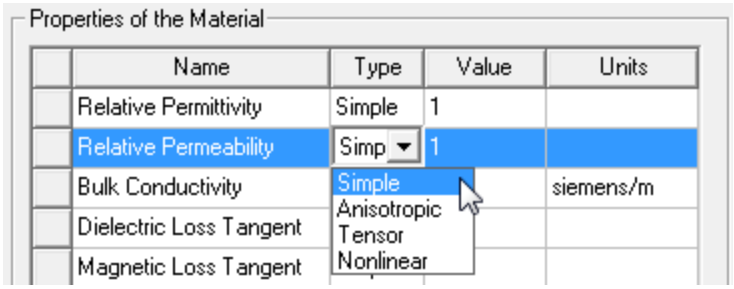
By default, only properties commonly used by the selected product are displayed. To view the complete table of properties, see Filtering Materials.

2. Type a name for the material in the **Material Name** text box, or accept the default.
3. Use the radio buttons in the **View/Edit Material for** section to specify whether the new materials apply to Active Design, Active Project, or All Properties. When **All Properties** is selected, **Physics** classification options are enabled to show or hide properties based on simulation type (Electromagnetic, Thermal, or Structural).

> **Note:**
>
> If a material is edited in a design type for which the **Physics** type has not been set (e.g., an HFSS design but Electromagnetic physics type was not set), the **Physics** type will be automatically set in the material.

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.

4. Enter a **Material Name**, or accept the default.
5. For each available material property, select a **Type** from the drop-down menu. Only applicable types appear for each material property. Some properties only use the Simple type, while others offer four or more types.

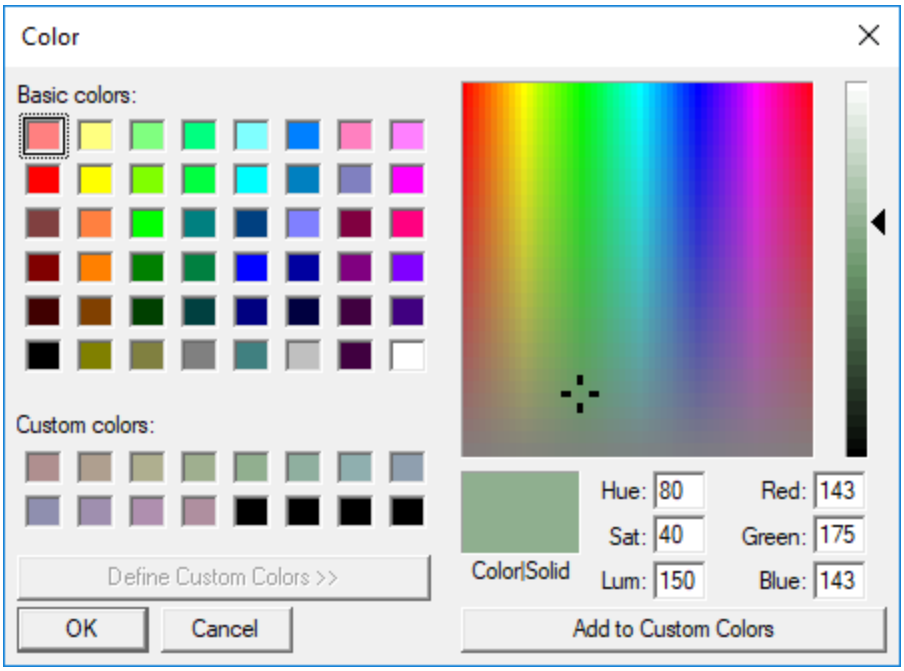


6. If the material is *linear*, enter a **Value** for the following material properties:
   - Relative Permeability
   - Relative Permittivity
   - Bulk Conductivity
   - Dielectric Loss Tangent
   - Magnetic Loss Tangent

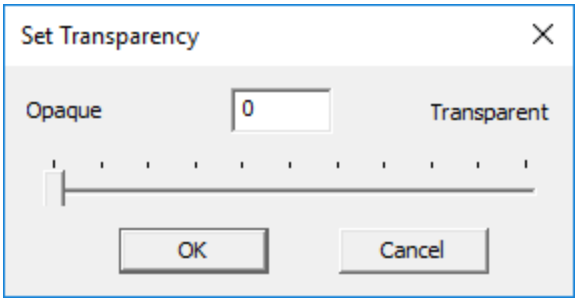
   If the material is a *ferrite*, enter a value greater than 0 in the Magnetic Saturation **Value** box. You may also choose to enter values in the Lande G Factor and Delta H **Value** boxes. Because Delta H values are measured at specific frequencies, you should also enter a - Measured Frequency value (default 9.4 GHz).

> **Note:**
>
> You may enter a variable name or mathematical expression in the **Value** box.

7. If one or more of the material properties are dependent on frequency, click **Set Frequency Dependency**, and then follow the directions for defining frequency dependent materials.
8. To modify the units for a material property, double-click the **Units** box and select a new unit system.
9. For Material Appearance, you can check the box to enable the fields for you to specify a color and transparency. Clicking the color bar opens a color selection window:

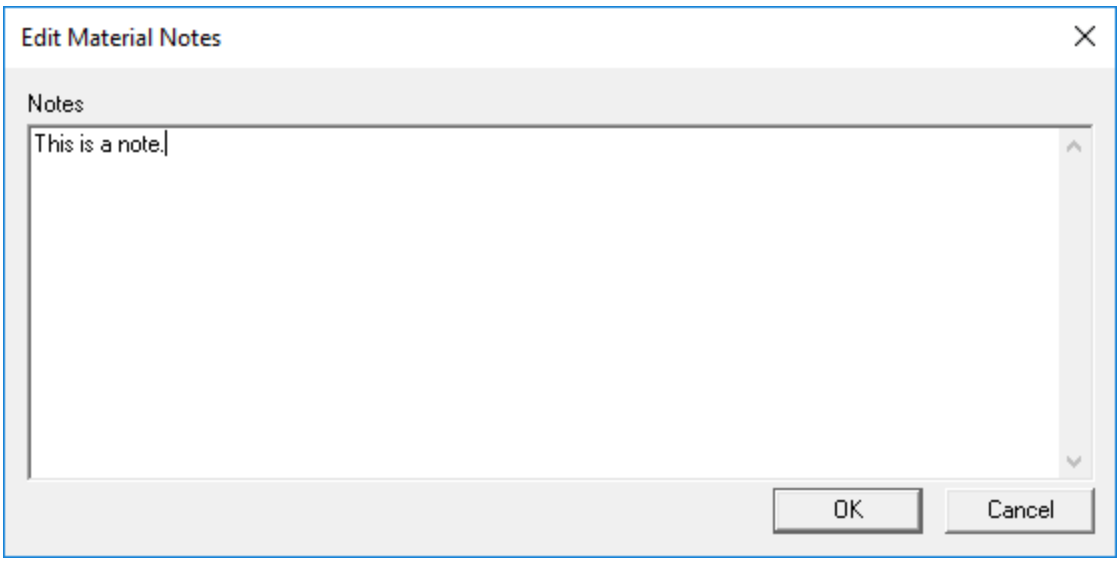


10. Clicking the Transparency box opens a Transparency dialog box with a text field and slider bar for selection.



11. Click **OK**.
12. To modify the units for a material property, double-click the **Units** box and select a new unit system.
13. Click **OK**.

14. If you want to add descriptive notes for the new material, click the ellipsis button [...] next to the **Notes** field. This opens a dialog box in which you can enter text.



15. Click **OK** to add the Notes.
16. Click **OK** on the **View/Edit Materials** dialog to add the new material to the material library.

Assigning Material Property Types

Material properties can be assigned using the **View/Edit Material** dialog box, using one of the following material property types, based on the applicability to the material property:

Simple	The material is homogeneous and linear.
Tensor	A full 3x3 matrix general tensor, which may or may not be symmetrical.

Select a material property type for each property from the **Type** drop-down menu. Some properties only use the Simple type

Defining Variable Material Properties

When defining or modifying a material's properties, each material property value in the **View/Edit Material** dialog box 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 **Relative Permittivity**

Value box for the material. Be sure to include the prefix **$** before the project variable name, which notifies the software 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.

Defining Frequency-Dependent Material Properties

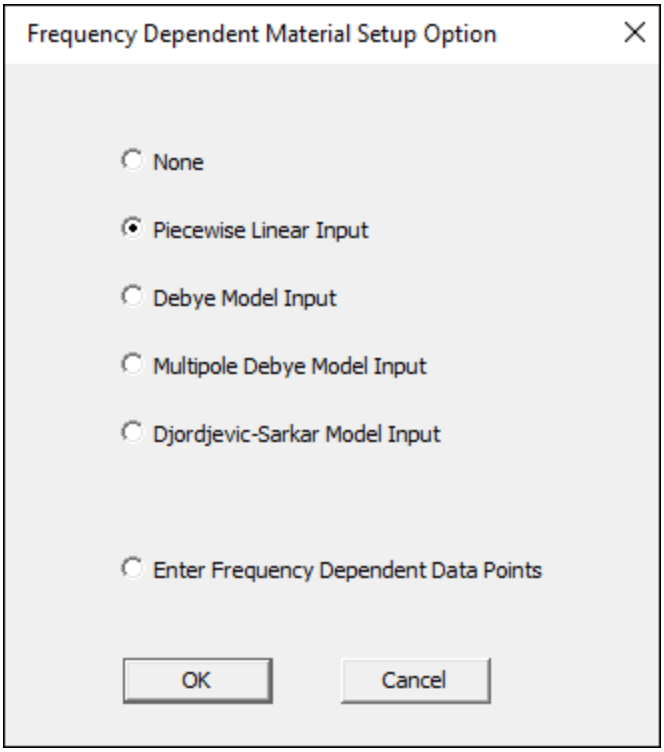
Q3D Extractor provides several frequency-dependent material models. The Piecewise Linear and Frequency Dependent Data Points models apply to both the electric and magnetic properties of the material. However, they do not guarantee that the material satisfies causality conditions, and so they should only be used for frequency-domain applications.

The Debye and Djordjevic-Sarkar models apply only to the electrical properties of dielectric materials. These models satisfy the Kramers-Kronig conditions for causality, and so are preferred for applications (such as TDR or Full-Wave Spice) where time-domain results are needed. The Design Settings also include an automatic Djordjevic-Sarkar model to ensure causal solutions when solving frequency sweeps for simple constant material properties.

Assigning Frequency-Dependent Properties

1. From the **Select Definition** window, select a material and click **View/Edit Material**.
2. Click **Set Frequency Dependency**.

   The **Frequency Dependent Material Setup Option** dialog box appears.



The options are:

- **Piecewise Linear Input** – defines the material property values as a restricted form of piecewise linear model with exactly 3 segments (flat, linear, flat). You specify the property's values at an upper and lower corner frequency. Between these corner frequencies, Q3D Extractor linearly interpolates the material properties; above and below the corner frequencies, Q3D Extractor extrapolates the property values. This dataset can be modified with additional points if desired.
- **Debye Model Input** – a single-pole model for the frequency response of a lossy dielectric material. In some materials, up to about a 10-GHz limit, ion and dipole polarization dominate and a single pole Debye model is adequate. Q3D Extractor allows you to specify an upper and lower measurement frequency, and the loss tangent and relative permittivity values at these frequencies. You may optionally enter the permittivity at optical frequency, the DC conductivity, and a constant relative permeability.
- **Multipole Debye Model Input** – allows you to provide the data of relative permittivity and loss tangent versus frequency. Based on this data, the software dynamically generates frequency-dependent expressions for relative permittivity and loss tangent through the Multipole Debye Model. The input dialog plots these expressions together with your input data through the linear interpolations. The generated expressions provide the new value for the material properties of relative

permittivity and loss tangent. Both the expressions and data triples can be saved and reloaded.

- **Djordjevic-Sarkar Model Input** – for low-loss dielectric materials (particularly FR-4) commonly used in printed circuit boards and packages. In effect, it uses an infinite distribution of poles to model the frequency response, and in particular the nearly constant loss tangent, of these materials. This allows you to enter the relative permittivity and loss tangent at a single measurement frequency. You may optionally enter the relative permittivity and conductivity at DC.
- **Enter Frequency Dependent Data Points** – allows you to enter, import or edit frequency dependent data sets for each material property. Any number of data points may be entered. This is an arbitrary piecewise linear model.

3. Select the input type and click **OK**.

   An input type-specific dialog box appears, based on your selection.

   Piecewise Linear Frequency Dependent Material Input, Debye Model Input, and Djordjevic-Sarkar Model Input remember the values previously used and also include plots to show the property curves in real time as changes to the input are made. Input values for each are saved as material attached data for the material being edited. These data items are saved with materials when they are written into a project file or exported to a material library. Note that when a frequency-dependent setup method is used and the values are pre-populated with saved data, the dialog box title will have "(Update)" appended.

After you have entered the data for your selection, you return to the **View/Edit Material** window. New default function names appear in the material property text boxes. Q3D Extractor automatically creates a dataset for each material property. Based on a varying property's dataset, Q3D Extractor can interpolate the property's values at the desired frequencies during solution generation.

To modify the dataset with additional points, see *Editing Datasets*.

> **Note:**
>
> Neither the piecewise nor the loss model asks for frequency-dependent conductivity because the constant sigma represents DC loss and the frequency-dependent loss tangent represents polarization losses.

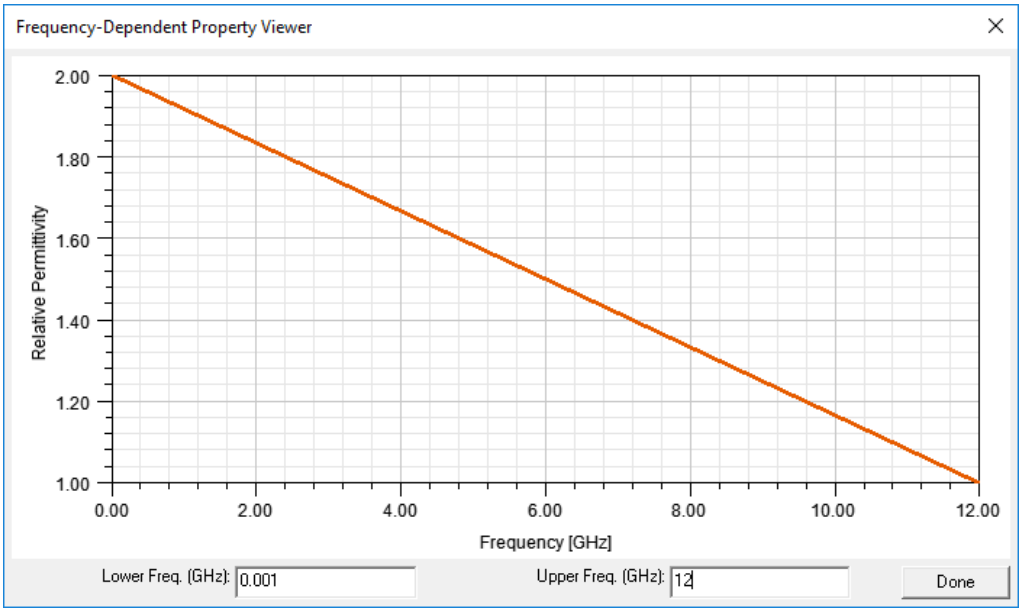
Frequency Dependence Visualization

When you view or edit material properties, it is important to have a sense of how properties may vary with frequency. Frequency-dependent properties come in a variety of forms, ultimately resulting in some value expression or dataset. Plots of properties as a function of frequency are available through the **View/Edit Materials** window.

To view the plot:

1. From the **Select Definition** window, select a material and click **View/Edit Material**.
2. Right-click a property (for example, relative permittivity or dielectric loss tangent), and select **View Property vs. Frequency**.

   The **Frequency-Dependent Property Viewer** appears, displaying the plot.



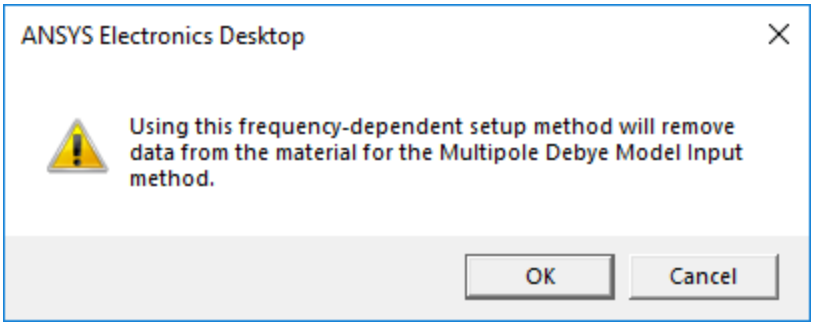
The window contains boxes where you can edit the frequency range for the plot. If the property was set by one of the nonlinear frequency setup methods, the frequency range will be derived from the data for that method, and edits to the lower/upper frequencies are not saved. Otherwise, the frequency range lower/upper frequency limit defaults are stored in the registry, and are updated if you modify the values. If values are not yet stored in the registry, the range defaults to 1MHz-10GHz. Note that **the View Property vs. Frequency** menu option is not displayed for choice properties because frequency-dependence doesn't apply to those.

Saved Input Data Invalidation For Frequency-Dependent Setup

You will be prompted with a warning in two circumstances:

- Data associated with one of the frequency-dependent setup methods is attached to the material definition, and a property which would be set by this method is modified.
- Data associated with one of the frequency-dependent setup methods is attached to the material definition, and a different setup method is subsequently used.

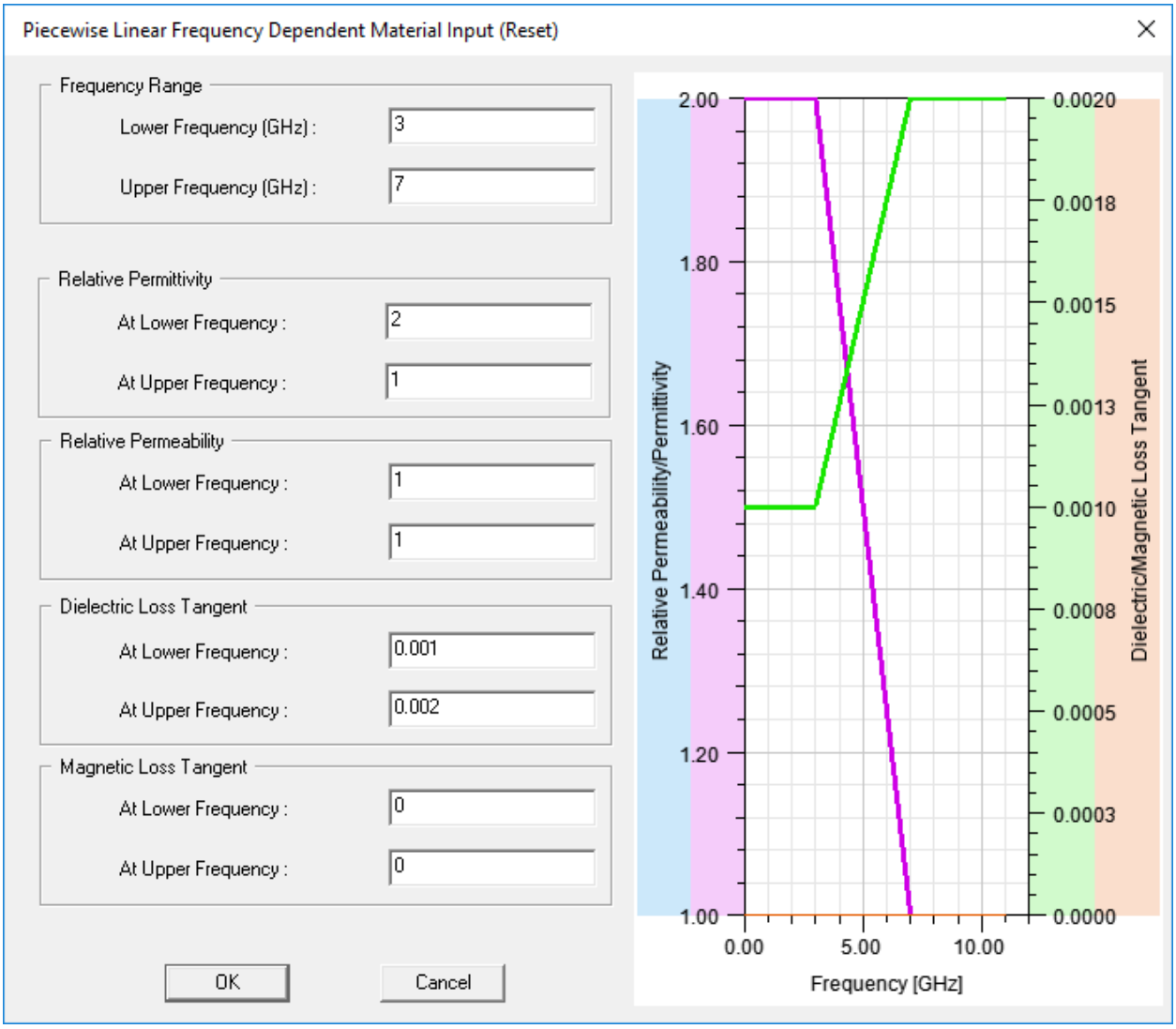
You can choose **OK** to continue with the edit and remove the invalidated setup data, or choose **Cancel** to discard any changes.



Assigning Frequency Dependent Material: Piecewise Linear Input

When you select **Piecewise Linear Input** as the model for the frequency dependent material property, the **Piecewise Linear Frequency Dependent Material Input** window appears.

From this window, you can enter material property values and view a property vs. frequency plot.



Use the fields to enter lower and upper frequency values for:

- **Frequency Range** – Q3D Extractor assumes that the material's property values remain constant between these frequencies.
- **Relative Permittivity** – If the permittivity of the material does not vary with frequency, enter the same value you entered for the Frequency Range Lower Frequency.
- **Relative Permeability** – If the permeability of the material does not vary with frequency, enter the same value you entered for the Frequency Range Lower Frequency.
- **Dielectric Loss Tangent** – If the dieletric loss tangent of the material does not vary with frequency, enter the same value you entered for the Frequency Range Lower Frequency.
- **Magnetic Loss Tangent** – If the magnetic loss tangent of the material does not vary with frequency, enter the same value you entered for the Frequency Range Lower Frequency.

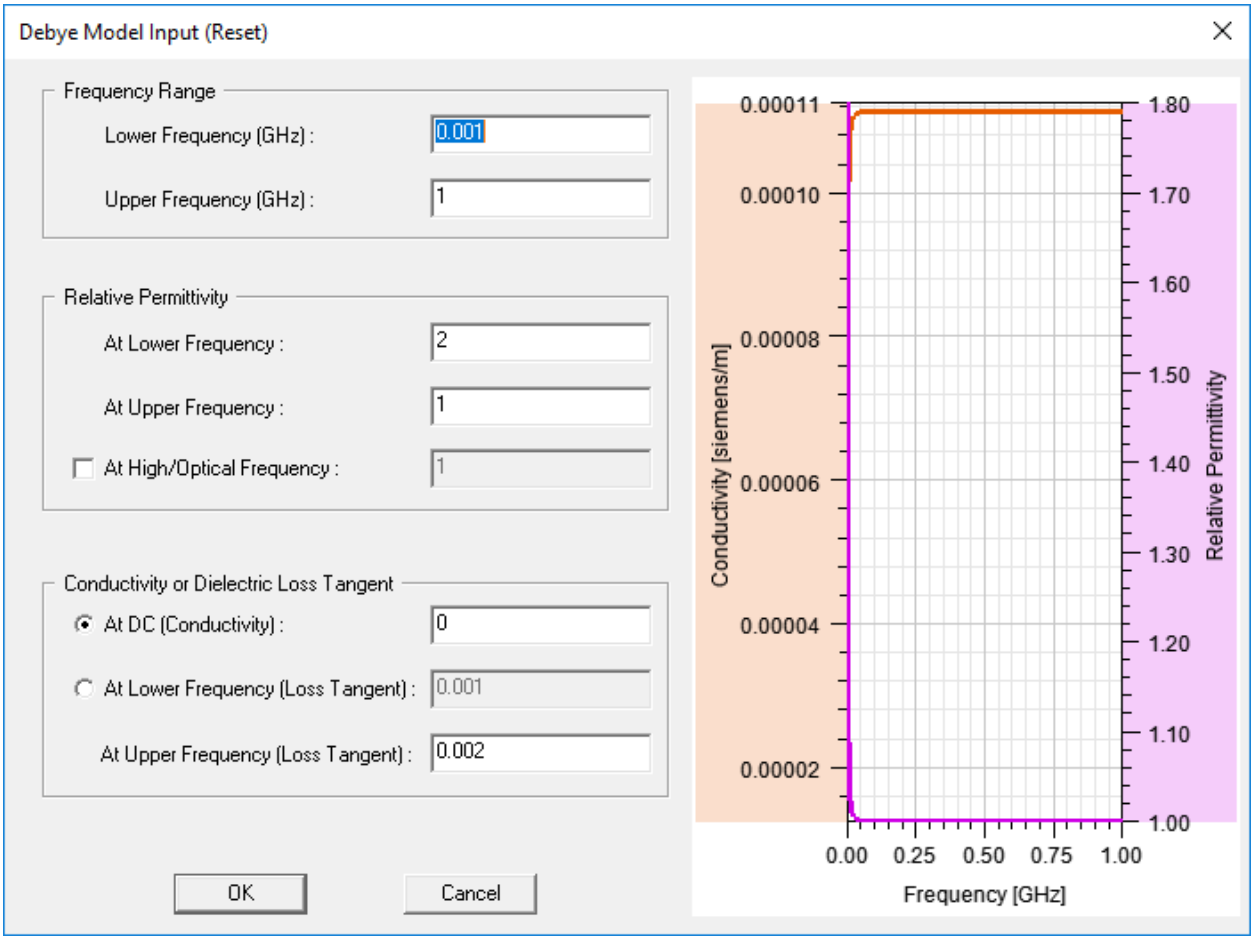
> **Note:**
>
> Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

To modify the dataset with additional points, see Editing Datasets.

Assigning Frequency Dependent Material: Debye Model Input

When you select **Debye Model Input** as the model for the frequency dependent material property, the **Debye Model Input** window appears.

From this window, you can enter material property values and view a property vs. frequency plot.



Use the fields to enter lower and upper frequency values for:

- **Frequency Range** – Q3D Extractor assumes that the material's property values remain constant between these frequencies.
- **Relative Permittivity** – If the permittivity of the material does not vary with frequency, enter the same value you entered for the Frequency Range Lower Frequency. If you need to specify a value for At **High/Optical Frequency**, select the check box and enter a value.
- **Conductivity or Dielectric Loss Tangent** – Select either **At DC** to specify conductivity, or **At Lower Frequency** to specify Loss Tangent.

After you have entered the data for your selection, click **OK** to return to the **View/Edit Material** window. New default function names appear in the material property text boxes. A dataset is automatically created for each material property. Based on a varying property's dataset, Q3D Extractor can interpolate the property's values at the desired frequencies during solution generation.

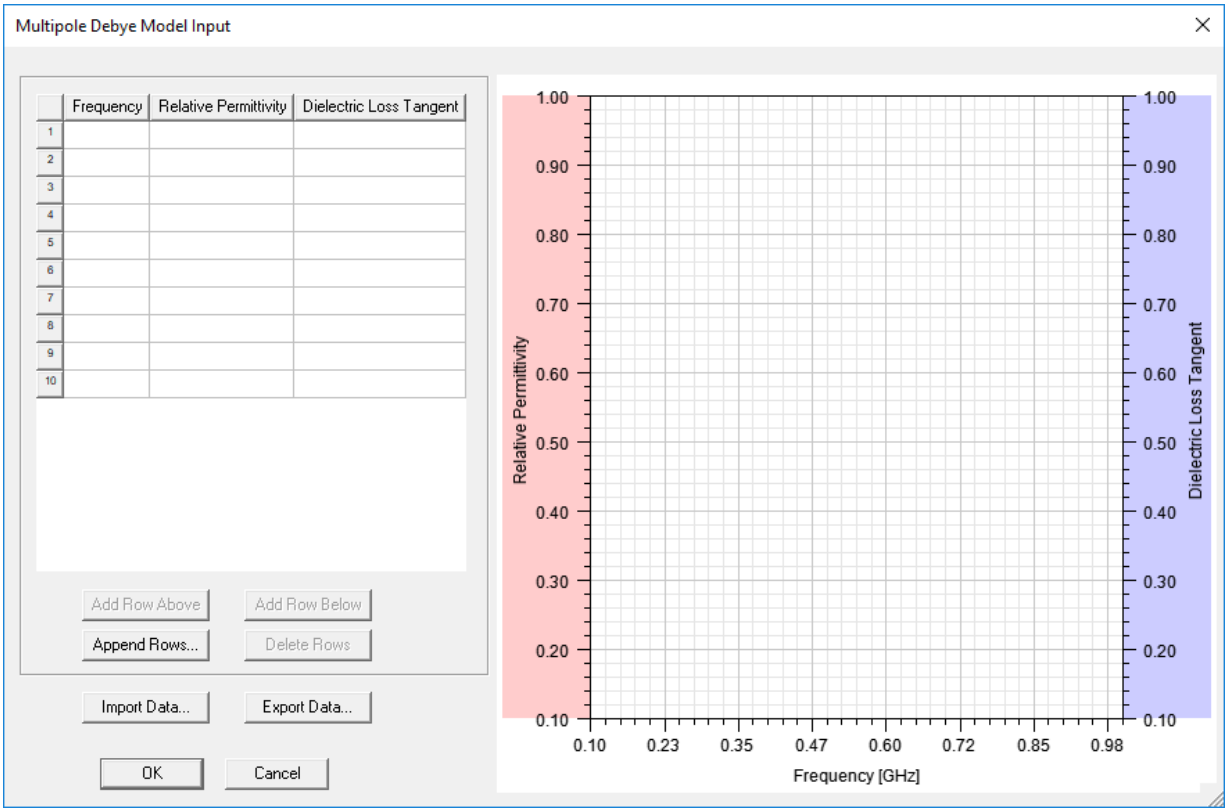
To modify the dataset with additional points, see Editing Datasets.

> **Note:**
>
> Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

Assigning Frequency Dependent Material: Multipole Debye Model Input

When you select **Multipole Debye Model Input** as the model for the frequency dependent material property, the **Multipole Debye Model Input** window appears.

From this window, you can enter material property values and view a property vs. frequency plot.



The window contains a table that allows you to specify **Frequencies**, and the material's **Relative Permittivity** and **Dielectric Loss Tangent** at each frequency.

Note:

- The minimum value for Frequency is 0. There is no upper limit.
- The minimum value for Relative Permittivity is 1. There is no upper limit.
- The minimum value for Dielectric Loss Tangent is 0. There is no upper limit.
- To get good results, provide at least 5 frequency points.

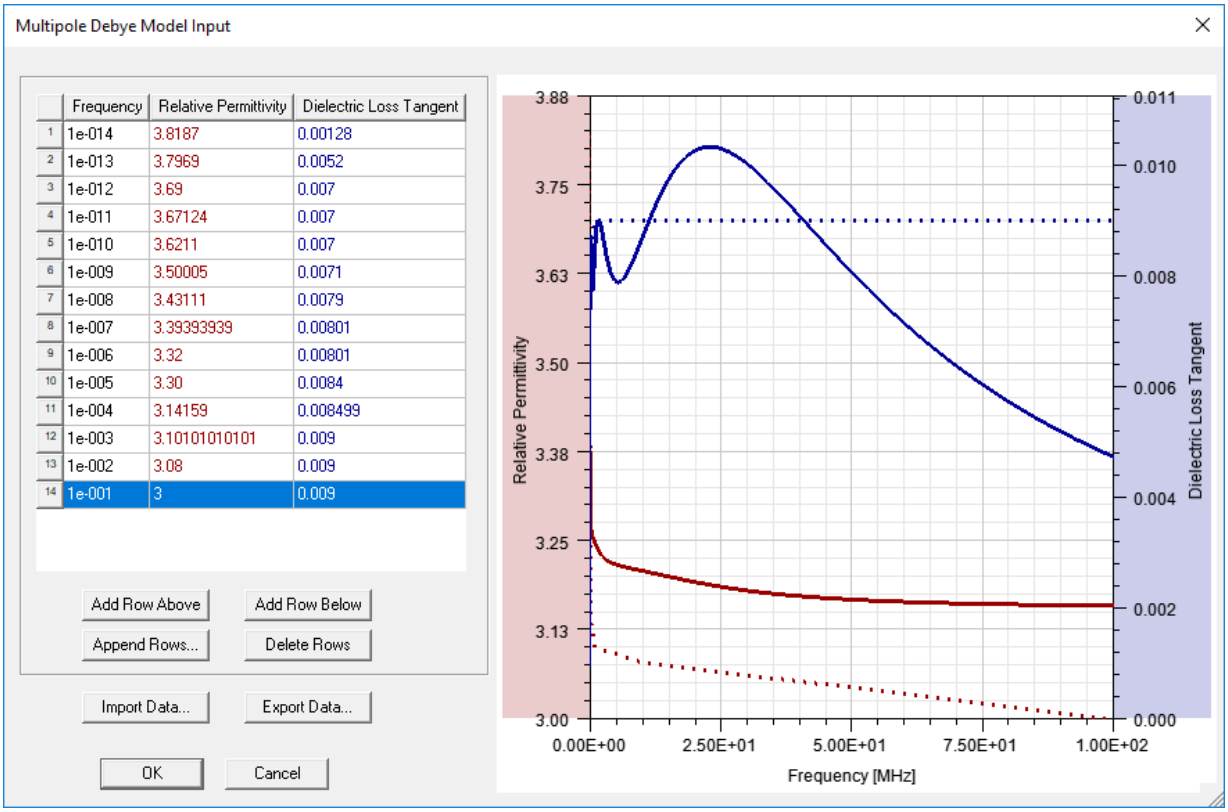
Enter data values manually, or click **Import Data** to import data from a *.tab file.

Below is an example of the file format. Each row provides Frequency (assumed to be Hz), Permittivity, and Loss Tangent.

```
0.00001 3.8136 0.00128

0.00010 3.7914 0.00520

0.00100 3.7500 0.00700
```

```
0.01000 3.7119 0.00700

0.10000 3.6742 0.00700

1.00000 3.6354 0.00700

2.15444 3.6346 0.00702

3.17000 3.6325 0.01073

4.64160 3.6186 0.01500

10.0000 3.5777 0.01750

21.5444 3.5458 0.01750

26.0000 3.5383 0.01750

46.4160 3.5148 0.01750

50.0000 3.5119 0.01750
```

As you enter data, the plot updates. The input data is linearly interpolated using the Multipole Debye model.



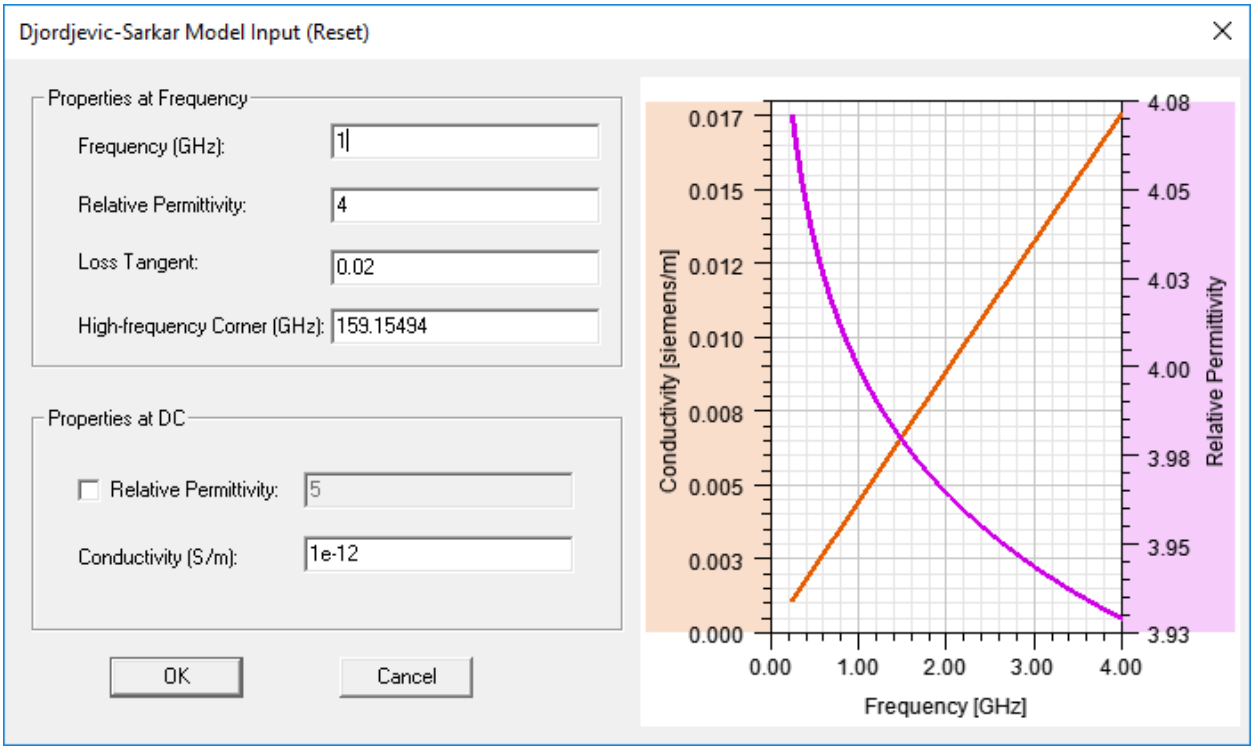
Double-click any plot element to change its properties (color, units, scaling, etc.).

To export the dataset in *.tab format for later use, click **Export Data**.

Assigning Frequency Dependent Material: Djordjevic-Sarkar Model Input

When you select **Djordjevic-Sarkar Model Input** as the model for the frequency dependent material property, the **Djordjevic-Sarkar Model Input** window appears.

From this window, you can enter material property values and view a property vs. frequency plot.



Use the fields to enter **Properties at Frequency** values:

- **Frequency**
- **Relative Permittivity**
- **Loss Tangent**
- **High-Frequency Corner** – the value of the High-frequency Corner should be at least 10 times higher than the Frequency. Reducing the upper corner frequency is not very attractive, because it has not been observed in experimental data. Therefore it is an upper bound on the loss tangents the Djordjevic-Sarkar model can handle. For technical details see *Technical Notes*: Djordjevic Sarkar Causal Dielectric Model.

Use the fields to enter **Properties at DC** values:

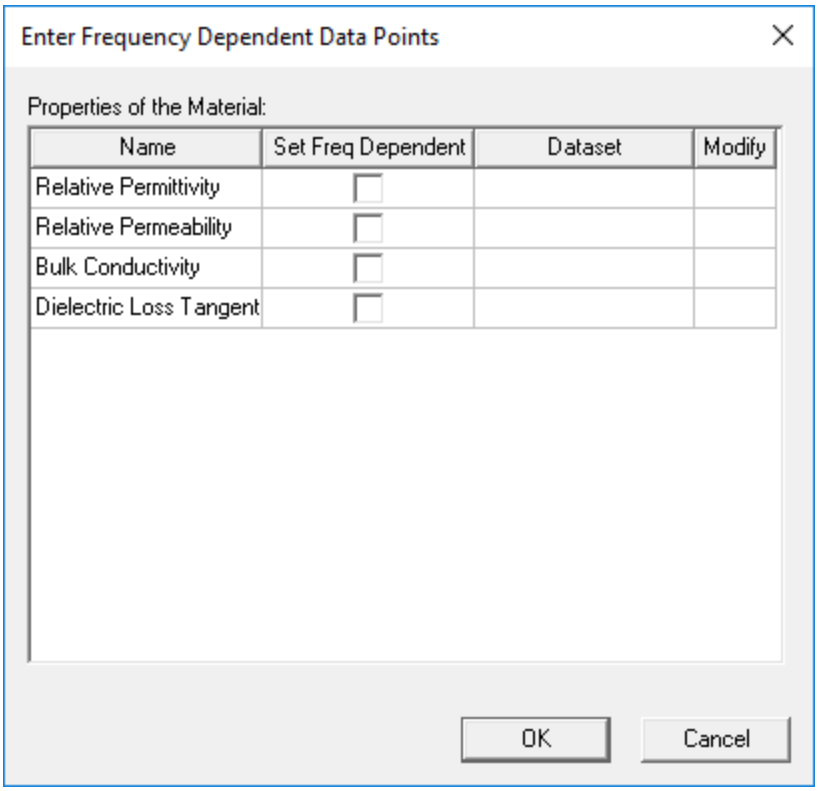
- **Relative Permittivity** – select the check box if you wish to specify relative permittivity.
- **Conductivity**

After you have entered the data for your selection, click **OK** to return to the **View/Edit Material** window. New default function names appear in the material property text boxes. A dataset is automatically created for each material property.

To modify the dataset with additional points, see Editing Datasets.

Assigning Frequency Dependent Material: Enter Frequency Dependent Data Points

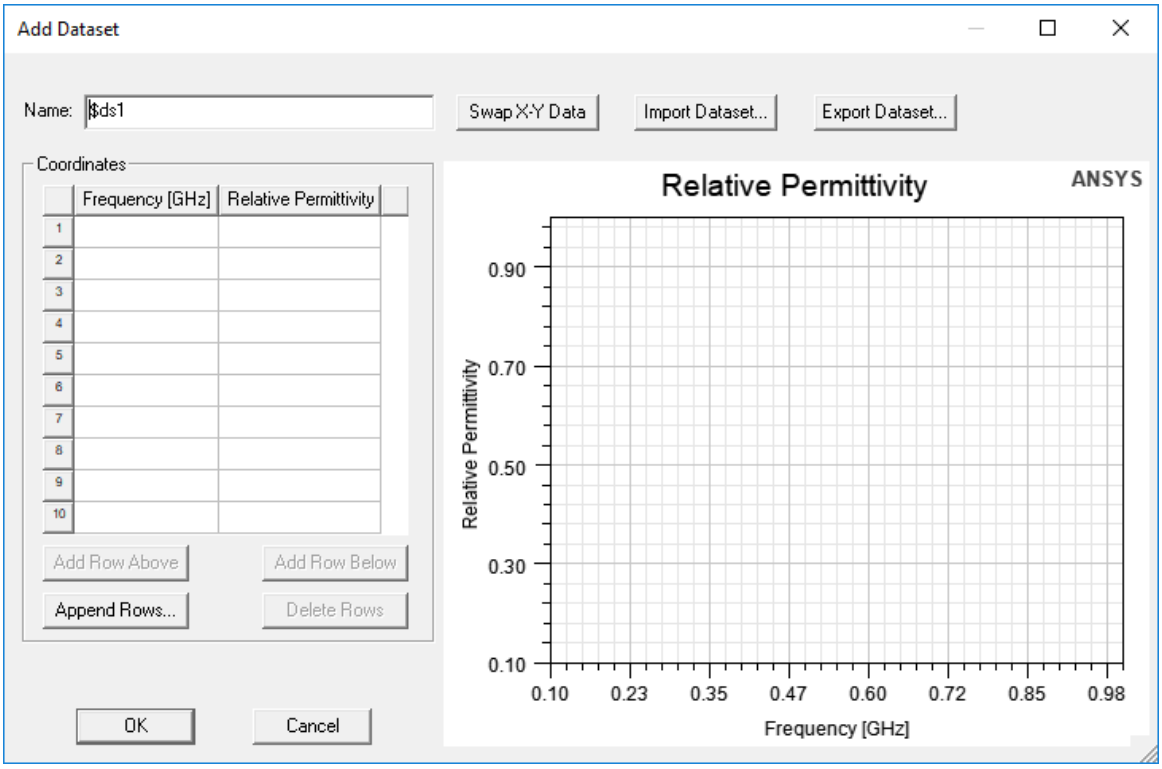
When you select **Enter Frequency Dependent Data Points** as the model for the frequency dependent material property, the **Enter Frequency Dependent Data Points** window appears.



From this window, you can use datasets to set frequency-dependent material properties.

1. Select the **Set Freq Dependent** check box next to the property you wish to edit. This enables the **Dataset** field.
2. Click the property's **Dataset** field and use the drop-down menu to select **Add/Import Dataset**.

   The **Add Dataset** window appears.

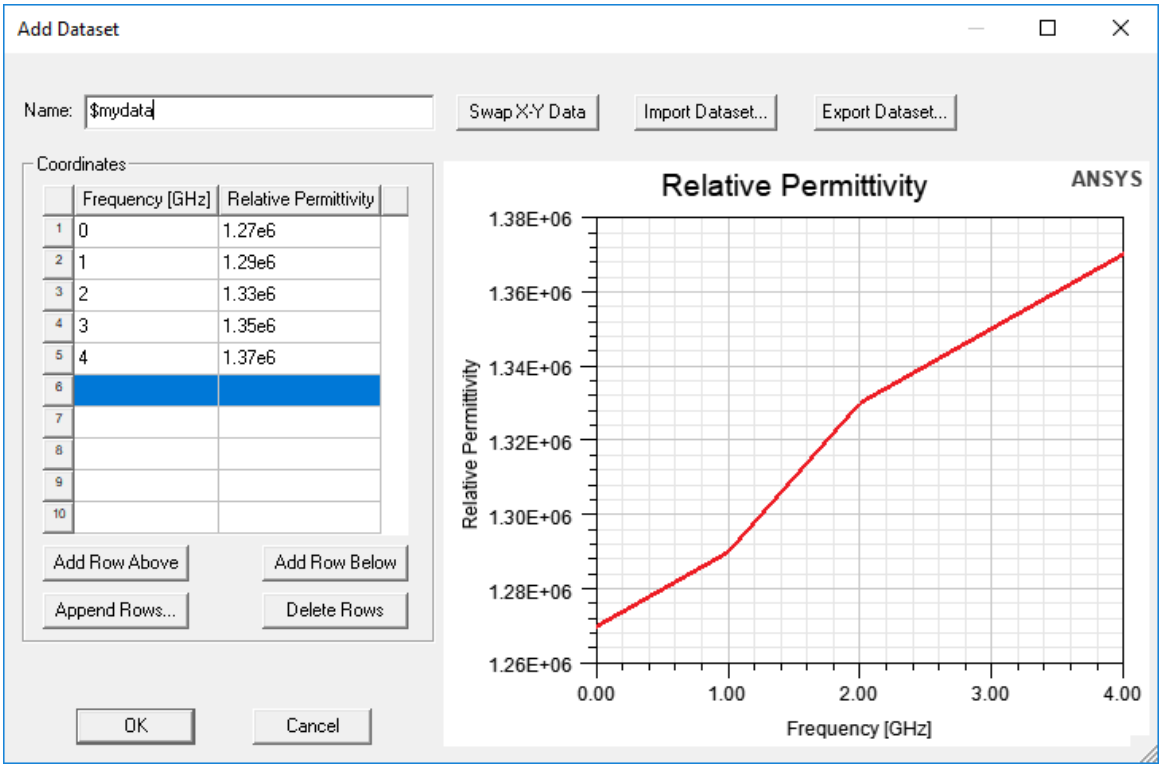


This window allows you to specify the following:

- **Name** – the default name is $ds1, but you can rename it. Note that the dollar sign ($) is automatically appended.
- **Coordinates** – contains a table that allows you to specify the Frequency (in GHz) and the specified material property's value at that frequency. Enter the data into the table manually, or click the **Import Dataset** button to populate it from a *.tab file.

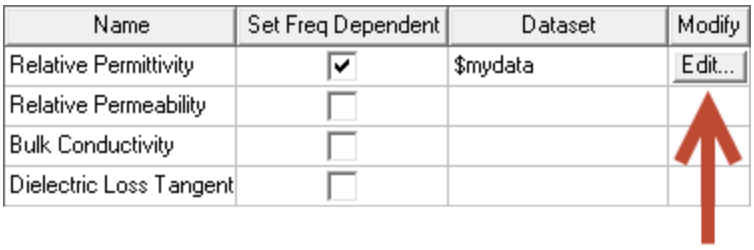
You can also:

- Rearrange table items using the **Add Row Above**, **Add Row Below**, **Append Rows**, and **Delete Rows** buttons.
- **Swap X-Y Data**
- **Export Dataset** as a *.tab file for later import.
- View the table data as a plot on the right side of the window.



3. When you have finished adding points, click **OK**.

   You are returned to the **Enter Frequency Dependent Data Points** window, where the **Modify** field is now active. If you need to modify the dataset, click **Edit**.



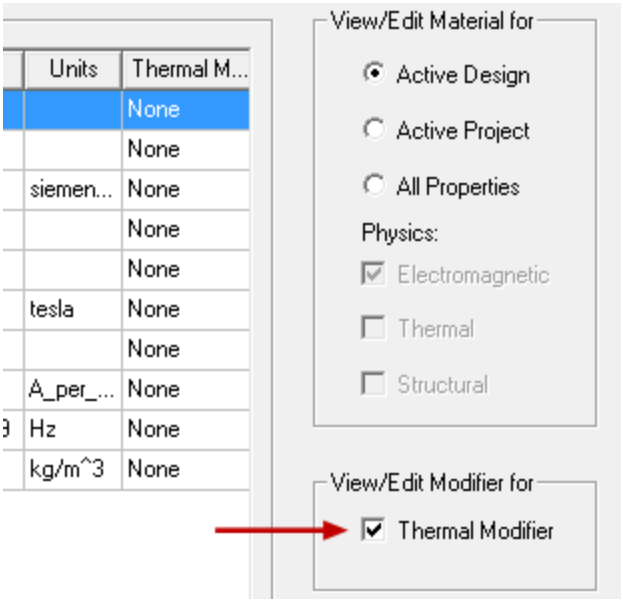
4. Click **OK** again to return to the **View/Edit Material** window.

Specifying Thermal Modifiers

Thermal modifiers are multiplicative in nature and assigning them introduces spatial dependence. Because you cannot have spatial dependence and causality, Ansys Electronics Desktop ignores Auto Causal Materials options when a thermal modifier exists.

To specify thermal modifiers for a material:

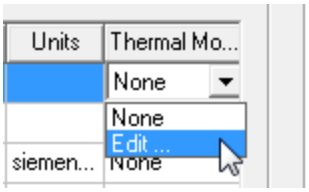
1. In the **View/Edit Material** dialog box, you must enable the **Thermal Modifier** check box:



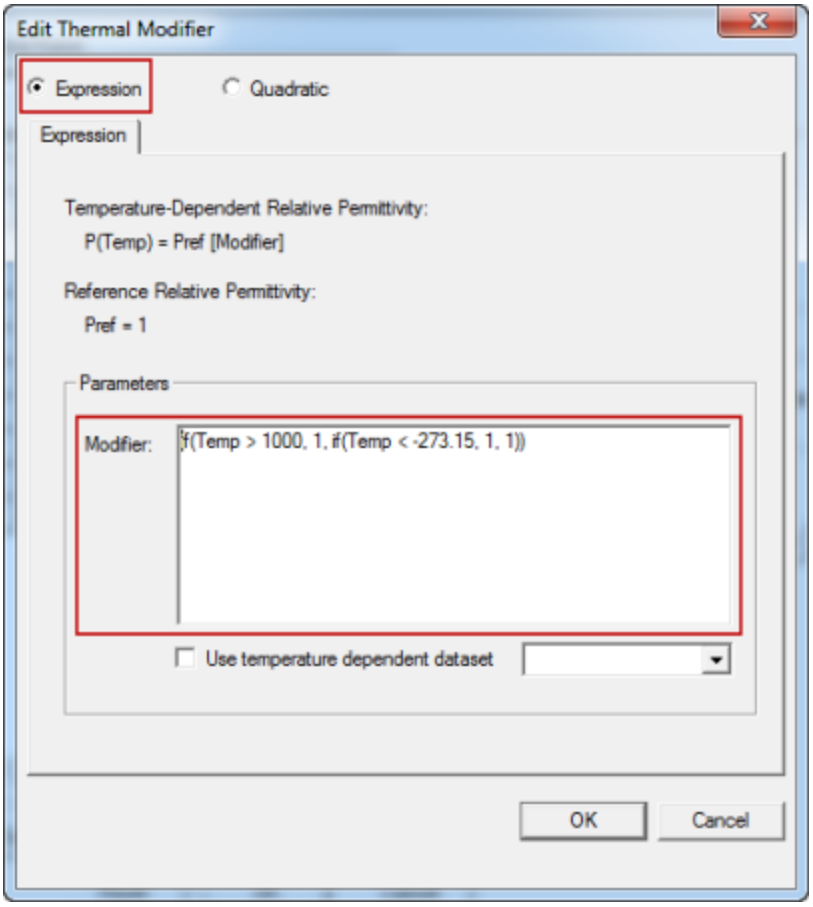
This option causes the properties table to expand to include a Thermal Modifier column.

By default, the **Thermal Modifier** option is set to **None** for all material properties.

2. Click a **Thermal Modifier** cell and select **Edit...** from the drop-down menu:



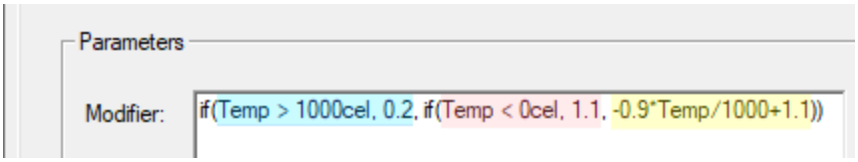
The **Edit Thermal Modifier** dialog box appears with the **Expression** option selected.



With this option selected, the **Modifier** text box is displayed under **Parameters**. This text box contains a conditional expression template, which you can use as a starting point for creating your own expression. The property for which the multiplier is being defined and its nominal value are indicated above the **Parameters** section.

- With **Expression** selected, you can write an equation for a thermal modifier in the **Modifier** text box.

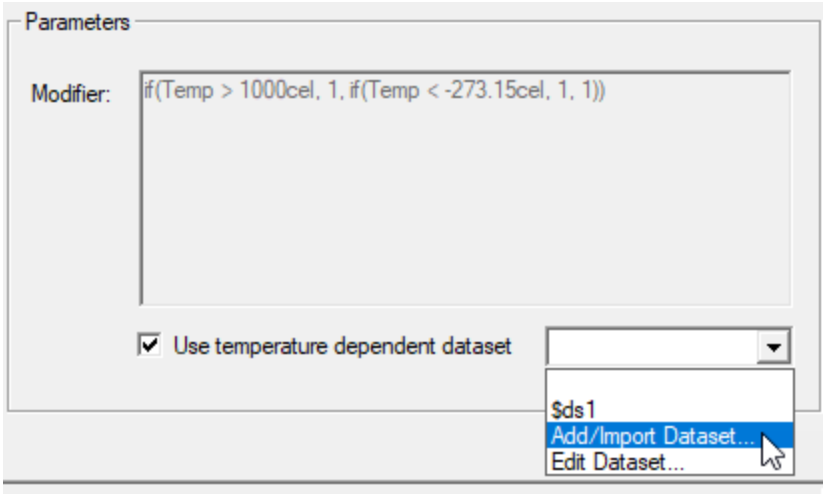
  The following example defines a multiplier that varies linearly from 1.1 (at 0° C) to 0.2 (at 1000° C). The multiplier is constant at 1.1 for temperatures below 0° C and constant at 0.2 for temperature above 1000° C:



In this example, the linear equation **-0.9 * Temp / 1000 + 1.1** (highlighted in yellow) defines the multiplier for the range 0° C ≤ Temp ≤ 1000° C.

The default unit of temperature in thermal modifier expressions is Celsius (cel). However, you can override the units by appending the appropriate abbreviation to the numbers in the expression (mkel, ckel, dkel, kel, cel, rank, fah). Model temperatures, material properties, and thermal modifiers need not have consistent units; values are converted automatically as needed.

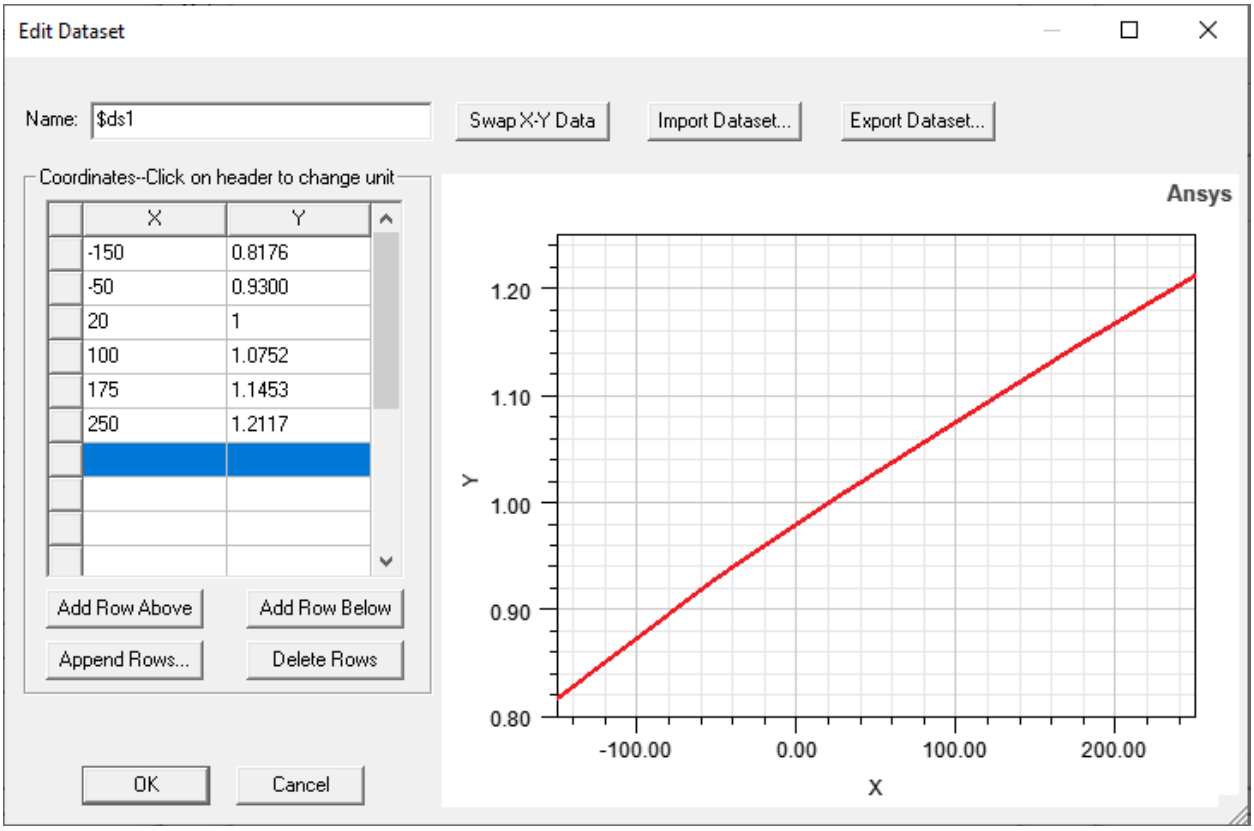
- Check **Use temperature dependent data set** to disable the Modifier text box. You can then use the drop-down menu at the bottom-right corner of the dialog box to select an existing project dataset or to create a new one by choosing **Add/Import Dataset**.



This command lets you define the thermal modifier dataset (a set of X and Y values). Alternatively, if a suitable project dataset already exists, it will be available for selection from the same drop-down menu.

When creating a new dataset, the **X** values are temperatures, and the **Y** values are the corresponding thermal modifiers. The value of the nominal material property is multiplied by this thermal modifier to determine the resultant property at a given temperature. The modifier is interpolated linearly between specified data points.

The following example is a dataset defining thermal multipliers for the thermal expansion coefficient (TEC) of an aluminum-beryllium alloy. The temperature range is -150 to 250° C, and the nominal value (corresponding to 20° C) is 1.37e-5/°C:



Based on these multipliers, the resultant TEC at -150° C is approximately 1.12e-5 (= 1.37e-5 * 0.8176), and the TEC at 250° C is 1.66e-5 (= 1.37e-5 * 1.2117).

In addition to filling the table of values manually, you can also **Import** an existing tab-delimited file to create the dataset.

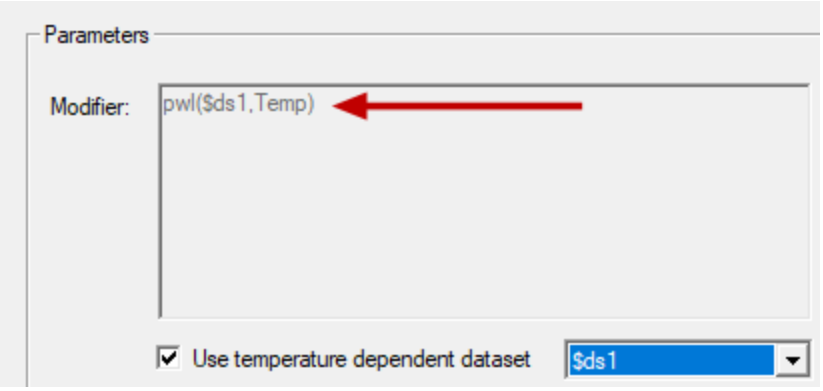
Important:

When you add a dataset from the **Edit Thermal Modifier** dialog box, you cannot specify the temperature units. The numbers in the **X** column will be interpreted as degrees Celcius and cannot be changed. Also, you cannot append a unit abbreviation (such as "fah" or "cel") to the numbers. You must enter temperature values in degrees Celcius for correct results. The values in the **Y** column are multipliers and therefore dimensionless.

Alternatively, you can create a dataset based on any desired temperature units before defining your temperature-dependent material properties. To do so, click **Project> Datasets** from the menu bar and then choose **Add**. If your project includes a design type that supports 3D datasets, a submenu appears offering a choice of a 1D or 3D dataset. If you see this choice, click **1D**. In either case, the **Add Dataset** dialog box appears.

For the dataset to be used later as a thermal modifier, you must click the **X** column header and choose **Temperature**. Then, click the **X** column header again and choose the desired temperature units. Finally, populate the table with your temperature and multiplier values and then click **OK** and **Done**. The dataset will now be available for selection from the drop-down menu in the **Edit Thermal Modifier** dialog box.

After choosing or creating a dataset, it appears in the **Modifier** field of the **Edit Thermal Modifier** dialog box, as shown in the following image:



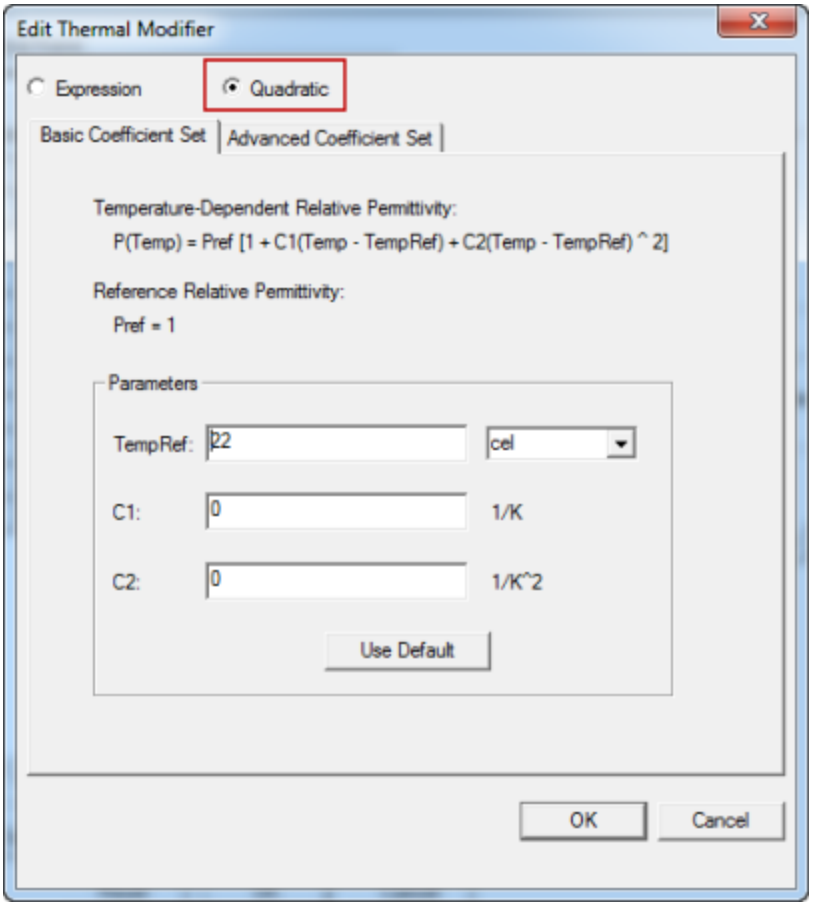
The syntax of the dataset expression is explained as follows:

- **pwl:** Type of dataset – Piecewise linear
- **$ds1:** Name of an existing dataset – "$" indicates a *project* dataset, which is a requirement for thermal modifiers
- **Temp:** Type of X value – Temperatures

  The Y values in this case are simple multipliers and therefore unitless.

> **Note:**
>
> To edit an existing dataset, select **Edit Dataset** from the drop-down menu in the **Edit Thermal Modifier** dialog box. The **Datasets** dialog box appears. Select the dataset to modify and click **Edit**.

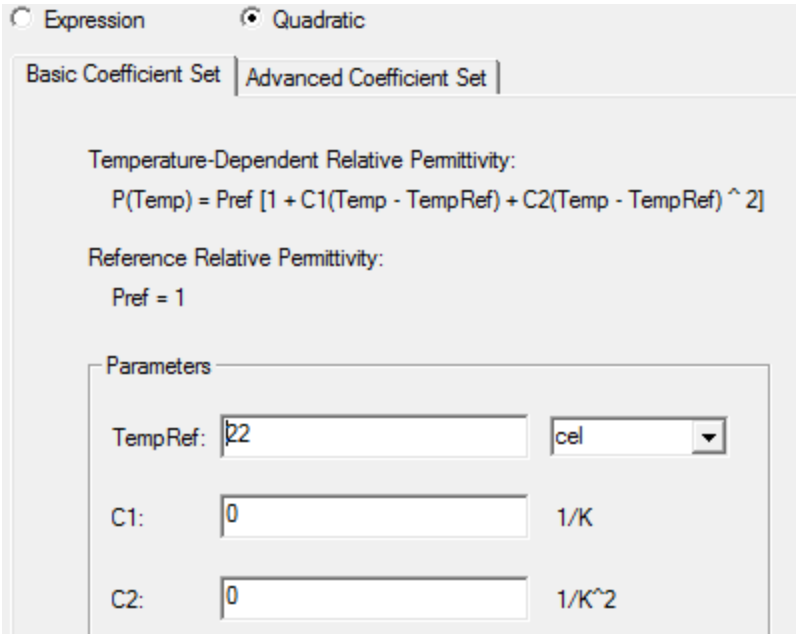
3. Alternatively, select the **Quadratic** option to display the tabs for **Basic Coefficient Set** and **Advanced Coefficient Set** (as shown below):



- Under the **Basic Coefficient Set** tab, you can edit fields for the TempRef and units, and fields for C1 and C2 for the following equation:

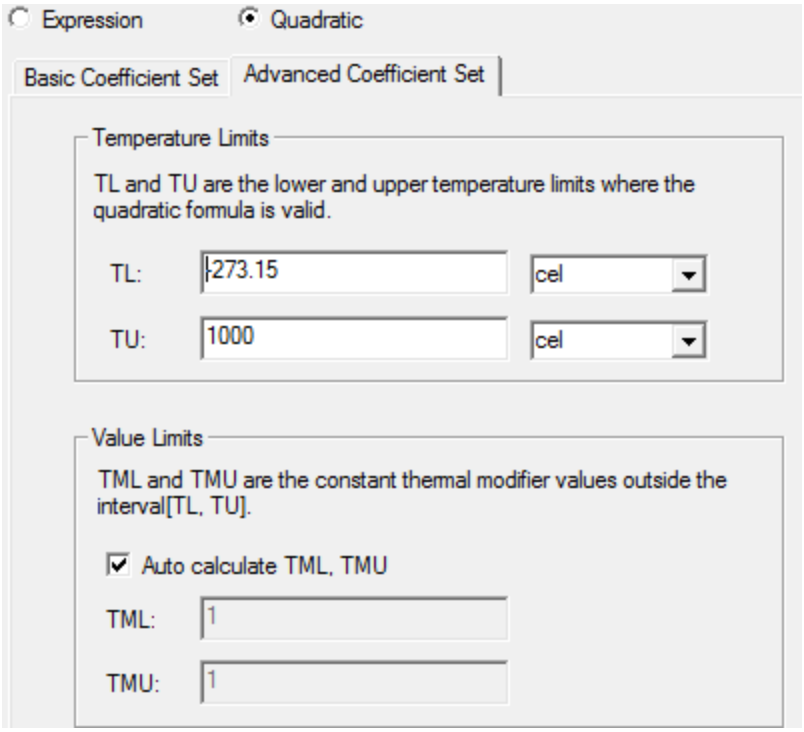
P(Temp) = Pref[1+ C1(Temp - TempRef) + C2(Temp - TempRef)^2]

where TempRef is 22 cel by default and where the Pref is defined as the reference relative permittivity.



> **Note:**
>
> The coefficients, C1 and C2, should be negative to yield physical results.

- Under 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 menu.



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 your edits and return to the View/Edit materials dialog box.

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.

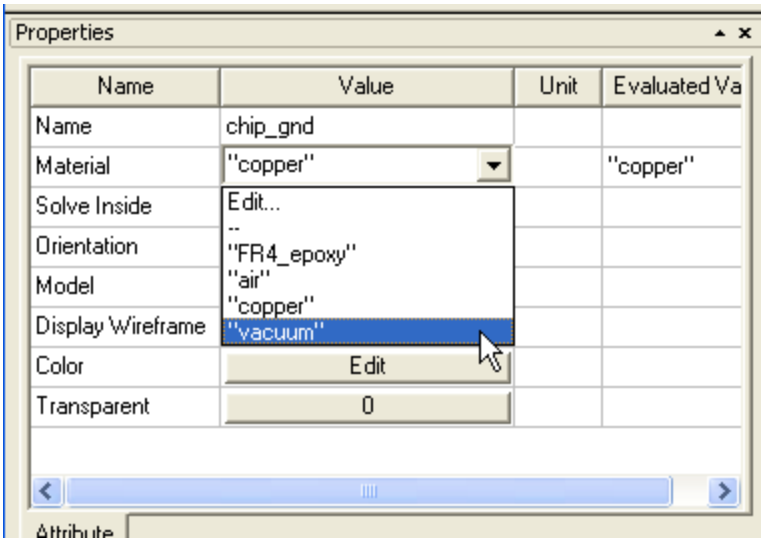
Defining Functional Material Properties

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.

> **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.

Assigning Materials from the Object Properties Window

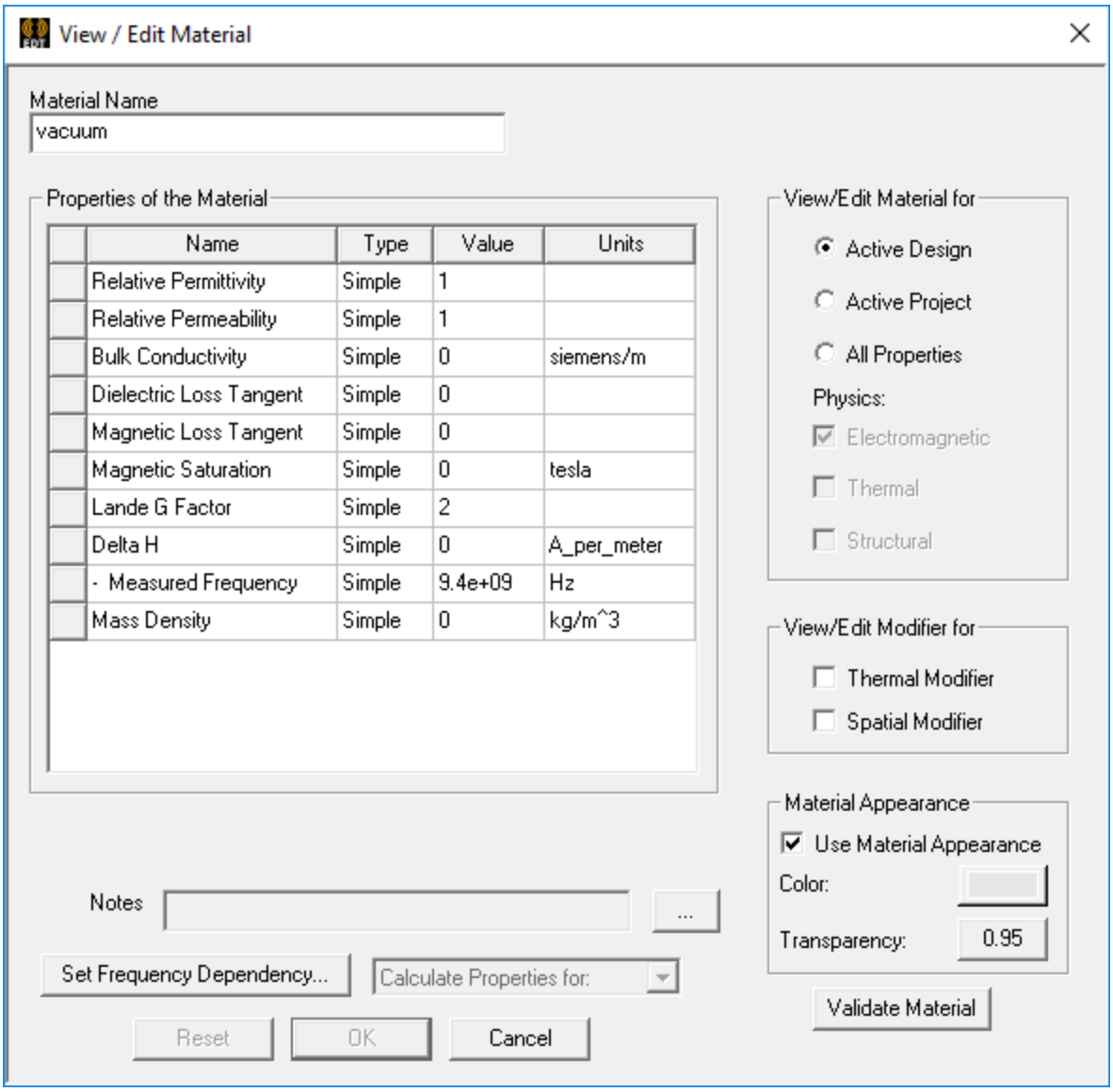
The Properties dialog for each object includes a materials property. If you click on the current material property you see a drop down list that includes an *Edit* command and a list of materials in the current project. You can select from the list of current materials to assign the selected material to that object.



Viewing and Modifying Material Attributes

1. From the **Select Definition** window, select a material to view or modify, and click **View/Edit Materials**.

The **View/Edit Material** window 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.

2. 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. You can use the scroll bars or size the dialog to see all properties. When **All Properties** is selected, the following **Physics** classification options are enabled to show or hide properties based on simulation type:
        - **Electromagnetic**
        - **Thermal**
        - **Structural**

    > **Note:**
    >
    > If a material is edited in a design type for which the **Physics** type has not been set (e.g., an HFSS design but Electromagnetic physics type was not set), the **Physics** type will be automatically set in the material.

    > **Note:**
    >
    > For Permeability, Q3D Extractor does not support frequency-dependent values. Entering a frequency-dependent value in this field will result in an error.

3. 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** , , **Vector** and **Vector Mag**, or for Relative Permeability, **Non-linear**, as required for that property.

       For **Simple**, you provide a value or variable.

       For **Vector**, you provide a **Vector Mag**.

       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.
    e. Enable the **Use Material Appearance** check box to specify a color and transparency.

> **Note:**
>
> Materials stored in the global material library cannot be modified.

4. If you want to add descriptive notes for the new material, click the ellipsis button [...] to the right of the **Notes** field. This opens a dialog box in which you can enter text. Pressing **Enter** or clicking **OK** saves the note. To enter multiple lines of notes, use **Ctrl+Enter** at the start of each new line.
5. Click **OK** on the **View/Edit Materials** dialog to add the new material to the material library.
6. Click **OK** to save the changes and return to the **Select Definition** window.

> **Warning:**
>
> If you modify a material that is assigned in the active project after generating a solution, the solution will be invalid.

Validating Materials

The Ansys Electronics Desktop can validate a material's property parameters for an Ansys EM software product. For example, the software will check 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 **Active Design**, **Active Project**, or **All Properties**, and then click **Validate Material**.

Copying Materials

1. In the **Select Definition** window, select the material you want to copy, and then click **Clone Material** or right-click the selected material and select **Clone** from the shortcut menu.
2. To modify the material's attributes, follow the directions for modifying materials.
3. Click **OK** to save the copy in the active project's material library.

Removing Materials

1. In the **Select Definition** window, select one or more materials you want to remove from the active project's material library.
2. Click the **Remove Material (s)** button or right-click the selected material and click **Remove** on the short-cut menu.

The material is deleted from the project material library.

> **Note:**
>
> The following materials cannot be deleted:
>
> - Materials stored in Ansoft's global material library.
> - Materials that have been assigned to objects in the active project.

In a project library, you may want to use the **Tools** > **Project Tools** > **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** window, select the material you want to export.
2. Click the **Export Material to Library** button, or right-click the selected material and click **Export** from the short cut menu.

   The **Export to material library** file browser appears.
3. 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 and then click **Save**.

Sorting Materials

You can change the order of the materials listed in the **Select Definition** window. You can sort the list of materials by name, library location, color, or material property value. To change the order of the listed materials:

- Click the column heading by which you want to order the materials.

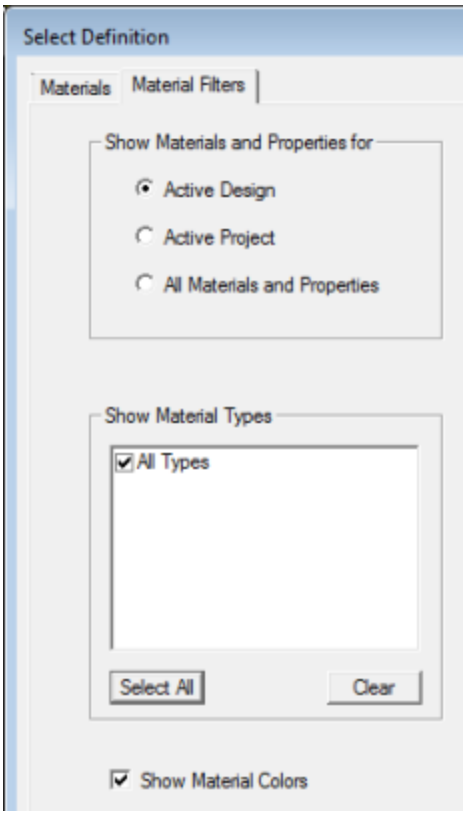
If the arrow in the column heading points up, the material data will be listed in ascending order (1 to 9, A to Z) based on the values in the column you chose. If you want the material data to be

listed in descending order (9 to 1, Z to A), click the column heading again. The arrow will point 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.

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 text 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 materials/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 materials/properties used in the active design.
- **Active Project** to display materials/properties used by the active project.
- **All Materials and Properties** to display all materials and properties available. Selecting this enlarges the table of materials shown under the **Materials** tab to show all materials possible. You can use the scroll bars or size the dialog to see all materials.

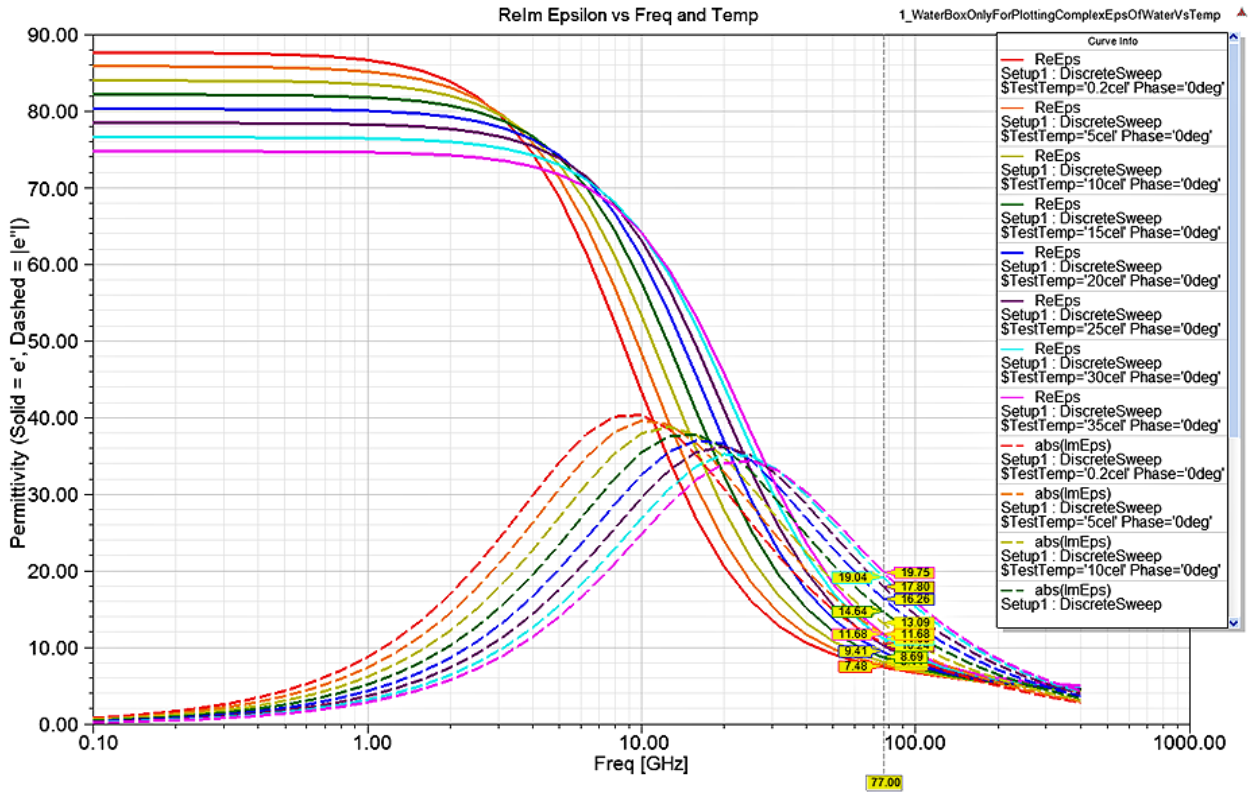
3. Under **Show Material Types**, select materials types to display on the **Materials** tab. Click **Select All** to select all of the types listed. Click **Clear** to clear all selections.
4. If you check Show Material colors, the Materials tab of the **Select Definition** window includes a color for material colors. See Viewing and Modifying Material Attributes.
5. Click the **Materials** tab to save your selections.

Click **Cancel** to revert back to the last saved selections.

Water (Temperature- and Frequency-Dependent Properties)

Applicability: The temperature-dependent and frequency-dependent water properties are applicable to HFSS, Q3D, and 2D Extractor designs.

The complex permittivity of water changes significantly in the 1 – 100 GHz range. Many HFSS applications involve frequencies within this range and also involve water. Appropriate water properties are needed for accurate electromagnetic modeling. The following image demonstrates the widely varying properties of water versus temperature and frequency:



These varying properties of water are included in HFSS through the addition of the material, *Water 0.2-35C 0.1-400GHz*.l

Application Example: Companies test automotive radar at 24 GHz or 77 GHz with thin layers of water applied on the target fascia. A thin layer of water may increase both signal reflection and attenuation, significantly affecting the radiation pattern.

A relative permittivity (ϵ_r) of 81 is used as a typical value for water. If you were to assume this value of ϵ_r for the radar example just described, the simulation would produce inaccurate results. To solve this problem, a temperature-dependent and frequency-dependent material library, *Water_0p2-35C_0p1-400GHz*, has been included within the *syslib* folder of the Ansys Electronics Desktop application. Rather than containing multiple versions of water suitable for different temperature and frequency combinations, the complex permittivity is expressed as a function of temperature and frequency in a single material choice, *Water 0.2-35C 0.1-400GHz*.

Application Procedure

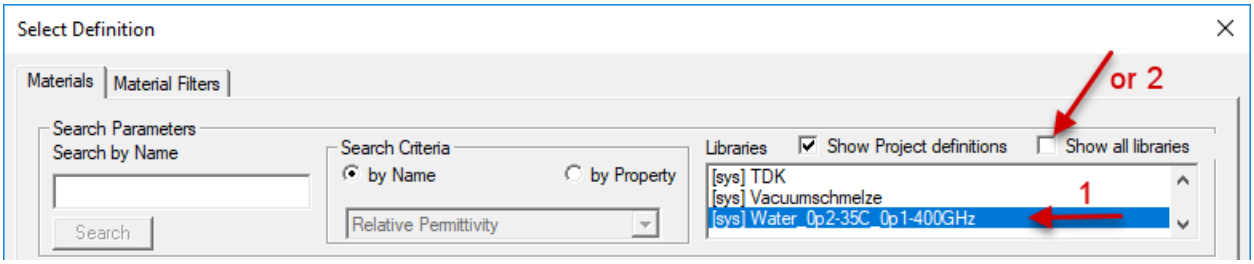
As with any material, you can choose the temperature-dependent and frequency-dependent water as the *Default material* or apply it to specific solid objects, as follows:

1. To apply the material for all subsequently created objects:
    - On the **Draw** ribbon tab, choose **Select** from the **Default material** drop-down menu.

2. Alternatively, to apply the material to one or more selected objects, use *one* of the following three methods:
   - From the **Attribute** tab of the *Properties* dialog box or docked *Properties* window, choose **Edit** from the **Material** drop-down menu.
   - Select one or more solid objects in the History Tree. Then, right-click a selected object and choose **Assign Material** from the shortcut menu.
   - Graphically, in *Select Objects* mode, select one or more objects in the Modeler window. Then, right-click in the window and choose **Assign Material**.

   Regardless of whether you use the step 1 method or one of the three methods under step 2, the same *Select Definition* dialog box appears.

3. In the **Libraries** section within the **Materials** tab, take *one* of the following three actions:
   - With **Show all libraries** *not* selected, click the library **[sys] Water_0p2-35C_0p1-400GHz** to select it (A single *SysLibrary* entry will appear in the material list along with any *Project* materials).
   - Select the **Show all libraries** option. (All *SysLibrary* and *Project* materials will be listed in the dialog box.



4. Select the material **Water 0.2-35C 0.1-400GHz**:

Name	Location	Origin	Relative Permittivity	Relat Permea
vacuum	Project	Materials	1	1
vacuum	SysLibrary	Materials	1	1
Water 0.2-35C 0.1-400GHz	SysLibrary	Water_0p2-35C_0p1-400GHz	1	1
water_distilled	Project	Materials	81	0.999991
water_distilled	SysLibrary	Materials	81	0.999991
water_fresh	SysLibrary	Materials	81	0.999991
water_sea	SysLibrary	Materials	81	0.999991
ZEONEX RS420 (tm)	SysLibrary	Materials	2.3	1

5. Click **OK**.

Include Temperature Dependence:

Temperature dependency is an optional feature that is *not* enabled by default. For the subject water properties to work, you must select the *Include Temperature Dependence* option and define each water object's temperature. The additional procedure is as follows:

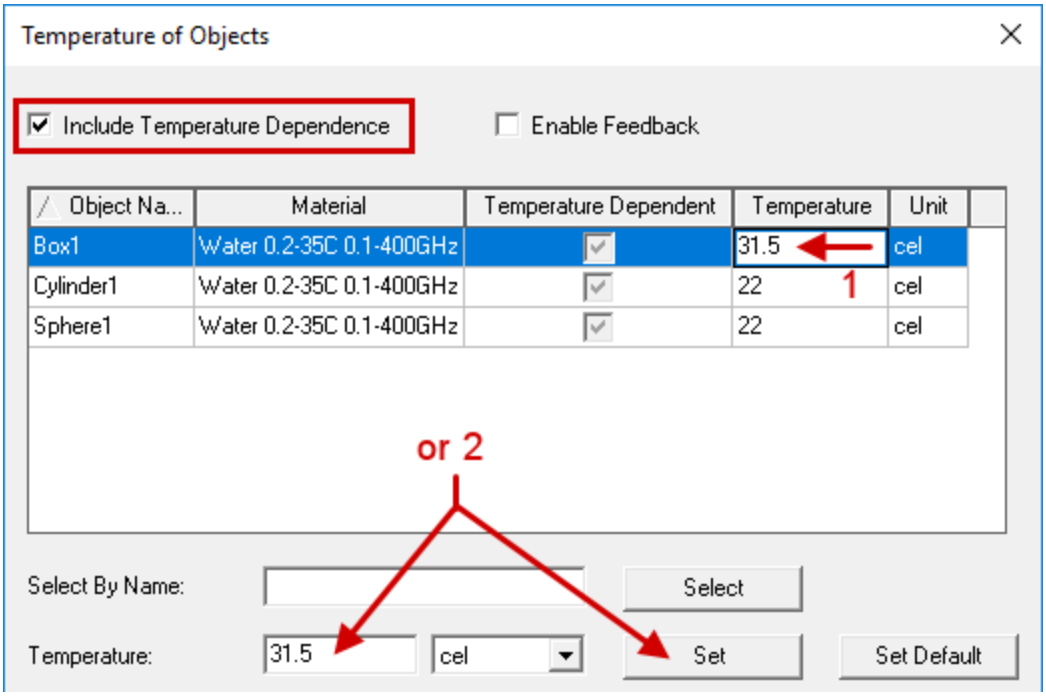
6. Using the menu bar, click one of the following commands, depending on the design type:
   - **HFSS** > **Set Object Temperature**
   - **Q3D Extractor** > **Set Object Temperature**
   - **2D Extractor** > **Set Object Temperature**

   Alternatively, right-click the top-level design entry in the Project Manager (*HFSSDesignx, Q3DDesignx,* or *2DExtractorDesignx*, by default), and choose **Set Object Temperature**.

   The **Temperature of Objects** dialog box appears.

7. Select the **Include Temperature Dependence** option.
8. In the **Temperature** column for all objects using the material *Water 0.2-35C 0.1-400GHz*, specify the object temperature in your preferred units.

   Alternatively, select one or more rows of the table, specify a value in the **Temperature** text box beneath the table, and click **Set**.

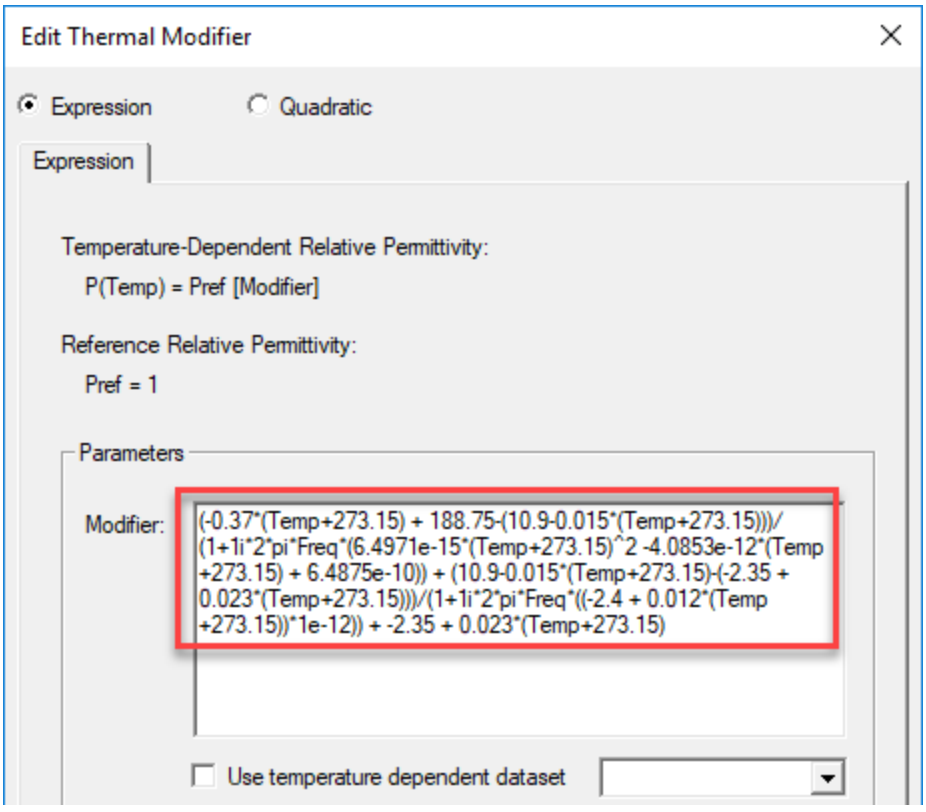


> **Note:**
>
> The internal temperature unit in Ansys Electronics Desktop is Celsius. For any other temperature unit you might specify, the value of *Temp* is automatically converted to Celsius internally prior to the execution of the ϵ_r equation, which ensures that the equation is valid for all temperature units.

9. Click **OK**.

Relative Permittivity Calculation

A nominal value of 1 is specified for the *Relative Permittivity* of the water. However, this value is modified by the equation shown in the *Thermal Modifier* column of the *View / Edit Material* dialog box. While the equation is listed as a "thermal" modifier, it also includes frequency-dependent factors. To access this equation, select the material, as shown in step 4 above, and click **View / Edit Materials**. Then, click the *Thermal Modifier* equation and choose **Edit**:



This equation is comprised of several different factors, each defined by a specific part of the overall expression. The equations for each factor were determined using available empirical data and curve fitting techniques and were built upon previously published work (see the

Reference section near the bottom of this topic). Let's look at the Relative Permittivity (ϵ_r) equation and the individual parameters that comprise it.

Relative permittivity equation:

$\epsilon_r = [(\varepsilon - \varepsilon_2) / (1 + 1i * \omega * \tau_1^2) + (\varepsilon_2 - \varepsilon_\infty)] / (1 + 1i * \omega * \tau_2) + \varepsilon_\infty$

where ω is the frequency in radians/second (equal to 2π * Freq in Hz), and the other factors are as defined below.

Dielectric relaxation parameters of water:

- **Linear, temperature-dependent parameters:**

  $\varepsilon = -0.37 * T + 188.75$

  $\varepsilon_2 = 10.9 - 0.15 * T$

  $\varepsilon_\infty = -2.35 + 0.23 * T$

  $\tau_2 = (-2.4 + 0.12 * T) * 1E-12$

- **Quadratic, temperature-dependent parameter:**

  $\tau_1 = 6.4971E-15 * T^2 - 4.0853E-12 * T + 6.4875E-10$

For each of the preceding five equations, T is the absolute temperature in degrees Kelvin (°K). Since the internal calculation unit for temperature in Ansys Electronics Desktop is Celsius, 273.15° must be added to *Temp*, and this sum (Temp + 273.15) must be substituted for *T* in each equation. The resultant complete expression for the relative permittivity, formatted appropriately for the software, is as follows:

ϵ_r = (-0.37 * (Temp + 273.15) + 188.75 - (10.9 - 0.015 * (Temp + 273.15))) / (1 + 1i * 2 * pi * Freq * (6.4971E-15 * (Temp + 273.15)^2 - 4.0853E-12 * (Temp + 273.15) + 6.4875E-10)) + (10.9 - 0.015 * (Temp + 273.15) - (-2.35 + 0.023 * (Temp + 273.15))) / (1 + 1i * 2 * pi * Freq * ((-2.4 + 0.012 * (Temp + 273.15)) * 1e-12)) + -2.35 + 0.023 * (Temp + 273.15)

Reference:

4 June 1999, Chemical Physics Letters 306 (1999) 57-63

The dielectric relaxation of water between 0°C and 35°C
R. Buchner, J. Barthel, J. Stauber
Institut für Physikalische und Theoretische Chemie, Universität Regensburg D-93040 Regensburg, Germany

Working with Material Libraries

There are three different materials libraries in Q3D Extractor: a *system library*, a *user library*, and a *personal library*.

The library files that ship with Electronics Desktop 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.

In addition to the system libraries, Electronics Desktop recognizes two user-configurable library structures, called the *User Library* and the *Personal Library*. These are used to add user (or company)-defined materials. Customarily, **`userlib`** is a network repository for proprietary or corporate definitions available to all seats in an enterprise, while **`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 Electronics Desktop directory.

Materials from all libraries are available for use in projects.

Editing Libraries

There are two different methods of editing libraries:

- Right-click **Materials** in the project window to display the **Edit Libraries** shortcut menu. Clicking displays 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.

Importing and Converting Materials in 2D Extractor

To preserve custom materials, you need to import them from a .mat file into the latest version of the user library.

> **Note:**
>
> The **.mat** extension shows up as a shortcut in the file browser.

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

To translate custom materials to the latest version:

1. Click **Q3D Extractor** > **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.
3. Enter a name for the new database in your current project in the **User Material DB Name** box.
4. Click **Translate**.

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 lambda refinement or 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 prior to solving.

- On the **Q3D Extractor** > **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.

5 - High Performance Computing

Ansys Electronics Desktop affords High Performance Computing (HPC) options that allow you to speed up processing. HPC leverages multiple cores through matrix multiprocessing, distributed frequency points (called spectral decomposition method or SDM), domain decomposition (DDM), parallel hybrid FEM/IE solving or the finite antenna array DDM. In addition, hierarchical HPC solving is possible where frequency points can be distributed with each frequency point using multiple cores or machines for large scale DDM analysis at each frequency point, all in parallel. Q3D Extractor intelligently determines which jobs are to be performed and how to distribute them for the simulation. It automatically apportions the jobs during the simulation process and makes optimum use of the available resources. This computing technology enables generating accurate solutions for large, complex, higher-fidelity models.

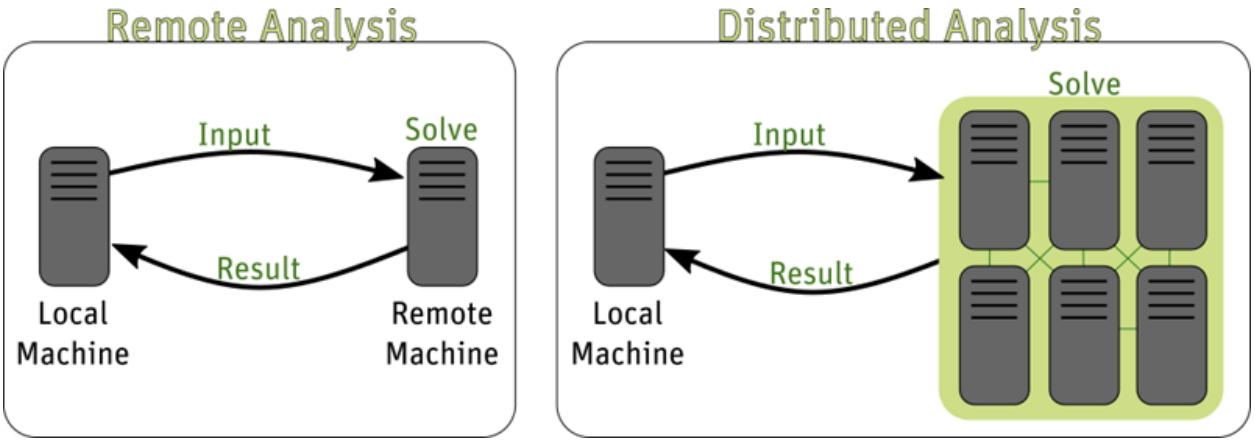
See the following topics for more information and configuration tips:

- Remote Analysis
- Distributed Analysis
- Ansys Cloud Direct Support for HPC Job Management
- Large Scale DSO for Parametric Analysis
- Interactive Scheduler Jobs
- HPC Integration
- Multi-Step Job Submission
- Windows to Linux Job Submission
- Distribution Command Line Options
- GPU Acceleration

Remote Analysis

It is possible to solve a project on a different machine from the one on which you are running Ansys Electronics Desktop. 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 will perform the solve (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 machines.



Important:

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.

The rest of this topic covers the following:

- Prerequisites for Remote and Distributed Analysis
- Configuring the Local Machine to Solve Remotely
- Remote Analysis Options
- Running Remote Analysis

The **Export Options Files** command writes XML files containing options settings at all levels to the specified directory. **Tools** > **Options** > **Export Options** makes it easier for different users to use Ansys EM tools installed on shared directories or network drives. For some example use cases, see: Example Uses for Export Options Features.

Prerequisites for Remote and Distributed Analysis

1. You must have installed Ansys Remote Simulation Manager (RSM) or a supported High Performance Computing (HPC) management software program (See: High Performance Computing (HPC) Integration).

   For a list of currently supported HPC software, select the topic, *Ansys 2025 R1 - Job Schedulers and Queuing Systems Support* from the Platform Support page.

2. 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 Desktop 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.

3. If you use RSM, it must be accessible from all remote machines. In addition, the Ansys Electronics Desktop engines must be registered with each initialization of RSM.

   To do this on each machine:

   - From Windows, click **Start** > **Programs** > **Ansys EM Suite 2025 R1** > **Register with RSM**. You can also run **RegisterEnginesWithRSM.exe**, located in the product subdirectory (for example, C:\Program Files\ANSYS Inc\v251\AnsysEM\RegisterEnginesWithRSM.exe).

     In each case, you see a dialog box confirming the registration. Click **OK** to confirm.

   - From Linux, run **RegisterEnginesWithRSM.pl**, located in the product installation directory (for example, /opt/ansys_inc/v251/AnsysEM/RegisterEnginesWithRSM.pl).

     If the RSM service cannot run due to permission issues for the configuration file, it issues an error message and exits.

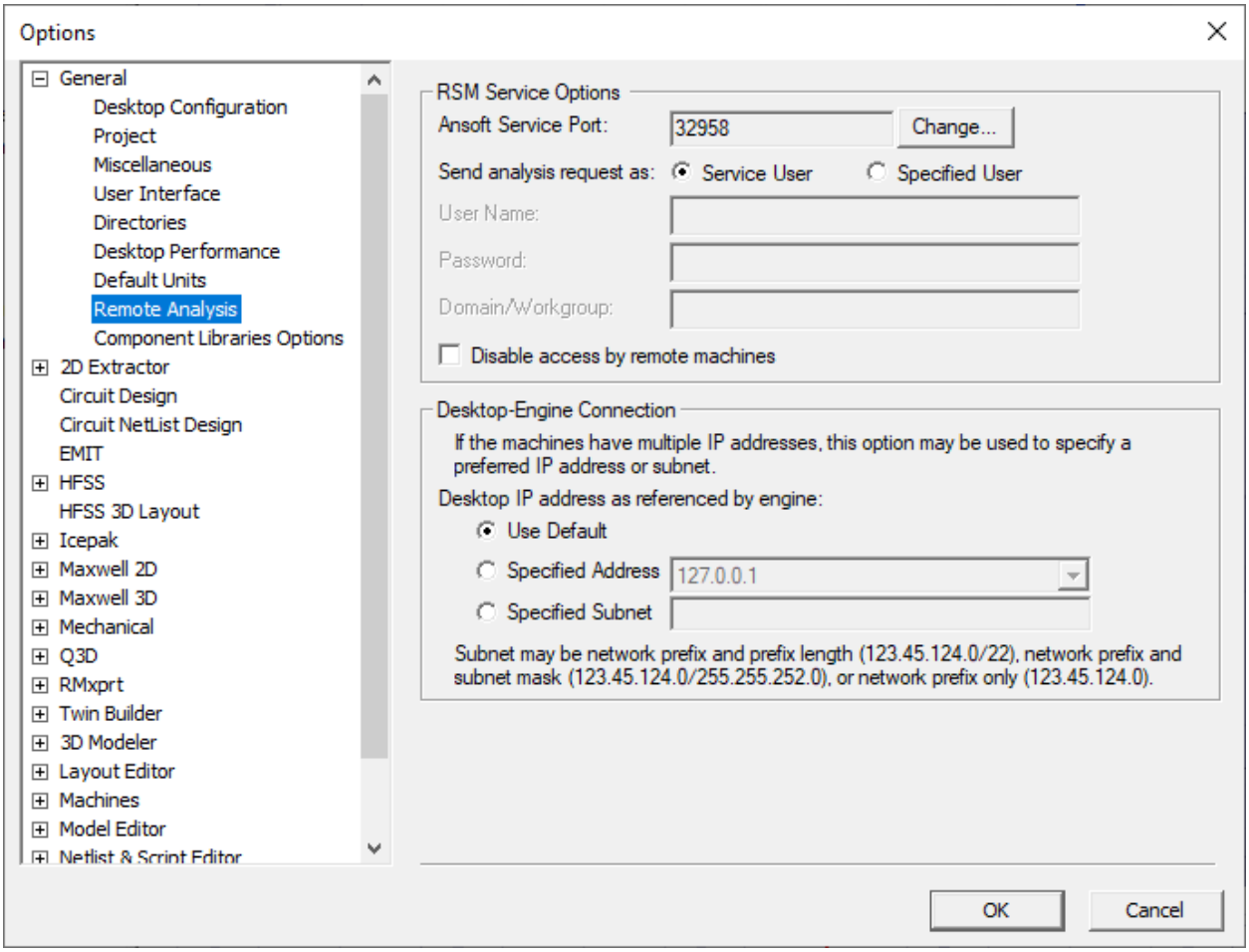
     If your product is not registered with RSM, the analysis will run locally.

Configuring the Local Machine to Solve Remotely

To set Q3D Extractor Analysis options, see: Distributed Analysis.

Remote Analysis Options

You can also set Remote Analysis Options from **Tools** > **Options** > **General Options** > **General** > **Remote Analysis**.



Select whether to run simulation processes as the user running RSM (Service User), or sd 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 Messages window issues a warning, 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 simulation continues. When the simulation is complete, you will see a message in the **Messages** window.

Troubleshooting

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

Problem:

When you try to solve from local to remote machine, a COM engine process starts on the remote machine, but the 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 turned 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, 2019)

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 Desktop 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.

Problem:

The command `ansoftrsmservice stop` will not work as root.

Resolution:

Use systemctl command to stop the service:

```
sudo systemctl stop ansoftrsmservice
```

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 Service Port in **Tools** > **Options** > **General Options** > **General** > **Remote Analysis**. If you have, change it back to the default (32958), restart Ansys Electromagnetics Desktop, 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 hang, 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 RSM Service. From the Windows menu, choose **Start** > **All Programs**> **Ansys EM Suite 2025 R1** > **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 Ansoft RSM service which is running on this machine."

Resolution:

1. To register the engine from the Windows menu, select **Start** > **All Programs** > **Ansys Electromagnetics Suite 2025 R1** > **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 Service Port in **Tools** > **Options** > **General Options** > **General** > **Remote Analysis**. If you have, change it back to the default (32958), restart Ansys Electromagnetics Desktop, 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 hang, the connection was successful.
4. Check to make sure Remote Simulation Manager is running:
   a. Go to the rsm subdirectory of the 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 Electronics Desktop installation directory. If the status query indicates "Not registered", type:
   `./RegisterEnginesWithRSM.pl add`
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 Ansoft RSM service which is running on this machine."

Resolution:

1. 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.

Q3D Extractor supports the following form(s) of distributed analysis:

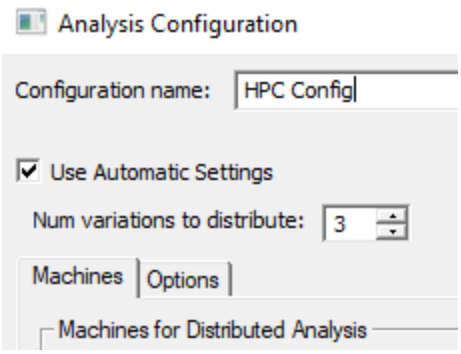
- Distributing rows of a parametric table, either as a regular DSO or as Large Scale DSO performed through command line.
- Distributing array solves.
- Distributing domain solves.
- Distributing single or discrete interpolating sweeps.

> **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.

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 Setting HPC and Analysis Options for setting MPI Licensing and the appropriate Ansys Windows or Linux Installation Guide for details on installing MPI.

Beta Feature: Parallel Component Mesh Adapt for 3D Component Array

If you enable the Beta Feature for Parallel Component Mesh Adapt, you can have parallel mesh adaption for 3D Component arrays. This feature works only with Use Automatic Settings enabled on the active Analysis Configuration.



When running a suitable design with this feature enabled, the Solution Profile will show multiple frequencies being adapted at the same time.

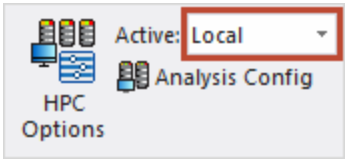
Configure a Distributed Analysis

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.



Distributed Analysis Configuration

After you have selected the active distributed machine configuration, you can edit its configuration in the **Analysis Configuration** window.

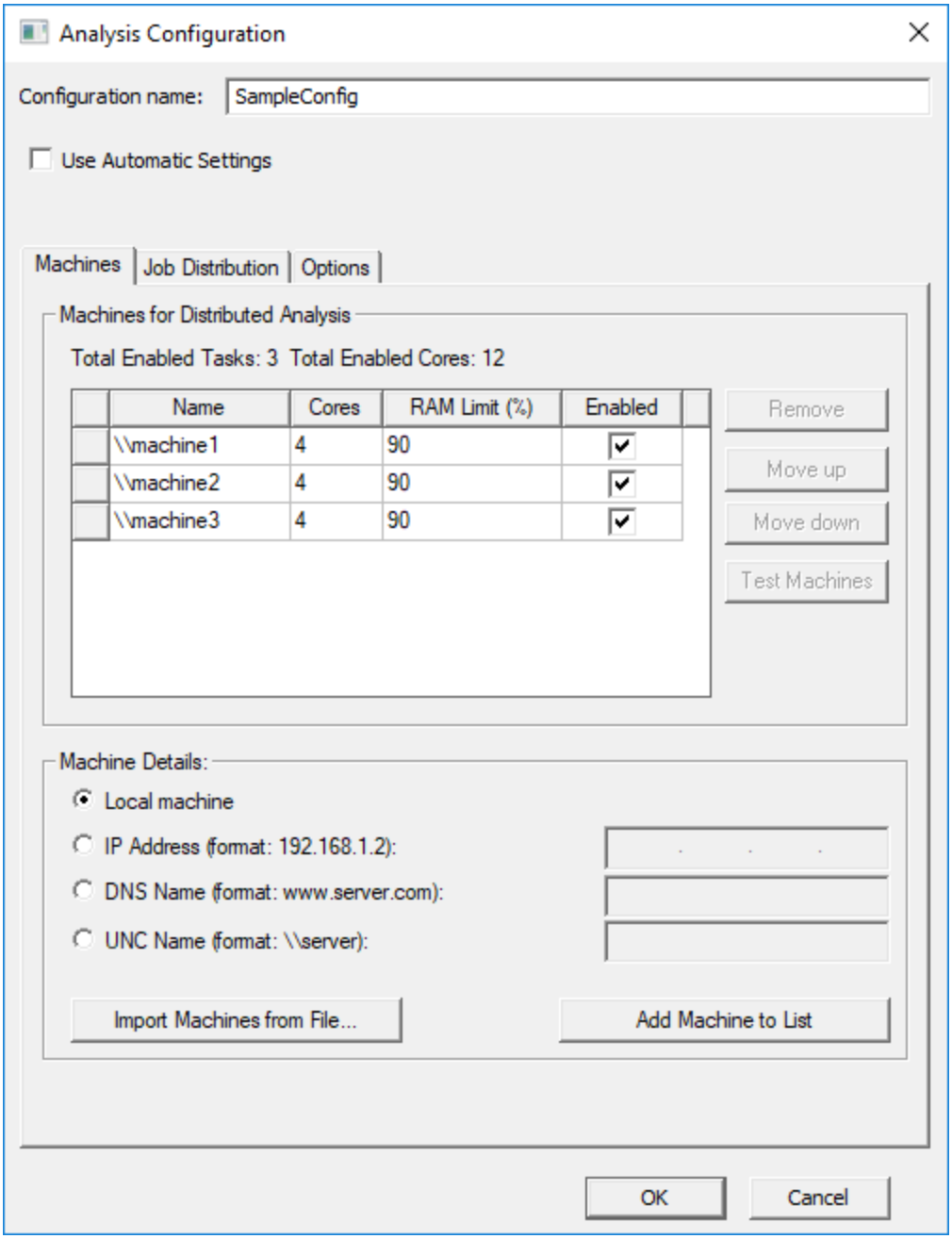
To edit the configuration:

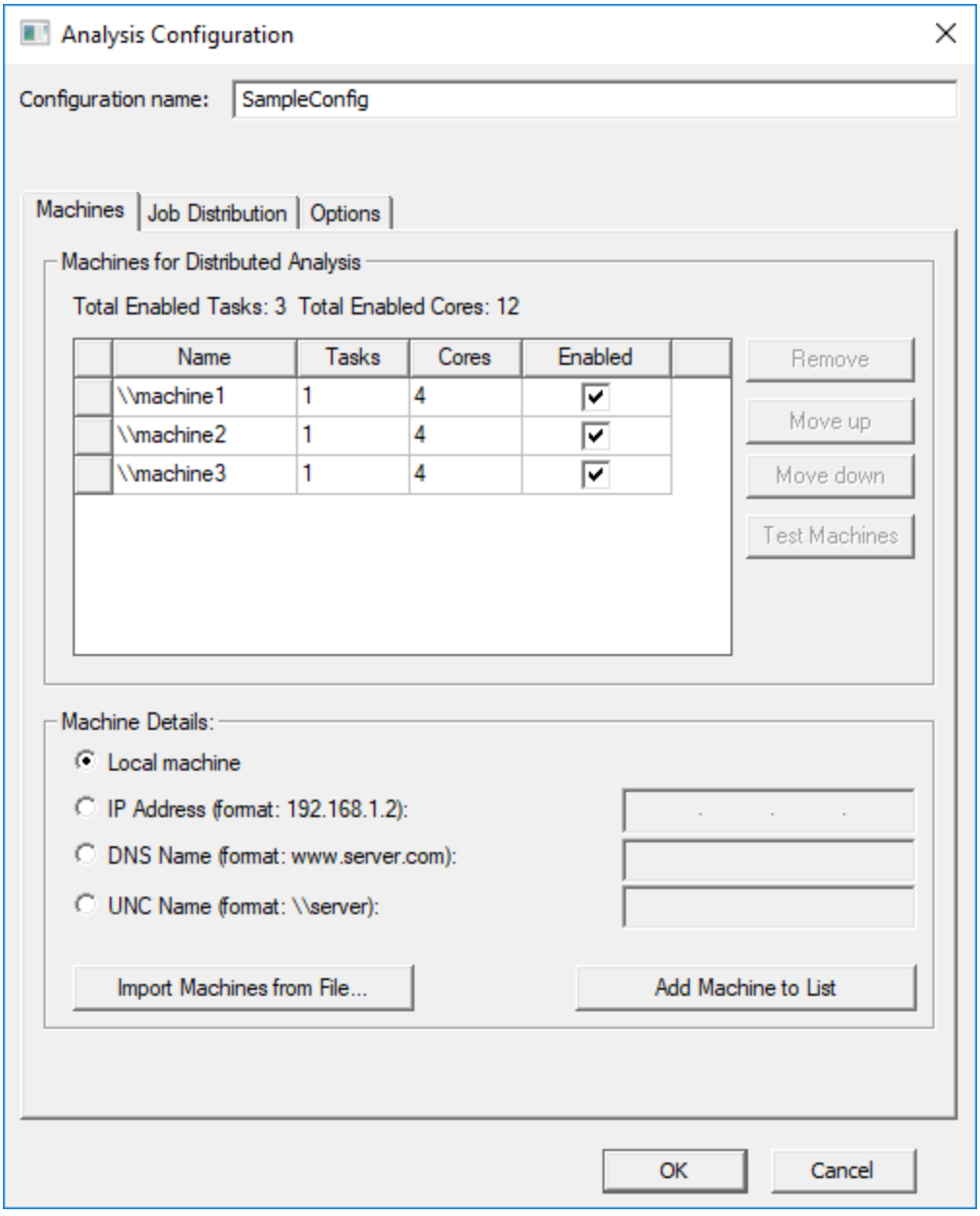
1. Access the **Analysis Configuration** window one of three ways:
    - From the **Simulation** tab, click **Analysis Config**.
    - Click **Tools** > **Edit Active Analysis Configuration**.

- From the **HPC and Analysis Options** window (opened from the **HPC Options** icon on the **Simulation** ribbon or **Tools** > **Options** > **HPC and Analysis Options**), select the configuration and click **Edit**.

> **Note:**
>
> You can also access the Analysis Configuration window from the **HPC and Analysis Options** window by clicking **Add** or **Copy**. When using **Add** or **Copy**, the steps are the same as below, but you will need to specify a **Configuration Name**. The name cannot be empty and it cannot be a reserved name.

The **Analysis Configuration** window appears.





> **Note:**
>
> Available options may vary slightly depending on the solver and design.
>
> 2D Extractor does not permit Automatic Settings.

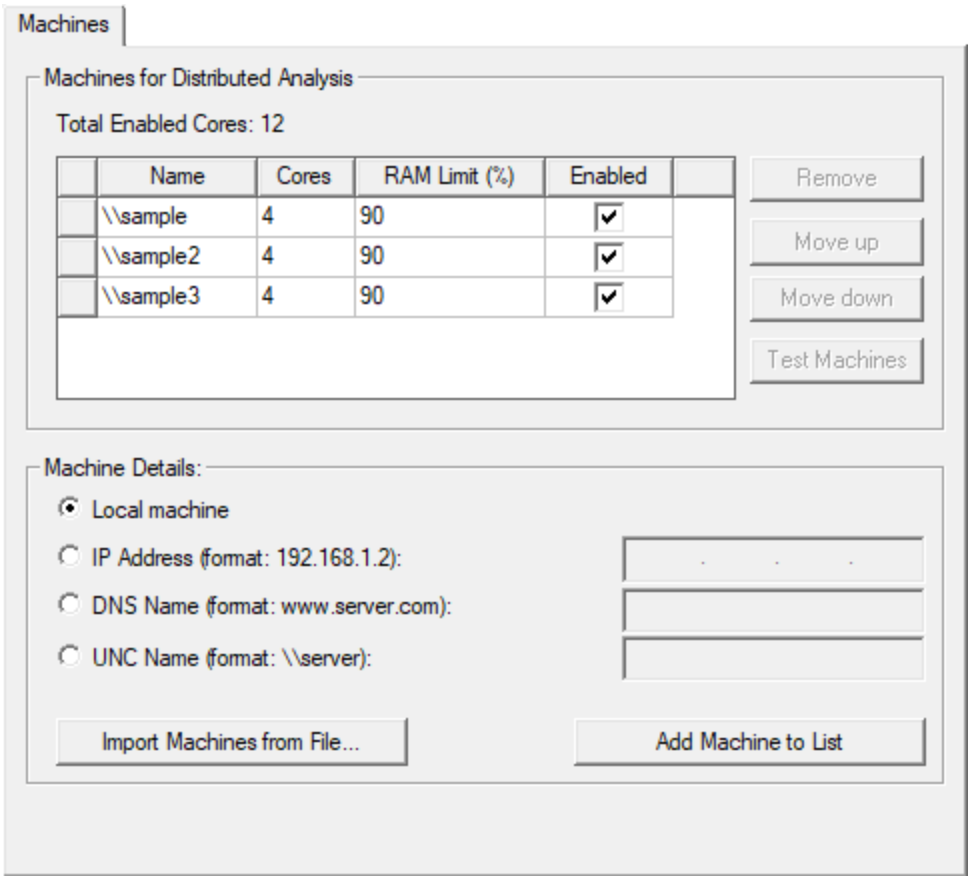
2. If automatic settings are supported, the **Use Automatic Settings** check box allows you to select this option and set parameters based on the best use of available resources for the

current analysis. Deselecting it enables the **Job Distribution** tab, where you can assign resources manually.

3. Use the tabs to navigate through the rest of the settings:
   - **Machines** – contains the machine list. You can add, enable, remove, test, and reorder machines from the list.
   - **Job Distribution** – when **Use Automatic Settings** is deselected, this tab appears and allows you to select job distribution settings manually.
   - **Options** – allows you to specify additional options, depending on the design type.
4. Click **OK** to save the configuration settings.

Distributed Analysis Configuration - Machines Tab

On the **Analysis Configuration** window, the **Machines** tab allows you to provide machine information, either by specifying remote machine details, or by importing a list of machines from a file. You can then add, enable, remove, test, and reorder machines from the list.



Manually Adding a Machine

You can add the **Local machine**, or add a machine by **IP Address**, **DNS Name**, or **UNC Name**. Select the applicable radio button, enter the machine's information in the specified format, then click **Add Machine to List**.

> **Important:**
>
> The remote machines must have the same Ansys Electromagnetics Suite version installed on the same OS version, and have active RSM service.

The added machine(s) will appear under **Machines for Distributed Analysis**.

Importing a Machine List

You can import a machine list from a text file. Each line of the file should contain a machine's information in the following format:

```
<MachineName>:<NumTasks>:<NumCores>
```

Machine List Details

The **Machines for Distributed Analysis** area lists machines in the order in which you entered them, irrespective of the load on the machines. To control the list order, select one or more machines, and use the **Move up** and **Move down** buttons. To remove one or more machines, select the machine(s) and click **Remove**.

The following columns can appear in the list:

- **Name** – the machine name.
- **Tasks** – this column allows you to specify the number of tasks a given machine will perform simultaneously. *Each separate solver or instance counts as one task.*

  If you have selected **Use Automatic Settings**, this column does not appear because there is no need for it.

- **Cores** – specifies the total number of cores that will be used on the given machine. The total number of Tasks and Cores are described just above the machine list.

  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. *The number of Cores must always be greater than or equal to the number of Tasks.*

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.

- **RAM Limit (%)** – specifies the maximum percentage of each machine's RAM you would like to be in use by the solver.
- **Enabled** – use the check boxes to enable or disable machines.

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) from the Toolbar menu and you launch multiple analyses from the same UI, Ansys Electromagnetics Suite solvers select the machines that are running the fewest number of engines 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).

Testing 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 that no other Ansys EM processes are running on the machine. To do this, select one or more machines and click **Test Machines**.

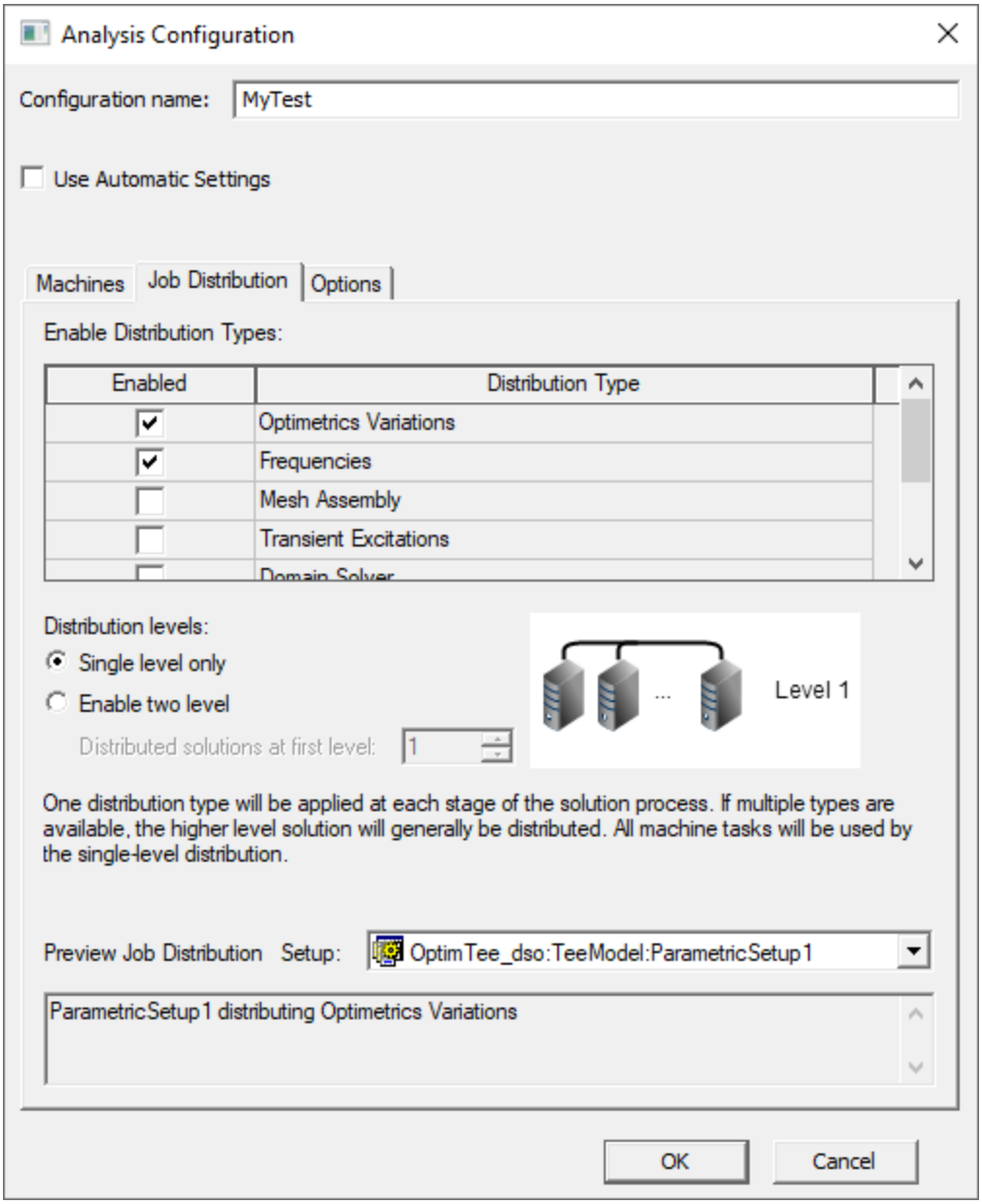
The **Test Machines** dialog box appears. The test goes through the current machine list and gives a report on the status of each machine. A progress bar shows testing progress. When the test is complete, click **OK** to close the dialog box. If you need to disable or enable machines based on the report, you can do so in the **Analysis Configuration** window.

Distributed Analysis Configuration - Job Distribution Tab

On the **Analysis Configuration** window, the **Job Distribution** tab allows you to manually enable specific types of job distribution and enable multi-level solves.

The **Job Distribution** tab is disabled if you selected **Use Automatic Settings**.

> **Note:**
>
> Different design types have different job distribution types.



The job distribution list allows you to specify which job distribution types to allow for the current analysis configuration. Use the check boxes enable and disable distribution types. At solve time, Q3D Extractor automatically selects the best distribution type from the enabled types.

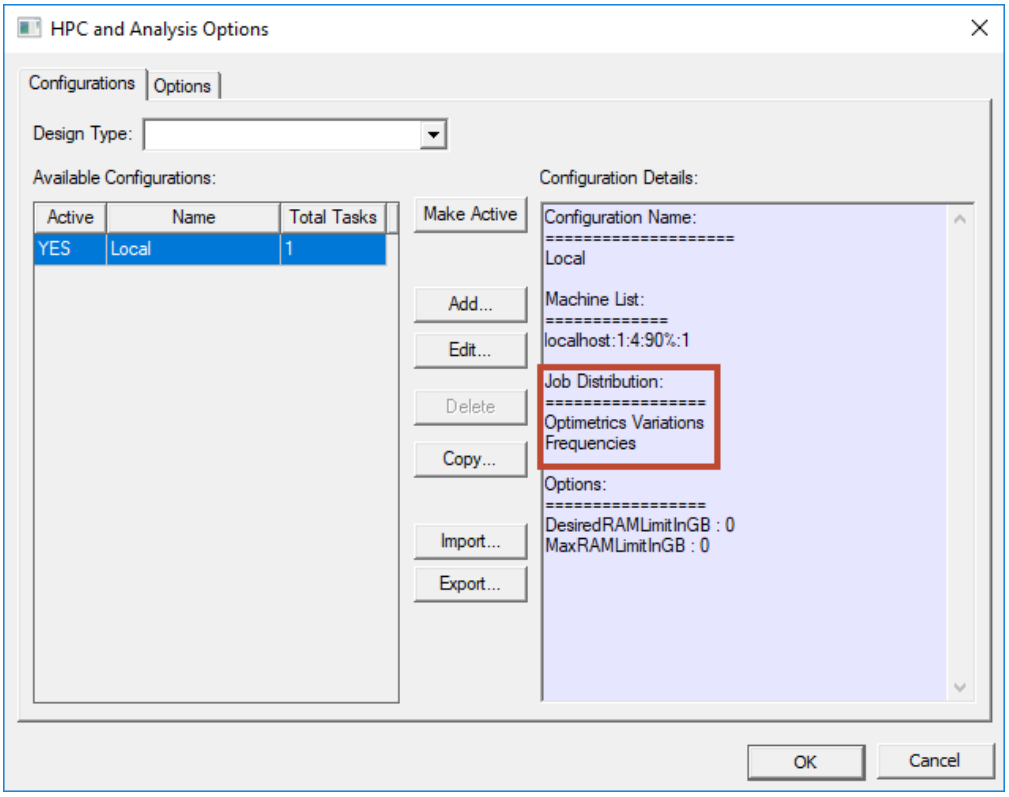
Enabling a distribution type does not mean it will be used. It must be also allowed by the solve setup. If you enable a distribution type for a given setup, and distribution is allowed, the preview window updates to describe the distributions. Note that enabled distribution types apply to all setups of the given design type, so it is possible for different setups in a design to be solved using different distribution types.

The concurrent initial mesh generation workflow with Distributed Mesh Assembly relies on the MPI based distribution technology inside the MeshAssemblyManager. The decision whether to launch sequential or parallel mesh generation is based on the combination of the number of individual meshes present, the HPC setting, and the number of tasks available.

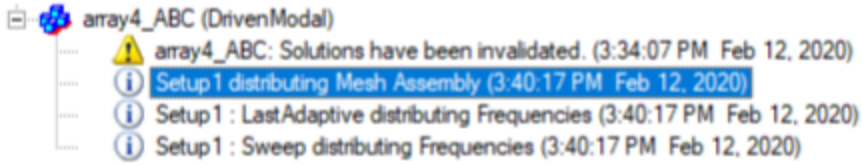
- If there's only one single mesh or if curvilinear contact exists: then normal initial mesh workflow with single G3dmesher will be launched.
- If there are more than one individual mesh and no curvilinear contact exists:
    - If Use Automatic Setting is not selected, under the **Job Distribution** tab, if "Mesh Assembly" is not enabled, then meshes will be generated sequentially. If "Mesh Assembly" is enabled and user assigns more than 1 tasks, then parallel mesh generation will be launched.
    - If Use Automatic Setting is selected and user assigns more than 1 core, then parallel mesh generation will be launched.
    - If Use Automatic Setting, the number of MPI tasks will be the number of cores user assigned or the number of geometries to be meshed, whichever is smaller. If there are more geometries than cores, the geometries will be dynamically assigned to tasks. If there are more cores than geometries, the remaining available cores will provide some of the tasks with multi-thread capabilities.

All the detailed progress information from the Mesher is suppressed and the progress will report the number of meshes being finished. All the mesh profiles will be available under profile report and the mesh feedback will also be available under mesh feedback tool.

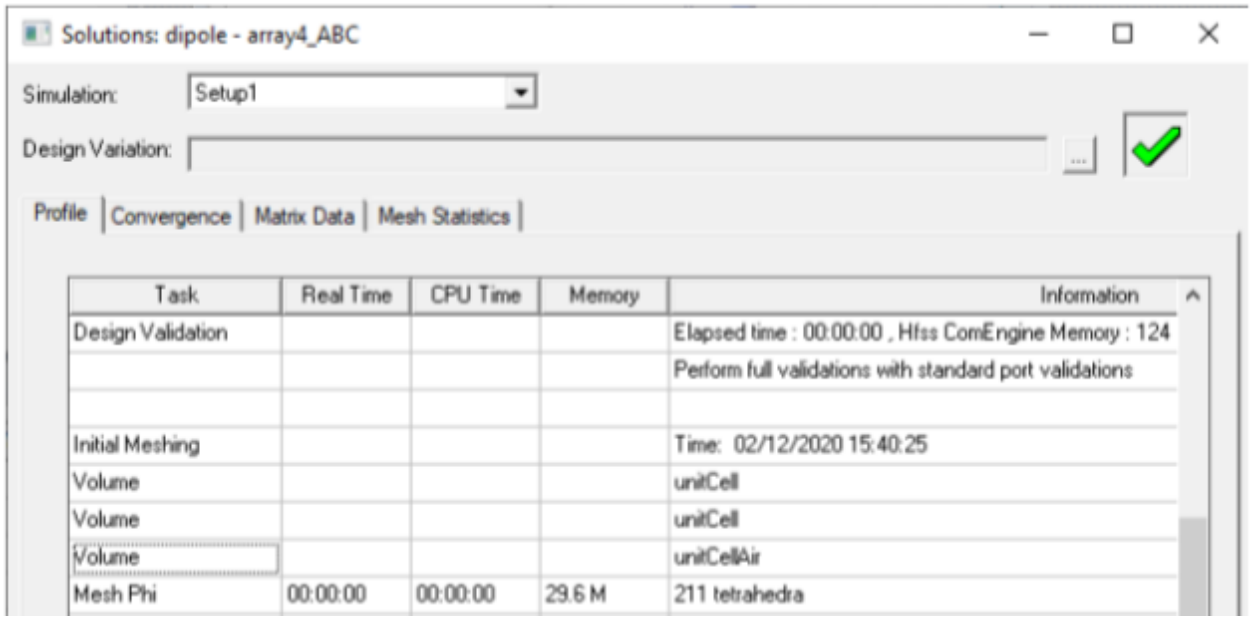
Enabled distribution types are listed when you select a configuration in the **HPC and Analysis Options** window.



When you run a simulation, the **Messages** window describes distributions.



When you view the Solution Profile, distributions for unitCell show as parallel Volume Tasks.



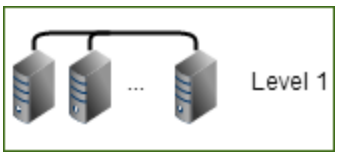
Distribution Levels

For products and designs that support two-level distribution, you can select either single or two-level distribution.

If you select single level, 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

In a single-level distribution, one distribution type is applied at each stage of the solution process. Common stages include LastAdaptive, Sweep, and Parametric. All machine tasks will be used by the single-level distribution.



Supported distribution types include:

- Optimetrics Variations
- Frequencies
- Mesh Assembly

- Transient Excitations
- Domain Solver
- Iterative Solver Excitations
- Direct Solver Memory

Solver distributions require MPI. See: Setting HPC and Analysis Options.

Parallel distribution types (such as Optimetrics Variations, Frequencies, and Excitations) do not require distribution. If these types are not able to distribute, the simulation can run sequentially.

Memory distribution types (such as Direct Solver Memory and Domain Solver) are set to require distribution. If these types are enabled, the software will assume that distribution is necessary to extend the simulation scale or add fundamental solution capabilities.

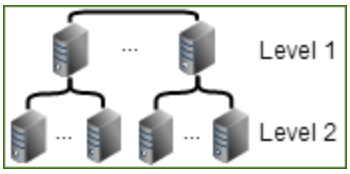
When multiple distribution types are available, the higher-level solution will generally be distributed. For example, when both Optimetrics Variations and Iterative Solver Excitations are enabled, Optimetrics Variations will be distributed. When both Optimetrics and Mesh Assembly, are selected Optimetrics are distributed. Domain Solver and Direct Solver Memory are exceptions because they are required; even though they are lower level, these types are distributed instead of parallel distribution types.

Two-Level Distributions

Selecting **Enable two-level** enables the **Distributed solutions at first level** box.

In a two-level distribution, the first level distributes the specified number of solutions. Each solution will then use a subset of machine tasks to distribute the second level. A solver distribution type must be available for the second level; otherwise, single-level distribution will be applied.

For two-level distribution, the total number of tasks must be greater than or equal to the number of tasks for level 1.



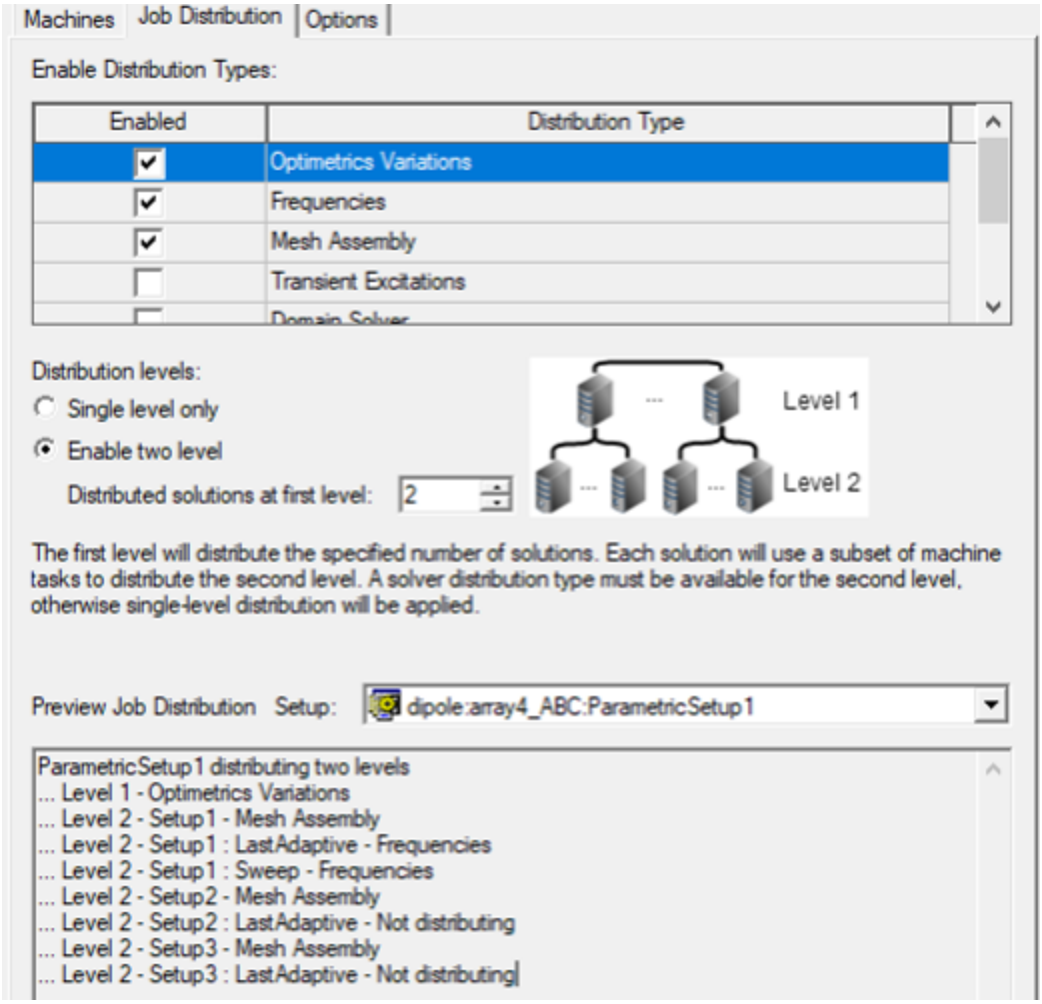
The following are examples of two-level distributions:

- A parametric setup distributing Optimetrics variations as level 1 and iterative solver excitations as level 2.
- A parametric setup for a non-transient problem distributing optimetrics variations as level one and frequencies as level 2.
- A frequency sweep distributing frequencies as level 1 and direct solver memory as level 2.

- A parametric setup for a transient network problem distributing optimetrics variations as level one and transient excitations as level 2.

The **Analysis Configuration** window displays a preview of the job distribution, if allowed for that configuration:

Here is another example:

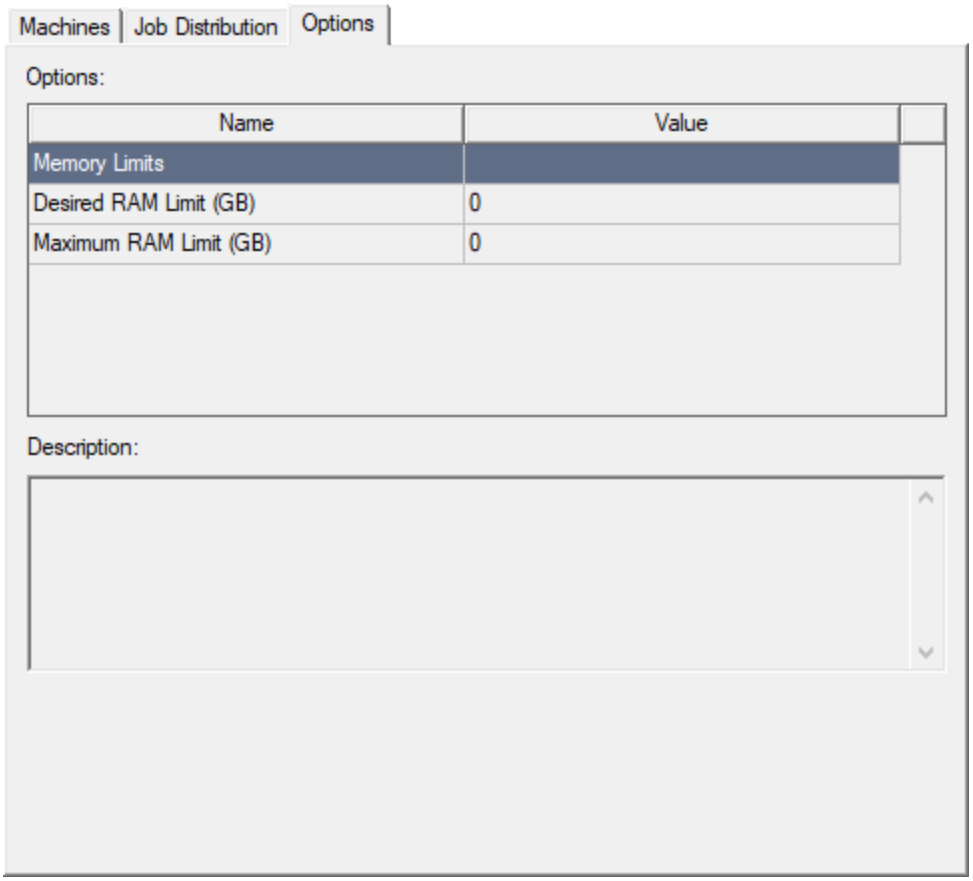


For more information, see: Two-Level Distribution Guidelines.

Distributed Analysis Configuration - Options Tab

On the **Analysis Configuration** window, the **Options** tab allows you to specify options for the current analysis configuration.

> **Note:**
>
> Different design types have different options.



These options settings will be in effect only when all the following are true:

- A design is being solved whose design type matches the analysis configuration design type.
- The analysis configuration is the active configuration for its design type.
- You have not specified corresponding `batchoptions` on the command line.

Relation to Batchoptions

Command line `batchoptions` can be used to override the options specified by the active configuration.

Analysis configuration option settings can be overwritten by specifying the option name and value inside a `-batchoption` string.

To add batchoptions from the UI, click **Q3D Extractor** > **Submit Job**. Under **Analysis Options**, click **Add**. This opens the **Add Batchoption** window.

See: Running Electronics Desktop from the Command Line and Batchoptions Command Line Examples.

Adding Configurations or Accepting Edits

Click **OK** to accept the changes and close the **Analysis Configuration** window. Only machines marked **Enabled** appear in the machine list.

Regardless of the machine(s) on which the analysis is actually run, the number of processors and RAM Limit (%) settings, and the default process priority settings are now read from the machine from which you launch the analysis. See: Setting HPC and Analysis Options.

Selecting Optimal Configurations for Distributed Analysis

This section provides some basic guidelines for configuring a distributed analysis, including how to choose the number of tasks for Distributed Direct and Distributed Iterative solvers, and when to use two-level distribution. In this section, assume that a fixed number of machines and cores are available, and that the goal is to use them as efficiently as possible.

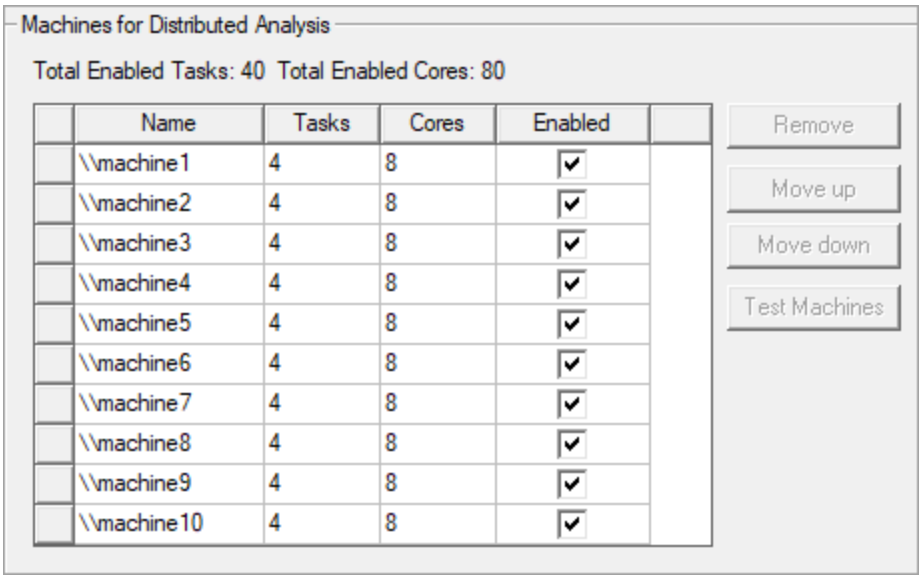
In the majority of situations, configuring a Distributed Solution Option (DSO) in combination with the multiprocessing option to take optimal advantage of the available hardware results in improved speed.

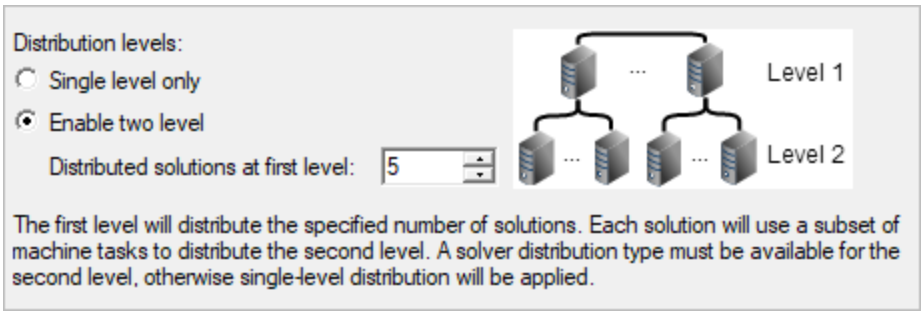
> **Note:**
>
> This topic assumes that there is enough hard drive space and memory for DSO simulations. For multiple DSO simulations on a single machine, the total memory needed is the sum of the memory used by each simulation. For example, consider a discrete frequency sweep in which each frequency point needs 3.5GB. A quad core system with 8GB of RAM would rely heavily on swap, which is highly inefficient. In this case, updating the number of tasks in the machine list is ideal.

Two-Level Distribution Guidelines

Ansys recommends that you focus resources for the highest level of distribution (level one). In other words, allocate the resources so that a maximum number of frequencies or variations are solved in parallel. First, determine the minimum number of machines required to solve one frequency or variation based on your prior knowledge of the type of model being solved. Then allocate the resources so that a maximum number of solutions can be solved in parallel.

For example, suppose that a given model requires at least 100 GB of memory to solve using DDM. Assume that you have 10 machines, each with 60 GB and 8 cores. One solution will require two machines for the DDM solver. This means that, at most, 5 frequencies can be distributed. Each DDM solution requires at least 3 tasks, but since more cores are available you should set 4 tasks per machine, resulting in 2 cores for each Domain . A machine configuration based on the machines and number of tasks described above is shown in the figures below.





Distributed Direct Solver Guidelines for HPC Configuration

The dependence of Distributed Direct solver on number of tasks is very similar to the Q3D Extractor solver. For distributed direct solver, the number of tasks increases the total memory but also typically increases the speed of the simulation.

Some general guidelines:

1. If memory usage is critical, use 1 task/machine using all cores.
2. If memory usage is not critical, use 4 cores/task for optimal speed.

Distributed Iterative Solver Guidelines for HPC Configuration

The distributed iterative solver distributes excitations for faster simulations and to access less memory to solve a large problem. Each task corresponds to a number of excitations and, therefore, more excitations are solved in parallel when the number of tasks increases.

The first consideration for the number of tasks is the memory requirement. Each task uses a similar amount of memory, and the total memory usage doubles as the number of tasks doubles. Thus the number of tasks on each compute node should be restricted to avoid an "Out of Memory" failure.

The second consideration for the number of tasks is the number of excitations. Q3D Extractor designates one of the tasks as the master, which does not participate in the iterative process. ***Thus the number of tasks should be no more than the Total Number of Excitations + 1***.

In summary, assuming that you have enough memory, if the number of excitations N is less than the total number of cores M and you choose 4 cores per task, then the number of tasks is ceil (N/4) + 1, where ceil() stands for the ceiling function and the extra "1" is dedicated to the master task; if the number of excitations N is more than the total number of cores M and you choose 4 cores per task, then the number of tasks is ceil(M/4). For example, if you have a design with 64 excitations and you have 32 cores available, you could define a possible setup having 8 tasks with 4 cores per task.

Distributed Mesh Assembly Guidelines for HPC Configuration

The concurrent initial mesh generation workflow with Distributed Mesh Assembly relies on the MPI based distribution technology inside the MeshAssemblyManager. The decision whether to launch sequential or parallel mesh generation is based on a combination of the number of individual meshes present, the HPC setting, and the number of tasks available.

- If there is only a single mesh or if curvilinear contact exists: the normal initial mesh workflow with single G3dmesher will be launched.
- If there is more than one individual mesh and no curvilinear contact exists:
    - If **Use Automatic Setting** is *not* selected under the **Job Distribution** tab, and if **Mesh Assembly** is not enabled, then meshes will be generated sequentially. If

Mesh Assembly is enabled and user assigns more than 1 tasks, then parallel mesh generation will launch.

- If **Use Automatic Setting** is selected and user assigns more than 1 core, then parallel mesh generation will launch.
- If **Use Automatic Setting** is selected, the number of MPI tasks will be the number of cores user assigned or the number of geometries to be meshed, whichever is smaller. If there are more geometries than cores, the geometries will be dynamically assigned to tasks. If there are more cores than geometries, the remaining available cores will provide some of the tasks with multi-thread capabilities.

All detailed progress information from the mesher is suppressed and the progress will report the number of meshes being finished. All mesh profiles will be available under profile report and mesh feedback will also be available under mesh feedback tool.

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. See Running Ansys Electronics Desktop 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 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.

  > **Note:** RAMLimitPercent and RAMLimitPerCoreinGB are not supported for 2D Extractor designs.

- **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 valid batchoptions specified in the command line will override the associated registry settings. Batchoptions 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 Desktop 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, SGE or SLURM scheduler, and only if the MPI Vendor is 'Intel'.

When submitting a job using the AnsysEM 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, SGE or SLURM scheduler, and only if the MPI Vendor is 'Intel'. To specify the value 'Scheduler' for this option for a 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.

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, Microsoft Excel). 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 rerun the analysis.

> **Note:**
>
> For a machine with *n* cores, it should be expected that running *n*, single core, distributed simulations in parallel will encounter additional overhead due to the need to spawn *n* unique solve processes. Therefore, it should not be expected to observe an *n* times speed up over the time taken to run *n* analyses in series with a single core. The relative impact of this overhead increases as the size of the simulation and time to complete a single solve shrink.

The basic Large Scale DSO process involves

1. Preparing the model for Large Scale DSO Analysis.
2. Submitting the Large Scale DSO job through **Tools** > **Job Management** or via command line.

3. Monitoring the job's progress.
4. Postprocessing the results.

For details, refer to the following sections:

- Prerequisites for Large Scale DSO
- Job Management Interface for Large Scale DSO
- Command Line Syntax
- Deployment/Configuration
- Tutorial Example for HFSS
- Job Outputs
- Job Monitoring
- Process Limits for Large Scale DSO on Windows
- Known Issues/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 reassigned 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.

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.
- 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, either Windows or LINUX.
- On the Windows platform, Ansys RSM is started as an admin account, rather than as a system account.

  > **Important:**
  >
  > Large Scale DSO does not support RSM Service running with system login credentials.

- On each machine of the cluster, the 'desktopjob' application is registered with Ansys RSM service using the command shown below:

  Windows: `<installation-directory>/<platform> desktopjob.exe -regserver`

  LINUX: `<installation-directory>/<platform> desktopjob -regserver`

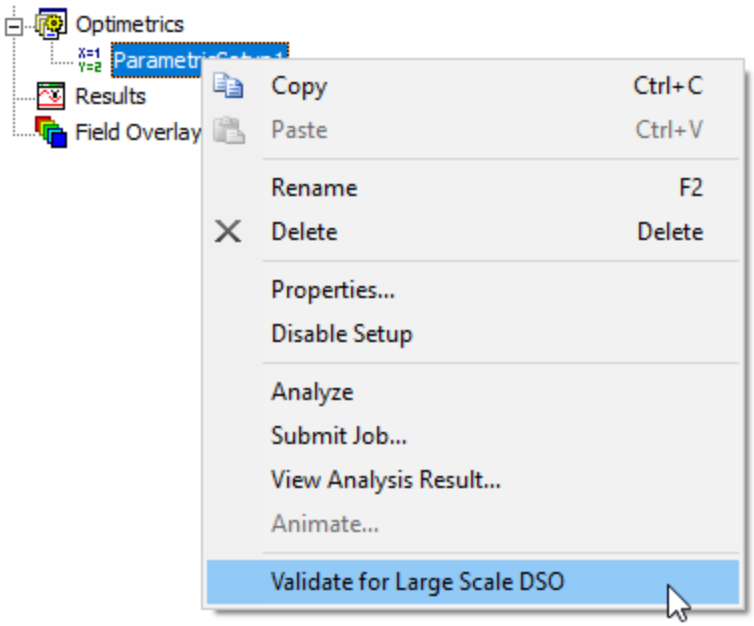
Job Management Interface for Large Scale DSO

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.

When you have selected the scheduler, 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. Close Electronics Desktop.
4. Copy the project (or folder, if the project references external files) from a personal workstation to a shared drive on cluster.

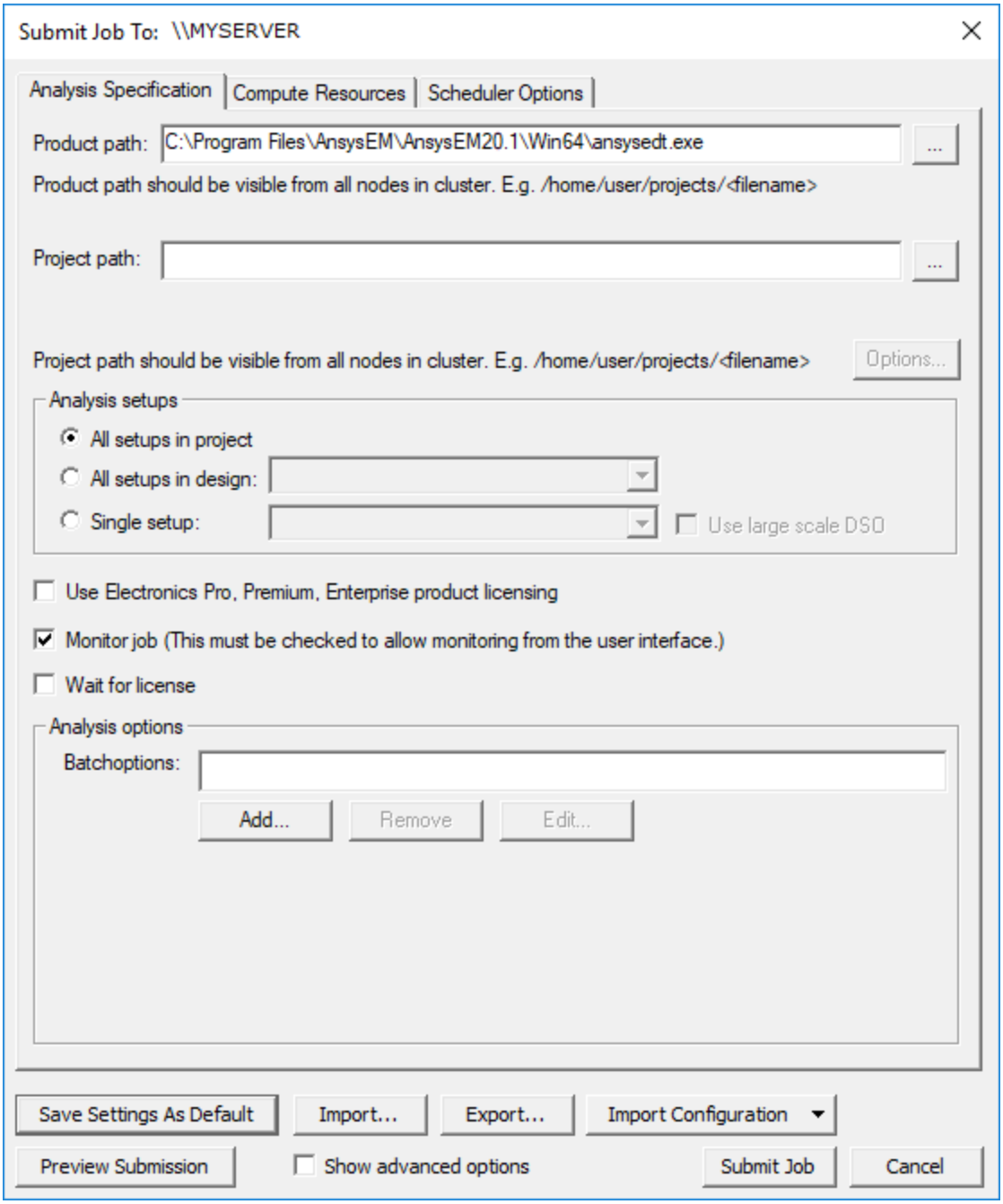
   In the RSM environment, you must specify a machine list (See: HPC and Analysis Options).

   In a Linux scheduler environment, a cluster must have a designated Postprocessing Node.

5. 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).
6. Launch Electronics Desktop on that node and open the project.
7. Verify that the model has been prepared correctly.

8. Close the project.
9. Submit the job one of several ways:
    - Click **Tools** > **Job Management** > **Submit Job**.
    - Click **Project** > **Submit Job**.
    - Click **Q3D Extractor** or **2D Extractor** > **Submit Job**.
    - Select the **Simulation** tab and click the **Submit** icon.

    The **Submit Job To** window appears, on the **Analysis Specification** tab.



Note:

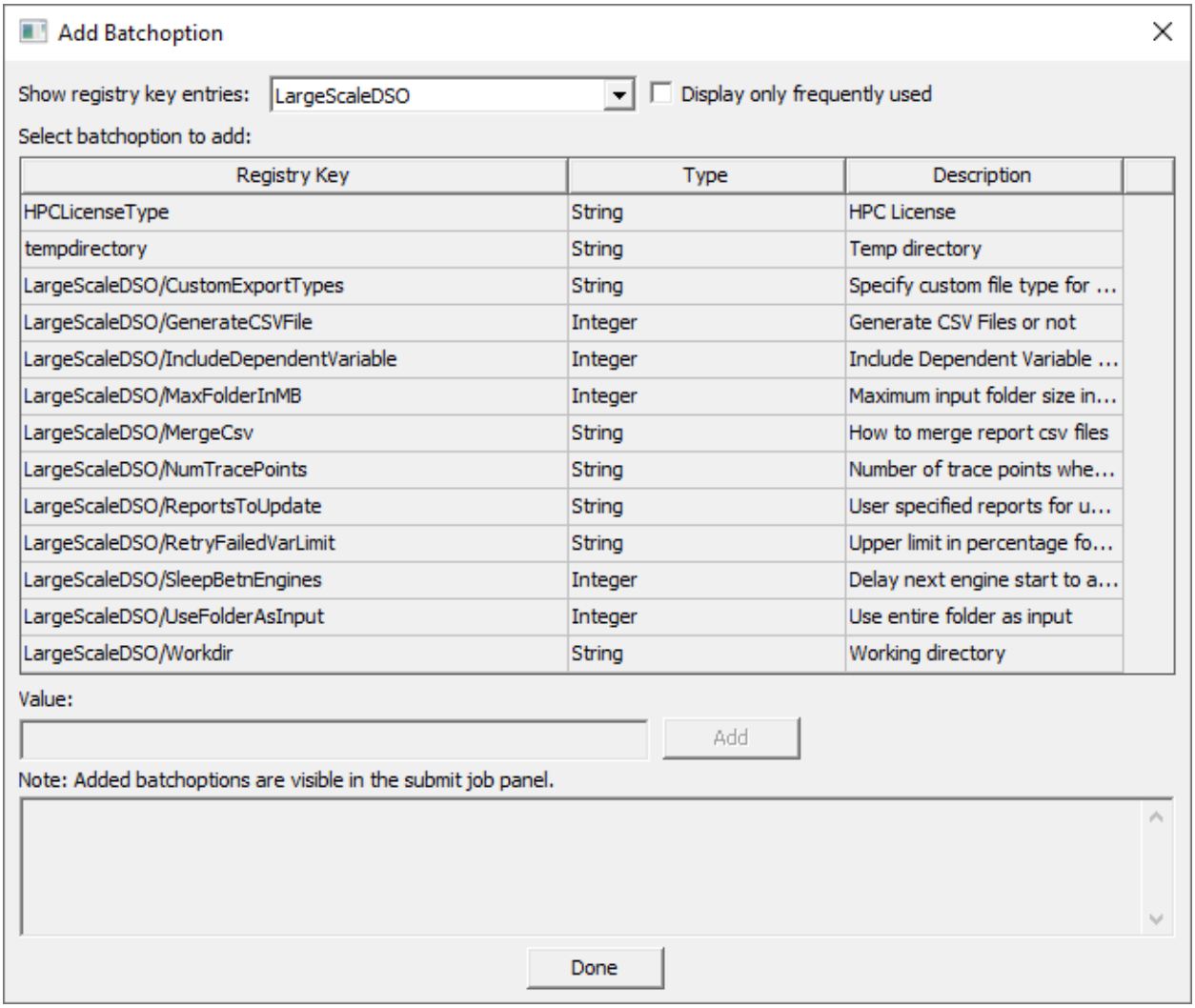
Options vary slightly depending on the selected scheduler. See: HPC Integration.

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.

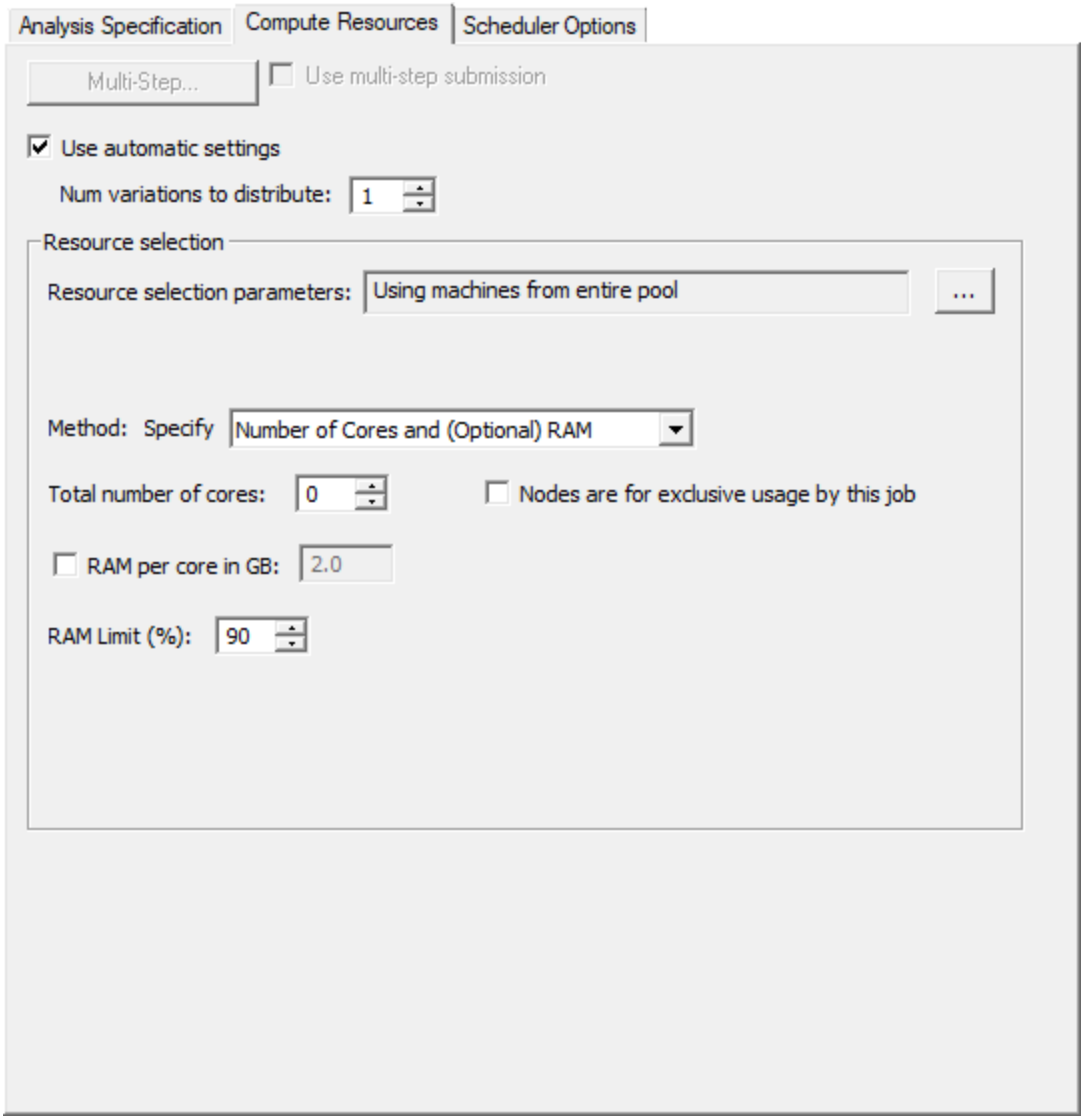
    - Determine whether to **Use Electronics Pro, Premium, Enterprise product licensing**, whether to **Monitor job**, and whether to **Wait for license.** If you wish to monitor jobs through **Tools** > **Job Management**, you must enable job monitoring.
    - Add **Batchoptions**, if desired.

      The following shows batchoptions specific to Large Scale DSO.



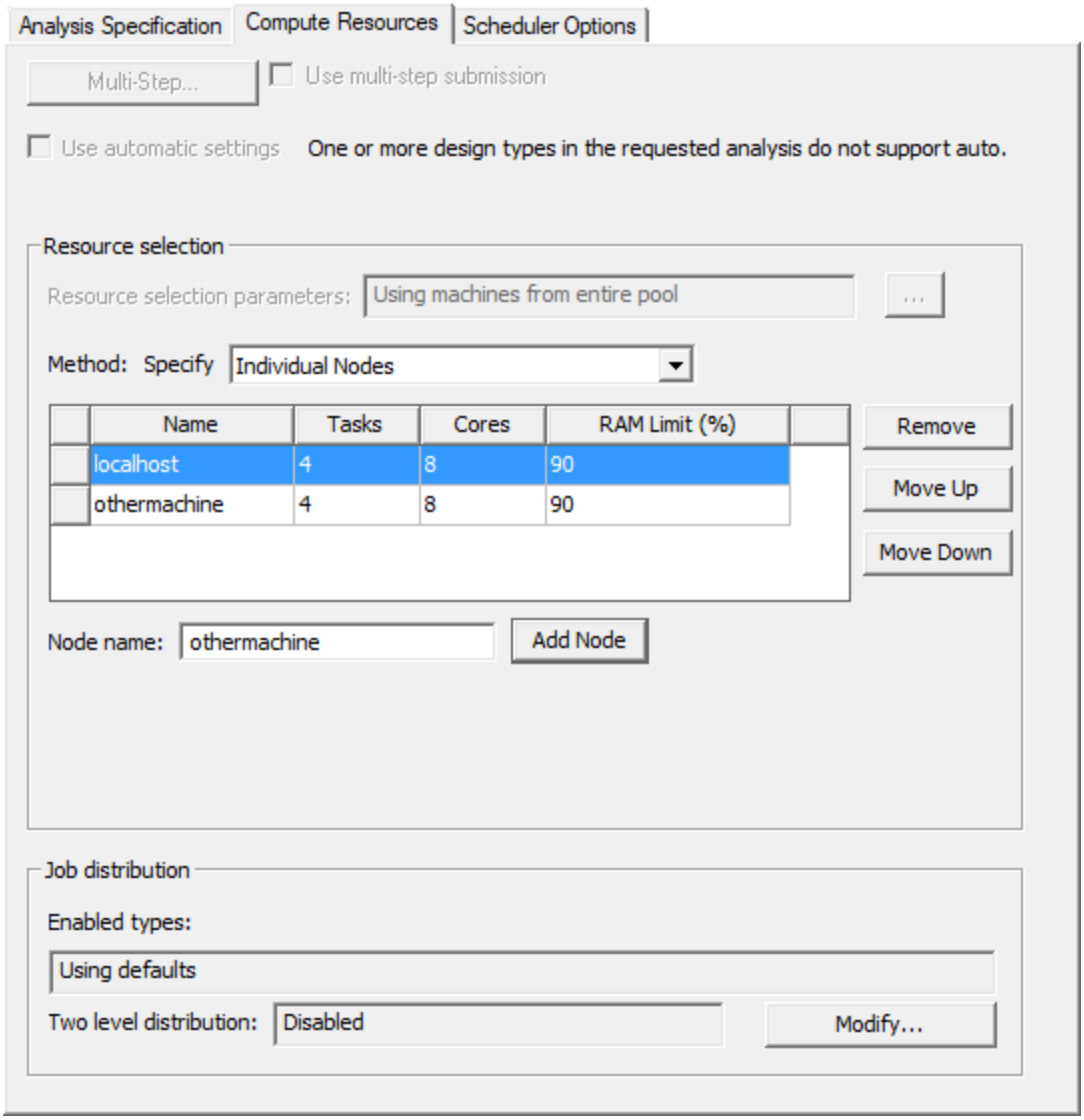
11. Select the **Compute Resources** tab.

    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 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 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**

  Prints the help text.

- **-cmd**

  Specifies the command to run.

  Available choices: dso

- **-ng**

  Runs the analysis in non-graphical mode.

- **-monitor**

  Outputs progress and messages to standard output/error.

- **-waitforlicense**

  Queues the job until licenses are available.

- **-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**

  Overrides the Tools/Option entries through either a batchoptions file or batchoptions string.

  -batchoptions specific for Large Scale DSO include:

  To retry failed variations, new batch option LargeScaleDSO/FailedVarRetryCount.

  For each task to re-simulate its failed variations when its assigned variations are finished, specify the FailedVarRetryCount batch option with positive integer. Default 0; Value-Zero(0) will disable the re-simulation for failed variations. Value ranges from 0 or positive integer.

  Redistribution batch option, LargeScaleDSO/VarRedistribution .Value 0 disables redistribution (default). Value 1 enables redistribution.

  Redistribution limit batch option, LargeScaleDSO/RedistributionLimit. Minimum estimated remaining time (in minutes) for variations to redistribute to another task. Value must be positive integer. Default 3 minutes

  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 **Ansoft 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 3:

```
-machinelist "list=m1:1:3,m2:2:2"
```

Example 4:

```
-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.

  Manual Example:

  ```
  -machinelist "Num=10"
  ```

  Auto Example:

  ```
  -machinelist "NumCores=40"
  ```

  Note: In a scheduler environment, the machinelist argument can be omitted, but the results may not be as expected or desired. If there is no machinelist argument, the scheduler-assigned resources will be used, assuming one core per task. E.g., if the scheduler assigns 40 cores, there will be 40 variations solved in parallel, with only one core used per variation. Since one core per task may not make sense in many cases or may not be what the user desires, a warning message that machinelist is missing will be output, but the job will proceed.

- **-auto**
  - Run the leaf jobs in auto mode. See Submitting Large Scale DSO Job Examples.
- **-numdistributedvariations**:

  Specify the number of parallel leaf jobs to run in auto mode. Required when running under a scheduler. If not specified when not running under a scheduler, then the number of tasks for each machine must be specified in the machine list. The total number of tasks will be the number of parallel variations (leaf jobs) to run.

  Example:

  ```
  -machinelist NumCores=40 -auto -NumDistributedVariations 10
  ```

- **-usefolderasinput**

  Choose this option if the job's input represents an entire folder rather than just the project file.

- **-maxfolderInMB**

  Specify the maximum size input folder that is allowed for a valid job (in MBytes). By default, the maximum size allowed for input is 10MB. Specify a value of 0 to remove this size restriction and enable inputs of any size. This option applies when `-usefolderasinput` is used.

- **-workdir**

  Specifies 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 traced values and 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;ReportNam
e1:Trace1:100;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**

  Solves the specified parametric setup.

  Syntax for the setup:

  `<design-name>:Optimetrics:<parametric-setup>`

Large Scale DSO Job Outputs

A Large Scale DSO analysis does not support the output of full parametric results. Instead, it extracts a subset of results using predefined rectangular plots, which are created by the user before running the job. The extracted columns of data are saved as CSV files. Typically, there is one CSV file per-trace, per-variation. 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).

> **Note:**
>
> Non-Rectangular plots of the design (for example, statistical eye and digital plot) are not extracted.

Output Location and Organization

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\tee.aedt` is `\\shared\projects\tee.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. Each variation's folder contains a CSV file for each trace.

CSV File Contents

The initial header rows of the CSV file define the solved variation. For each such row, the first column contains a variable name and the second contains a variable 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.

These examples provide context:

- **Traces of an S-parameter Report** – The data portion of the CSV file contains two columns of data: the first contains Freq values and the second contains values for the trace's s-parameter component.
- **Trace of a Far Field Report** – Suppose there is a far field report with a trace (magrE), whose primary sweep is phi and whose 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 containing phi values, the second containing magrE values for the first value of theta, and the third column containing values for second value of theta. The magrE output columns are titled as 'magrE_crv1' and 'magrE_crv2' respectively.

- **Advanced 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.

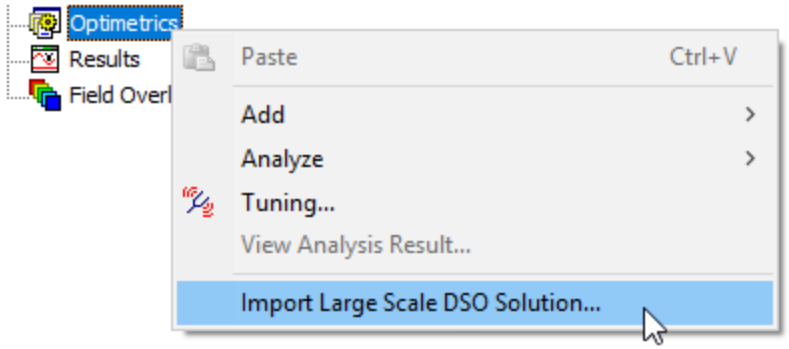
Postprocessing Large Scale DSO Dataset Solutions

In Ansys Electronics Desktop, the **Import Large Scale DSO Solution** command allows you to post-process Large Scale DSO dataset solutions.

Importing a Large Scale DSO Solution

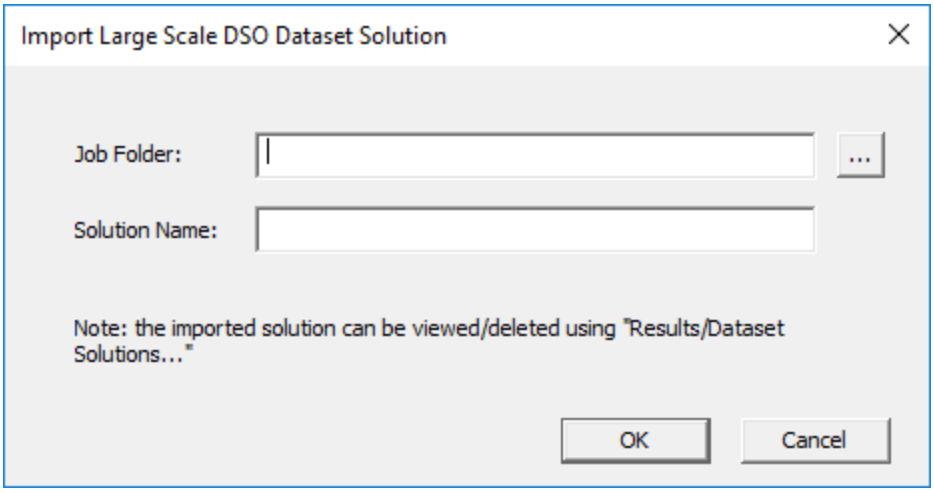
To import a Large Scale DSO solution:

1. In the Project Manager, right-click **Optimetrics** and select **Import Large Scale DSO Solution**.



> **Note:**
>
> You can also access this window by right-clicking **Results** > **Dataset Solutions**.

The **Import Large Scale DSO Dataset Solution** window appears.



2. Click the ellipsis button (**...**) and browse to select a job folder. To select a results dataset, double-click the results folder name. Results folders are organized by the scheduler prefix (for example, RSM) and job number.
3. Click **Open** to return to the **Import Large Scale DSO Solution** window.
4. Review the **Job Folder** path and **Solution Name**, then click **OK**.

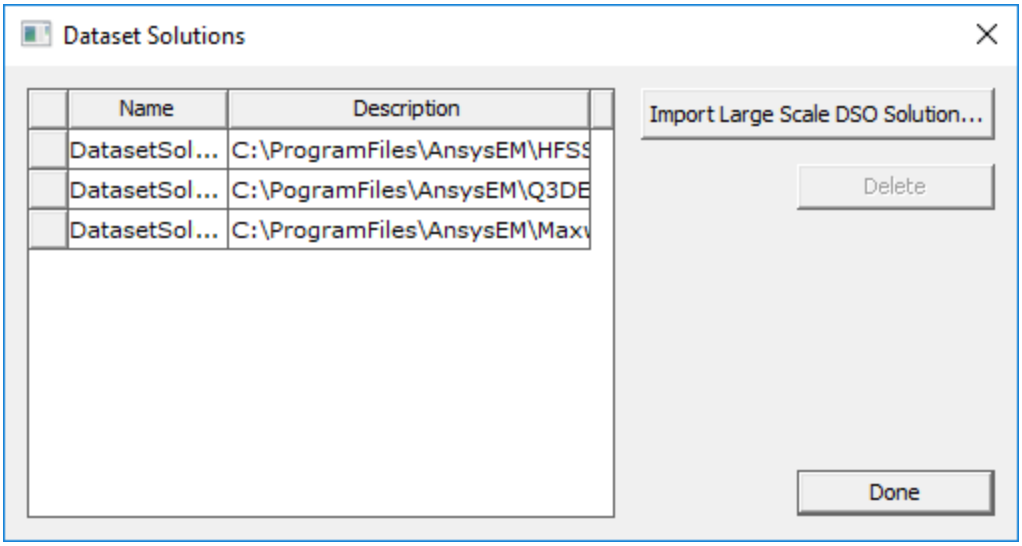
   The dataset is imported.

Viewing Imported Solutions

To view a list of imported datasets:

1. In the Project Tree, right-click **Results** and select **Dataset Solutions...**

   The **Dataset Solutions** window appears.



Selecting a dataset enables the **Delete** button, which you can click to remove it.

2. Click **Done** to close the window.

Creating a Dataset Report

After importing one or more DSO solutions, you can create a dataset report.

To do so:

1. In the Project Tree, right-click **Results**. Select **Create Dataset Report** > **[Report Type]**.

   The **Report** window for the report type opens. See Creating Reports.

Cloning a Dataset Solution

If you reopen a project that was solved using Large Scale DSO, you can quickly clone the solution report:

1. In the Project Tree, under **Results**, right-click the report and select **Clone from Dataset Solution** > **[Solution Name]**.

   The cloned solution appears. You can now reuse the existing report definition rather than creating a new report.

Large Scale DSO Job Monitoring

The **Monitor Job window** allows you to monitor the progress and status of Large Scale DSO jobs, including information on variations solved so far, variations currently solving, and the number of variations remaining.

Additional Resources for Large Scale DSO Monitoring

Large Scale DSO avoids detailed intra-variation monitoring, as it increases network traffic for large-scale jobs. Additional monitoring resources include:

- **Cluster Monitoring Tools** – Standard cluster monitoring tools are ideal for job-neutral resource monitoring as they use negligible network bandwidth.
- **Detailed Monitoring of Analysis of a Variation** – For detailed monitoring, you may want to examine a job's log files. Large Scale DSO writes detailed logs about the machines where engines are running and the local storage location of per-engine distributed databases. You can log in 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, and number of engines started on this node. It is located in <workdir>/<jobid>/r<nodeIndex>. For example, <workdir>/<jobid>/r0 contains the desktopjob.log corresponding to the engines running on the first node of job, while <workdir>/<jobid>/r2 contains the log corresponding to engines running on the 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 contains the log corresponding to first engine running on first node, while <workdir>/<jobid>/r1/r2 contains the log corresponding to the third engine running on the 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> 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 desktopjob.log file 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.

Large Scale DSO Deployment/Configuration

LINUX Cluster Configuration

- 'Temp directory' configuration.

  Temp directory is either on 'local storage' or on storage that has equivalent speed characteristics. 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 example, 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.

- Ansoft RSM environment: In the case of supported scheduler environments, there is no extra configuration needed. In the case of Ansoft RSM environment, following additional steps are needed:

  Ansoft 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 Ansoft 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 Ansoft RSM environment, Large Scale DSO can only be enabled for one product.

Troubleshooting hints (Ansoft RSM environment only): "shared drive read/write" requirement is a new constraint introduced in Large Scale DSO. So if 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:

- Ansoft RSM and Ansys Electromagnetics products are either installed locally on each node of cluster OR installed on a single shared-drive available to all nodes of cluster.

- 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.

> **Important:**
>
> Ansoft 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.

Large Scale DSO Known Issues/Troubleshooting

Node Order

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, otherwise Large Scale DSO solve with RSM will fail. Set the order in the **Submit Job To** window, **Compute Resources** tab:

	Name	Tasks
	localhost	1
	othermachine	1

Parallel Task Limitation for LS-DSO Parametric Variations

There some limitations of running short parametric runs in parallel and using a subset of cores and/or running across multiple machines instead of just one. LS-DSO does help in getting close to linear scalability for parametric runs in many scenarios, it has some overheads and typically they are small in comparison to total time of run. However, a few factors in particular scenarios can make them prominent.

The startup of each task involves copying the project and launching ansysedt to solve a subset of parametric table. Typically, one task solves several variations and this startup cost is negligible in comparison to total time. Consider a case where each task is solving only single variation. Additionally the actual solve time of one variation is relatively small, 3-5 minutes. These factors make the startup cost significant. There is also some variance in solve times of different variations. When all variations are solved in parallel, the total time is determined by

slowest variation, even though the expectation intuitively might be relative to average time. That could make the perception of overhead worse in this case.

Another important factor is the number of parallel tasks on each machine relative to the total number of cores on the machine. Each task is effectively running an ansysedt of its own and solving subset of parametric table on a copy of project. The solve process does involve significant disk IO. If you run too many tasks on a single machine, they may end up competing with each other for single disk, and that could cause them to slowdown. The conflict in memory access may also become a factor.

In a particular case, 26 tasks were run on one 26 core machine, and this caused significant slowdowns. Spreading the tasks across machines helped scalability as an example with a 32 core machines in a Linux Cloud, we ran this job with 26 tasks on a single machine and then another one over two machines with 13 tasks per machine. The single machine job finishes in about 14 mins, while the job on two machine finishes in about 8.5 mins. This indicates that spreading tasks across machines helps with scalability by minimize file read/write conflicts.

Job Restart

There is no provision for stopping and restarting a job. A new job does not reuse solved results; it always solves all rows in the table. An abort or failure of a job restarts from the beginning, unless a new parametric table with the unsolved rows is created.

Linux-Only Issues

- Deployment/Installation errors (such as mainsoft-related) are not captured. If there is such an issue, the Large Scale DSO job will fail without useful messages in the logs.
- Report-based extraction fails if traces and parametric-setup are not prepared as per the Getting Started guides.
- 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' (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.

Process Limits for Large Scale DSO on Windows

Windows has a default limit on the resource usage for non-interactive sessions, which limits the number of processes that can be started in parallel. This puts a limit on the number of Large Scale DSO tasks (Design Points) that can be started on a given Windows machine. The limit is defined by the third parameter of the SharedSection segment of Windows registry HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems\Windows.

`SharedSection=1024,20480,768` (In this example, the limit is set to 768)

To increase the number of tasks that can be run on a single machine, increase this limit gradually, as needed, in multiples of 1024 (but not exceeding 8192). Ansys in-house testing has shown success in running up to 64 tasks on a machine with the value of 4096.

For more information, refer to Microsoft documentation.

Interactive Scheduler Jobs

This document includes information, guidelines, and caveats 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.

Some customers use an alternative method for submission of scheduler jobs. For convenience, we call such jobs "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 selects a project and then 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.

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** window 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** window.

- 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 that are automatically generated for batch jobs submitted using the AnsysEM 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

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

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.

  > **Note:** RAMLimitPercent and RAMLimitPerCoreinGB are not supported for 2D Extractor designs.

- **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'
```

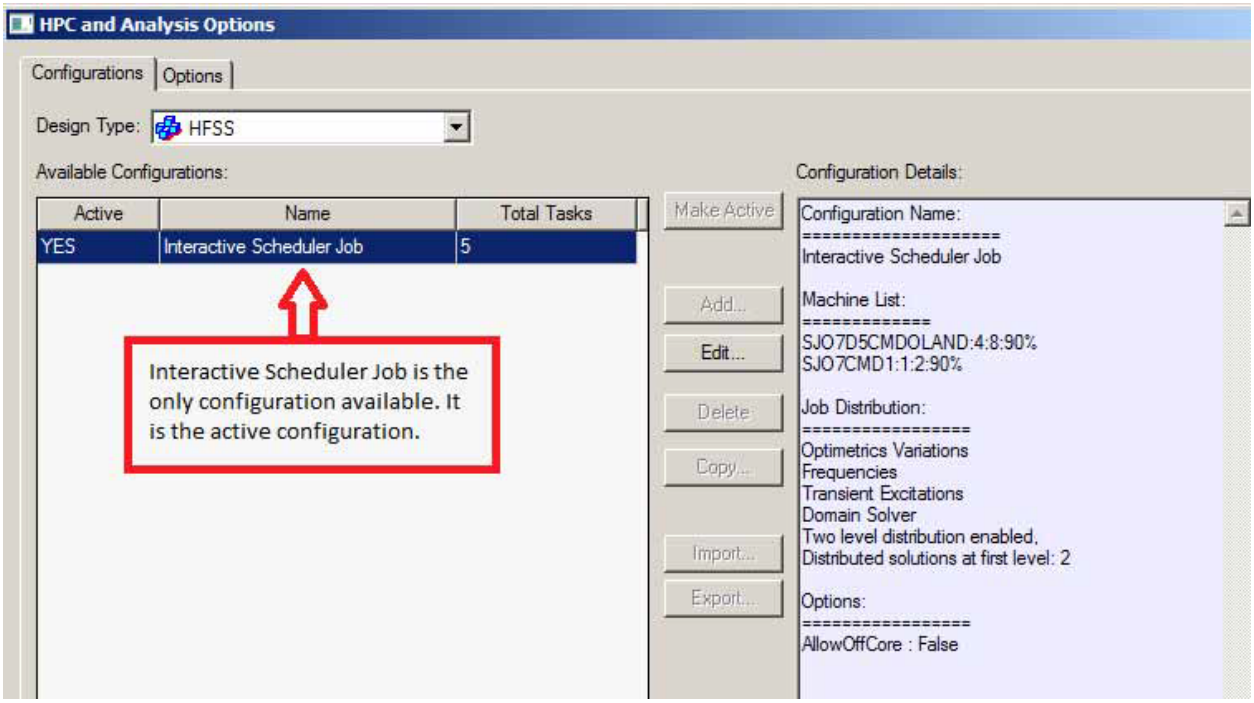
```
'Desktop/Settings/ProjectOptions/DoAutoSave'=1
'LargeScaleDSO/MaxFolderInMB'=100 "
```

DSO Configuration for Interactive Scheduler Jobs

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

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.

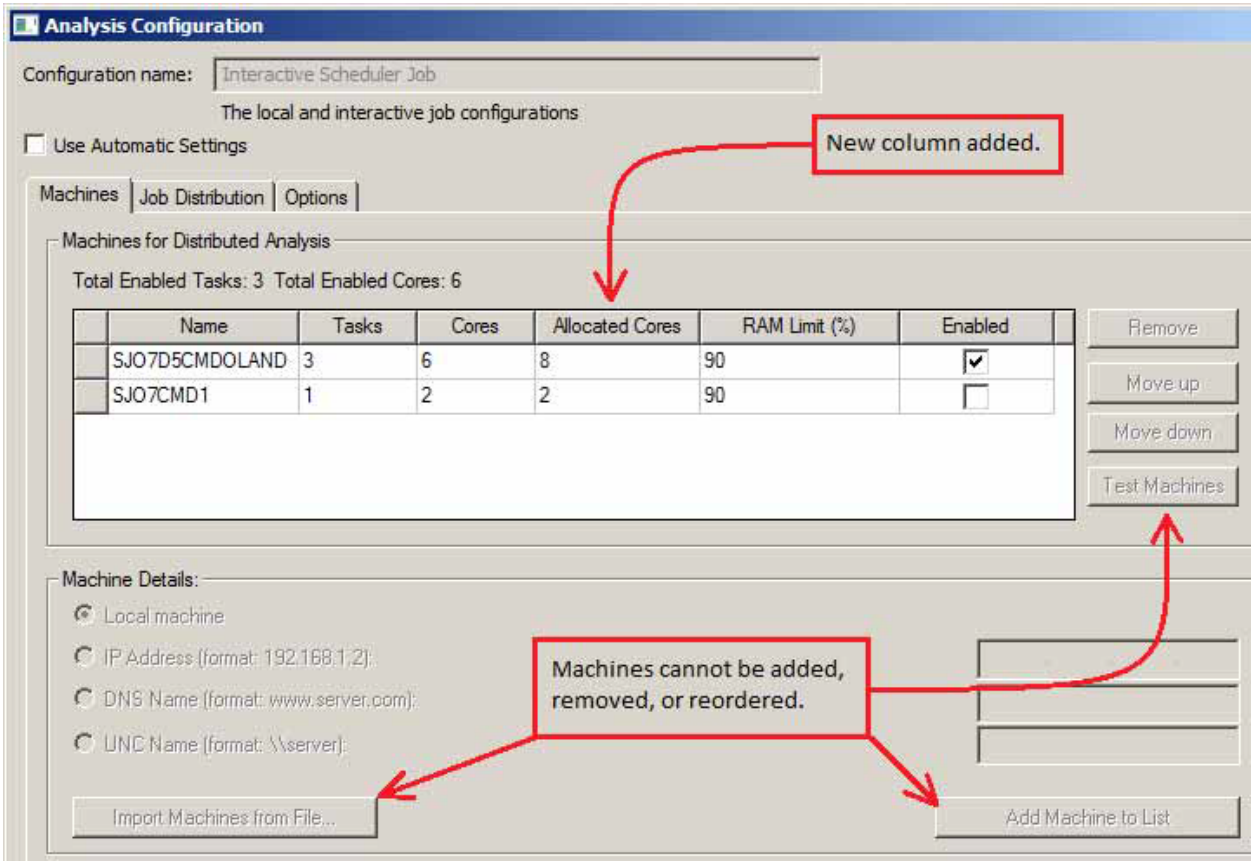


One major difference between interactive scheduler mode and batch mode or normal interactive mode is:

- 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** window. The grid of machine information is prepopulated 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, and 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** window only appears if the **Use Automatic Settings** check box is not checked. The state of this check box, 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** check box or the job distribution settings are:

- -distributed
- -local
- -auto
- -machinelist

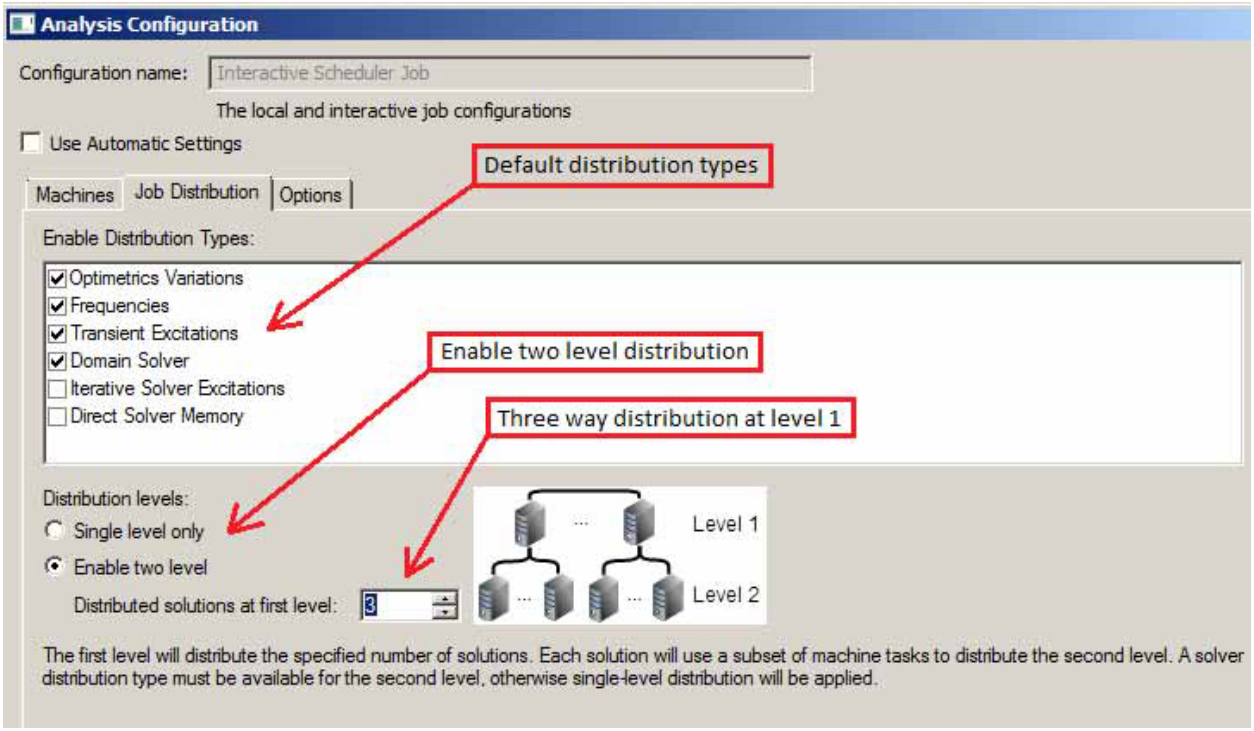
The registry contains the last value of these settings for an interactive scheduler job for the same user on the same host. Settings on the command line override settings from the registry. Any changes in the GUI will override the initial settings, even if the initial settings are from the command line. Any changes in the GUI also update the registry settings for the "Interactive Scheduler Job" configuration for the current design type.

Example

Command line:

```
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** window 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.

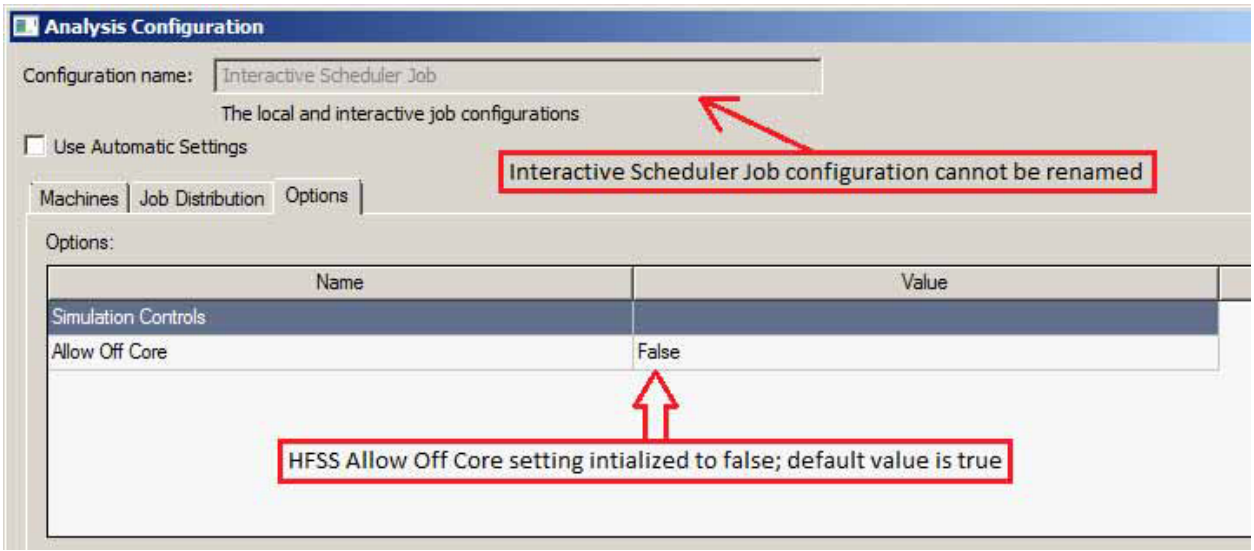
The settings on this tab are initialized from the command line (using the -batchoptions command line option) or from the registry. The registry contains the last value of these settings for an interactive scheduler job for the same user on the same host. Settings on the command line override settings 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

Command line:

```
ansysedt -batchoptions 'HFSS/AllowOffCore'=0
```

Initial Analysis Configuration Options:



Design Type Options for Interactive Scheduler Jobs

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

You can use the **Options** tab of the **HPC and Analysis Options** window 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

The following example includes two command line options, one that is applicable to HFSS ('EnableGPU') and one that is applicable to all design types ('HPCLicenseType'). These opitons would be included as part of the complete job submission command line syntax.

Command Line Excerpt:

```
-batchoptions 'HFSS/EnableGPU'=1 'HPCLicenseType'='Pack'
```

High Performance Computing (HPC) Integration

Ansys Electromagnetics products offer a direct integration with a number of High Performance Computing (HPC) software programs.

The list of currently supported schedulers includes:

- Ansys Cloud Direct
- Ansys Cloud Burst Compute
- Microsoft Windows HPC Server (Windows Only)
- Grid Engine (GE) (Linux Only)
- Platform's Load Sharing Facility (LSF) (Linux Only)
- PBS Pro or PBS Torque (Linux Only)

Electronics Desktop also supports custom integration.

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.

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 EM products. The general procedure is to specify the scheduler and head node, describe and submit the job, and monitor the results. Using

“Ansys Cloud Direct” or “Ansys Cloud Burst Compute will require a login through a browser instead.

Scheduler Terminology

This help uses some specific terminology when discussing HPC and schedulers. See definitions below.

Term	Definition
Core	Unit of processing.
Compute Cluster	Network of machines on which jobs run. Typically consists of head nodes and many compute nodes.
Distributed Processing	Multiple engines are launched simultaneously on the same machine *or* on different machines.
Engine	Electronics Desktop application (aka. executable) launched during analysis commands to generate analysis results.
Job	Application (aka. program, executable) with command line options that uses resources to produce results. For example, `hfss.exe -ng -BatchSolve`
Machine/Host/Node	Consists of one or more processors, memory, disk, etc.
Multiprocessing	A single engine uses multiple cores on the same machine.
Parallel Job	Job that runs on multiple cores belonging to the same or different machines.
Processor	Consists of one or more cores.
Resource	Machines, licenses, etc. that a job uses.
Serial Job	Job that runs on a single core.
Service	Program that runs in the background (for example, RSM Service) and listens on a port. The OS provides the programming interface through which Applications communicate with services, once the machine and port number are known. Launching an executable on a remote machine requires a service to run on the remote machine.

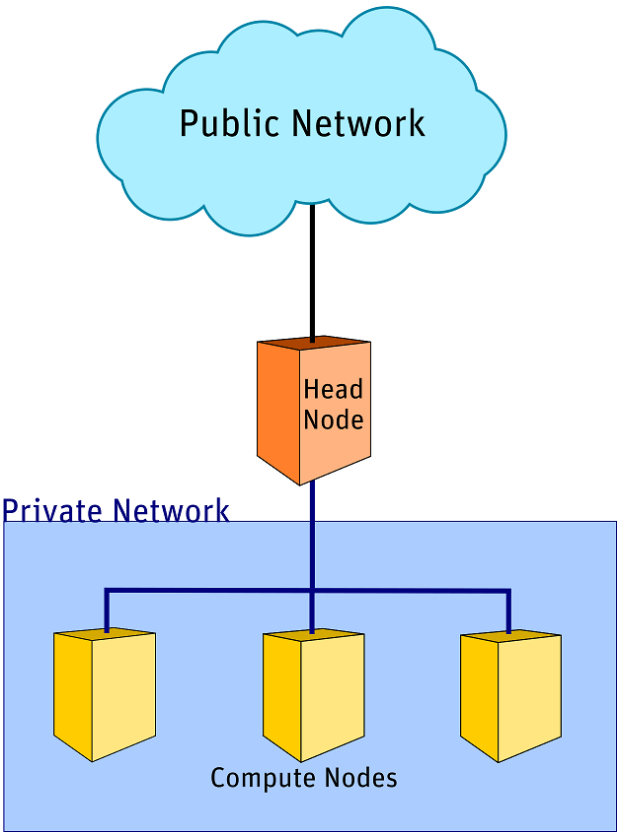
What a Scheduler Does

Schedules are responsible for the following:

- Enabling effective/efficient utilization of cluster's resources consistent with organization's goals
- Maintaining queue(s) of jobs
- Maximizing throughput of jobs by processing all jobs as quickly as possible (typically, one job per CPU)
- Allowing a choice of various scheduling policies (for example, First Come First Serve, Priority-Based, and Preemption)

- Providing a suite of tools or utilities (graphical or command line) for end users to submit, monitor, suspend, and abort jobs
- Managing a compute cluster by running various interacting services on head nodes and compute nodes
- Providing a programming interface to access services

Scheduler-Managed Compute Cluster



Head node(s) typically maintain 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 manage jobs.

Configuring Electronics Installation for HPC

In HPC applications, Ansys Electromagnetics Suite must be available on each cluster host where jobs may be run.

- On the Linux platform, Ansys EM may be installed on a shared drive that is accessible to all machines in the cluster.
- On the Windows platform, Ansys EM must be installed separately on each host of the cluster.

Ansys EM must be accessible using the same path on each host. All cluster users running Ansys EM 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 Electromagnetics 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 EM software. No additional configuration is required.

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*:

- For a Windows cluster: Configure firewall by adding exceptions that allow Ansys EM programs and services to communicate with each other. If you are using a standard Windows Firewall, this is automatically done for you by the installation program. If you are using a third-party firewall software, it needs to be configured in a similar manner.
- For a 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.

Installation Directory Examples

Microsoft Windows

Install the Ansys Electromagnetics Suite in directory "C:\Program Files\ANSYS Inc\v251\AnsysEM" on each node of the cluster. The same directory pathname must be used on all hosts.

Linux

Install the Ansys Electromagnetics Suite in a common directory that is accessible using the path /opt/ansys_inc/v251/AnsysEM on each execution node of the cluster.

Integration with Ansys Remote Simulation Management (RSM)

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).

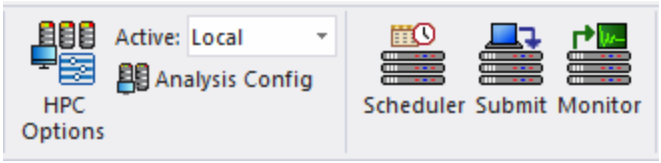
When do you need RSM?

RSM is generally 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, you may not need to install RSM. For details of installation and configuration of RSM, see the *Ansys Electromagnetics Installation Guides*.

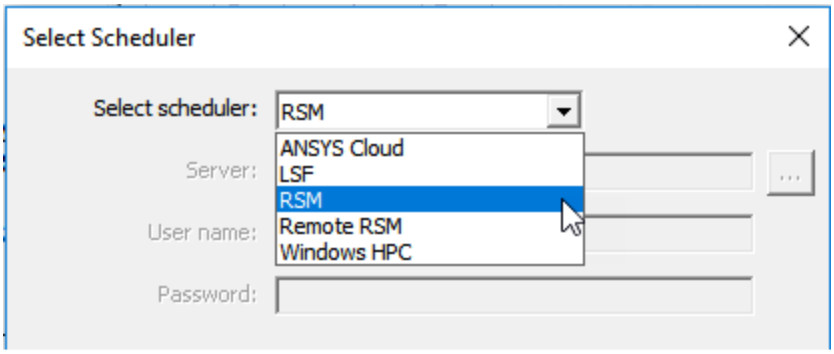
Job Management UI for RSM

You can use the Job Management UI to submit batch jobs to RSM. The Job Management UI is accessed by running Ansys Electronics 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.

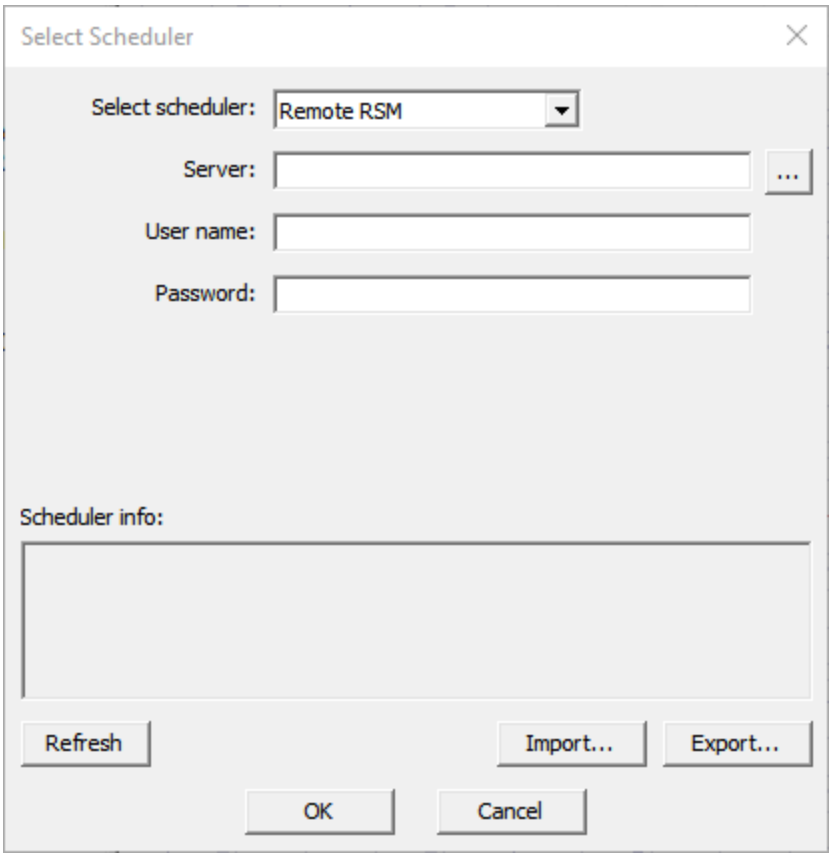


This displays the selection dialog box. The **Select scheduler** drop-down menu lists potential schedulers (which can include RSM, LSF, Windows HPC, or SGE, depending on the environment).

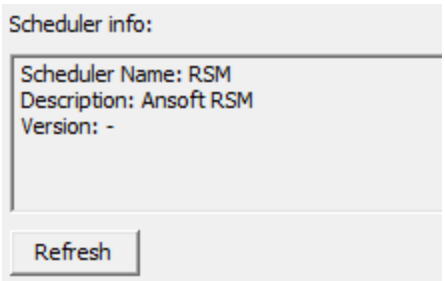


If you select a scheduler that is not supported in your environment, you receive a warning.

If you select Remote RSM and your environment has been configured , you can select a computer, user name, and password.



After selecting a scheduler, click **Refresh** to display information for that scheduler.



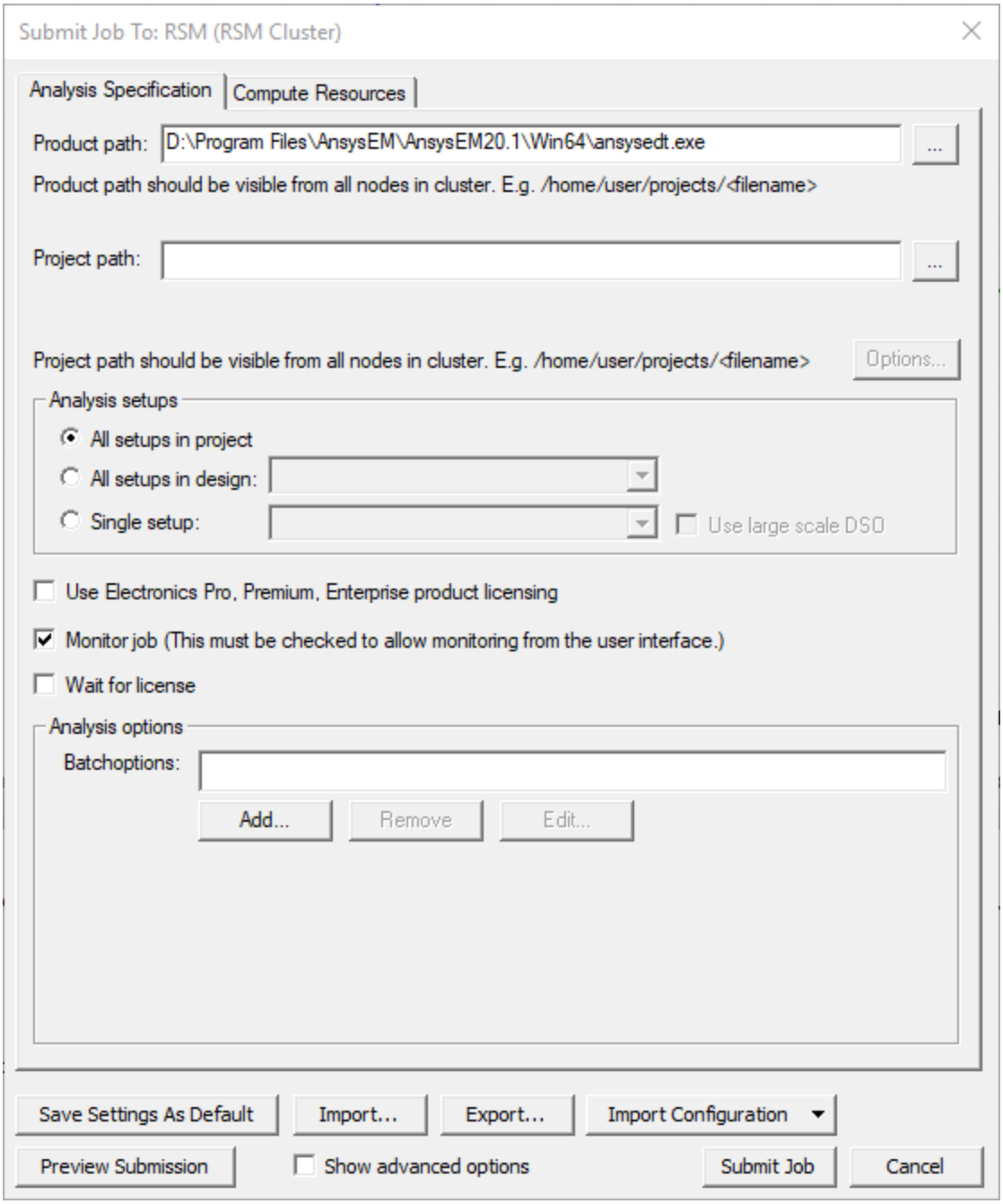
Once you have selected a scheduler supported in your environment, you can go through the following steps to submit a batch job.

1. Set up and prepare 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 (say project is copied to /home/projects/spool/test.adsn).
    - In the RSM environment, you are required to specify a machine-list. See: HPC and Analysis Options. 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** window, 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. 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:** window. 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:** window contains two tabs:

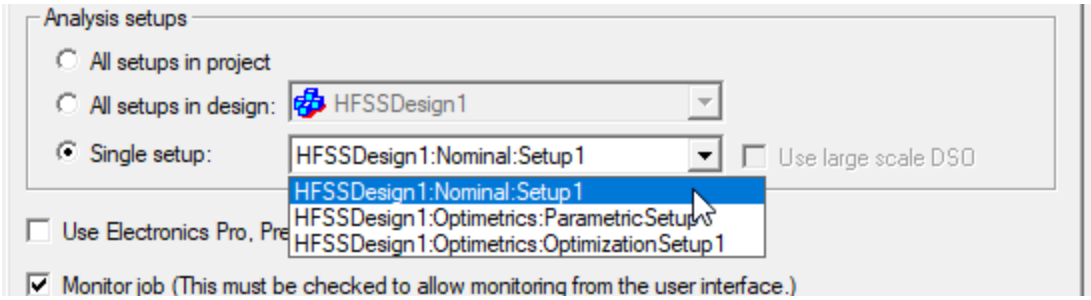
    - **Analysis Specification**– specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables. If you select the Analysis or Optimetrics setup, the Analysis Specification is pre-populated.
    - **Compute Resources** – specify the amount of compute resources and how to select specific resources from those available, and automatic settings, if supported by the design types.

    The standard Job Submission window displays.



5. Use the ellipsis button [**...**] to open a browser to select the project. The project can be an archive.
6. In the **Analysis setups** area, you can select radio buttons for **All setups in the project**, **All setups in the design**, or a **single setup**. For instance, the OptimTee example includes setups for Nominal, Parametric, and Optimization. If you accessed the **Submit Job** window from the right-click menus for Setup or Optimetrics Setup, this field can be

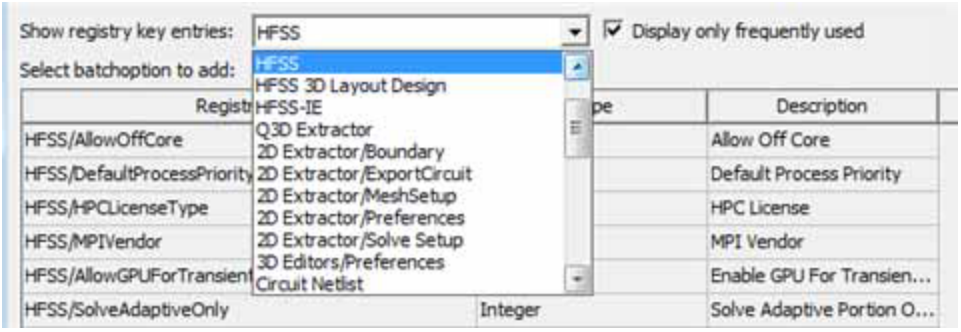
pre-populated.



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.

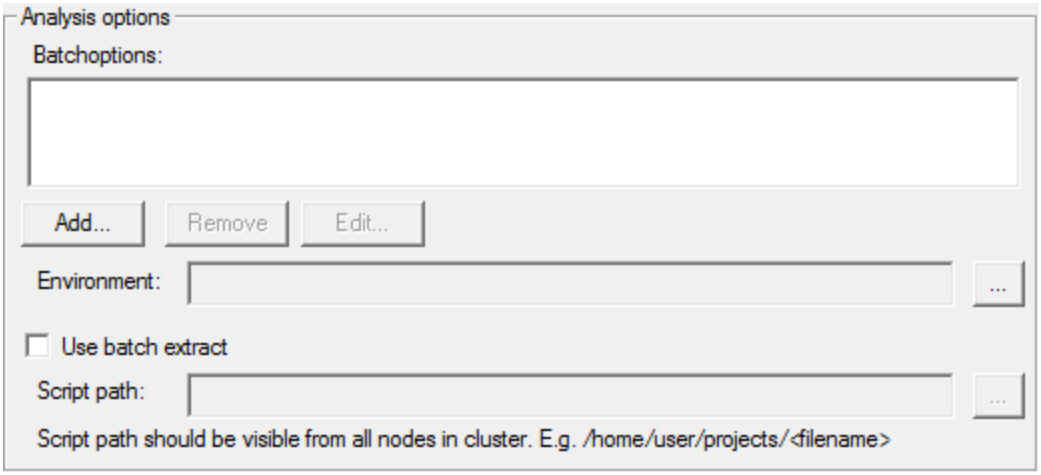
7. The Analysis options include check boxes for Monitoring the job, whether to wait for a license, and a field for adding Batchoptions via a graphical interface, or as text.
    - If you intend to monitor the job through a user interface, you must select **Monitor job**. You can then monitor this job 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**.

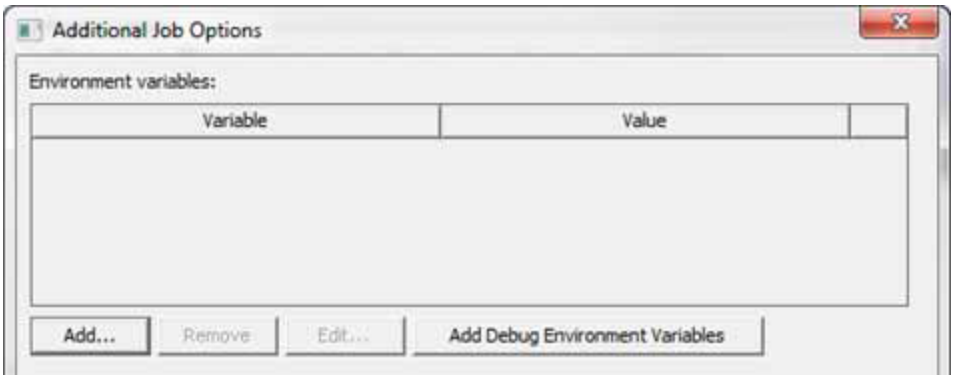


- When you have selected a batchoption, you can type the value in the field, and click the **Add** button to add the option to the batchcommand.
- In the **Submit Job To:** window, you can enable **Show advanced options** to display additional fields for environment variables, and whether to **Use batch**

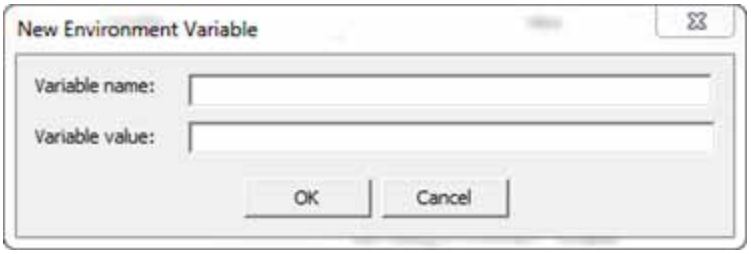
extract.



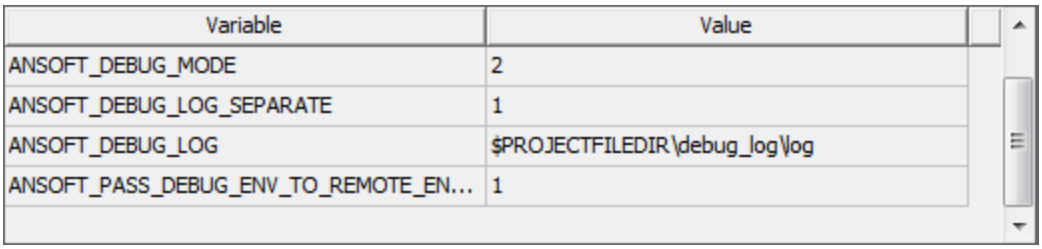
The Environment field is for environment variables, for instance, for debugging features or other variable controlled features. Click the ellipsis [**...**] button to open a dialog box for **Additional Job Options**.



The **Add...** button opens a **New Environment Variable** dialog box in which you can include a variable name and value.

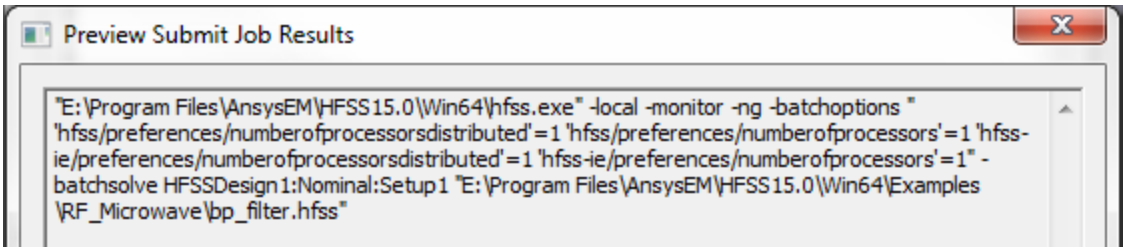


Clicking the **Add Debug Environment Variables** button automatically adds a set of debug variables. This can be useful in working with Ansys Application Engineering support.



Selecting a variable in the dialog enables the **Remove** and **Edit** buttons. The **Edit** button opens a dialog box where you can change the variable and value.

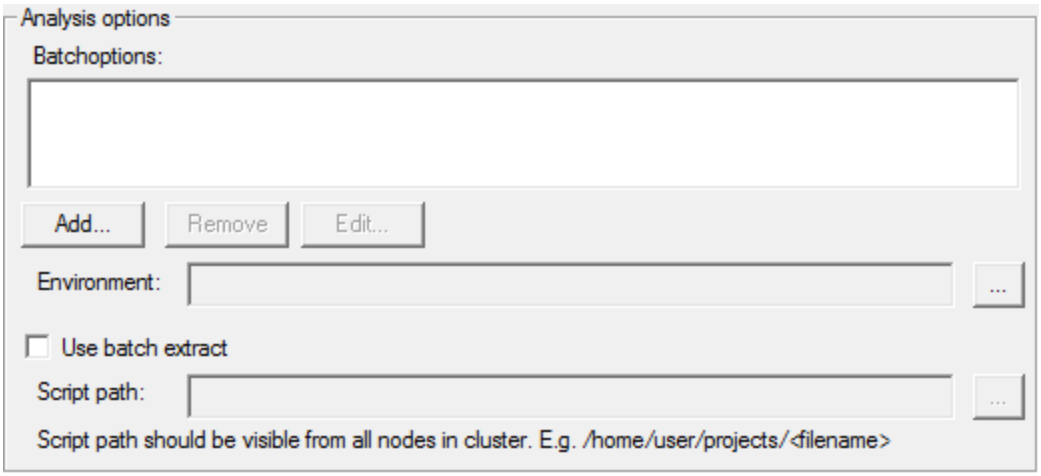
8. To see the command-line to be submitted to the scheduler, click **Preview Submission.** This opens a dialog box showing the command to be sent to the scheduler.



The text can be copied to the clipboard, if desired.

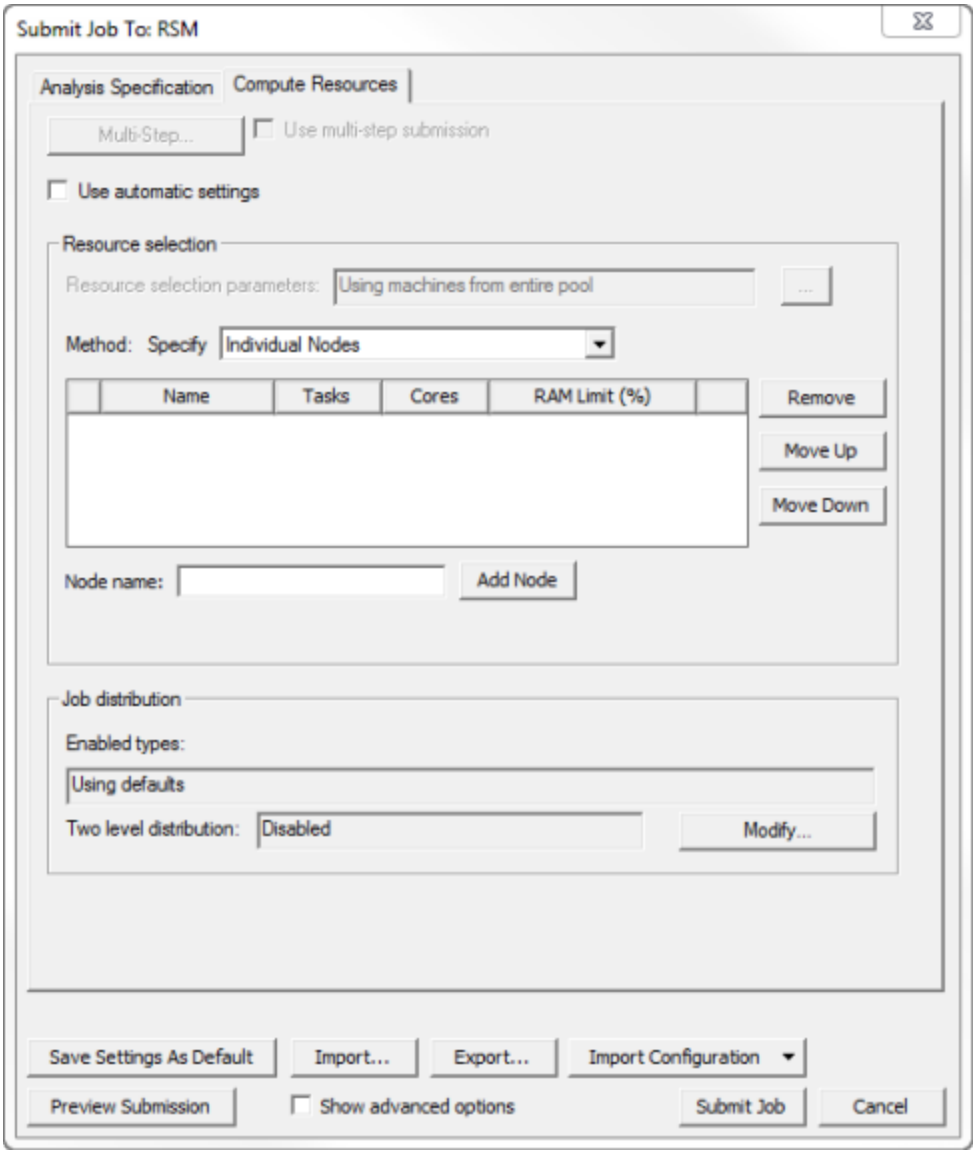
Use Batch Extract for RSM

Selecting **Show advanced options for RSM** also shows the Use batch extract fields.

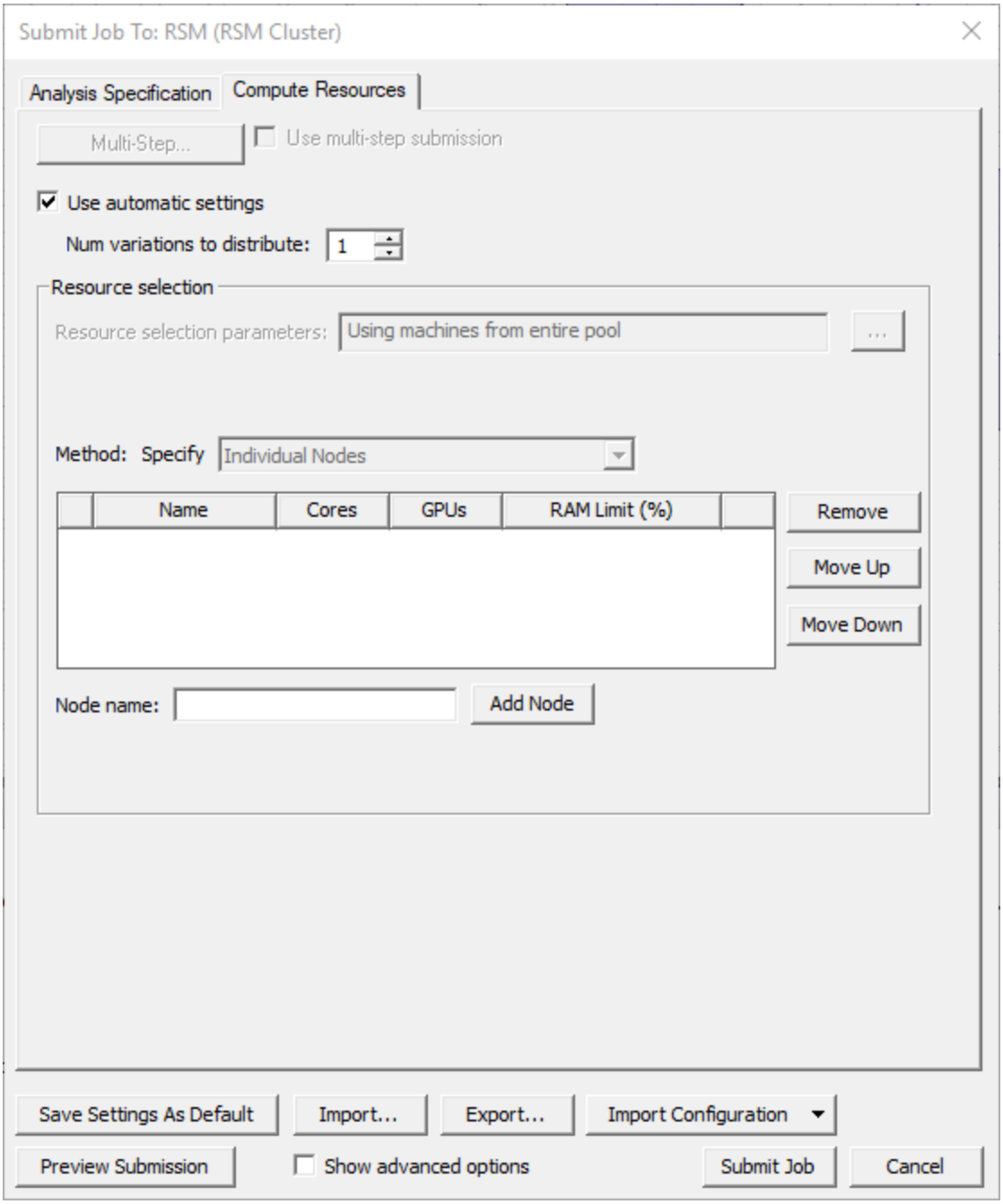


See the discussion on Running Ansys Electronics Desktop from a Command line for a discussion of the solve information available through batch extract.

9. The **Compute Resources** tab of the **Submit Job to: RSM** window 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** check box and **Num variations to distribute** field appear.

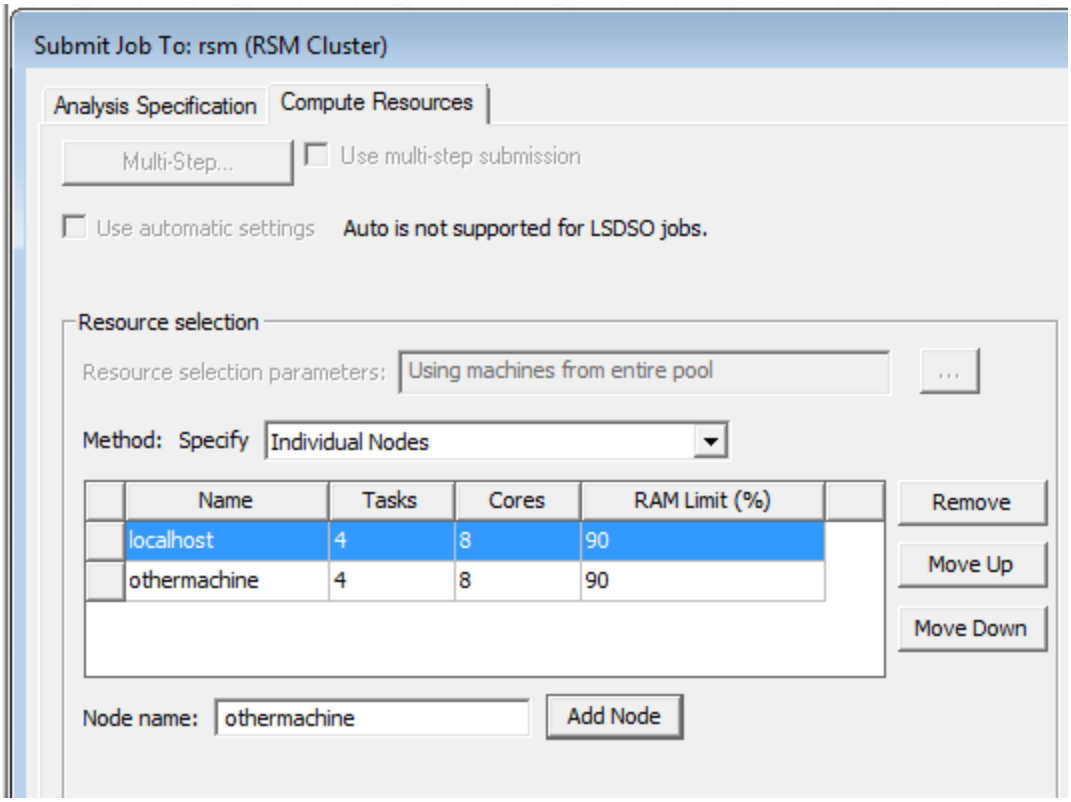


> **Note:**
>
> If you select **Use automatic settings** with **Num variations to distribute** set to 1, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. It does distribute frequencies, domains, and use of multiple level domains. If you set **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

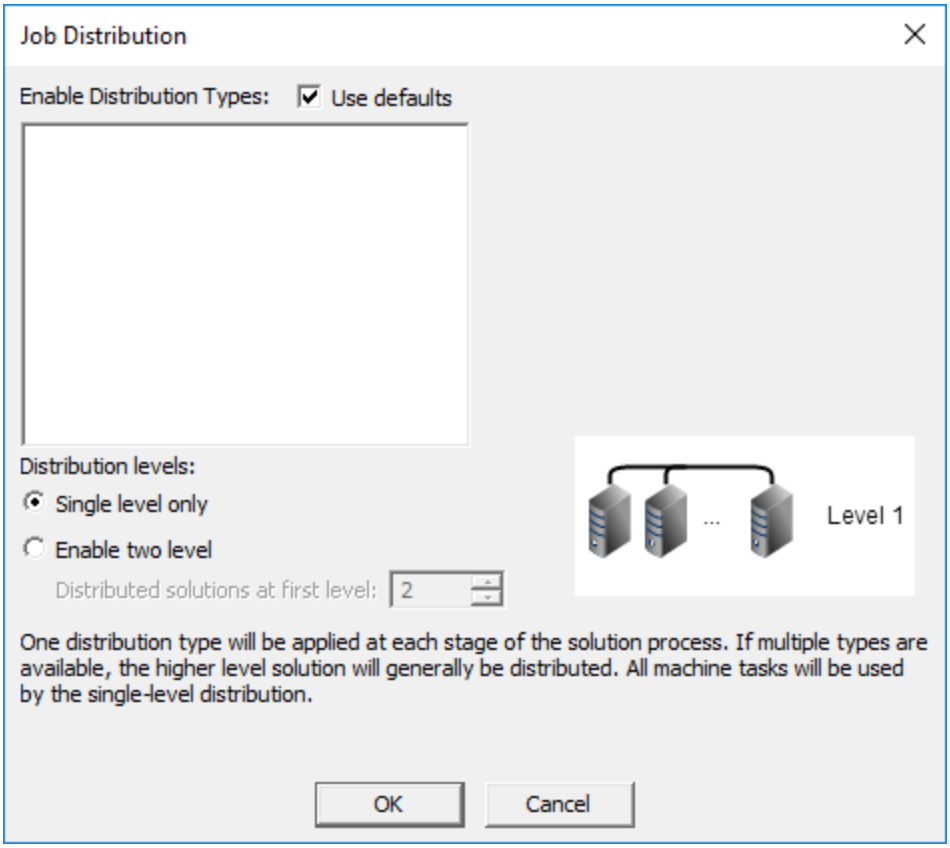
  Here you can specify a node list. In a computing environment where the available cores are not uniform, you can use this to control which resources your job will use. For use with Large Scale DSO for RSM, for jobs that are submitted from job submission panel, localhost must be the first node in the resource selection node list, other wise LSDSO solve with RSM will fail.



Job Distribution

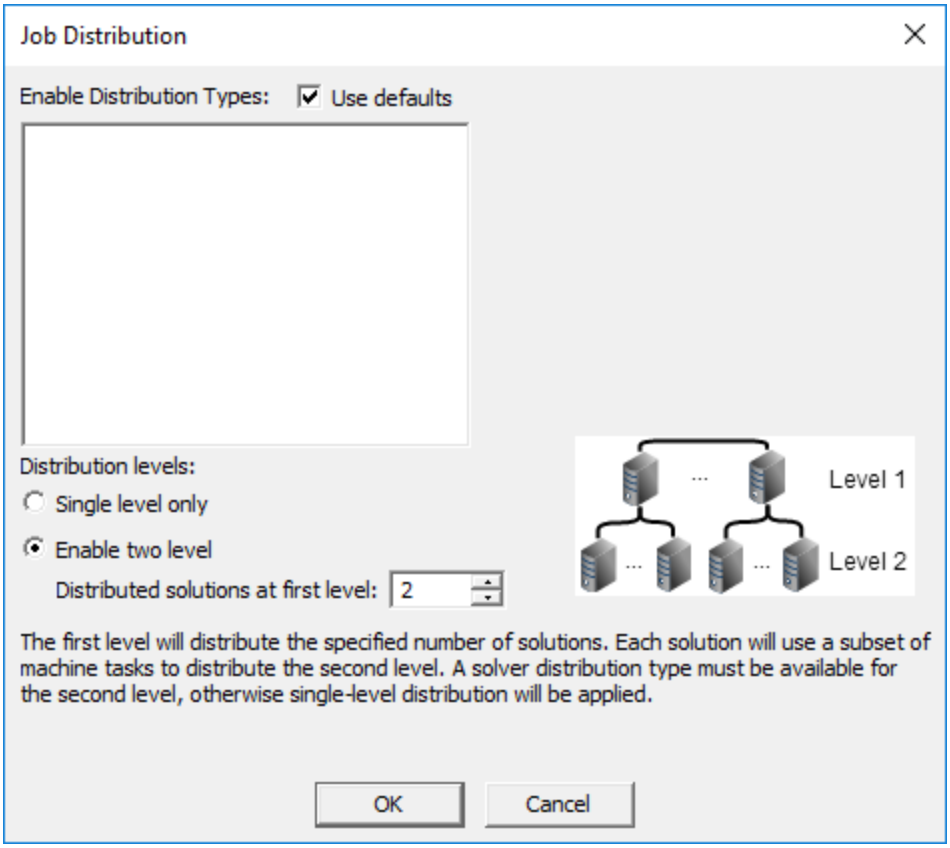
If you disable User automatic selection, you can modify the Job distribution settings.

- 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 appplicable and desired.



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.

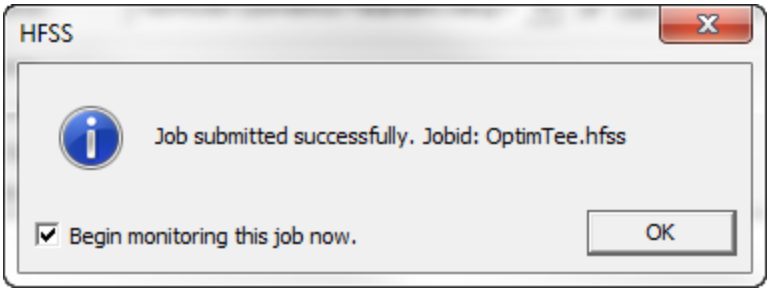


In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, 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.

1. 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 box displays. You can select **Begin monitoring this job now** and click **OK**.



2. You can monitor this job either automatically (by selecting the option above) or through the **Tools** > **Job Management** > **Monitor Jobs...** command. For more details, see Monitor Jobs window.

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.

Changing the AnsoftRSMService Listening Port

For Remote Analysis or Distributed Analsysis, processes may need to be started on multiple hosts. If the Ansys Electromagnetics Desktop needs to start a process on a remote host, the AnsoftRSMService is used to start these remote processes. By default, the AnsoftRSMService listens for socket connections from the Ansys Electromagnetics Desktop on port 32958.

This section describes how to change the port number used by the AnsoftRSMService.

To change the port number, both the AnsoftRSMService and the Ansys Electromagnetics Desktop must be configured to use the new port number. The same port number must be used for Ansys Electromagnetics Desktop and for the AnsoftRSMService process running on each host used for the analysis.

AnsoftRSMService Configuration

The AnsoftRSMService port number is configured in the ansoftrsmservice.cfg configuration file. This configuration file is located in the platform specific subdirectory of the RSM installation directory.

The default location of this directory is "C:\Program Files\ANSYS Inc\v251\EM_RSM" on Windows.

The default location of this directory is /opt/ansys_inc/v251/EM_RSM on Linux.

To modify the AnsoftRSMService configuration, first stop the ansoftrsmservice, then modify the ansoftrsmservice.cfg configuration file, then restart the ansoftrsmservice.

> **Note:**
>
> When using Linux, root maintains exclusive control over systemd-managed services. Once the startonboot command has been given, only root users can start and stop AnsoftRSMService, using the commands:
>
> ```
> sudo ansoftrsmservice start
>
> sudo ansoftrsmservice stop
> ```

The beginning of the configuration file should appear as follows:

```
$begin 'AnsoftCOMDaemon'
$begin 'Default:CommDetails'
$begin 'CommDetails'
'ListenPort'='32958'
$end 'CommDetails'
$end 'Default:CommDetails'
```

If there are additional lines between the following two lines, then they should not be modified:

```
$begin 'AnsoftCOMDaemon'
$begin 'Default:CommDetails'
```

To change the port number, modify the ListenPort setting within the single quotes from 32958 to the desired port number. The single quotes should not be removed or changed.

For previous versions of the software, the ansoftrsmservice.cfg file may not contain the lines:

```
$begin 'Default:CommDetails'
$begin 'CommDetails'
'ListenPort'='32958'
$end 'CommDetails'
$end 'Default:CommDetails'
```

If these lines are not present, then add them to the ansoftrsmservice.cfg file after the first line of the file, and change the ListenPort to the desired port number. For example, to change the port number to 32000, the beginning of the ansoftrsmservice.cfg file should look like the following after the changes:

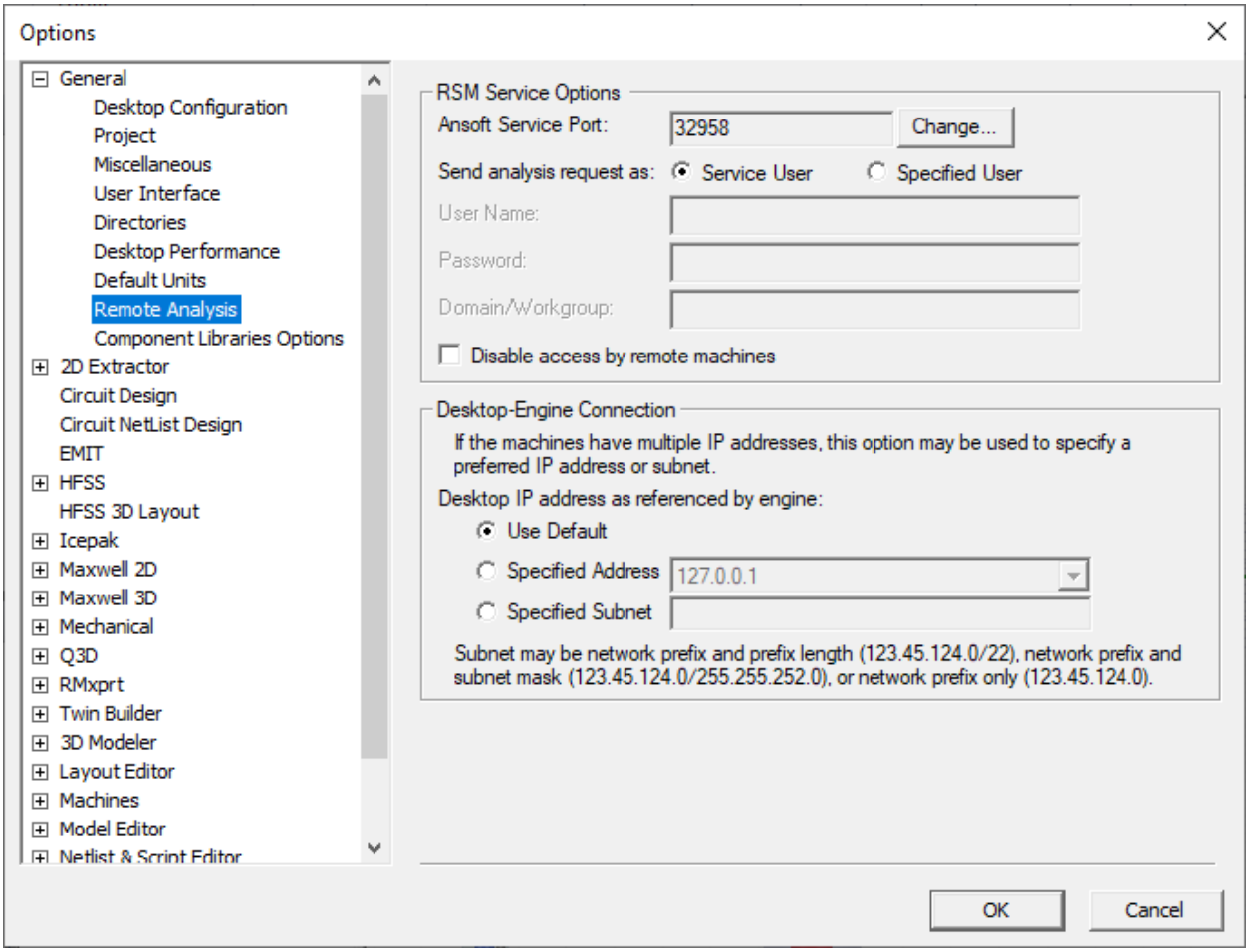
```
$begin 'AnsoftCOMDaemon'
$begin 'Default:CommDetails'
$begin 'CommDetails'
'ListenPort'='32000'
$end 'CommDetails'
$end 'Default:CommDetails'
```

Ansoft Electromagnetics Desktop Configuration

The port number for connecting to the AnsoftRSMService is configured from the **General Options** window.

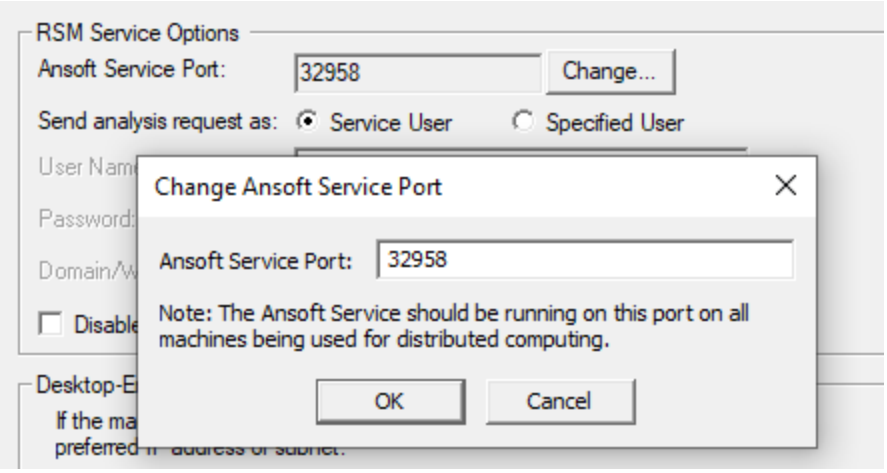
To access these options:

- Select **Tools** > **Options** > **General Options** to open the **General Options** window.
- If it is not already expanded, expand the **General** tree and click **Remote Analysis**.



Under **RSM Service Options**, the **Ansoft Service Port** field shows the current AnsoftRSMService port number.

1. Click **Change** to modify this setting. This opens the **Change Ansoft Service Port** dialog box.



2. Enter the new port number in the **Ansoft Service Port** field.
3. Click **OK** to close the dialog box and accept the new port number.
4. Click **OK** to close the **General Options** window and accept the changes.

Integration with Grid Engine (GE)

The Grid Engine (GE) scheduler (also known as OGE and SGE) is only supported on Linux. With GE, jobs may be submitted in any of the following ways:

- Using GE commands (qsub, etc.) or the SGE gui (qmon)
- Using the generic scheduler GUI in local mode
- Using the generic scheduler GUI in service mode

For additional information on supported schedulers, visit the Platform Support page. Then, locate and click the following link:

Ansys 2025 R1 - Job Schedulers and Queuing Systems Support

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 EM job doesn't require graphics. Ansys EM job's progress can be monitored through GE commands or through Electronics Desktop's Job Management interface.

GE Job Management

You can use Ansys Electronics Desktop to submit batch jobs to GE and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select SGE/OGE/GE as the scheduler.

2. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to GE.

   GE-specific Settings:

   - On the **Compute Resources** tab, click the **Resource Selection Parameters** ellipses button (**...**) to specify the **ParallelEnvironment** parameter. If you do not specify anything, the scheduler will select a parameter.
   - On the **Compute Resources** tab, select **Specify node list** if you wish to specify the nodes. 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 GE version as well as the particular scheduler settings. The output of the `qconf -sc` command shows all of the complexes available to GE schedulers. For a complex to be considered valid for memory resource selection, it must be of type "MEMORY", have relop "<=", be requestable (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.

     For Univa GE 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
   - On the **Scheduler Options** tab, you can **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.

   Click **Preview Submission** to view the qsub commands to be used to submit the job.

3. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.

Installation of Ansys EM 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 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 2025 R1, 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 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: Selecting a 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.

GE Commands for Information about Jobs and Cluster Configuration

The following GE 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 Grid Engine help for more information.

qconf -help: The first line displays the SGE version

qacct -j *job-id*: Displays a log of the completed job with the specified job ID (if accounting is enabled)

qstat -j *job-id*: Displays a log of the running job with the specified job ID

qconf -sc: Show all complex attributes

qconf -spl: Show a list of all parallel environments

qconf -sp *pe-name*: Show details of the parallel environment with the specified name

qconf -sql: Show a list of all queues

qconf -sq *queue-name*: Show details of the queue with the specified name

qconf -sconf: Show configurations

Submitting Ansys EM SGE Batch Jobs via the Command Line

The SGE qsub command may be used to submit Ansys EM jobs. Typical command formats are:

```
qsub qsub_args ansysEM_exe ansys_args
qsub qsub_args job_script
qsub qsub_args[ -]
```

where:

- *qsub_args* are the options of the qsub command,
- *ansysEM_exe* is the pathname of the Ansys EM tool executable to launch,
- *ansys_args* are the arguments to the Ansoft tool command, and
- *job_script* is a shell script containing the Ansys Electromagnetics desktop command to run.

In the first format, the Ansys EM 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 EM 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 EM Command or Arguments for SGE

If the Ansys EM tool executable pathname (*ansysEM_exe*) or any of the arguments of the Ansys EM tool command (*ansysEM_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 EM 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 EM batch jobs may be submitted as SGE serial jobs without any special considerations.

See Monitoring Ansys EM SGE Batch Jobs for options for monitoring Ansys EM batch jobs.

Parallel SGE Batch Jobs

When an Ansys EM 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 EM batch job environment. This is needed because the ANSOFT EM desktop uses the "qrsh -inherit" command to launch engine processes.

See Monitoring Ansys EM SGE Batch Jobs for options for monitoring Ansoft batch jobs.

Setting Up an SGE Parallel Environment (PE)

To allow Ansys EM 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 EM desktop to start analysis engines on hosts other than the local host.

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.

Monitoring GE Serial and Parallel Batch Jobs

You can monitor jobs through the Electronics Desktop user interface, or through the command line.

The suggestions below are for GE serial jobs and GE parallel jobs.

GE qstat Command

The SGE **qstat** command may be used to display information on jobs and queues. If the **-j** option is included, then information on jobs is displayed. If the option includes a job list (**-j [** *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.

Ansys EM 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.

Example SGE qsub Command Lines

All of the following examples show how to submit Linux HFSS jobs on SGE, but similar command lines will work for all Ansys Electromagnetics products.

Serial job using command line:

```
qsub -b y /opt/ansys_inc/v251/AnsysEM/ansysedt -ng -BatchSolve
 ~/projects/OptimTee.aedt
```

- The **-b y** option indicates that hfss 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/ansys_inc/v251/AnsysEM/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/ansys_inc/v251/AnsysEM/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/ansys_inc/v251/AnsysEM/ansysedt
 -ng -BatchSolve -Distributed -machinelist num=4
~/projects/OptimTee.aedt
```

- The **-b y** option indicates that hfss 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. 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".

Recommended Practices for GE Clusters

The following subsections contain recommendations on how to set up an GE cluster for efficiently running Ansys Electromagnetics Suite serial and parallel jobs. These recommendations require the cluster administrator to make configuration changes.

Submitting Exclusive Jobs

In many cases, clusters are used to run "large" Ansys Electromagnetics Suite 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 GE built in mechanism for specifying that a job is "exclusive". GE 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 GE 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 -mc to 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/ansys_inc/v251/AnsysEM/ansysedt -ng -
   BatchSolve -machinelist num=1 ~/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/ansys_inc/v251/AnsysEM/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

GE 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.

GE 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.

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.

> **Note:**
>
> Recent versions of UGE (Univa Grid Engine) come with "m_mem_free" and "mem_free" complexes already configured, and if so then there is no more configuration required. You can just use mem_free when per-host memory request is desired, and m_mem_free when per-core (per-slot really) memory request is desired. SGE may already have "mem_free" which can be used for per-host memory request.

Below shows how to configure the mem_free consumable resource.

Recommended attributes are:

- name : mem_free
- shortcut : mf
- type : MEMORY
- relop : <=
- requestable : YES
- consumable : YES
- default : 0
- urgency : 0

> **Note:**
>
> Ansys Electronics Desktop has the capability in auto cores and RAM to automatically select the memory complex (if one is available) and create this command line option for the user, if you check the "Use RAM constraint" and enter a non-zero amount of GB to use per core.

Serial Jobs in GE

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 GE

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 a GE 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, and pe_sph2 is a parallel environment in which two slots are allocated to the job per host, etc. The command qconf -ap pe_name may 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/ansys_inc/v251/AnsysEM/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 hfss 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 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 "\"".

Issue with qrsh (SGE)

Ansys EM 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 EM desktop software starts the MainWin Core Services on startup, if they are not already running.

After the Ansys EM 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 Ansoft 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 EM 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 EM 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 EM software.

Here is an example bash wrapper script that may be used to work around this issue. In this example, the product is hfss, but the same approach will work for any Ansys EM product. In this example, the script is named sge_hfss and is in the AnsysEM software installation directory. When an Ansys Electromagnetics desktop job is submitted to the SGE scheduler, the script (sge_hfss, in this example) should be submitted instead of hfss. The script will modify the environment, as needed, then start hfss. When the analysis finishes, the script returns the exit status of hfss.

An alternative is to place the script in an arbitrary directory, and modify the script to include an absolute path to the product (hfss in this example).

Script contents:

```
#! /bin/bash

# This script will not correctly process arguments containing
# spaces or other characters special to the shell.

# Create hfss command line
# In this example, sge_hfss and hfss are in the same directory
# An alternative is to use an absolute path for the hfss command
```

```
cmd0=$0
cmd="${cmd0/%sge_hfss/hfss} $@"

# Fix environment variables
export ANS_SGE_TMPDIR=${TMPDIR}
unset TMPDIR
unset TMP

# Run the hfss command and return the exit status
${cmd}
exit $?
```

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 Windows HPC GUIs from Microsoft: Job Manager or Cluster Manager
- Using Windows HPC command line tools
- Using Ansys Electronics Desktop UI commands for scheduler selection, job submission and job monitoring/control.

General Guidelines for Submitting Ansys EM 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 the command line that runs Ansys Electromagnetics desktop in non-graphical mode to perform analysis of a project.

Specifying the Number of Compute Resource Units for HPC Jobs

You can either select **Use automatic settings** on the *Compute Resources* tab, or you either enter the number of tasks and total cores per machine, or individual nodes. HFSS, HFSS 3D Layout, and Icepak have **Use automatic settings** selected by default. This release permits options in setting whether nodes are exclusive for the submission. See Windows HPC Non-exclusive Jobs.

Ansys EM 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 EM project file. The Project Directory (the directory containing the project file) must be accessible to all of the cluster hosts that may run Ansys EM 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 '
```

Submitting and Monitoring Ansys EM HPC Jobs

Jobs may be submitted to the Windows HPC Scheduler using any of the following methods:

- Using the *Submit HPC Job* dialog box
- 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 box 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 2025 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 ellipsis [...] 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

1. Windows HPC Jobs may be monitored using the *Monitor Job* dialog box, which is brought up by the **Tools > Job Management > Monitor Jobs...** command. This dialog bos 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. You can also select the **Simulation** tab of the ribbon and select the **Monitor** icon. 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 Jobs window.

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.

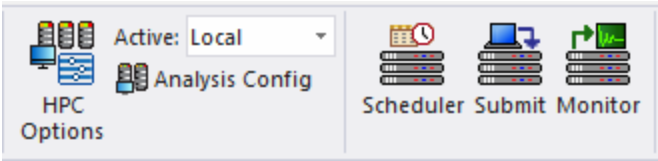
Job Submission User Profile on Cluster Compute Nodes

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.

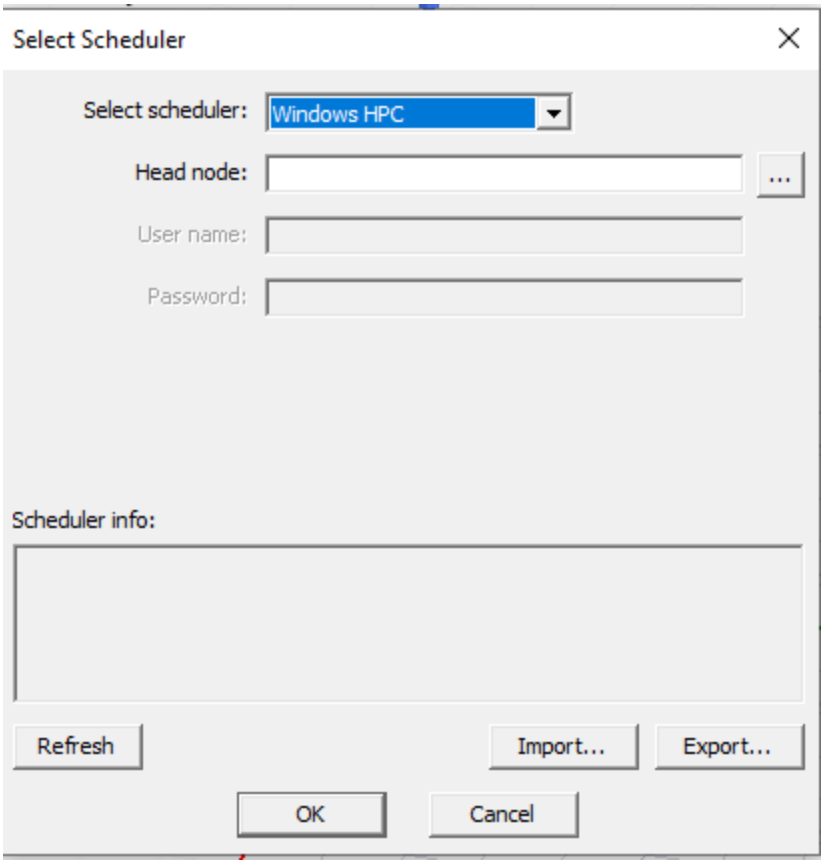
Submitting and Monitoring Jobs for Windows HPC

In order to submit jobs using **Windows HPC**:

- Click **Tools** > **Job Management** > **Select Scheduler**, or
- 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, 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 box will close. If there is a problem contacting the head node, the dialog box will not close and the changes will not be accepted.

After setting the job submission node, perform *one* of the following:

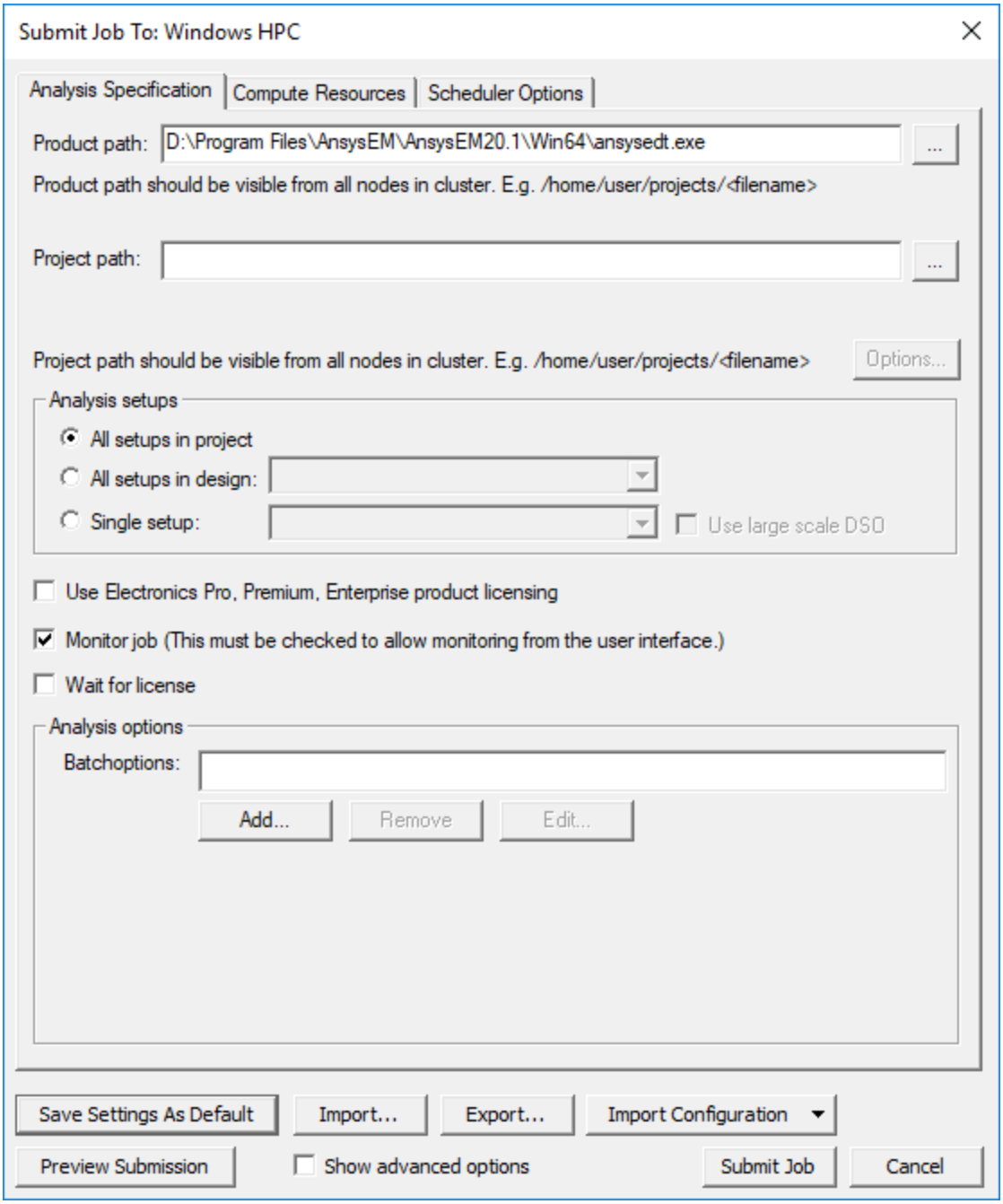
- Select **Tools** > **Job Management** > **Submit Job...**
- Click **Project** > **Submit Job...**
- Click **Q3D Extractor** > **Submit Job...**
- Select the **Simulation** tab and click the **Submit** icon.
- Use a shortcut menu to select **Submit Job**.

This opens the **Submit Job To** dialog box.

The **Submit Job To** dialog box 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.

On the **Analysis Specification** tab, enter path names for the Product and Project. 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 **Analysis Setups** area of the 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)

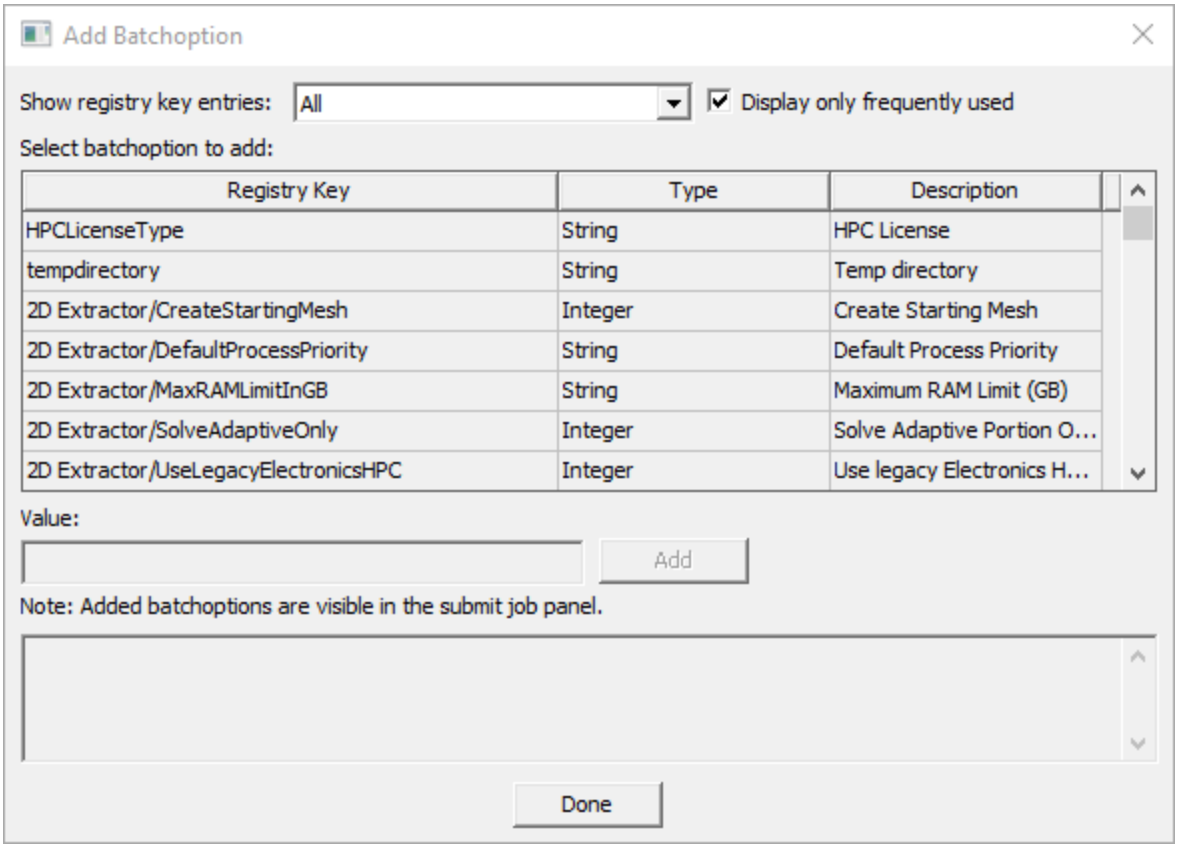
- Single setup (if you selected the Submit Job command from the shortcut menu, the setup name populates the field)

If you specify multiple setups, they will be processed sequentially in the order displayed in the edit box.

The Analysis options include:

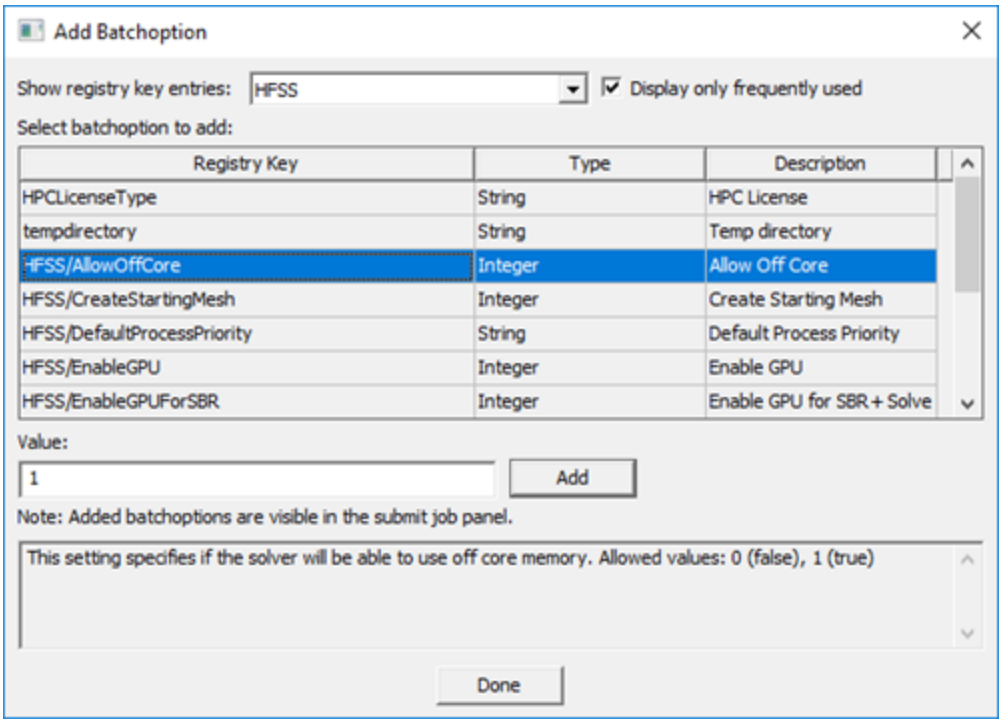
- Monitor job – you must enable this option to monitor the job from the user interface.
- Wait for license – specify whether to wait until a license if available before starting a simulation.
- Batch options – optionally specify -batchoptions in the text field. See detailed discussion of -batchoptions beginning under Running Ansys Electronics Desktop from a Command Line.

The **Add...** button opens the **Add Batchoption** dialog box.

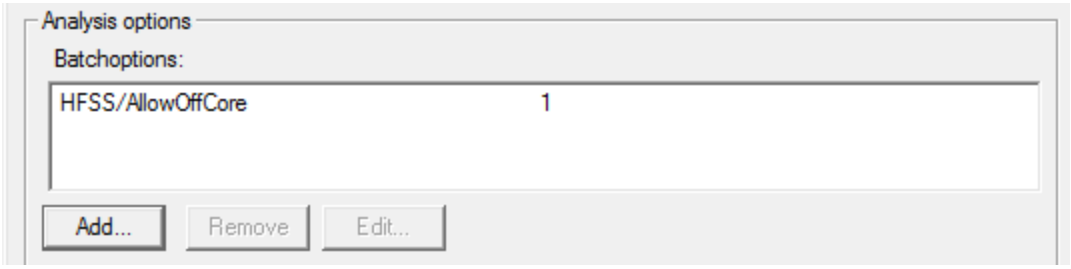


This dialog box provides access to all -batchoption commands. The drop-down menu lets you select specific categories, and you can choose to display only frequently used commands. You can edit and remove any batch options you specify.

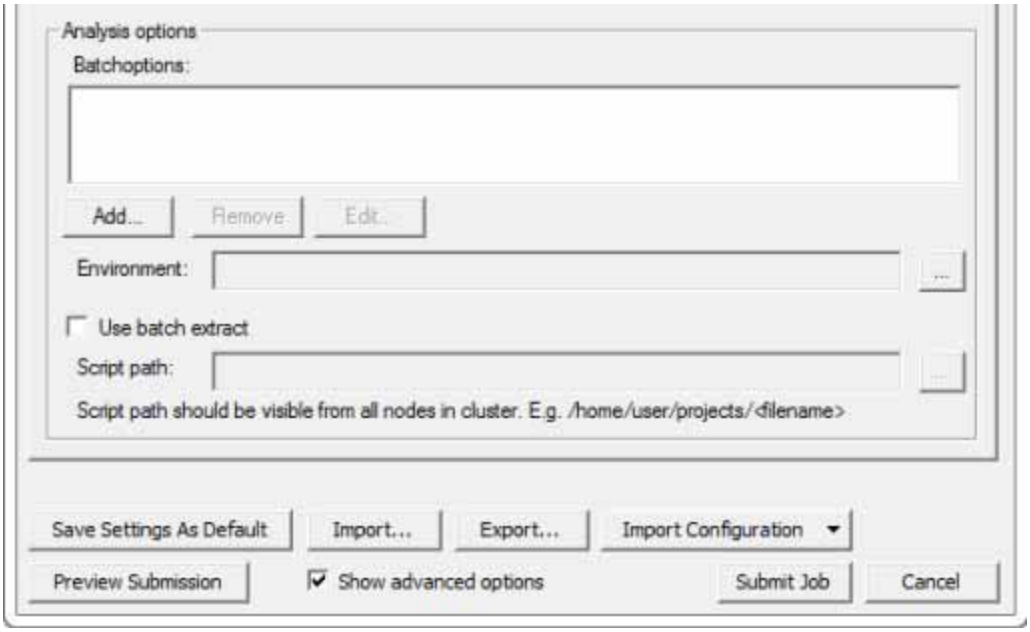
Select a Registry Key in order to show the current Value for the type. The lower field explains the meaning of the Type Value.



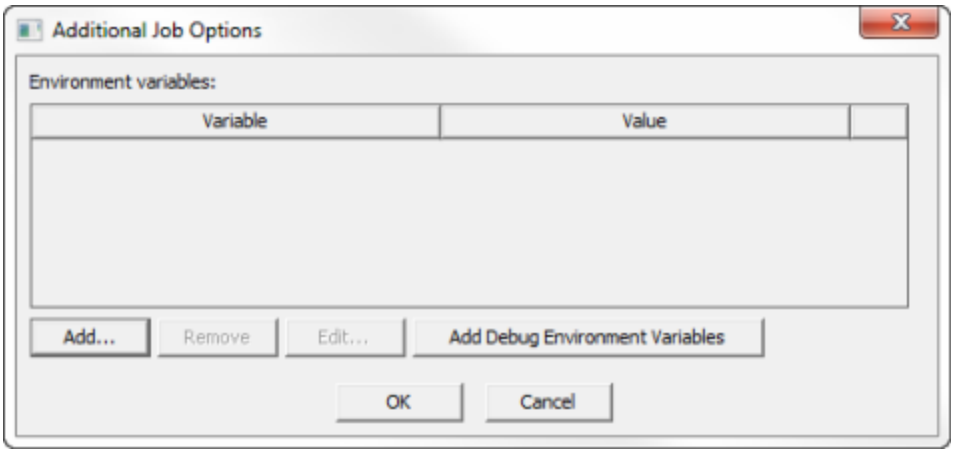
Any batchoptions for which you select **Add** will be visible in the **Submit Job** dialog box.



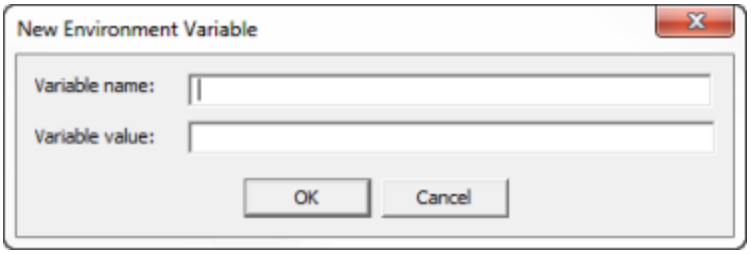
If you have the **Show advanced options** box checked in the **Submit Job** dialog box, the Environment field and the Use batch extract fields display.



The **Environment** field lets you specify any environment variables. Click the ellipsis button [...] to display the **Additional Job Options** dialog box.

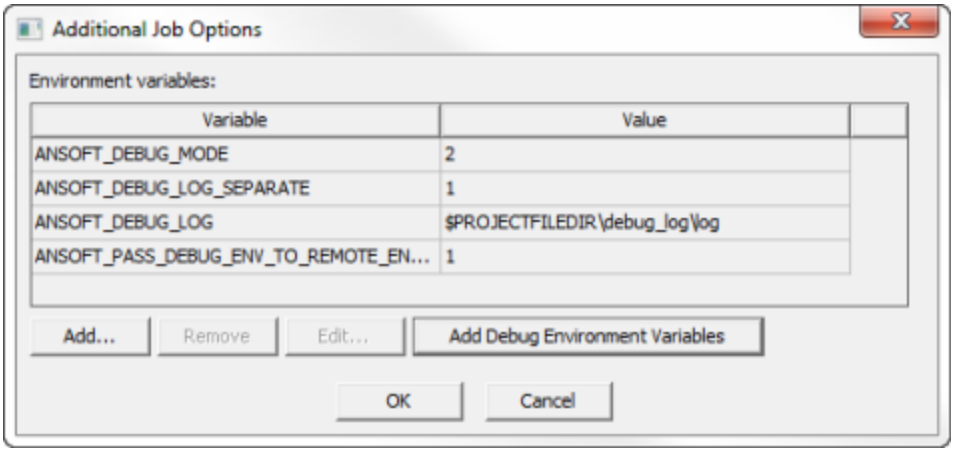


Click the **Add...** button to open the **New Environment Variable** dialog box.



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 enable the **Remove** and **Edit...** buttons. You can also click **Add Debug Environment Variables**.



Note:

For certain methods of resource selection for Windows HPC job submission, Ansys Electronics Desktop checks for nodes being online. In an auto-scaling cluster, nodes can fail the online check because they're not online until a job is running. An environment variable allows customers to bypass this check:

ANSYSEM_SKIP_NODE_ONLINE_CHECK

Set the variable value to 1 to disable the online check. Debug logging fully verifies what aspects of the cluster are causing this check to fail.

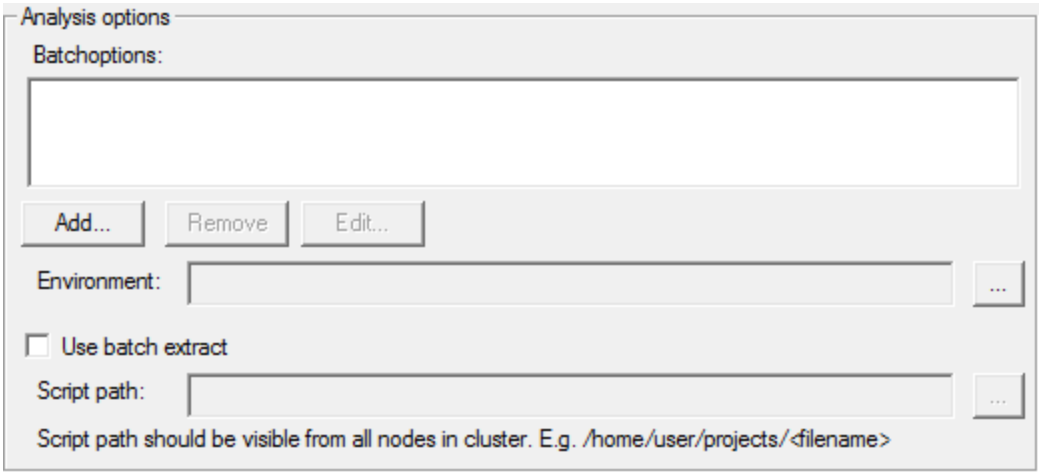
Job submission in Windows works without changing settings if environment variable ANSYSEM_SUBMIT_JOB_REQ_NODE_ONLINE is set with value 0. The installation doesn't set this environment variable. ANSYSEM_SUBMIT_JOB_REQ_NODE_ONLINE accepted

values: 0 or 1 (0 = disable; 1 = enable, i.e. require nodes to be online for submission). Default: 1 (if not set, Ansys Electronics Desktop requires nodes to be online for submission).

If you have enabled **Show Advanced Option**s, any variables that you add will be displayed in the **Environment** field of the **Submit Job** dialog box.

Use Batch Extract for Windows HPC

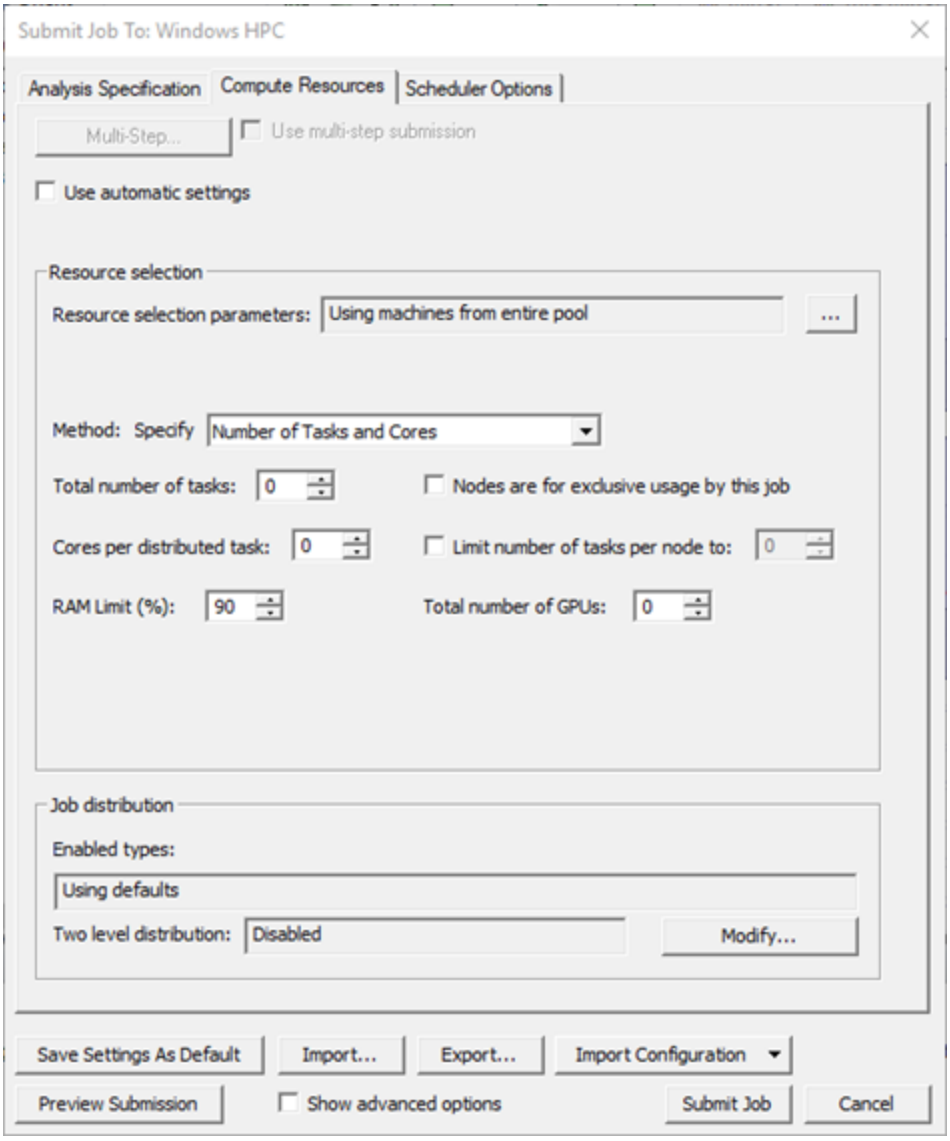
Enabling **Show Advanced Options** for Windows HPC also displays the **Use batch extract** check box and script path field.



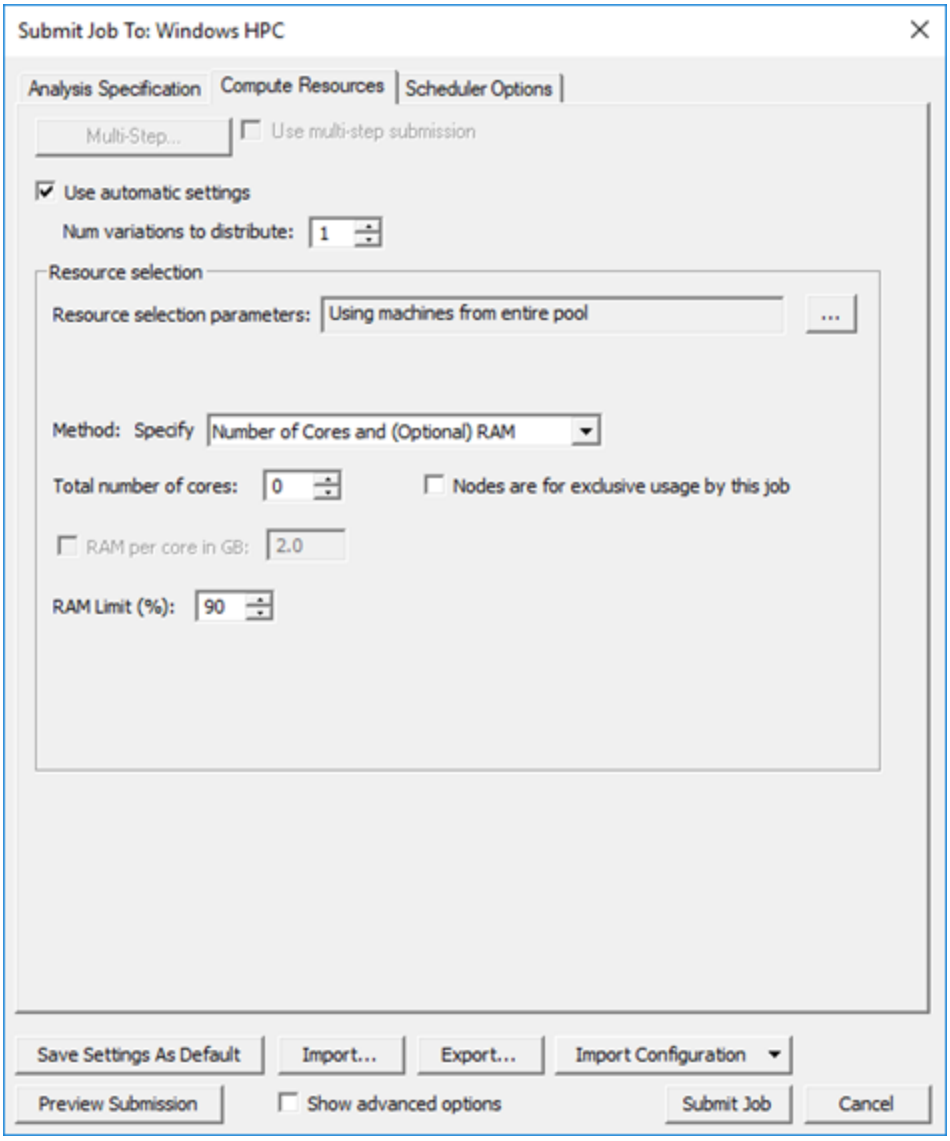
See Running Ansys Electronics Desktop from a Command line for an explanation of the solve information available through batch extract.

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.

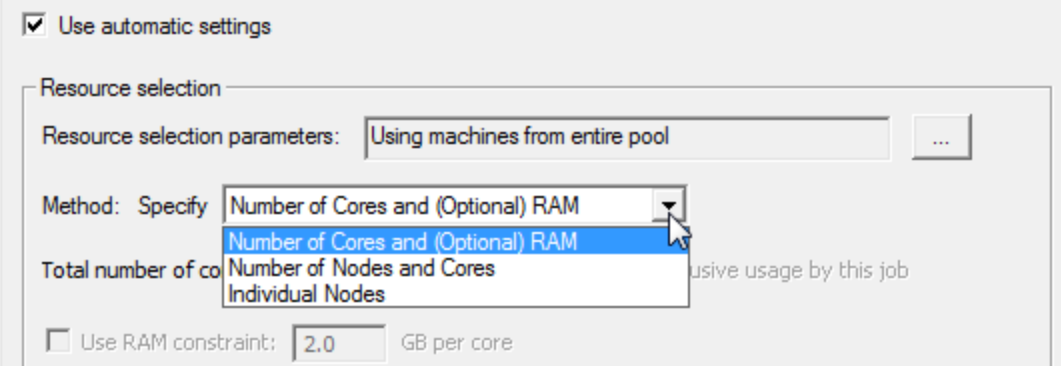


For Ansys Electronics Desktop configurations, the **Submit Job To** dialog box includes a **Use automatic settings** check box that simplifies the **Compute Resources** tab.



With **Use automatic settings** selected, the **Job distribution** field is removed. When using 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 entire pool."

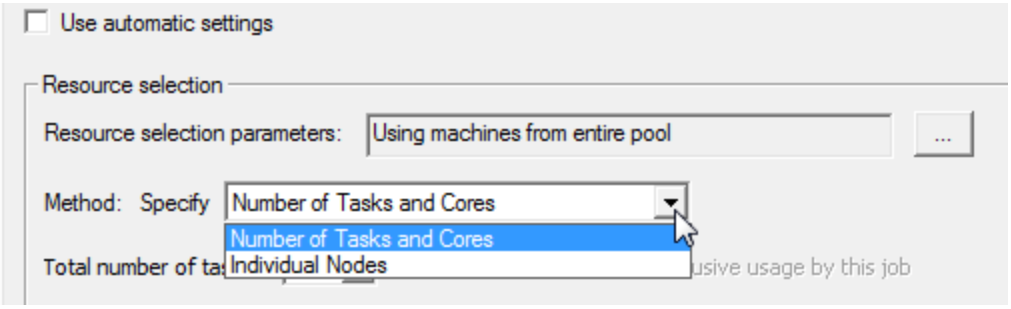
The Method field of the **Submit Job To** dialog box has a drop-down menu with two or three selections, depending on whether you select **Use automatic settings**.



Note:

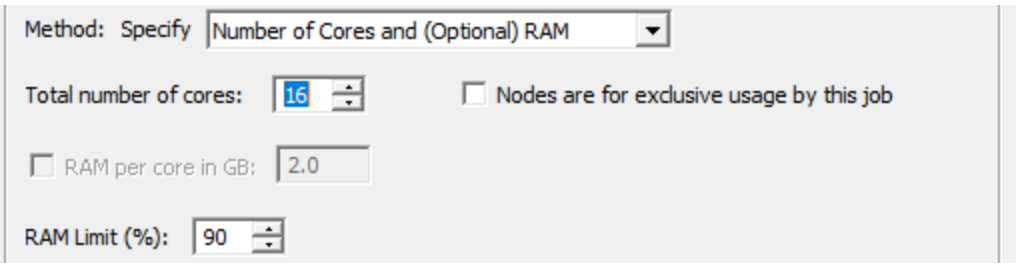
If you select Use automatic settings, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. It does distribute frequencies, domains, and use of multiple level domains.

If you uncheck or cannot access Use automatic settings, these two Methods are listed:

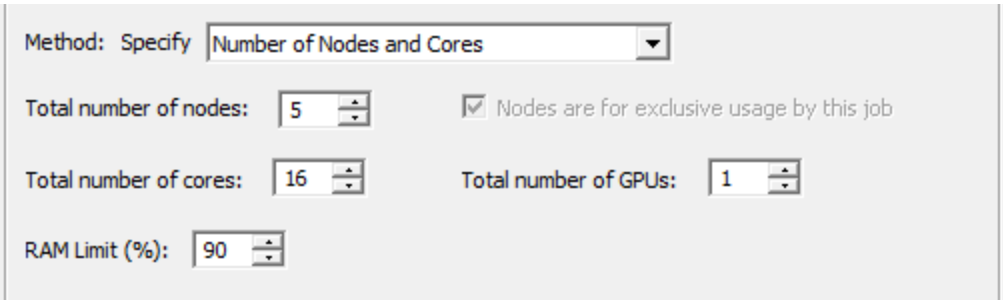


For Windows, there is a 64 core limit for a single machine. Each Method selection changes the available options listed:

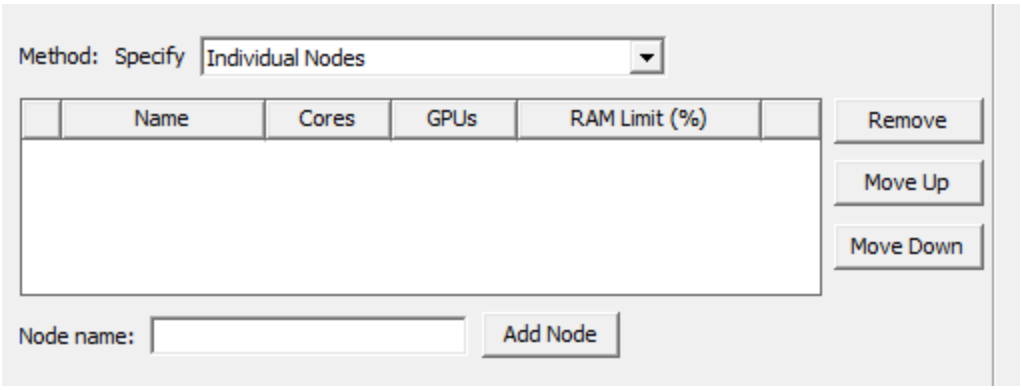
- Specify Number of Cores and (Optional) RAM



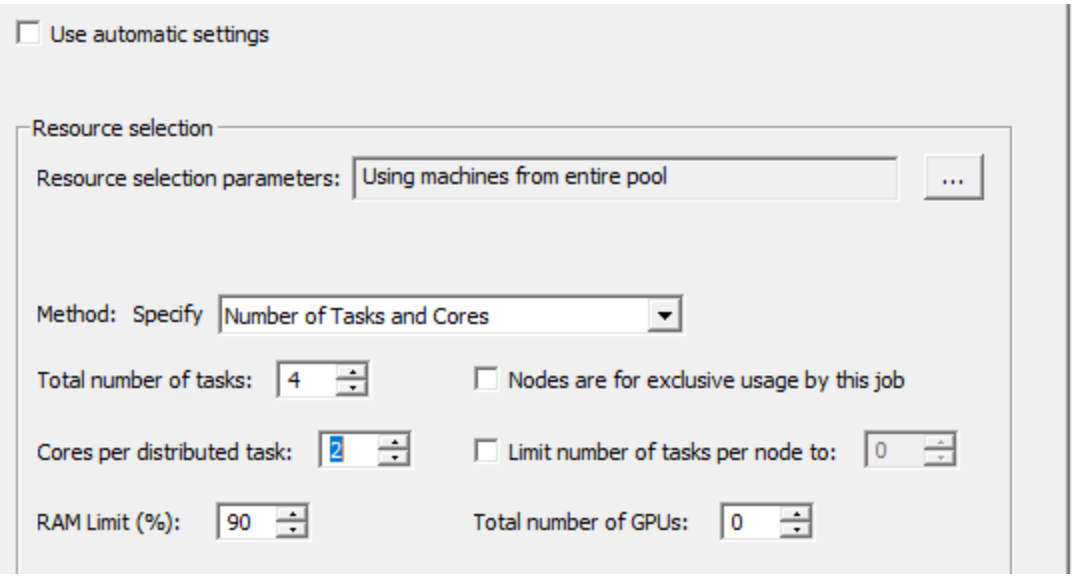
- Number of Nodes and Cores



- Individual Nodes



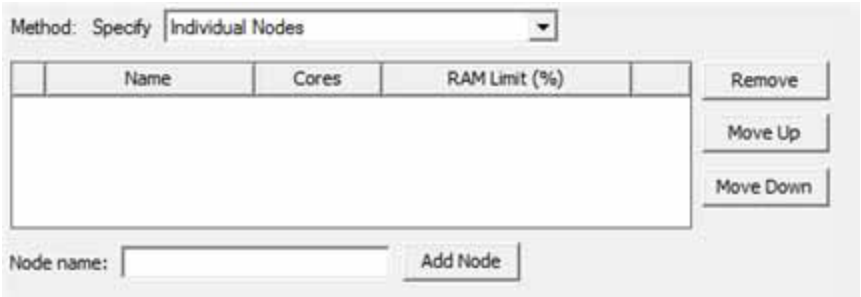
- Number of Tasks and Cores ("Use automatic settings" is unchecked for this option. Checking "Use automatic settings" means that you do not have to specify tasks or core parameters):



Individual Node List

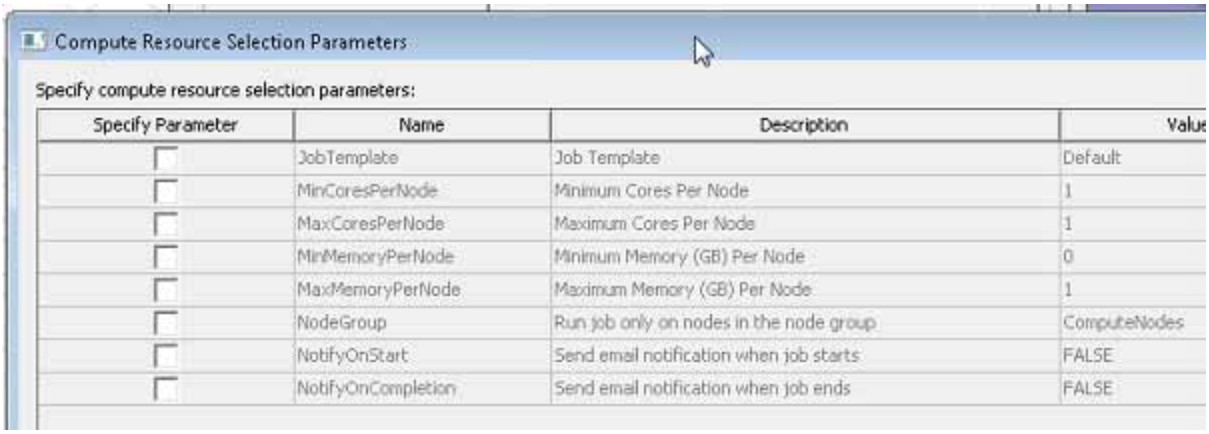
For Windows HPC jobs, you may either specify a node list, or specify job parallelization parameters, but not both.

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: Maximum 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.)

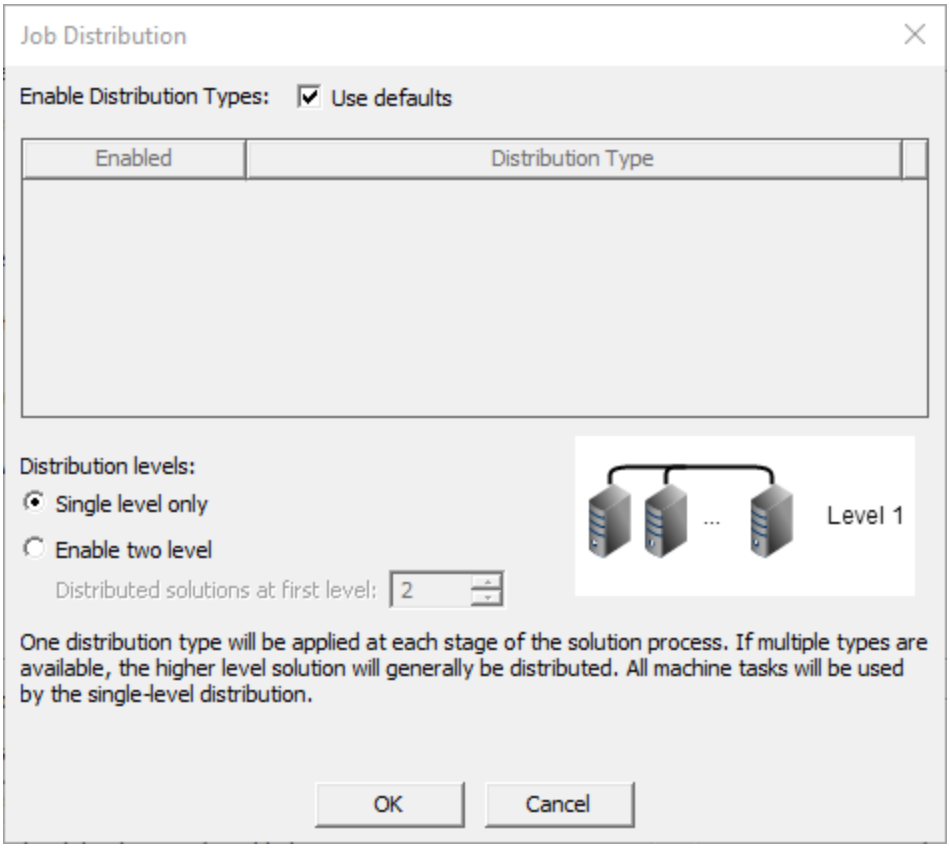
Job Parallelization

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:

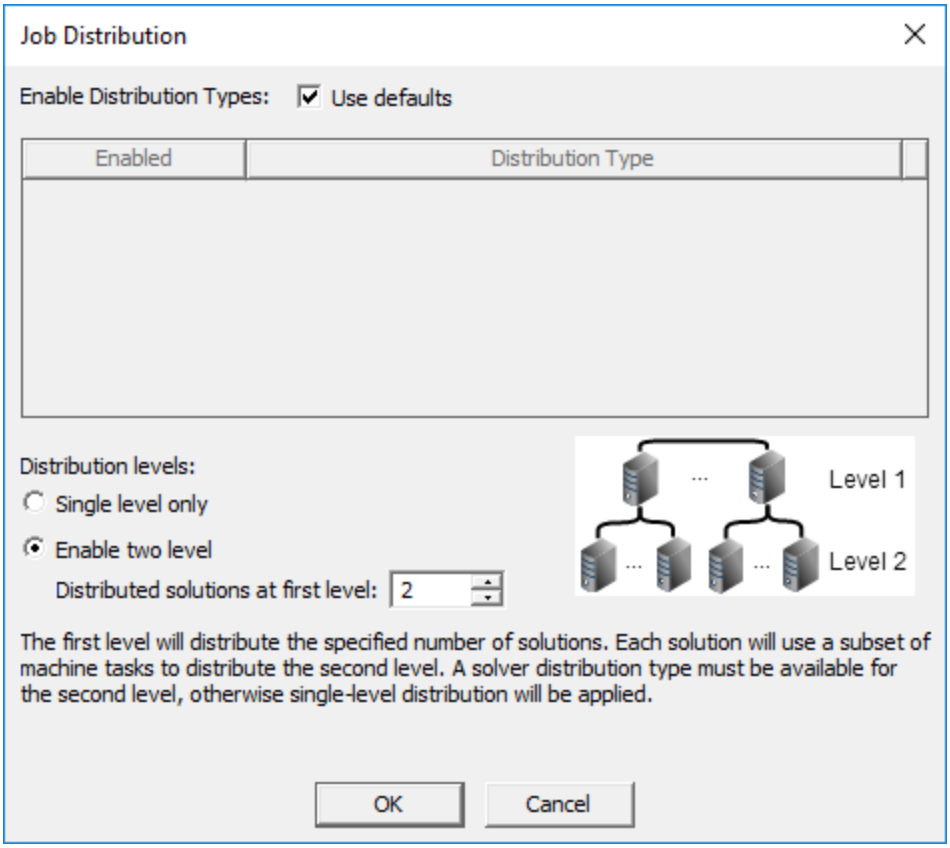
- Total number of tasks: The number of nodes requested for the job is the total number of tasks divided by limit on the number of tasks per node, rounded up if it is not an integer.
- Cores per distributed task. This determines the amount of multiprocessing per task.
- Whether nodes are for exclusive 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.



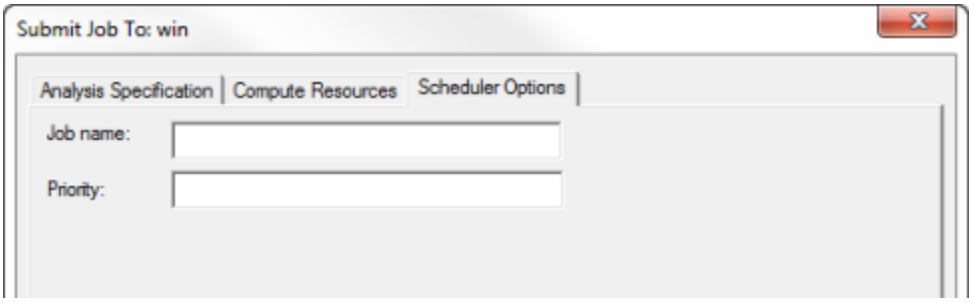
- Second level distribution operates within DSO. If available and enabled you can specify the number distributed solutions for level 1.



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.

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 *Microsoft HPC Pack 2019: Job Templates* white paper from Microsoft for additional details:

https://learn.microsoft.com/en-us/powershell/high-performance-computing/job-templates?view=hpc19-ps

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.

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. For Windows, there is a 64 core limit for a single machine.
- **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.

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

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.)

For additional information on supported schedulers, visit the Platform Support page. Then, locate and click the following link:

Ansys 2025 R1 - Job Schedulers and Queuing Systems Support

> **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, priorityuy, 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

For additional information on supported schedulers, visit the Platform Support page. Then, locate and click the following link:

Ansys 2025 R1 - Job Schedulers and Queuing Systems Support

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 in the installation directory.
- Turn OFF any 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 2025 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. See: Example Command Lines.

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 "C:\Program Files\ANSYS Inc\v251\AnsysEM\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]"
"C:\Program Files\ANSYS Inc\v251\AnsysEM\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]"
"C:\Program Files\ANSYS Inc\v251\AnsysEM\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 "C:\Program Files\ANSYS Inc\v251\AnsysEM\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
"C:\Program Files\ANSYS Inc\v251\AnsysEM\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 PBSPro 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)

For information about supported schedulers, visit the Platform Support page. Then locate and click the following link:

Ansys 2025 R1 – Job Schedulers and Queuing Systems Support

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.

Non Standard Installations for PBS

PBSPro

If the environment variable PBS_DEFAULT is set, then 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, then 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.

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, then there is no check for sufficient resources before submitting the job.
- If the user specified server is not the default server, then 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 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'
```

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.

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 EM -monitor Command Line Option for PBS

The Ansys EM **-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.

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 Ansoft 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 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.

Example PBS qsub Command Lines

All of the following examples show how to submit Linux hfss jobs on PBS, but similar command lines and job scripts will work for all Ansys EM products. Most of the following examples are PBS "Single-node jobs." The last example is a PBS "multi-node jobs"; this example demonstrates 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/ansys_inc/v251/AnsysEM/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/ansys_inc/v251/AnsysEM/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/ansys_inc/v251/AnsysEM/ansysedt -ng -BatchSolve -batchoptions
-machinelist num=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 Ansys Electronics Desktop 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/ansys_inc/v251/AnsysEM/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, 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 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/ansys_inc/v251/AnsysEM/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. This option is required for all batch jobs.

Integration with SLURM Scheduler (Linux Utility)

For Linux customers, Electronics Desktop 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. Using SLURM as a starting point, customers should be able to develop their own integrations for other schedulers. Electronics Desktop can check job states in SLURM scheduler within the current job monitoring GUI. 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 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".

For additional information on supported schedulers, visit the Platform Support page. Then, locate and click the following link:

Ansys 2025 R1 - Job Schedulers and Queuing Systems Support

See also: Integrating Ansys EM Tools with Third Party Schedulers.

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:

- ${ANSYS_EM_INSTALL_DIR}/schedulers/libgeneric_scheduler.so
- ${ANSYS_EM_INSTALL_DIR}/schedulers/proxies.cfg
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/persistence.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/SLURM_GetInfo_linux.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/SLURM_GetRunInfo_linux.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/SLURM_SubmitJob_linux.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/SLURM_ControlJob_linux.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/SLURM_GetJobState_linux.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/SLURM_LaunchProcess_linux.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/utils/ComputeResourceList.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/utils/ErrorAndDebug.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/utils/__init__.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/utils/RunCommand.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/utils/SiteCustomize.py
- ${ANSYS_EM_INSTALL_DIR}/schedulers/scripts/utils/slurm_srun_wrapper.sh

Environment Variables

There are some environment variables for the Generic Scheduler proxy. All environment variables apply on both Desktop and RSM Service.

`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.

`ANSYS_EM_EXEC_DIR`

Set this variable to the same value with ANSYS_EM_INSTALL_DIR (not the RSM service path). To ensure the RSM service can accurately locate the essential script files required for MPI integration, it is necessary to set this variable within the RSM service environment.

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= <INSTALL_DIR>/ansys_
inc/v251/AnsysEM/common/mono/Linux64/bin:<INSTALL_DIR>/ansys_
inc/v251/AnsysEM/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 the 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"
```

Integrating Ansys EM Tools with Third-Party Schedulers

This document indicates how to create a dynamically linked library to allow integration of Ansys EM 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, it is then automatically loaded and used by Ansys EM 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 EM 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) may be distributed to multiple analysis engines running in parallel.

- Serial Jobs
- Parallel Jobs

Serial Jobs

When an Ansys EM 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 EM provides an "out of the box" integrated solution that will work with the scheduler. In this case, the Ansys EM 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 EM Software Tools with other schedulers, the user may provide a way for Ansys EM 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 EM Tool and the scheduler. This library is loaded by the Ansys EM Tool at runtime, and if the Ansys EM Tool is running as part of a scheduler job, the Ansys EM 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 EM Tool and the scheduler.

In the second approach, the user creates an IronPython program to provide communication between the Ansys EM Tool and the scheduler. This program is loaded by the Ansys EM Tool at runtime, and if the Ansys EM Tool is running as part of a scheduler job, the Ansys EMs 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 EM 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. *Host requirements* apply to all hosts that may be allocated to an Ansys EM serial or parallel batch job.

Installation Requirements

The Ansys EM installation directory must be accessible from all cluster hosts using the same path. One way to achieve this is to place the Ansys EM 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 EM tool locally on each cluster host using the same local directory path.

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 path.

For cloud submission such as Cloud Direct, Ansys HPC Platform Services, and Ansys Cloud Burst Compute there are no requirements for shared drives or shared directories (installation or project).

Using a Shared Library (Linux) or a DLL (Microsoft Windows)

This section describes how to create a dynamically linked library to allow integration of Ansys Electromagnetics Suite 2025 R1 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 Suite 2025 R1.

Installation Details

The scheduler proxy library must be installed in the schedulers subdirectory of the Ansoft installation directory. For example, if the Ansys EM installation directory is "C:\Program Files\v251\AnsysEM", then the scheduler proxy library must be installed in directory "C:\Program Files\ANSYS Inc\v251\AnsysEM\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.

Build Information for Scheduler Proxy Library

This section contains the recommended compiler and linker settings for building a scheduler proxy library.

- 64-Bit Microsoft Windows
- Linux

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]
- 64-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 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 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
- GetMessageStringToRegisterForSigTerm
- 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.

> **Note:**
>
> 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 then 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.

Note:

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 then 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.

Note:

- 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.

GetMessageStringToRegisterForSigTerm

Purpose

Obsolete. The string copied to the buffer should be an empty string.

Signature

```
extern "C" bool FN_PREFIX_GetMessageStringToRegisterForSigTerm
(char * buffer, unsigned int* length);
```

Arguments

buffer: Pointer to a character buffer to contain the string 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, including the terminal null byte, then the string is copied to the buffer and true is returned. If the buffer length is insufficient for the string, then the buffer is unchanged, and false is returned.

> **Note:**
>
> To get the string, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the string. 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.

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. 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.

Note:

- 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 Ansoft 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 Ansoft RSM service executable per host, and the Ansoft RSM executable will launch all of the engine processes.

If the Ansoft 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 Ansoft 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 Ansoft RSM service should be used to launch engine processes for this scheduler. Returns false if the Ansoft RSM service should not be used to launch engine processes for this scheduler.

> **Note:**
>
> 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 then 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.

Note:

- 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 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.

Note:

- 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.

Testing Scheduler Integration

One way to test these functions is to run the analysis for an Ansys EM 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 EM 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

Many schedulers create a temporary directory for each job and delete the directory after the job finishes. One way to verify that this function is working correctly is to determine the pathname that the scheduler uses for the temporary directory and to monitor the contents of the temp directory as the job is running. If the analysis engines write files to this directory as the job runs, then this function is working.

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., frequency sweep, 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., frequency sweep, 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

- Troubleshooting LaunchProcess
- Troubleshooting GetUseRsmForEngineLaunch

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 Electronics Desktop installation directory.

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 EM tools.

Installation Details

The IronPython program must be installed in the schedulers subdirectory of the Ansys EM installation directory. For example, if the installation directory is "C:\Program Files\ANSYS Inc\v251\AnsysEM", then the IronPython program must be installed in directory C:\Program Files\ANSYS Inc\v251\AnsysEM\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 is 2.7.0.40

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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

> **Note:**
>
> 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.

Note:

- 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.

Scheduler Proxy Interfaces

Scheduler proxy supports 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.

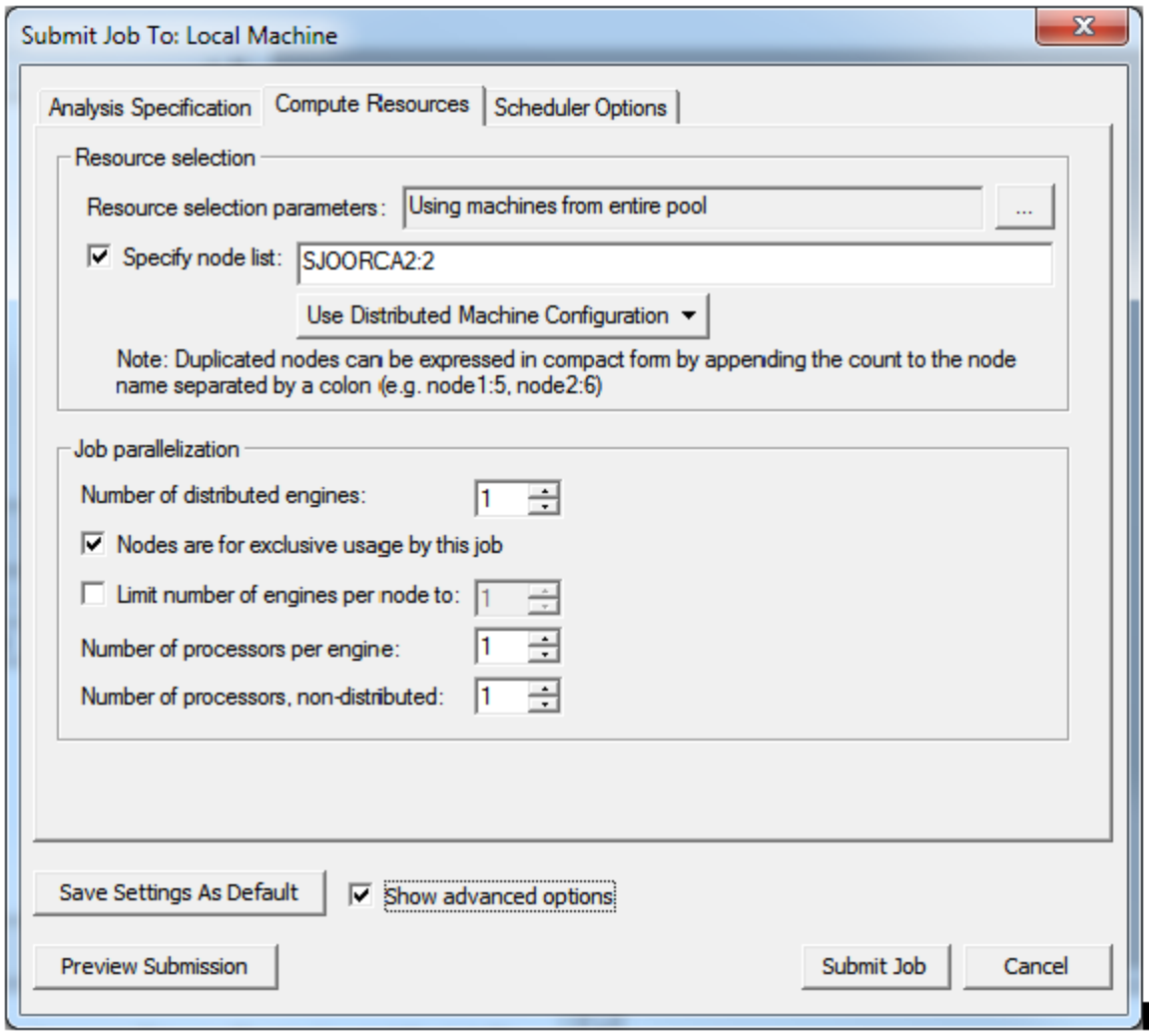
int GetSchedulerInfo(std::string& msg, std::string& schedulerName, std::string& schedulerDescription, std::string& schedulerVersion):

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* tab or the *Submit Job To* dialog box allows the user to specify scheduler specific resources. This function returns the information used to create and populate the *Compute Resource* tab.



Each line in the dialog box 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.

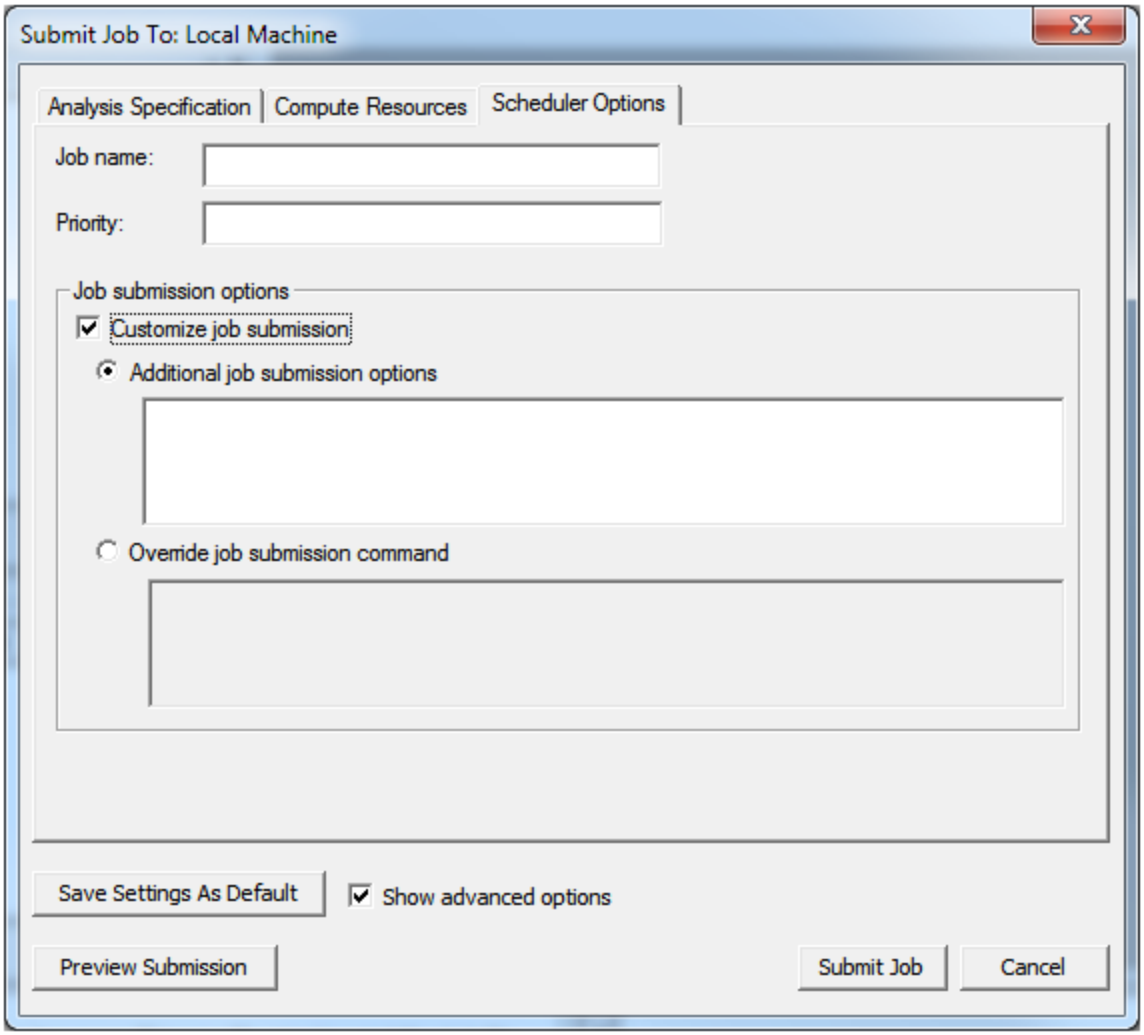
The **computeResources** argument is a reference to an object of type **UniformComputeResourcesStruct**. This **struct** contains zero or more resource attribute settings for the job. Each resource attribute setting consists or a resource name and a resource value. The resource name is the name of one 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* tab of the *Submit Job To* 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 *Scheduler Options* tab of the *Submit Job To* dialog box allows the user to either specify additional job submission options or to specify all submission options, replacing the settings from the other *Submit Job To* dialog box 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.

Ansys Cloud Direct Support for HPC Job Management

Ansys Electromagnetics Desktop supports the Ansys Cloud Direct 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**.

> **Important:**
>
> 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 Direct.

Uploads to and downloads from Ansys Cloud Direct 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 Direct 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.

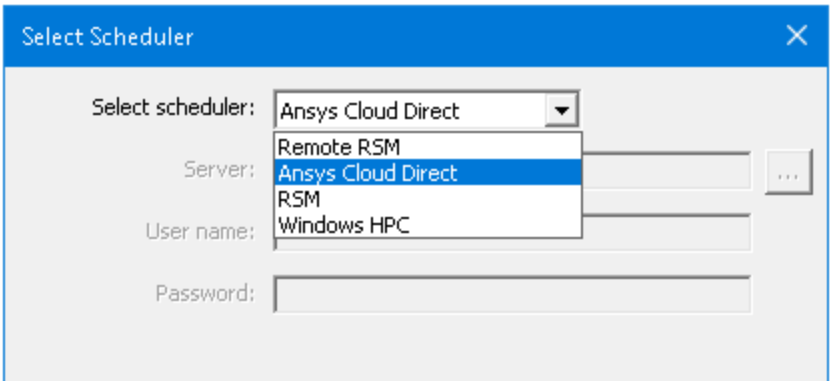
- You must select Ansys Cloud as your Job Management Scheduler, and, if necessary, log in to the Ansys Cloud Service using the Ansys Discovery Account and associated password.

Job Management UI for Ansys Cloud Direct

You can use Ansys Electronics Desktop to submit batch jobs to Ansys Cloud Direct and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select Ansys Cloud Direct as the scheduler.



2. Click **Log in** to launch a login window for Ansys Cloud Direct.

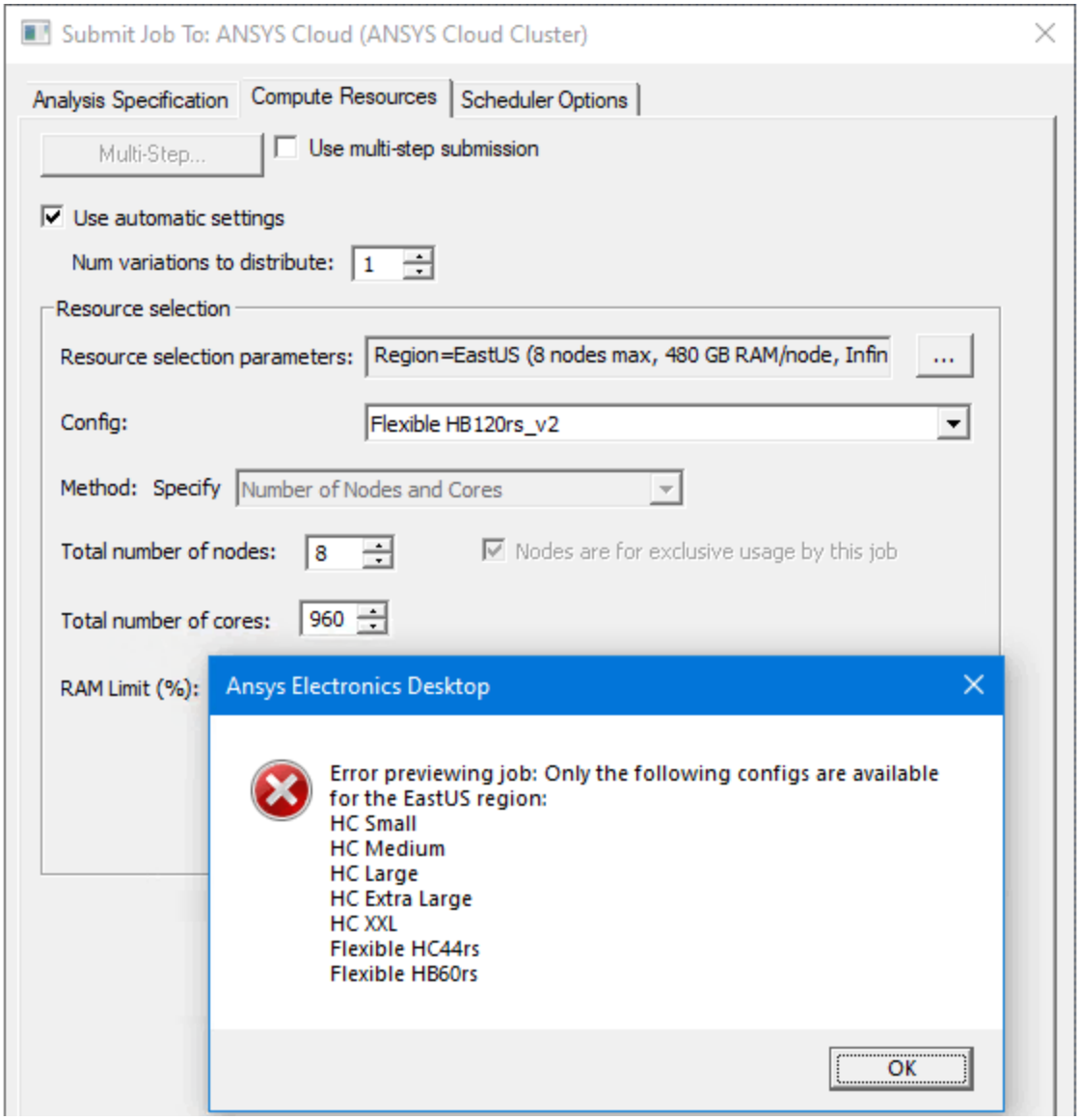
   After a successful login, you may not need to log in again for several days if you do not log out.

   If you wish to log out, click **Log out** from the **Select Scheduler** window.

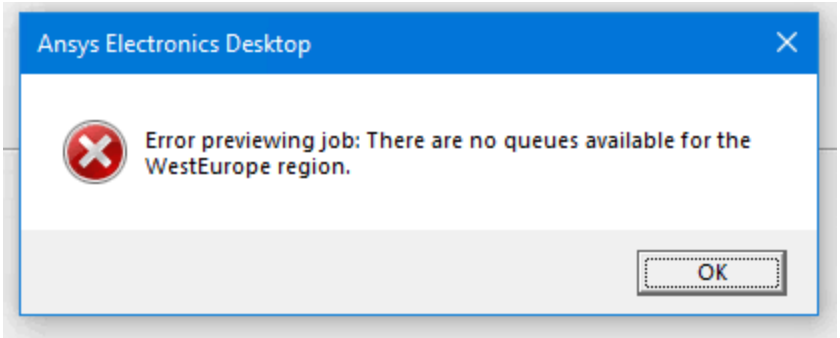
3. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to Ansys Cloud Direct.

   Ansys Cloud Direct-specific Settings:

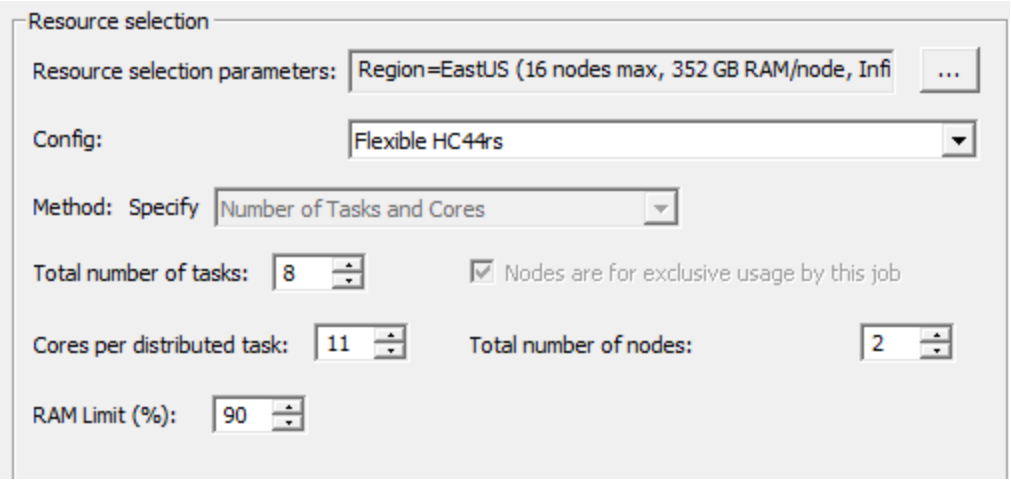
   - On the **Compute Resources** tab, **Resource Selection Parameters** will allow you to select a **Region** (for example, EastUS, JapanEast, WestEurope). The **Config** drop-down menu does not display configurations that are not available in any of the selectable regions. That is, if the regions are EastUS, WestEurope, and SouthwestUS, and none of these regions have the smallhc-based queues, then the "HC Small" configuration does not appear. 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.



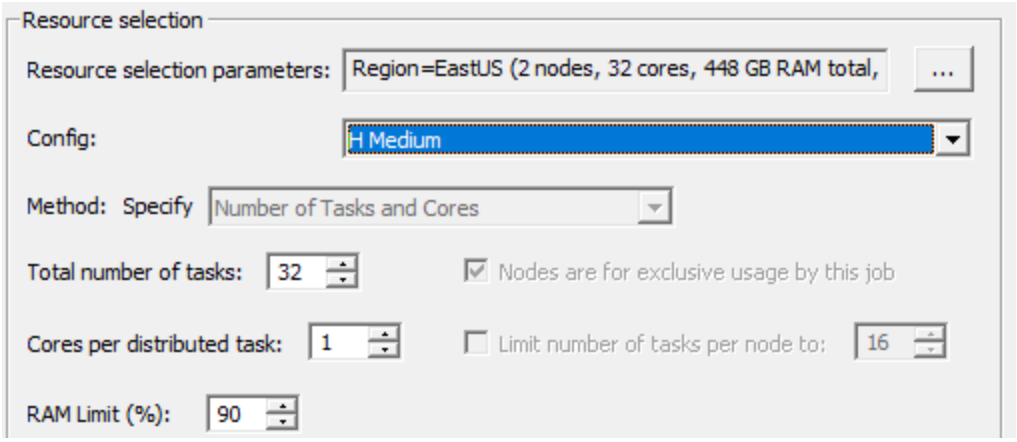
If no queues are available, an error message provides that message:



- 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.

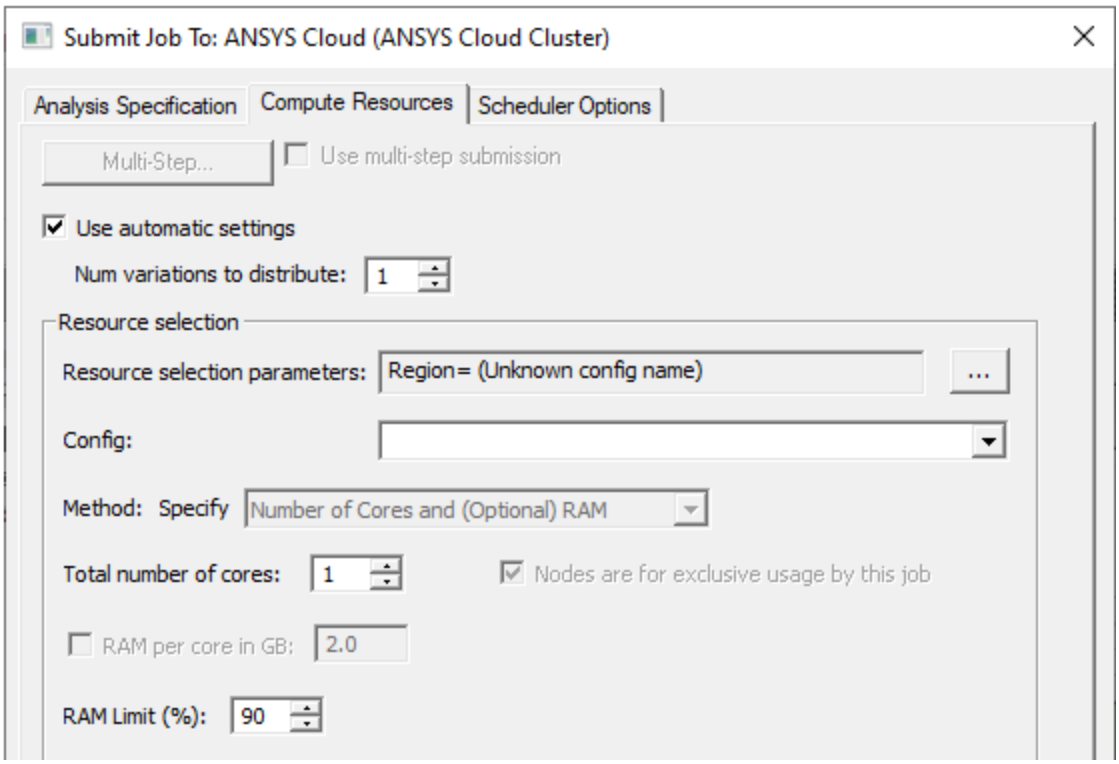


In comparison, if you selected a fixed configuration, the **Resource selection** panel appears like this:

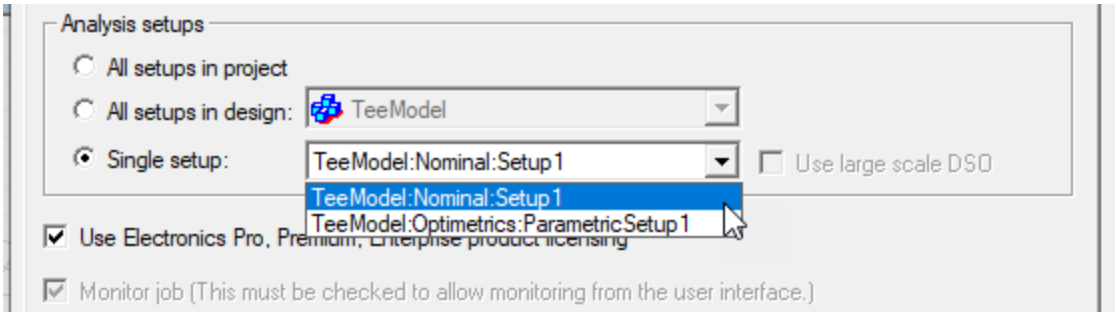


> **Important:**
>
> Attempts to use more for submission will result in error messages.

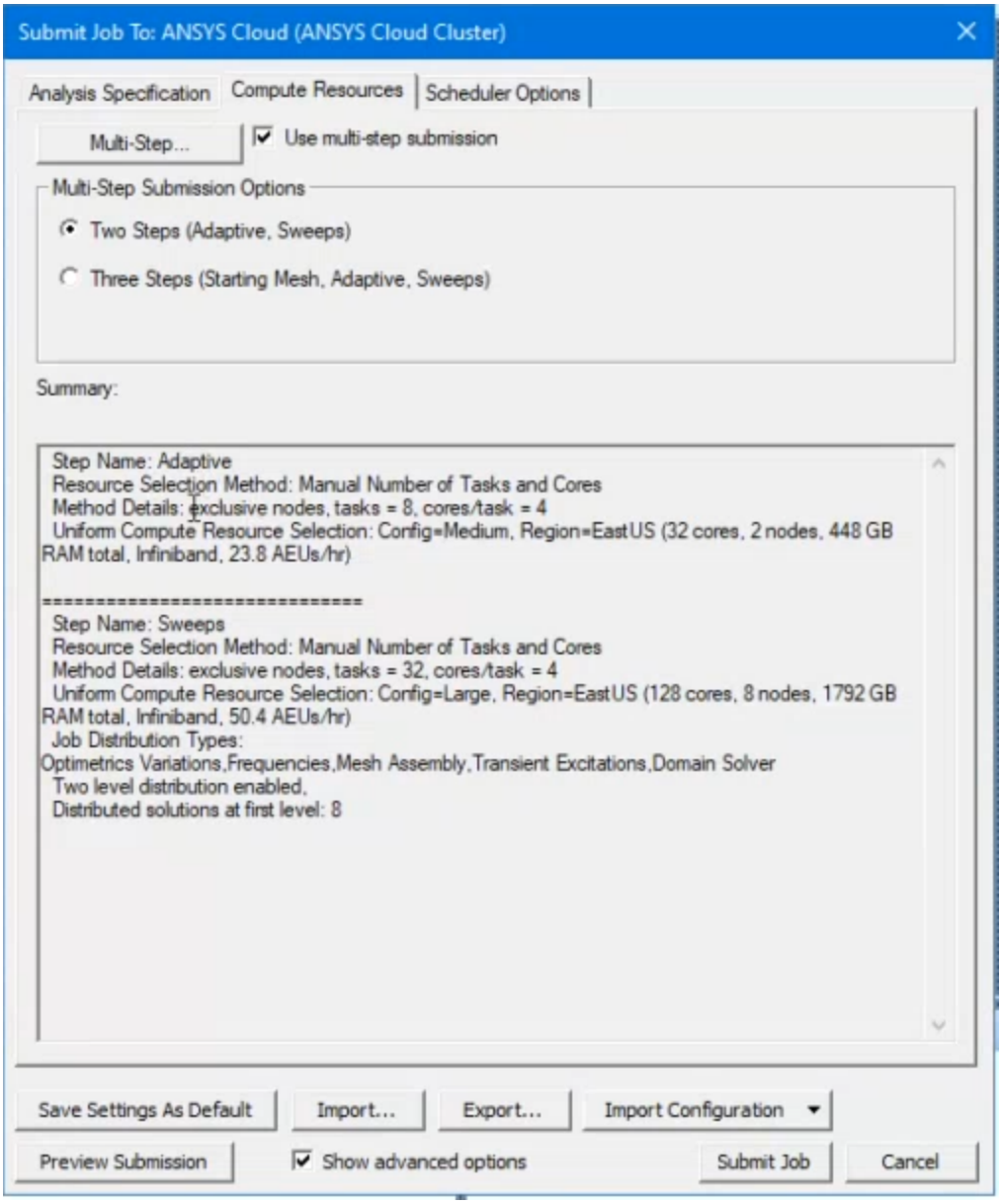
- On the **Compute Resources** tab, if you select **Use automatic settings** with **Num variations to distribute** set to 1, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. If you set **Num variations to distribute** to 2 or more, Optimetrics variations will be solved in parallel. Other distribution types will be distributed automatically.



- In response to a set of minimal constraints, Ansys Cloud Direct 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 Direct, you must specify a single setup on the **Analysis Specification** tab.

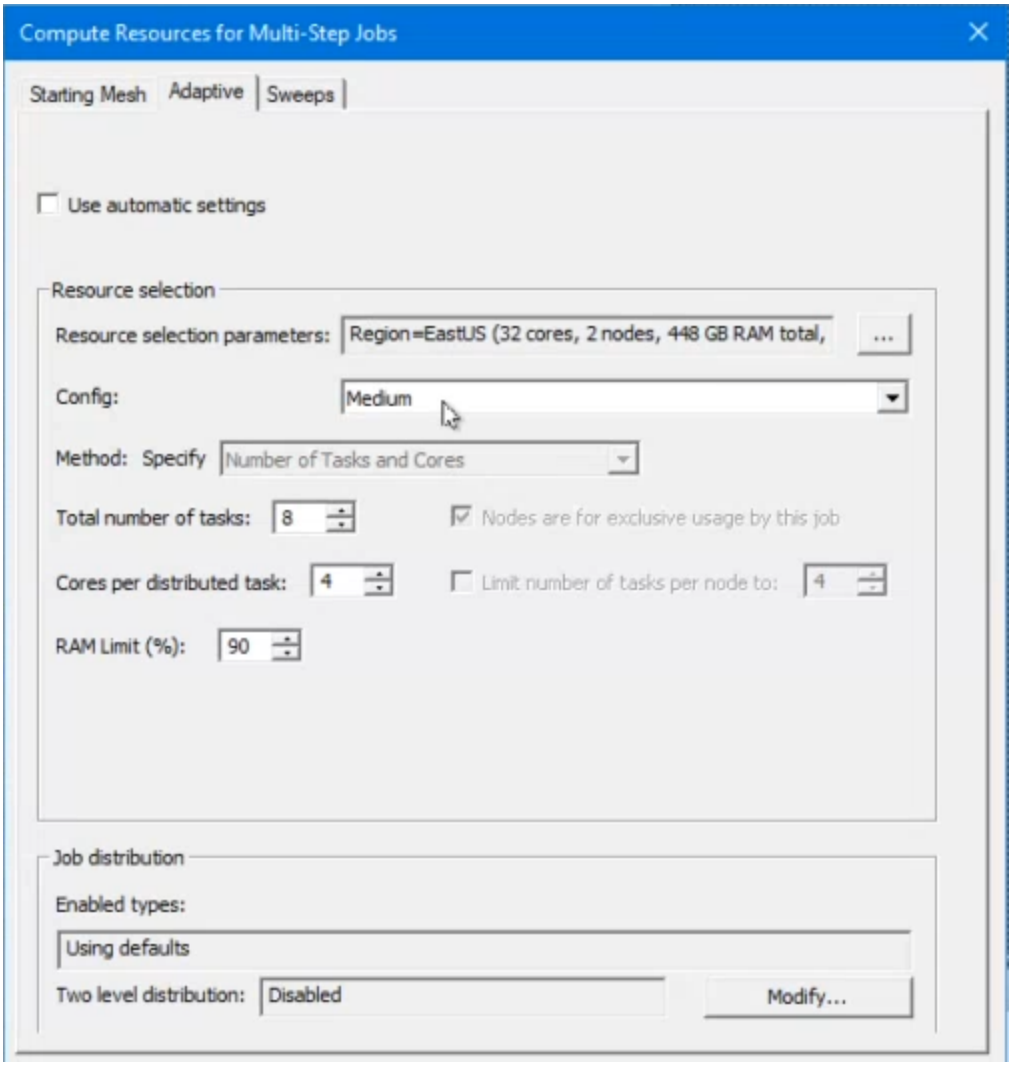


This enables **Multi-Step** options on the **Compute Resources** tab.

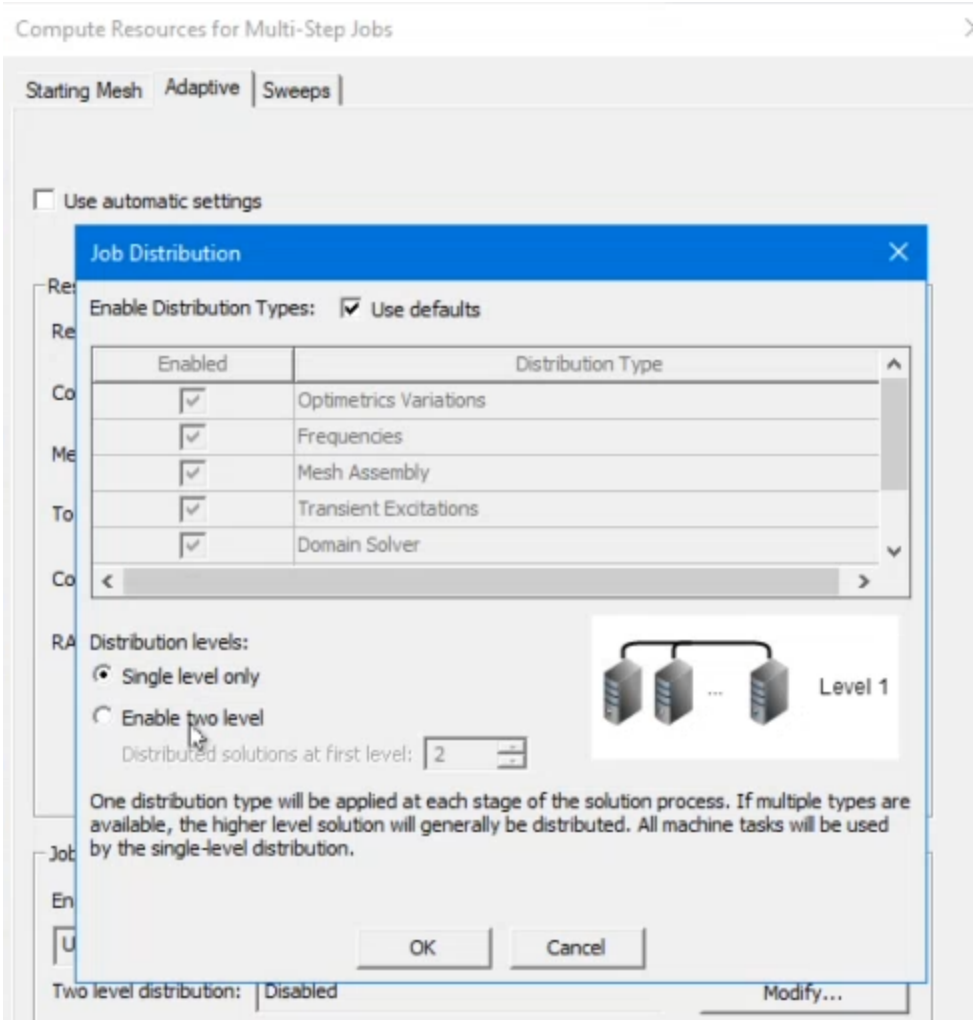


Click the **Multi-Step** button to open the **Compute Resources for Multi-Step Jobs** dialog box. It has tabs for **Starting Mesh**, **Adaptive**, and **Sweeps**.

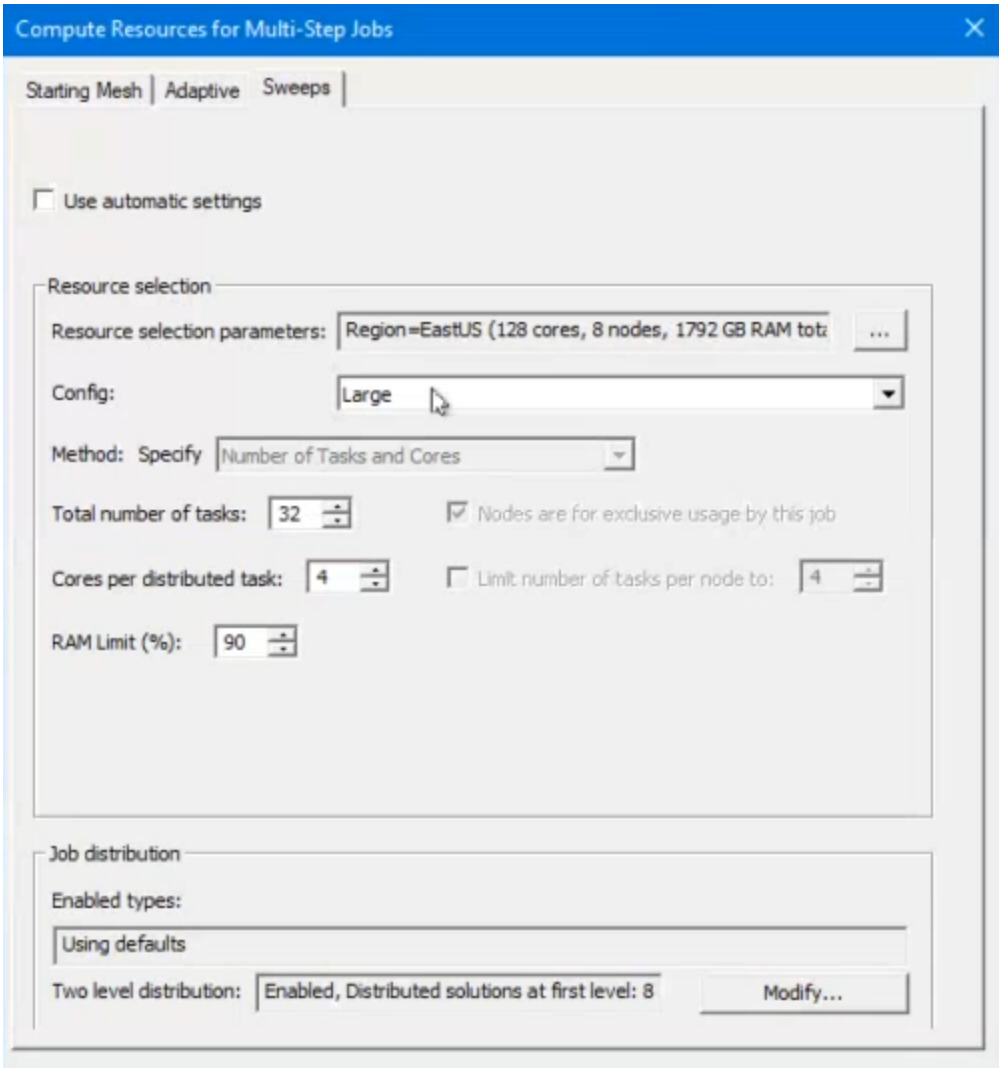
On the **Adaptive** tab, you can **Use automatic settings**, or specify **Resource Selection**



Click the **Modify...** button on the **Adaptive** tab to open the **Job Distribution** dialog box.

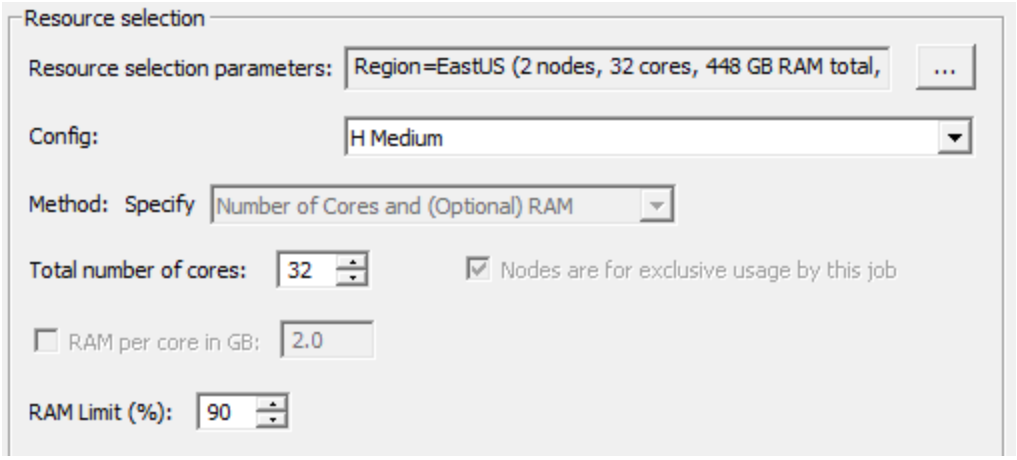


The **Sweeps** tab lets you specify resources or **Use automatic settings**.

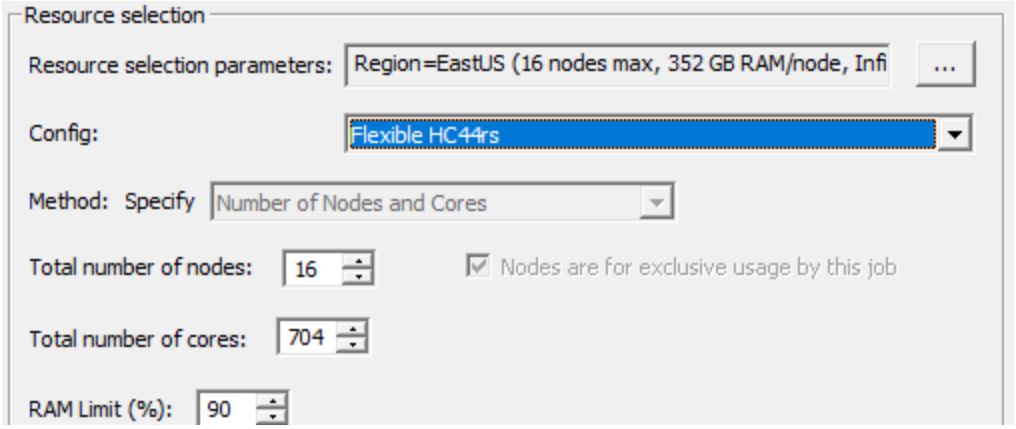


The **Sweeps** tab includes an independent **Modify...** button to specify **Job Distribution**.

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:

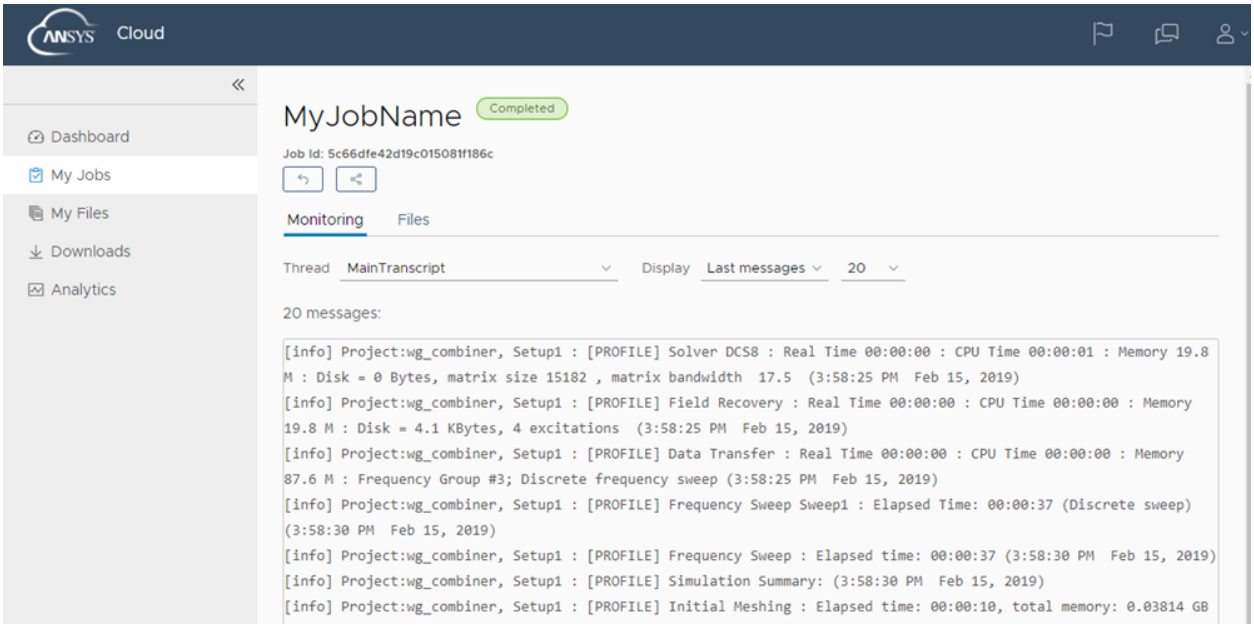


If you select flexible queue for automatic settings, The **Resource selection** panel only allows nodes and cores for a flexible queue:



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 Direct

When submitting jobs to Ansys Cloud Direct from Electronics Desktop, certain environment variables 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 Direct

The following option is automatically added for Ansys Cloud Direct 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

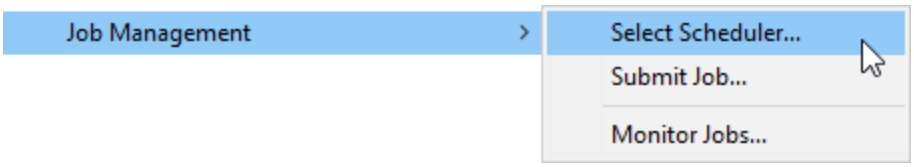
The license type batch option ("HPCLicenseType") is forced to be "Pack" because Cloud is configured to only work with pack licensing. The following MPI batchoptions are automatically set as "Intel" because Ansys Cloud Direct is configured to only work with Intel MPI.

- HFSS 3D Layout Design/MPIVendor
- HFSS/MPIVendor
- Icepak/MPIVendor
- Maxwell 2D/MPIVendor
- Maxwell 3D/MPIVendor
- Q3D Extractor/MPIVendor

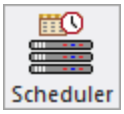
Selecting a Scheduler

Access the **Select Scheduler** window one of three ways:

- Click **Tools** > **Job Management** > **Select Scheduler...**

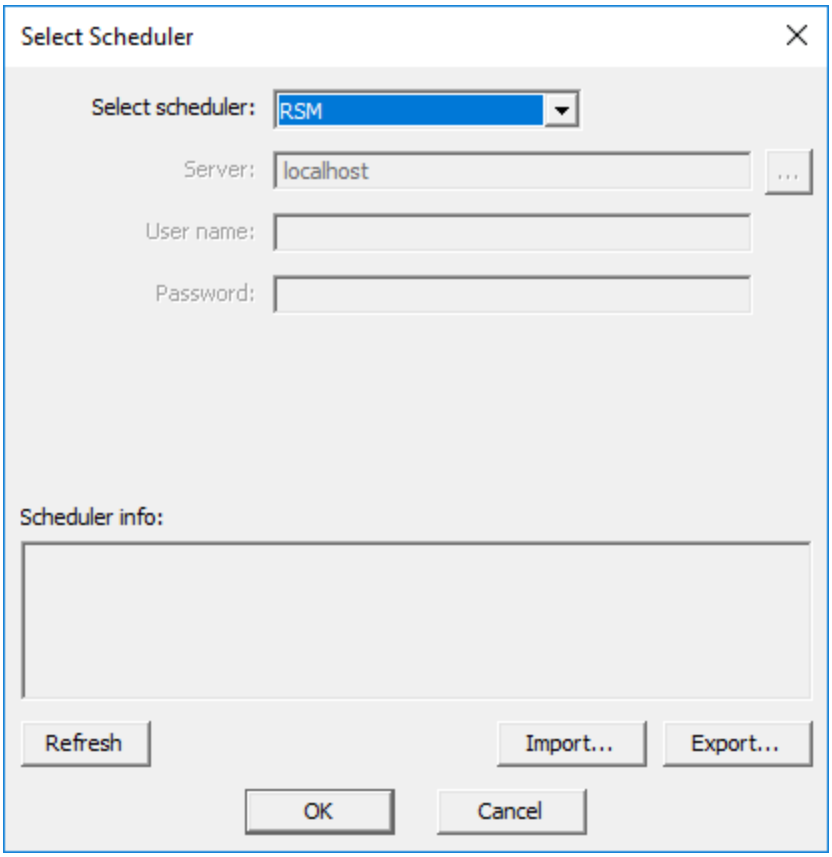


- Select the **Simulation** tab and click the **Scheduler** icon.



- From a command window, use the -showselectscheduler command.

The **Select Scheduler** window appears:



From the **Select Scheduler** window:

1. 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.

2. If applicable for the scheduler type, enter server and user information. If Ansys Cloud Direct or Ansys Cloud Burst Compute is selected then the “Login” button will open the default browser and prompt for login.

   Information about the selected scheduler appears in the **Scheduler info** field.

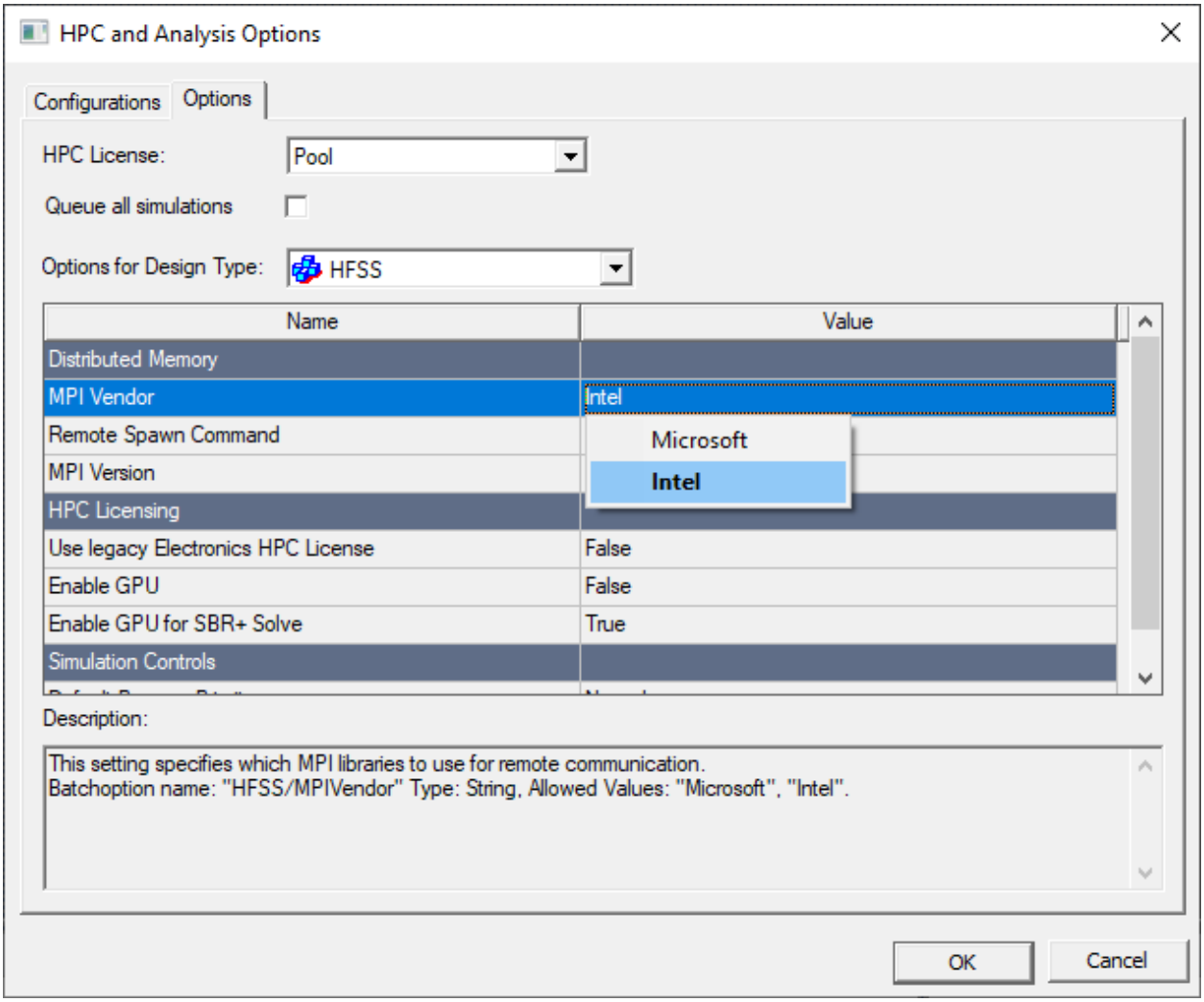
3. Click **OK** to complete your selection.

MPI Vendor and MPI Version

For HFSS, HFSS 3D Layout, Q3D Extractor, Maxwell, and Icepak design types:

These settings may be used to specify the MPI Vendor and Version for distribution via MPI.

Go to the **Options** tab of the **HPC and Analysis Options** dialog box, and select the design type. Then you can select the MPI Vendor and MPI version.



MPI Vendor

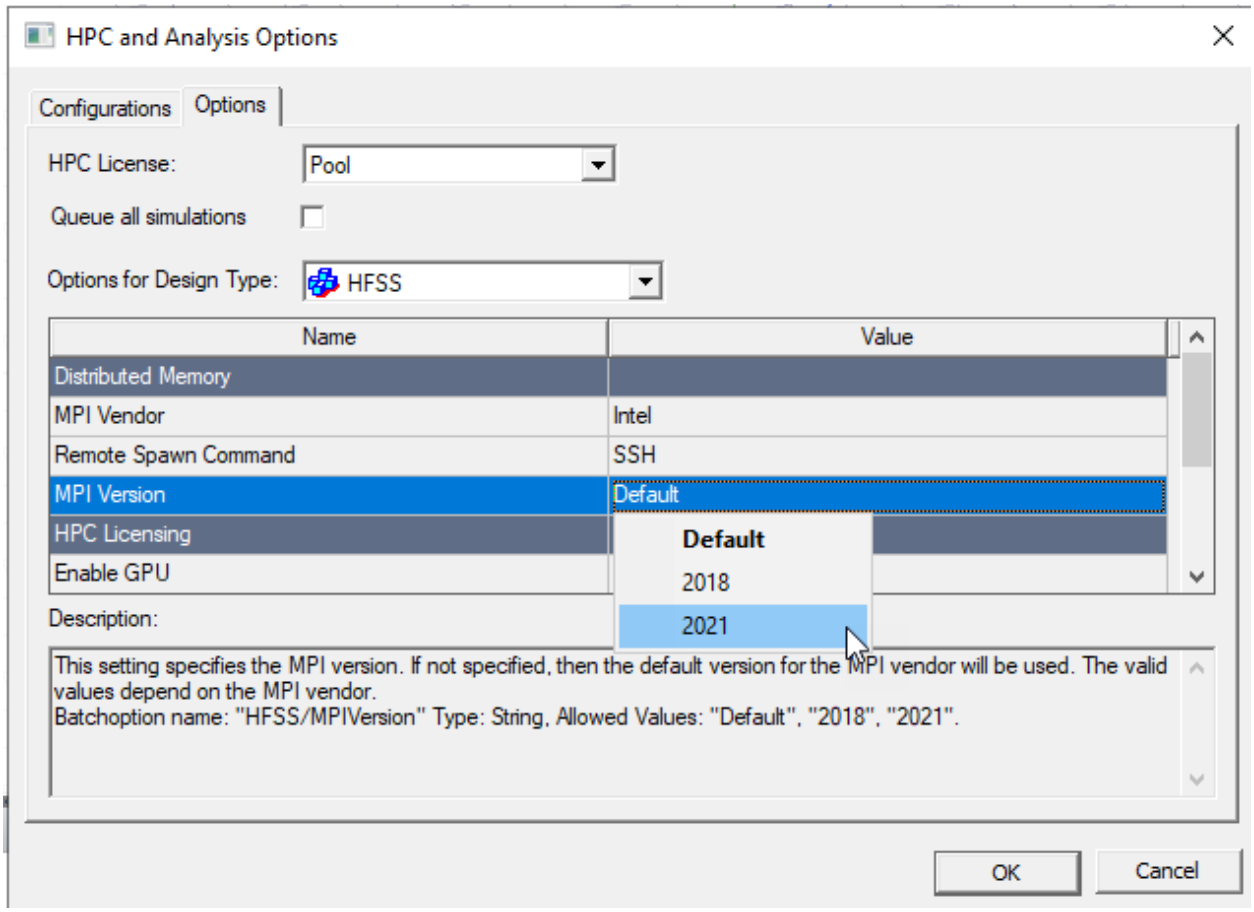
The default MPI Vendor is "Intel". On Linux, "Intel" is the only supported MPI vendor. On Windows, both "Intel" and "Microsoft" are supported MPI vendor options. For jobs distributed to multiple hosts, the "Microsoft" MPI Vendor is only supported when running as a Windows HPC job.

MPI Version

If only a single MPI Version is supported for the selected MPI Vendor, then the MPI Version option is ignored. 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 2021 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.



Also see Setting Intel MPI Interconnect for more details.

Submitting a Job

Ansys Electromagnetics Desktop supports its own Remote Simulation Management (RSM) and Ansys Cloud Direct 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.

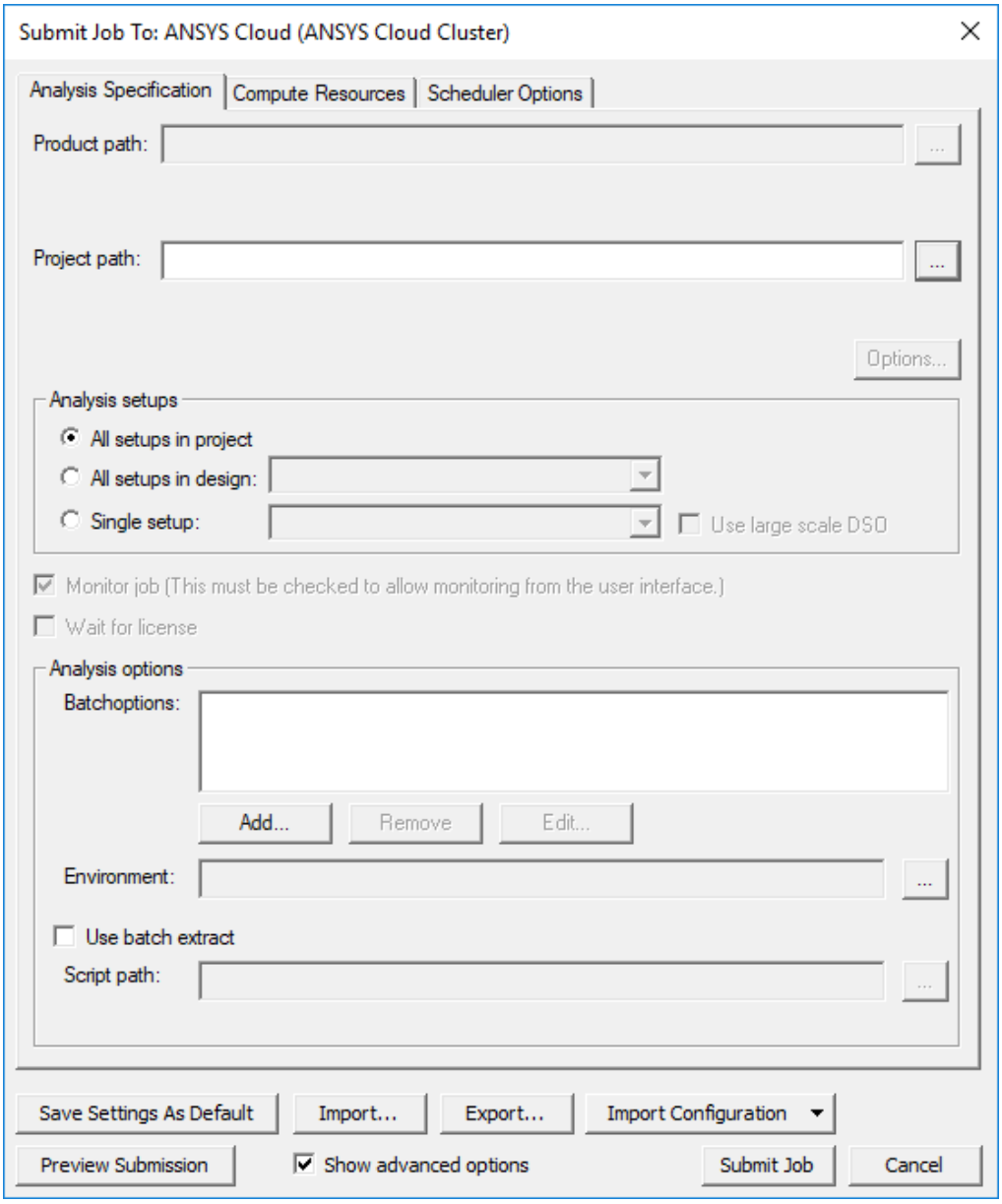
> **Note:**
>
> For certain methods of resource selection for Windows HPC job submission, Ansys Electronics Desktop checks for nodes being online. In an auto-scaling cluster, nodes can fail the online check because they're not online until a job is running. An environment variable allows customers to bypass this check:
>
> ANSYSEM_SKIP_NODE_ONLINE_CHECK
>
> Set the variable value to 1 to disable the online check. Debug logging fully verifies what aspects of the cluster are causing this check to fail.

Job submission in Windows works without changing settings if environment variable ANSYSEM_SUBMIT_JOB_REQ_NODE_ONLINE is set with value 0. The installation doesn’t set this environment variable. ANSYSEM_SUBMIT_JOB_REQ_NODE_ONLINE accepted values: 0 or 1 (0 = disable; 1 = enable, i.e. require nodes to be online for submission). Default: 1 (i.e. if not set, Ansys Electronics Desktop requires nodes to be online for submission).

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 **Q3D Extractor** or **2D Extractor** > **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.

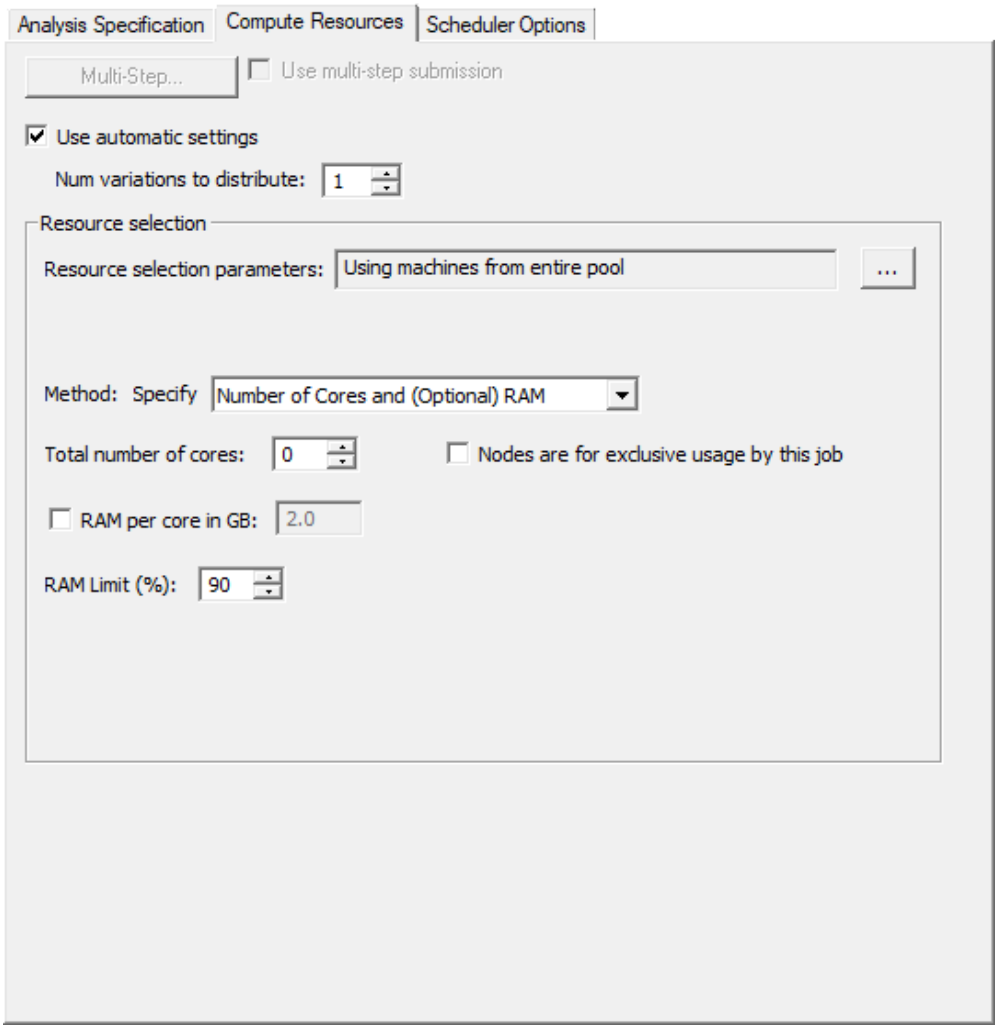
   The **Product path** and **Project** fields support mapped drives. Click the ellipses (**...**) to select files.

   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.

   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.



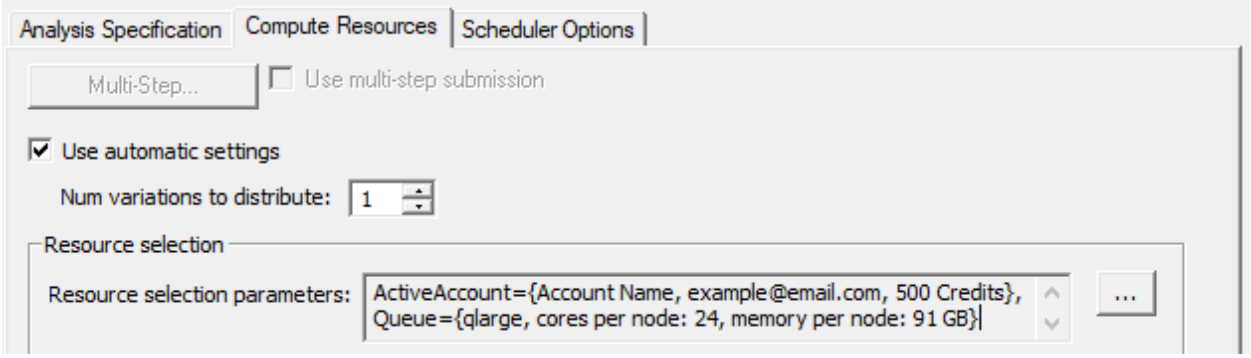
Options may vary slightly depending on your selected scheduler.

5. If applicable, determine whether or not to **Use multi-step submission**.
6. Determine whether or not to **Use automatic settings**.

> **Note:**
>
> Ansys strongly recommends using Automatic settings.

7. Use the buttons in the **Resource selection** area to allocate resources. See: Job Management for Large Scale DSO. If you are using Ansys Cloud Burst Compute select an active account using the ellipsis […] button to the right of the **Resource selection parameters:** text box and ensure that it has sufficient Core Hour Balance. This balance

will be displayed in the **Resource selection parameters:** text box.



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 Direct 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.

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.

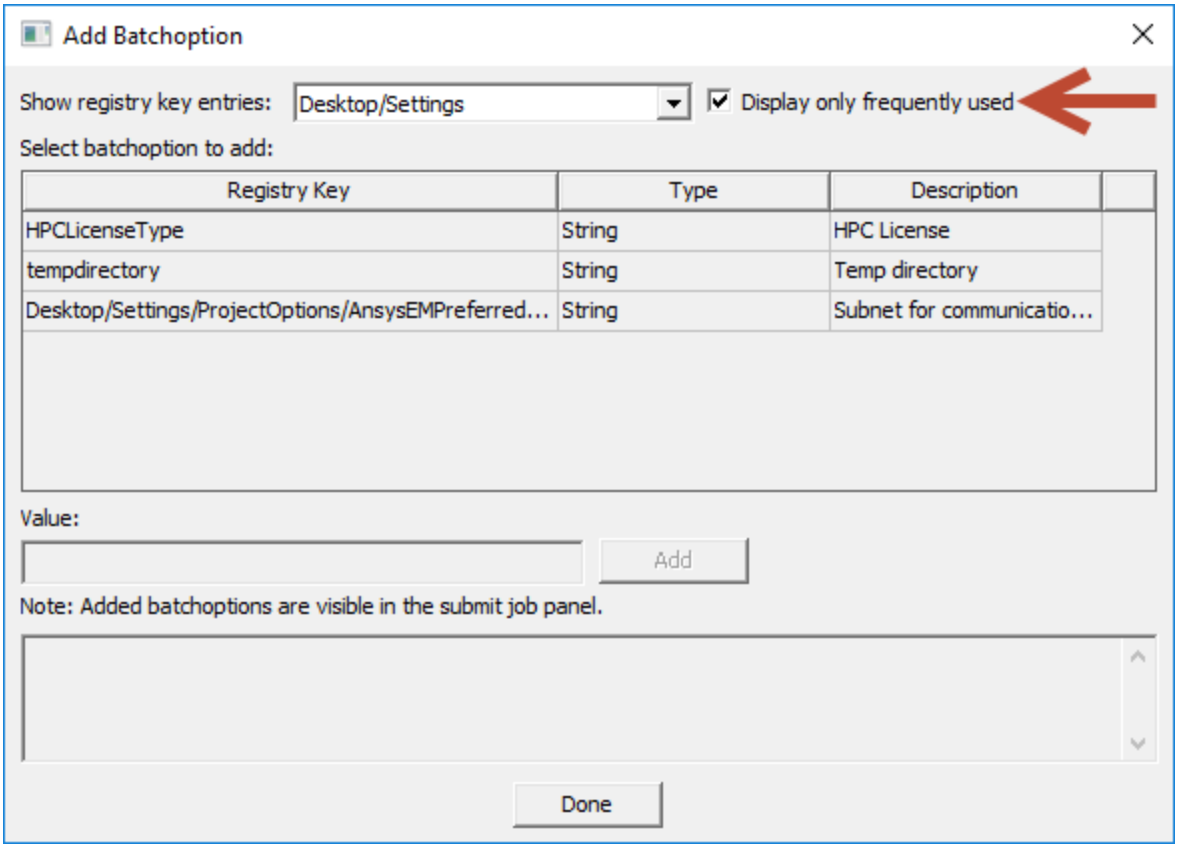
This topic covers the following:

- Batchoptions
- Environment Variables

- Batch Extract
- Customize Job Submission

Batchoptions

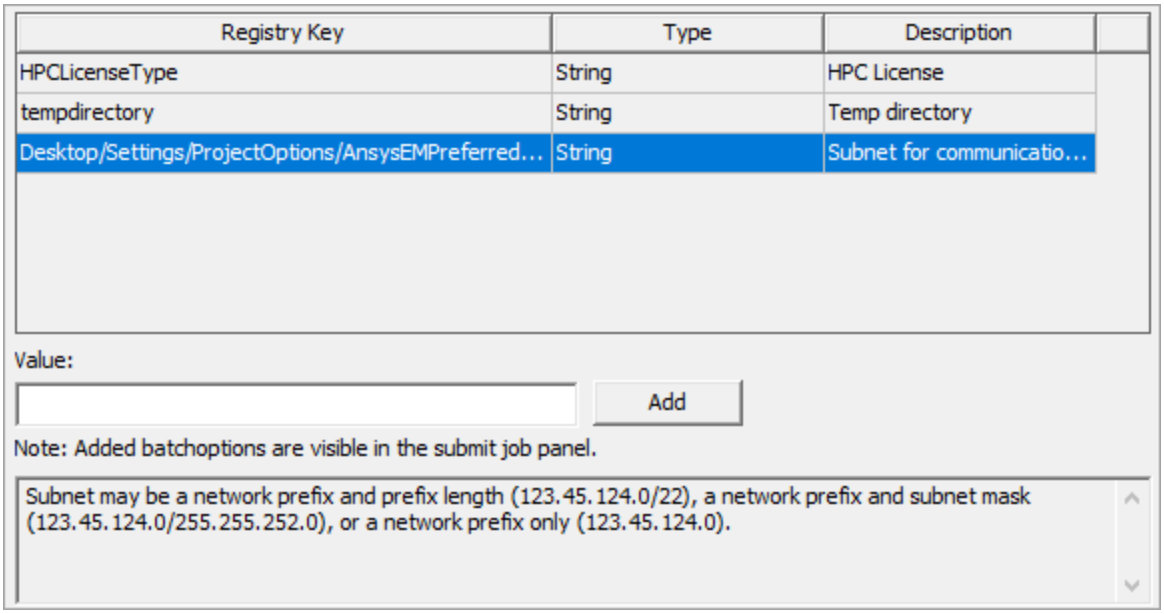
In the **Submit Job To** window, under **Batchoptions**, click **Add** to open the **Add Batchoption** window.



The **Show registry key entries** drop-down menu allows you to select categories of registry keys to display.

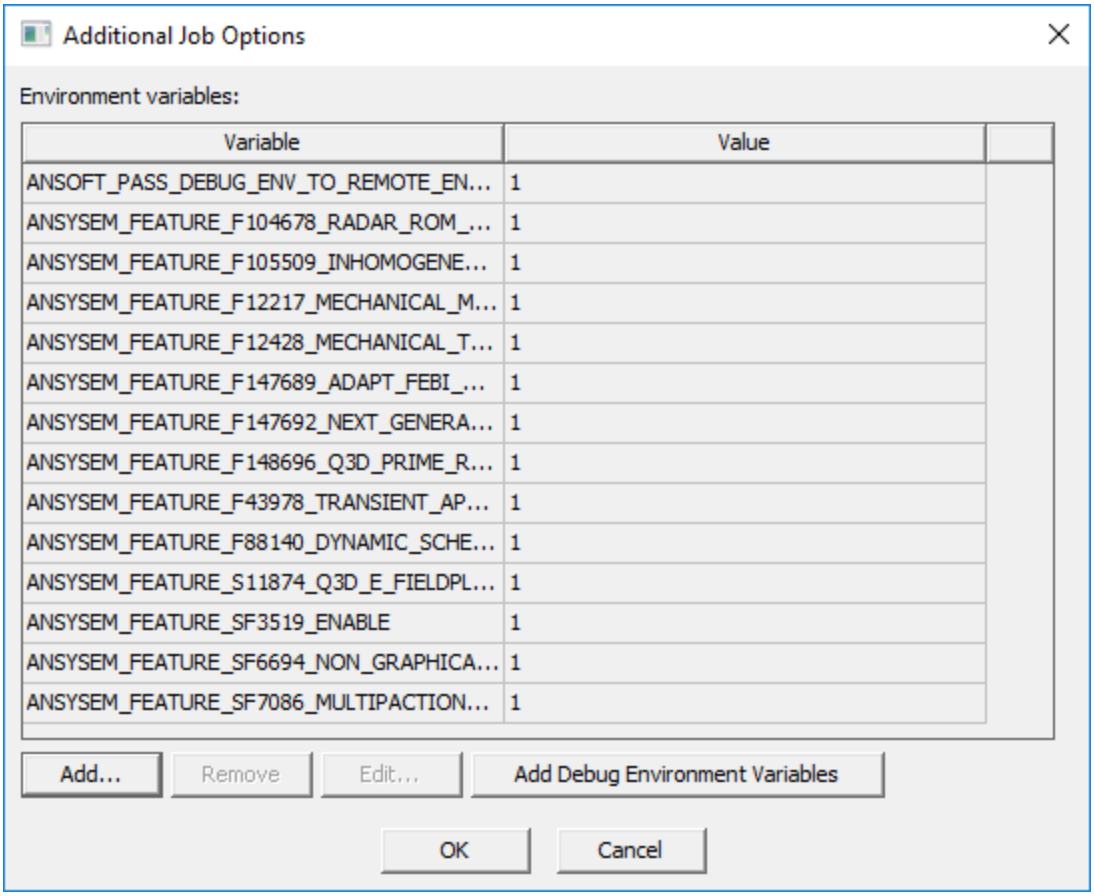
Note that **Display only frequently used** is selected by default. Deselect this to view all options for the selected category.

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.

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" on the facing page
- "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 page 5-196

Serial Job on a Single Processor

Suppose Ansys Electronics Desktop is installed at "C:\Program Files\ANSYS Inc\v251\AnsysEM\" and you are using RSM for DSO:

```
"C:\Program Files\ANSYS Inc\v251\AnsysEM\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\ANSYS Inc\v251\AnsysEM\ansysedt.exe" -ng
-BatchSolve -machinelist num=3 -monitor -local \\shared_
drive\projs\OptimTee.aedt
```

Distributed Job Using Four Processors

Ansoft RSM

```
"C:\Program Files\ANSYS Inc\v251\AnsysEM\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\ANSYS Inc\v251\AnsysEM\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\ANSYS
Inc\v251\AnsysEM\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\ANSYS
Inc\v251\AnsysEM\ansysedt.exe" -ng
-BatchSolve -Distributed -machinelist num=4
-batchoptions \\shared_drive\registry.txt
\\shared_drive\projs\OptimTee.aedt
```

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

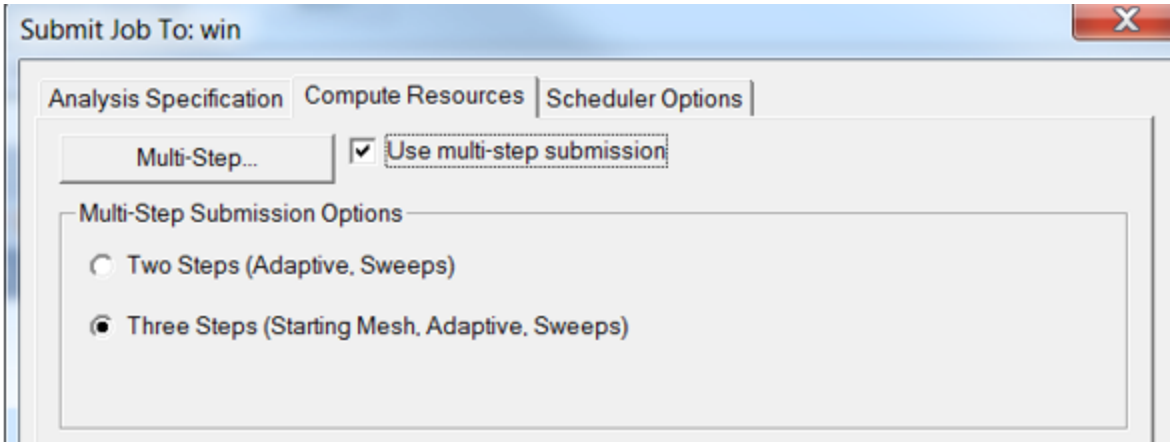
the first job to do initial meshing and adaptive passes, only reserving a single compute node, or maybe even reserving just a partial node. The second job can then do the frequency sweep(s), reserving and using multiple nodes. Note that while the first job runs, because it may only be using one node, 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

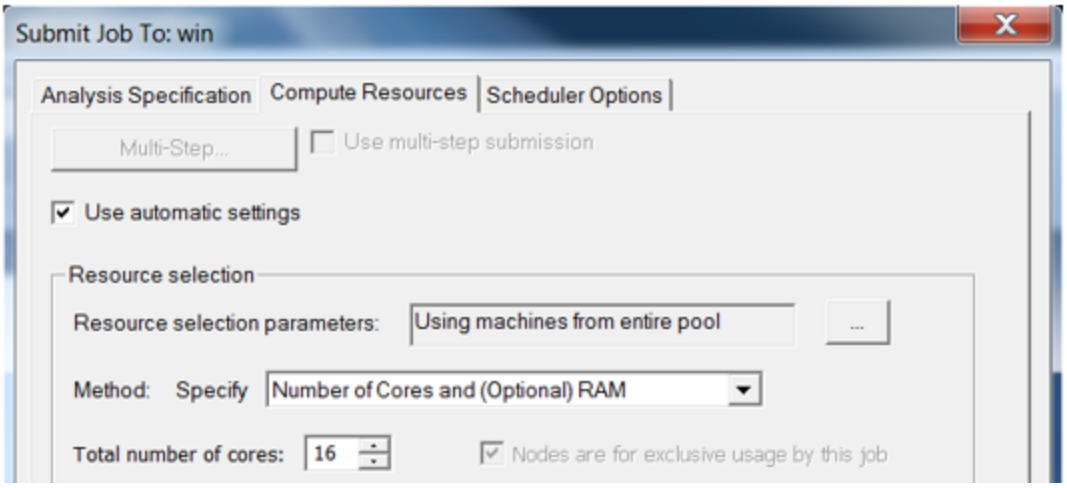
Multi-step job submission is only allowed for a single setup of a project. Only certain design types (or setup types for a given design type) offer this functionality. Ansoft RSM does not support multi-step jobs because it does not have queuing capabilities.

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.



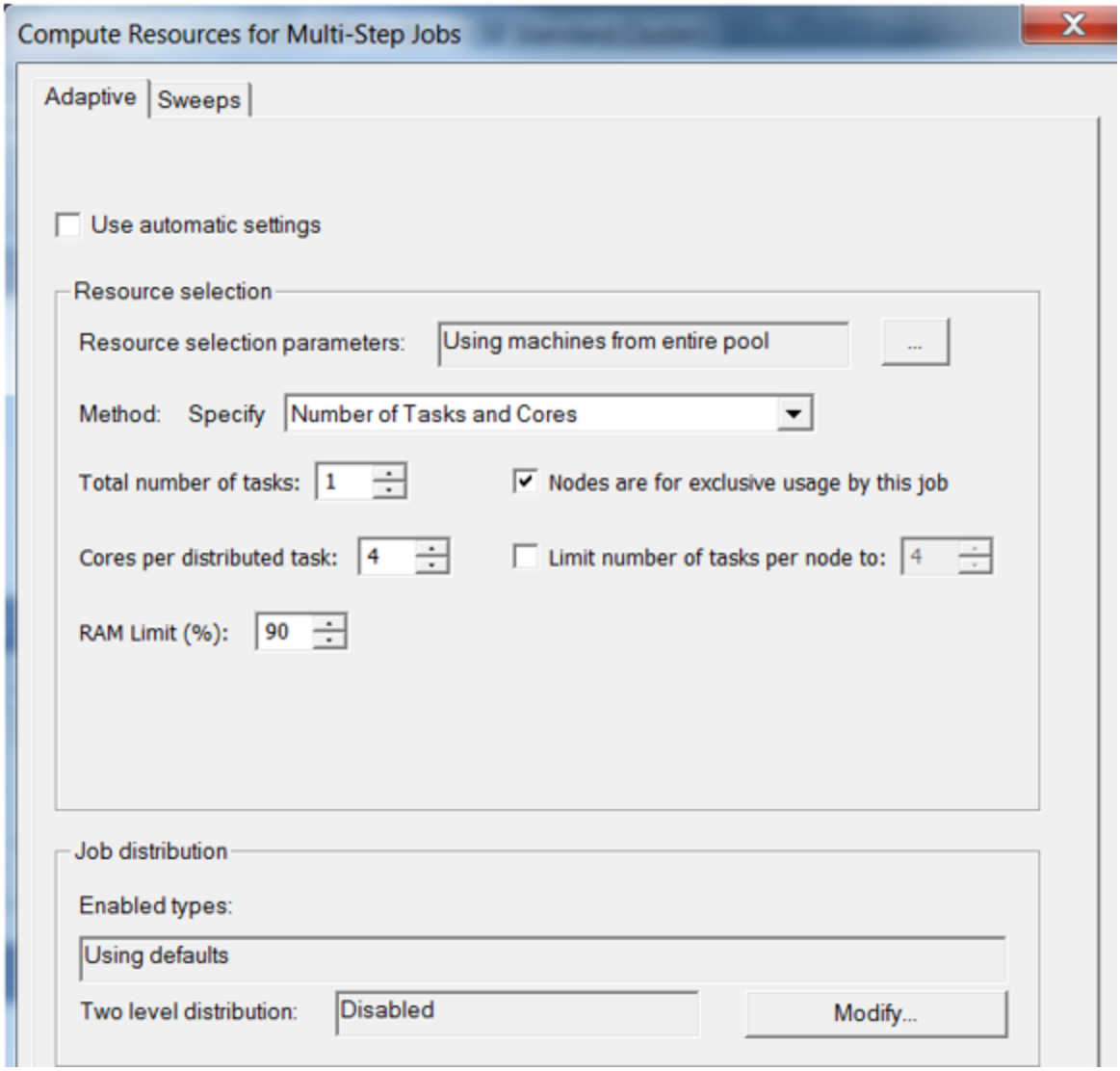
If the **Multi-Step** button and check box are not enabled, it could be because you have not selected a single setup, the design type of the setup does not support Multi-Step, or the scheduler type (e.g., RSM) does not support it.



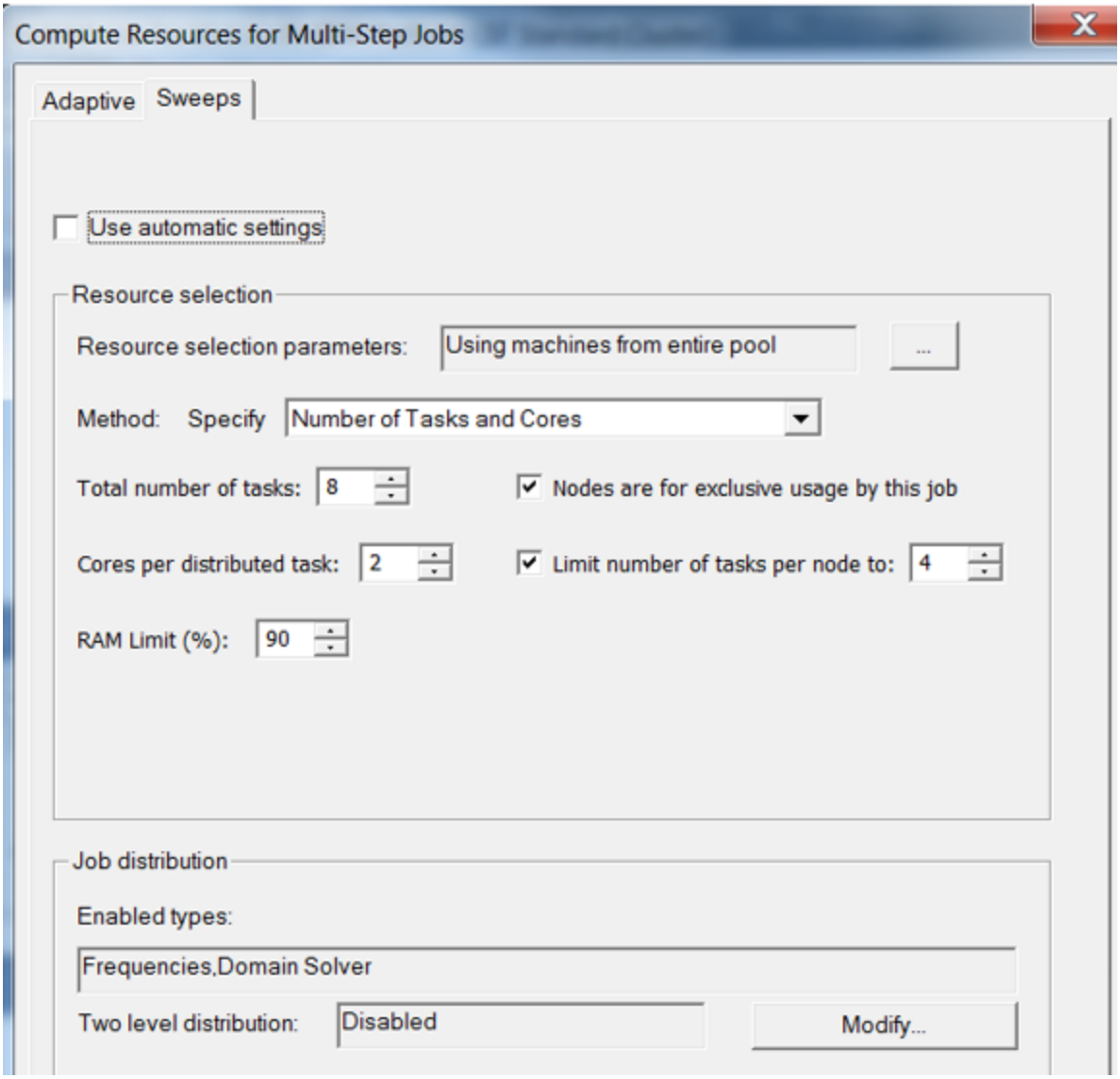
On the **Compute Resources** tab, check the **Use multi-step submission** box and select the appropriate submission option:

- 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.

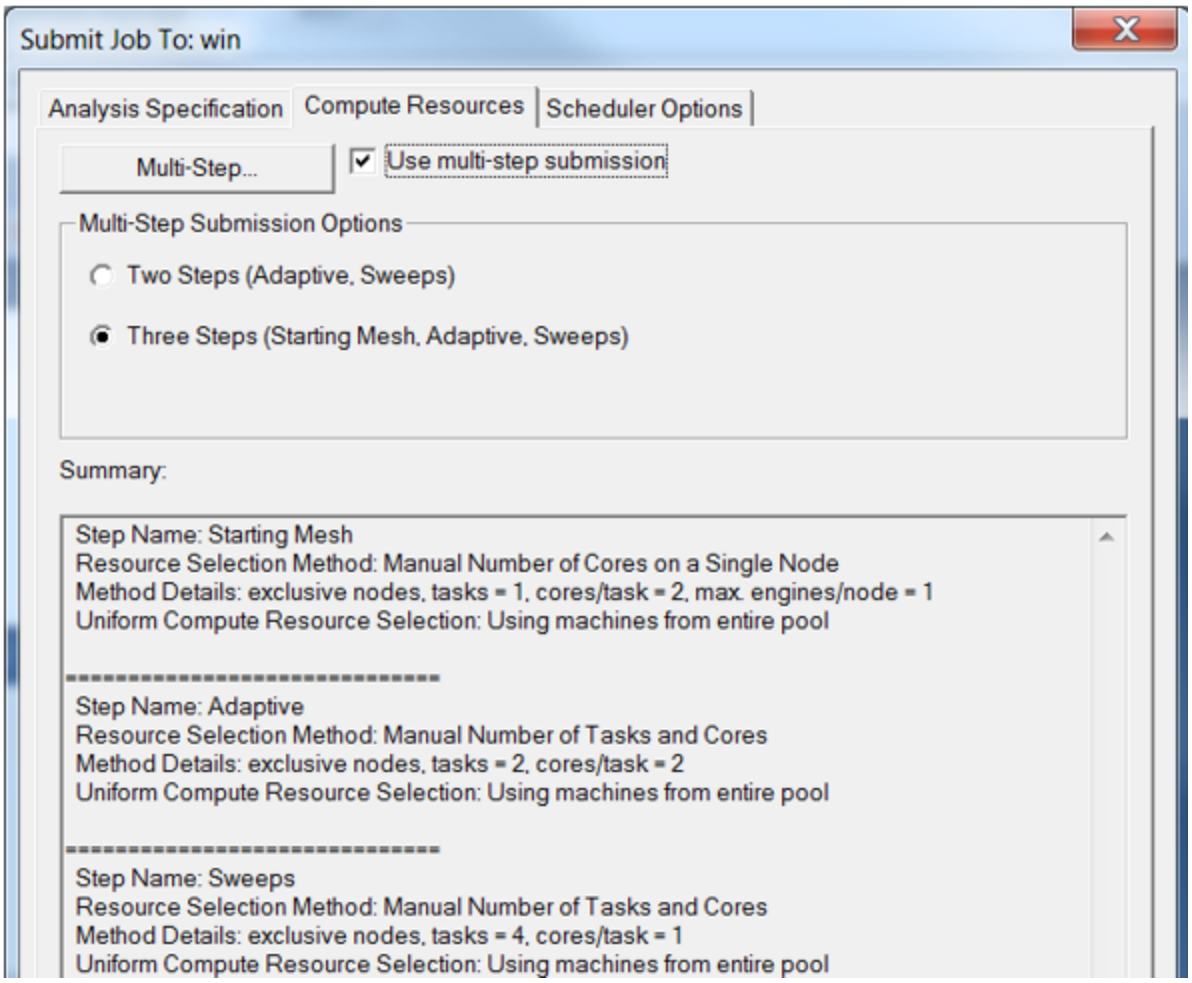


Notice, for example, that the **Sweeps** tab lists the same resources choices but they can be assigned differently.

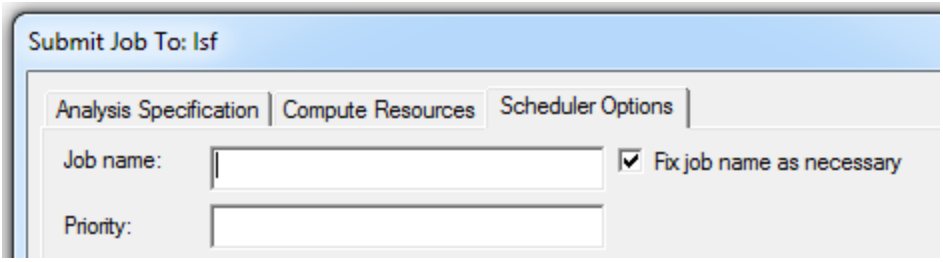


Set the desired compute resources for each step and click **OK**.

The Summary field of the **Compute Resources** tab shows a text summary of resource specifications used for each step.

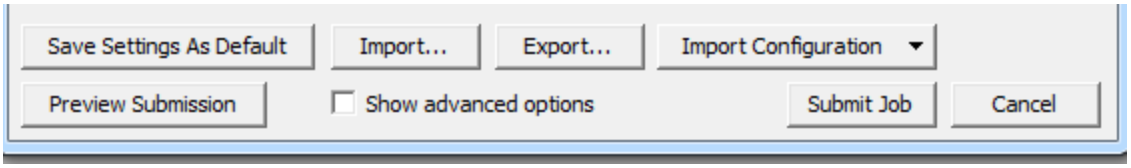


If desired, set the job name on the **Scheduler Options** tab. When the jobs are submitted, each will have a name given by job name (if any) appended with the name of the step for the job. For example, if the job name entered is “MultiStep”, then the individual jobs might be named “MultiStep-Adaptive” 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 window 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`’ Univa, 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.

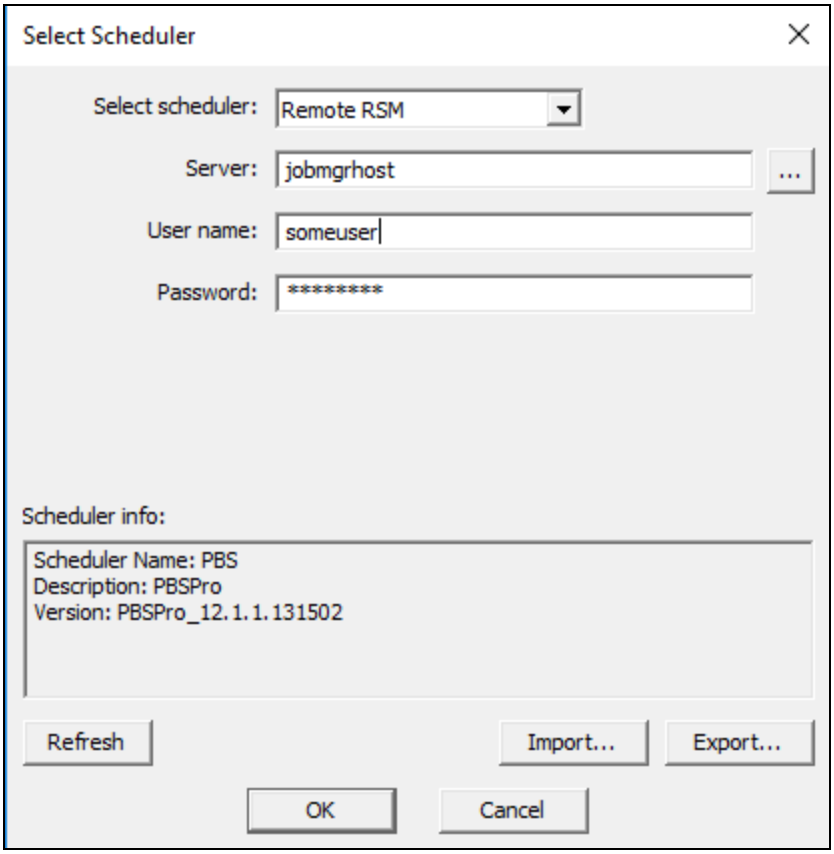
Supported Schedulers

This feature may be used with all Linux schedulers for which job submission from the GUI is supported:

- IBM Spectrum LSF
- Univa Grid Engine (GE)
- PBSPro
- Torque

Select Scheduler Dialog

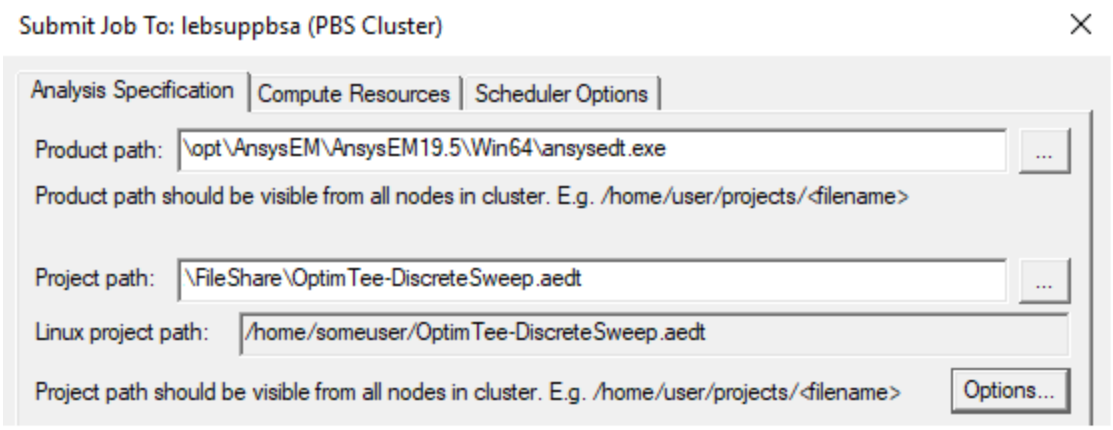
If you select the **Use a computer on the network** option, you can enter a username and password. This username and password are used when the job is submitted to the Linux scheduler.



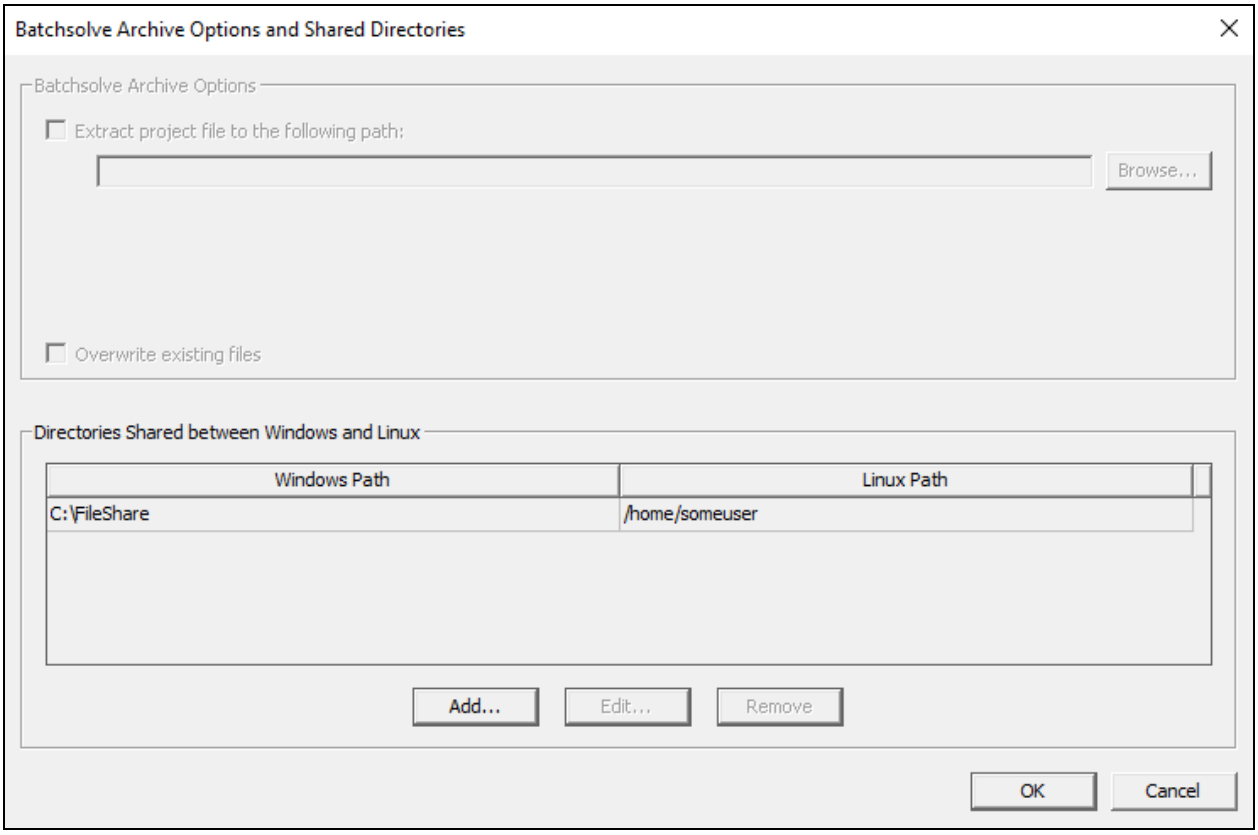
Submit Job

In the ***Submit Job To*** window, 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.

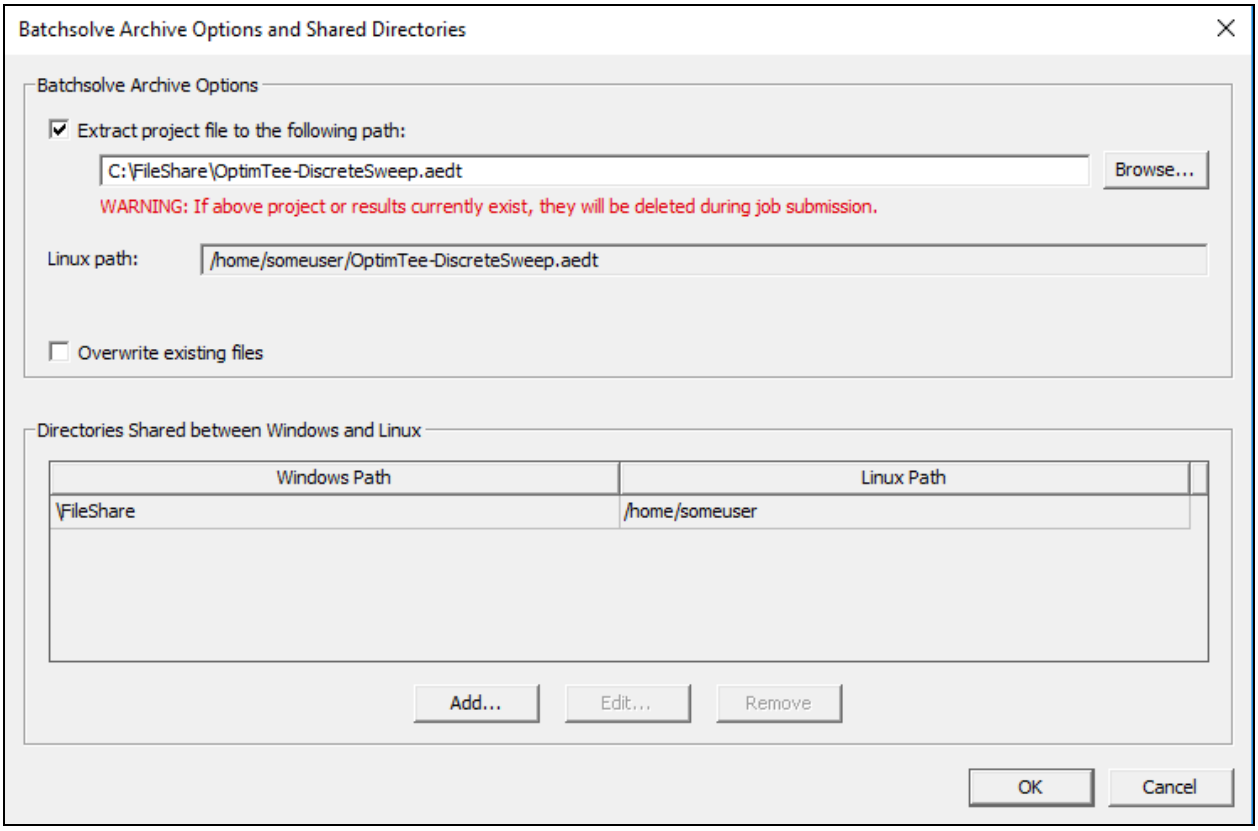
You must enter the Windows path of the project file in the Product path edit control or use the browse button (labeled "...") to select the Windows path of the project file. If the Linux path of the project file can be determined from the specified Windows path and the directories shared between Windows 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.



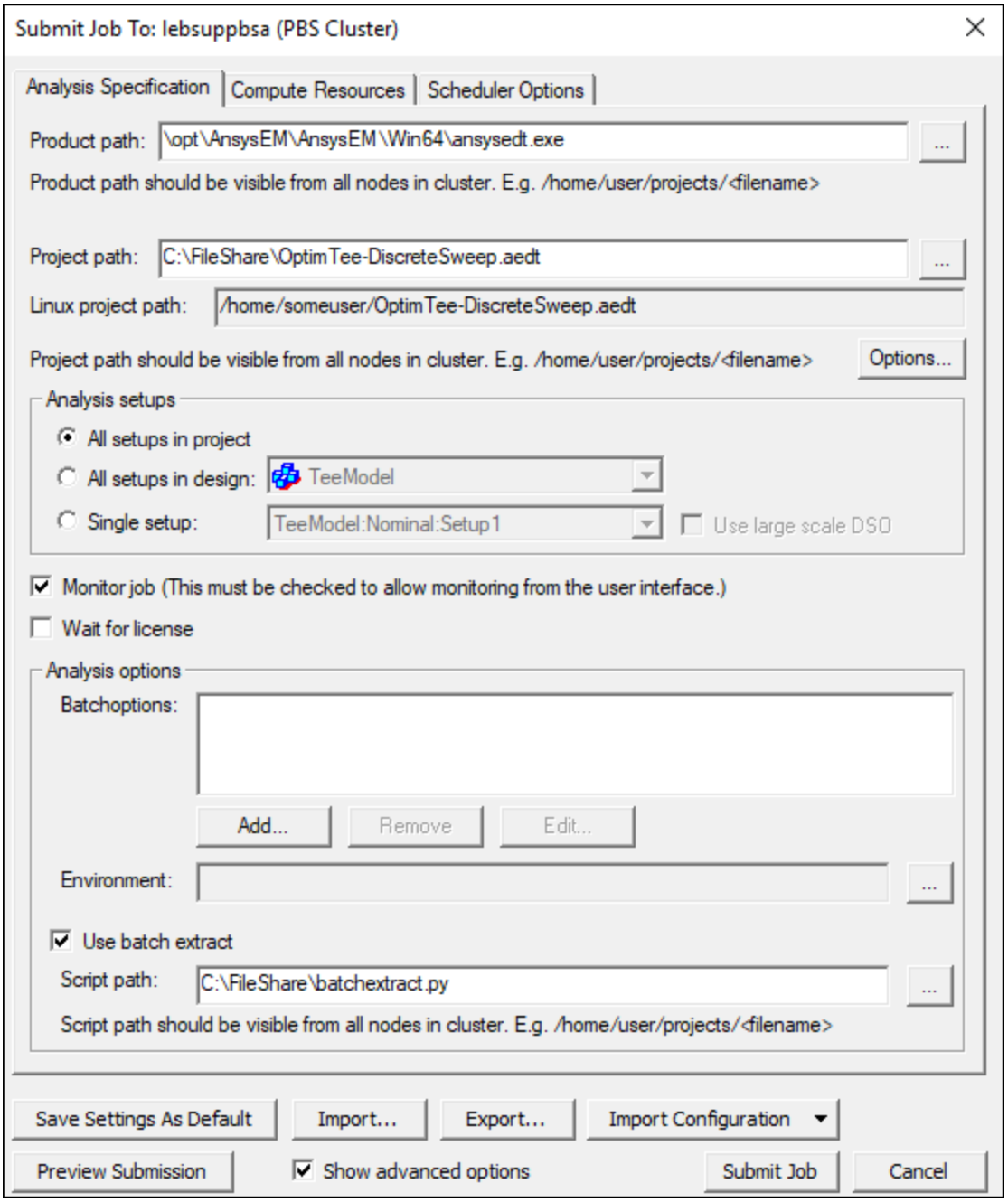
The **Options** button opens a window that you can use to specify archive options for a job. The lower portion of this window allows you to specify one or more directories shared between Windows and Linux. The mapping of directories between Windows and Linux is shown in a grid which displays the Windows 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 window 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 window.



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** window. 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.

Windows® HPC Non-Exclusive Jobs

Windows HPC allows three choices for the Cluster AffinityType setting:

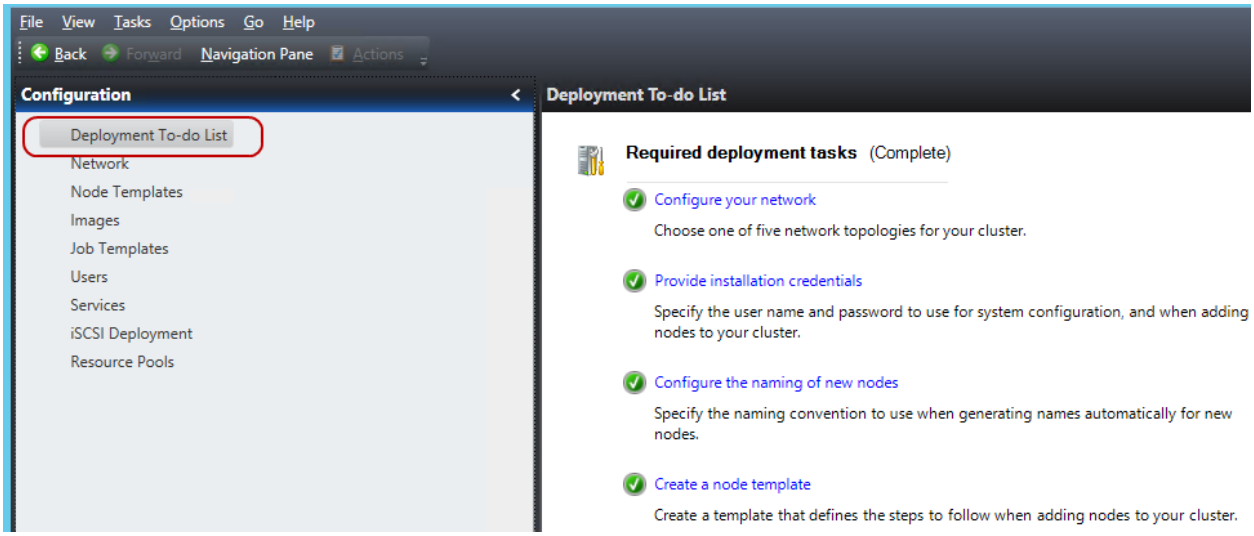
- No Jobs: Affinity is not set for any jobs
- Non Exclusive Jobs (default): Affinity is not set for exclusive jobs
- All Jobs: Affinity is set for all jobs

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.

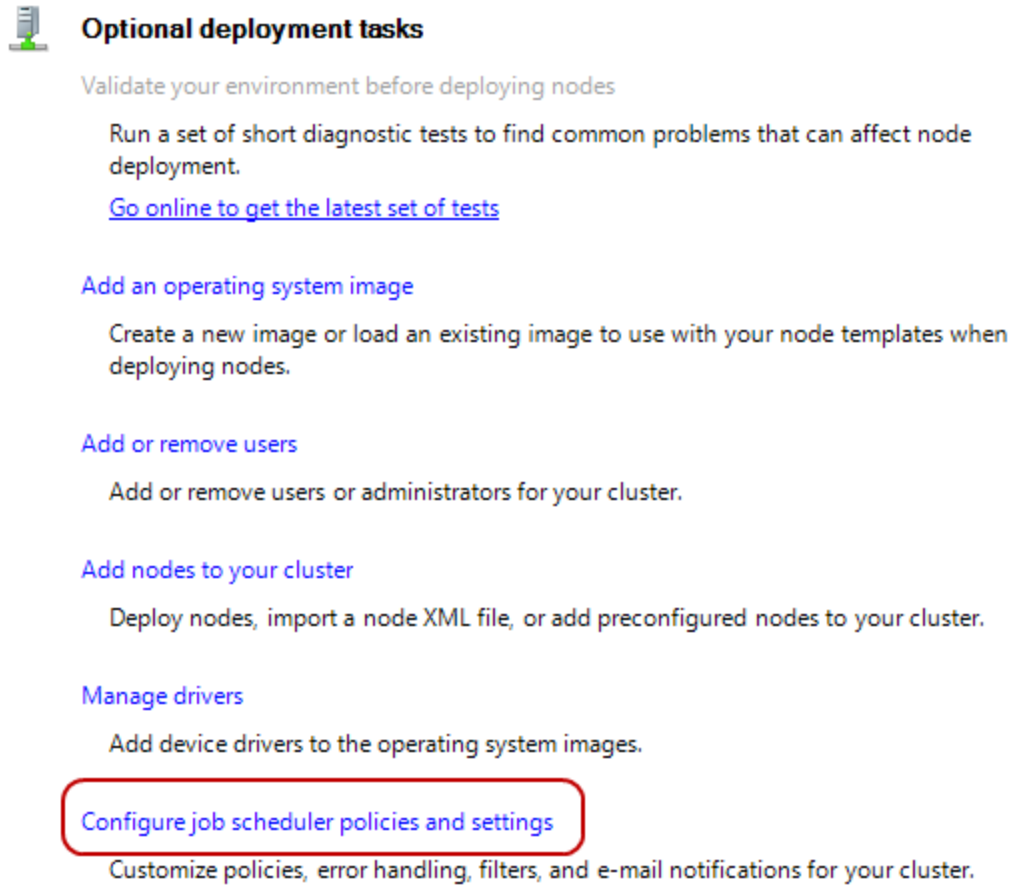
The previous job submission dialog always has the “exclusive” check box disabled for Windows HPC jobs, so that “exclusive” jobs cannot be submitted to a Windows HPC cluster. The “exclusive” check box is now enabled for Windows HPC jobs. This allows you to submit exclusive jobs to a Windows HPC cluster. If this check box 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

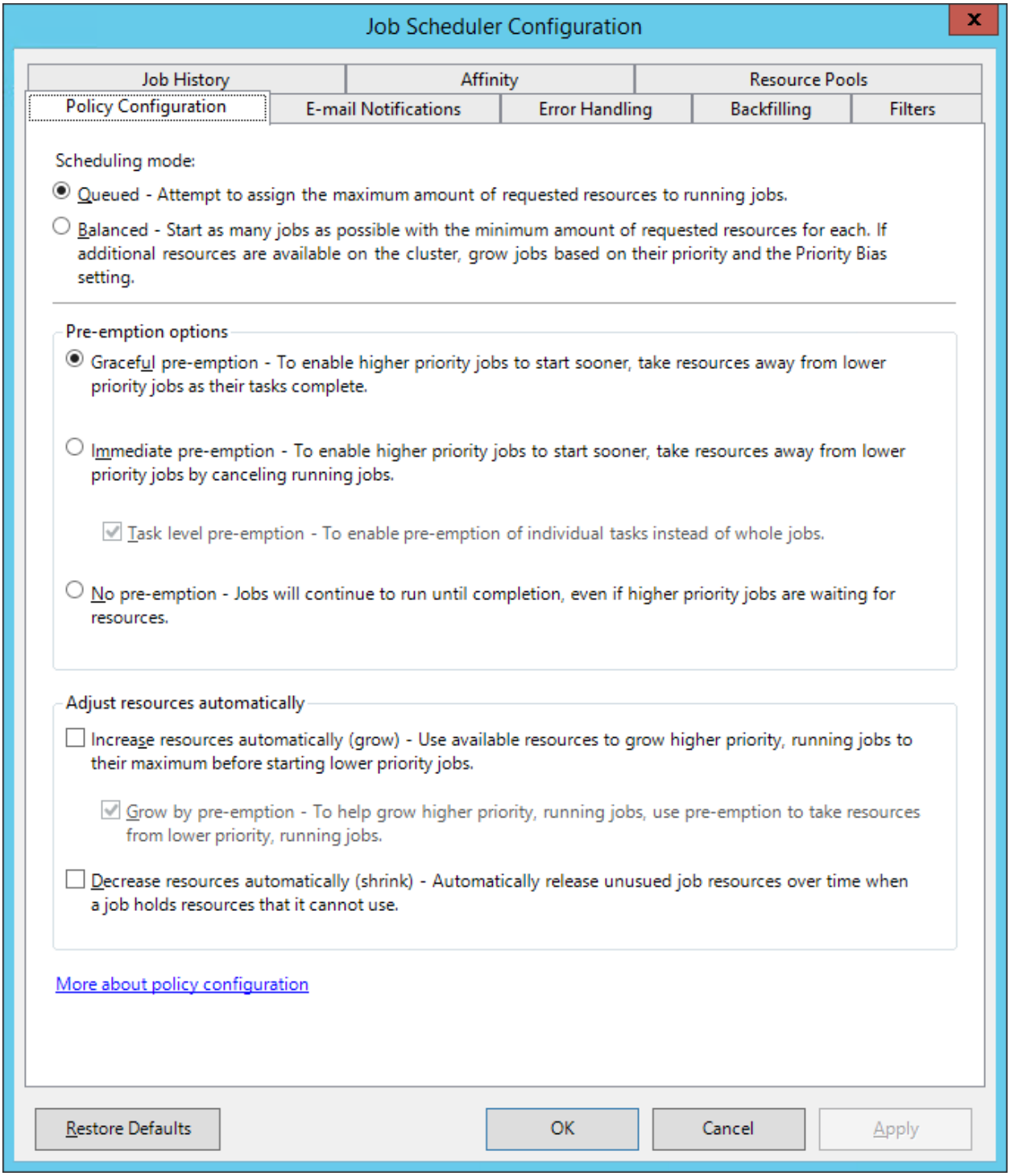
A Windows HPC Cluster administrator is able to modify the Affinity Type setting for the cluster. The administrator should start the Windows HPC Cluster Manager. The first step is to select the Configuration button on the lower left pane, then select the Deployment To-do List in the upper left pane. Then, the upper right pane displays the Deployment To-do List, as shown below.



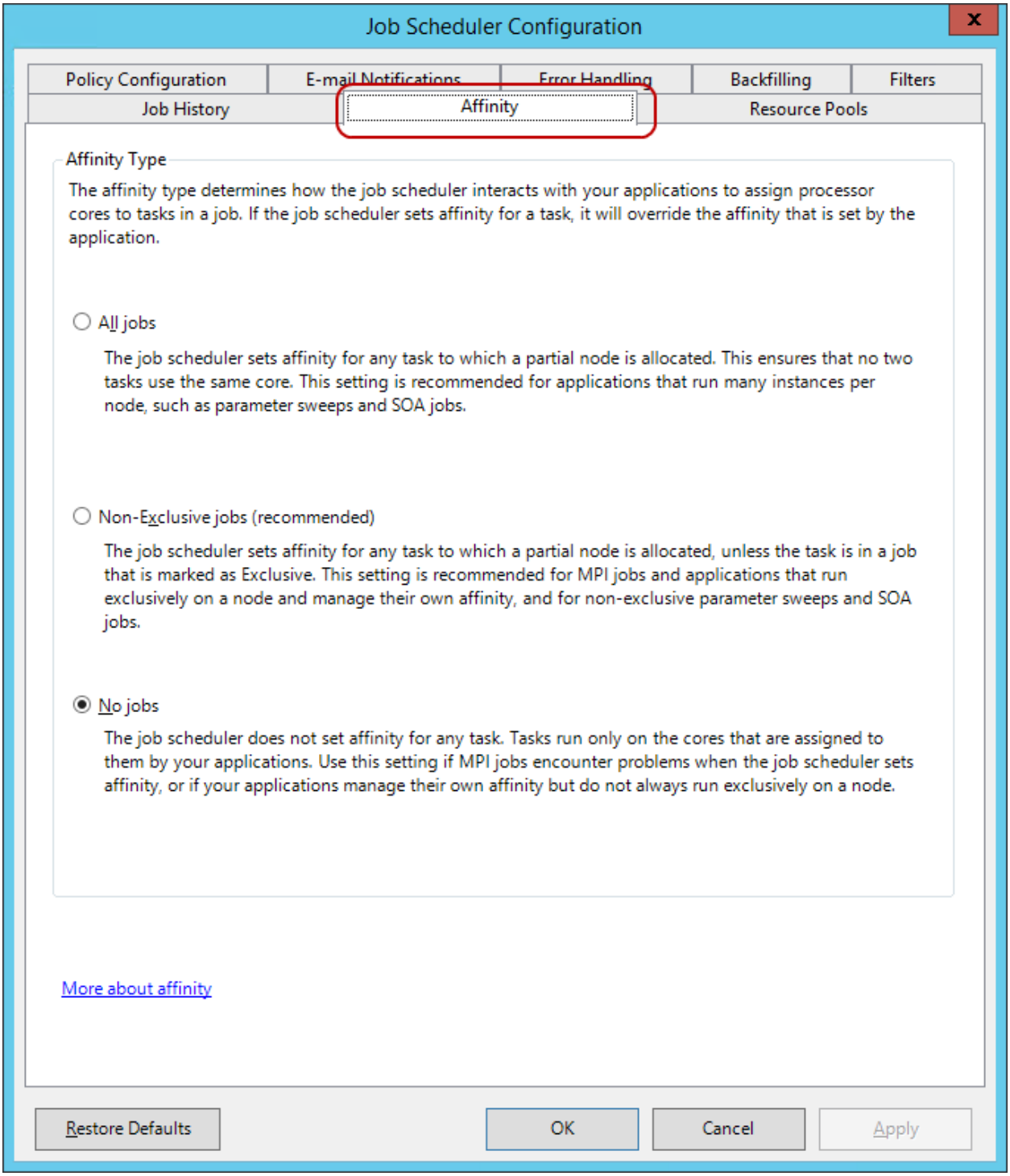
Next, the "Configure job scheduler policies and settings" should be selected. This is under the heading "Optional deployment tasks."



The *Job Scheduler Configuration* dialog box 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.

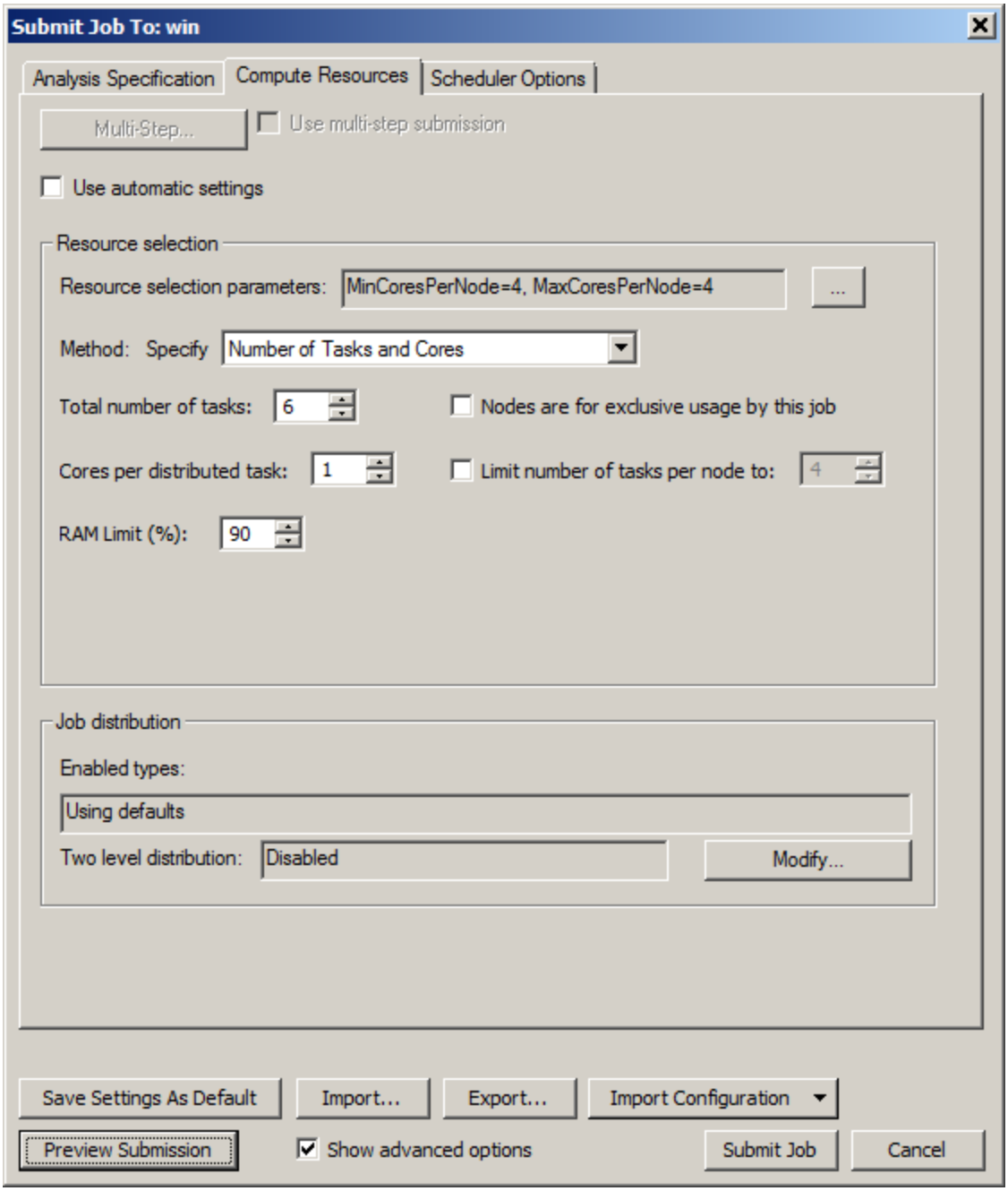
Examples with "Submissions with Nodes are for exclusive usage by this job" Unchecked and Checked

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:

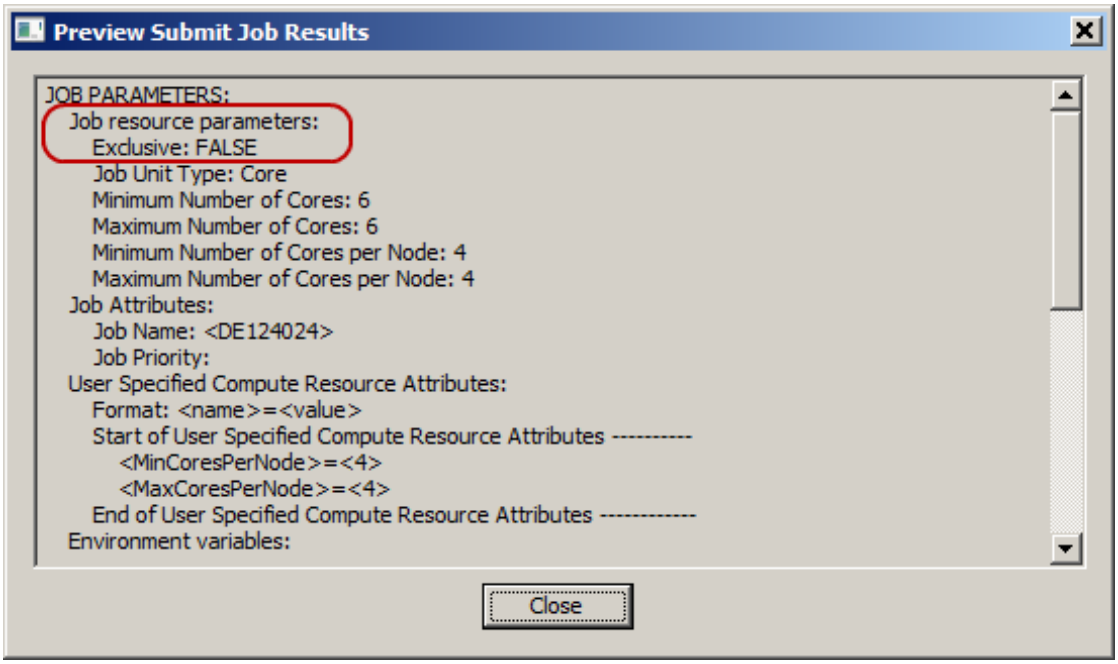
- MinCoresPerNode=4
- MaxCoresPerNode=4

The first set of screen shots show submission of a job to a Windows HPC cluster in which the "Use automatic settings" check box 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" check box. In previous releases, this check box 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.



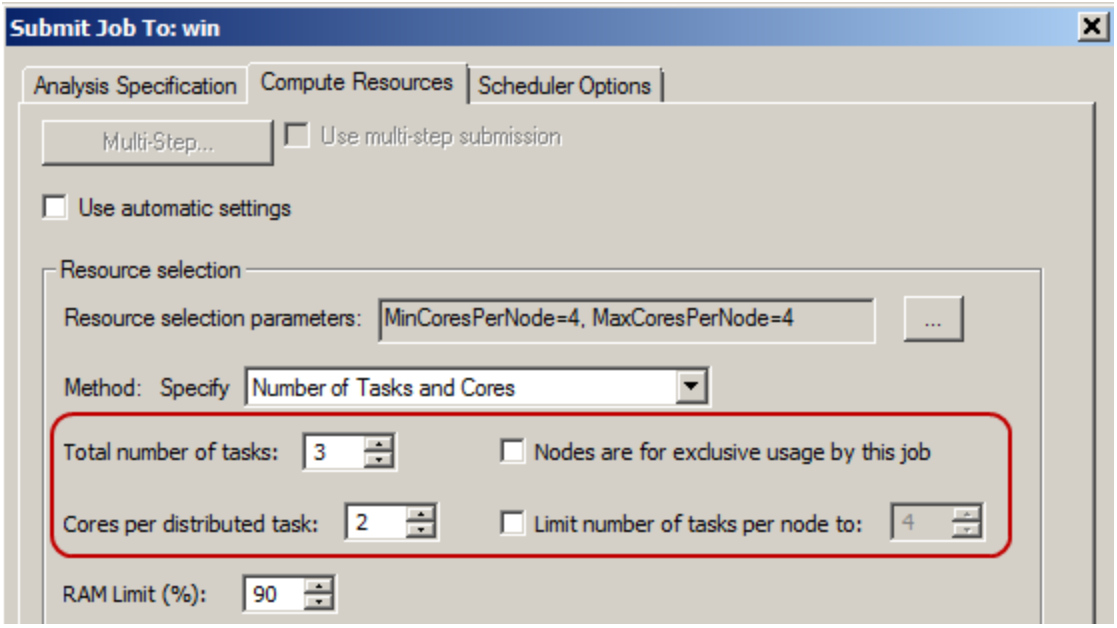
The next figure shows Preview for the job submission indicating that the exclusive parameter for the job is false.



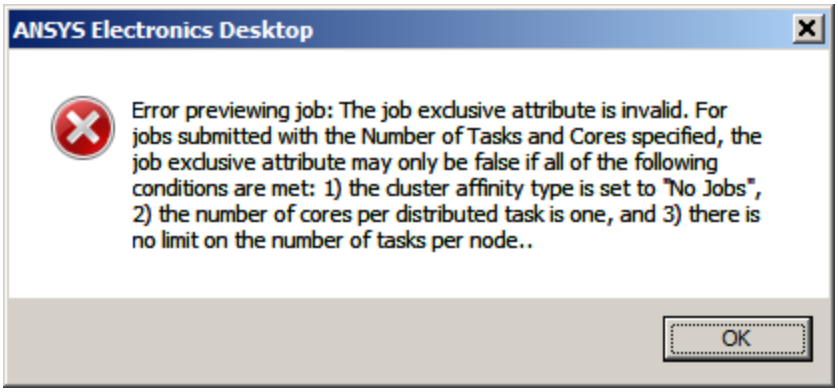
Jobs may only be submitted using this resource selection method with the exclusive parameter set to false if all three of the following conditions are met:

1. The cluster AffinityType is "No Jobs",
2. There is no multiprocessing, 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.

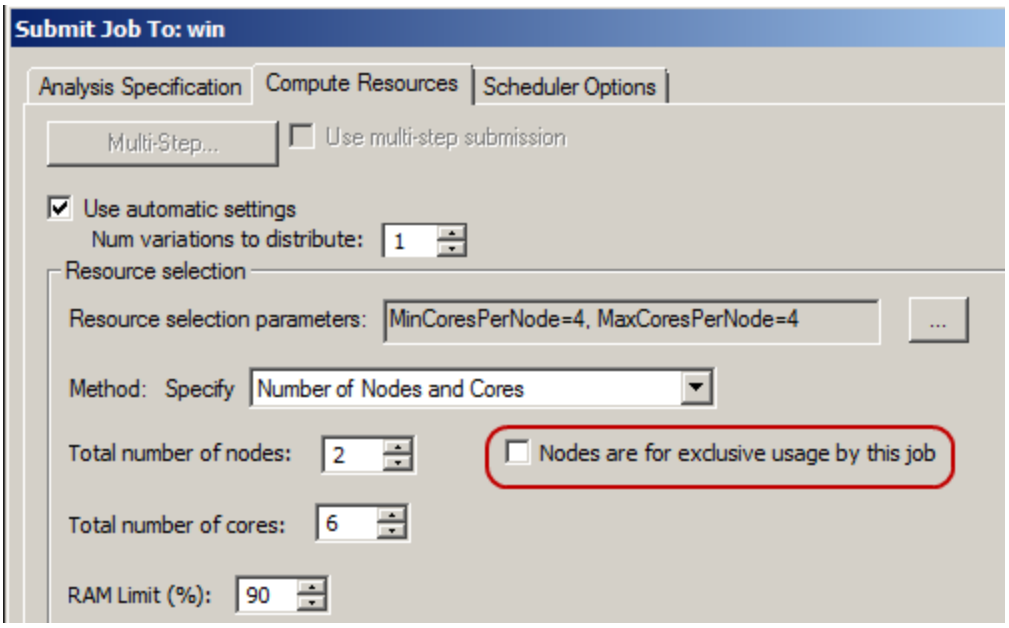


The following figure and screen shot ErrorMsgTasksAndCoresNonExMP.png shows the error message box shown to the user.

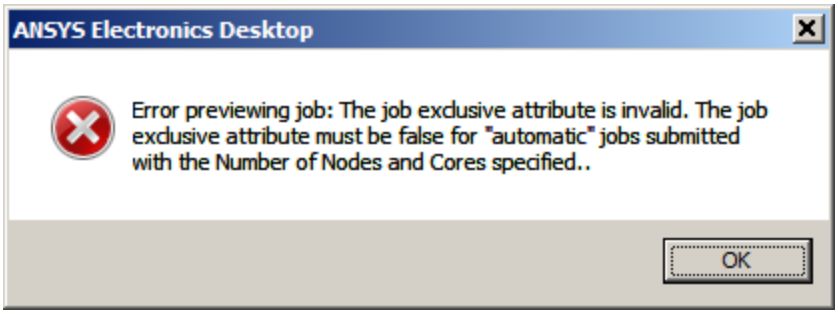


The next example shows submission of a job to a Windows HPC cluster in which the "Use automatic settings" check box is checked, and the resource selection method is "Specify Number of Nodes and Cores". If you select *Use automatic settings* with *Num variations to distribute* set to 1, Optimetrics variations 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. The exclusive setting for the job is controlled by the "Nodes are for exclusive usage by this job" check box. In previous releases, this check box 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

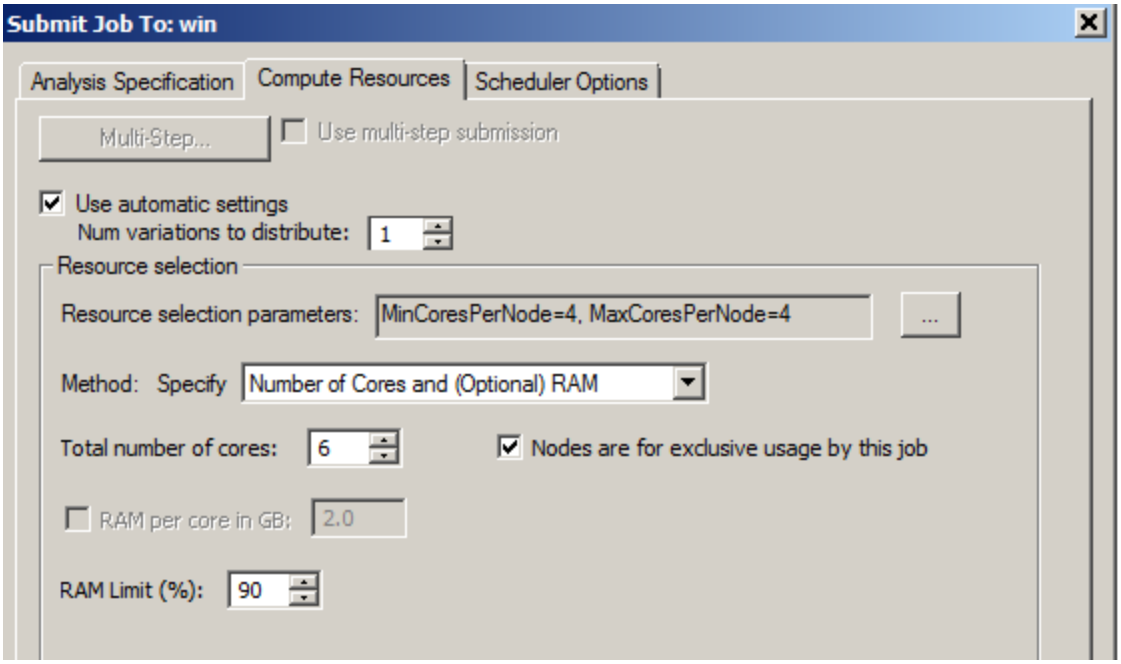
behavior is the same as previous releases. If it is unchecked, then the job is not submitted or previewed because only exclusive jobs are allowed when using this resource selection method, and an error message box is shown to the user in this case. The following figure shows this configuration:



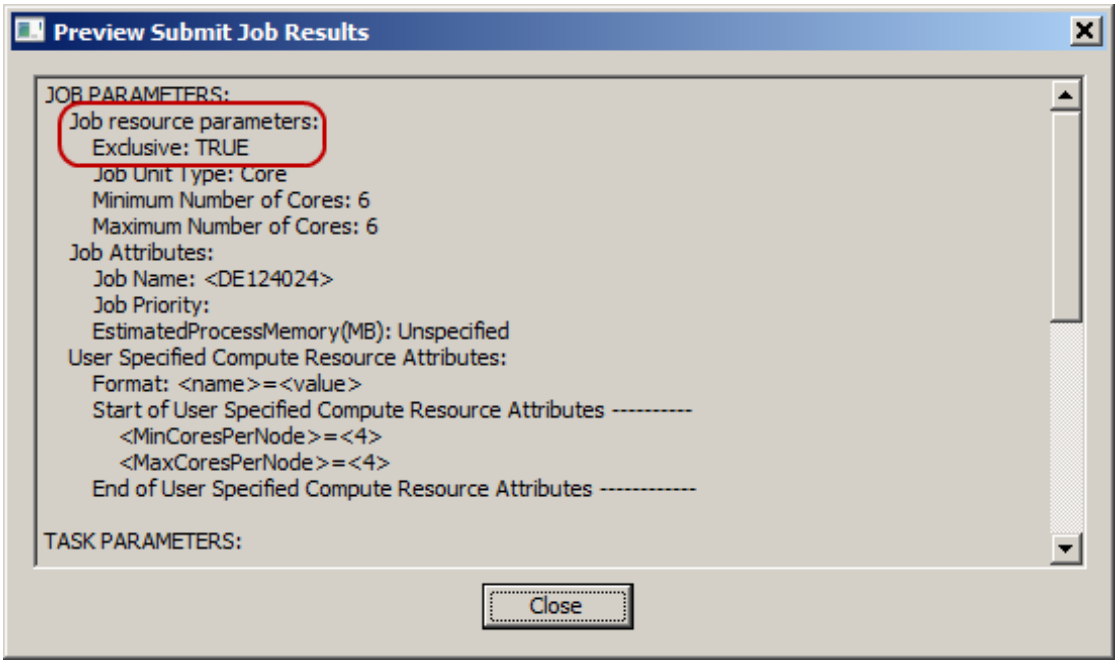
The following figure shows the error message.



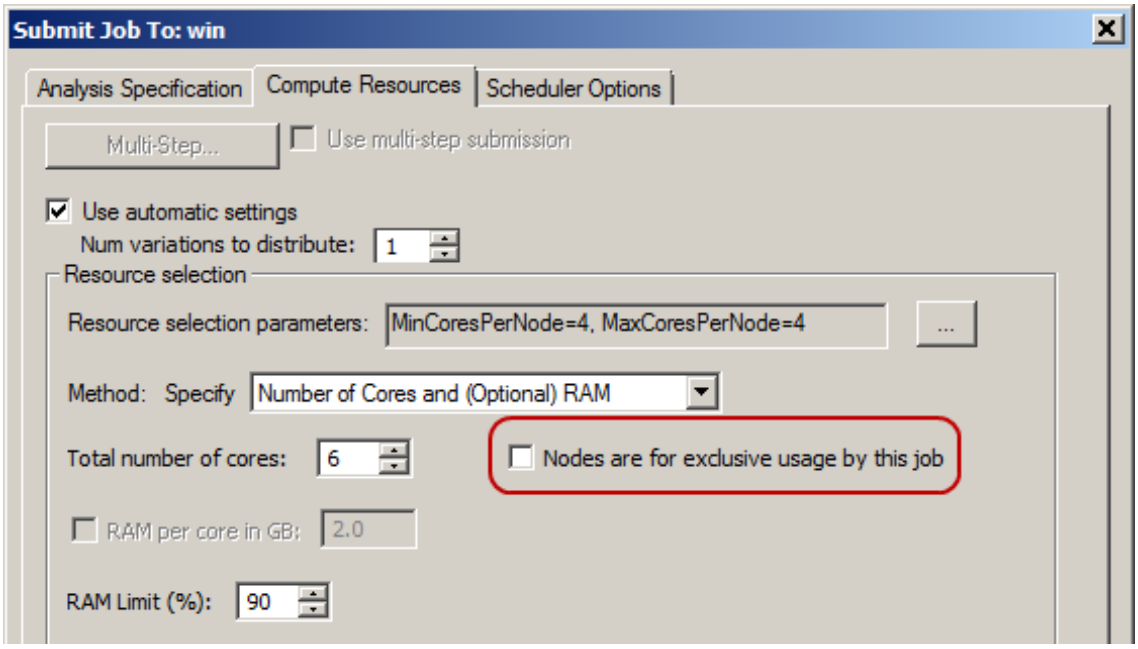
The next example show submission of a job to a Windows HPC cluster in which the "Use automatic settings" check box is checked, and the resource selection method is "Specify Number of Cores and (Optional) RAM". The exclusive setting for the job is controlled by the "Nodes are for exclusive usage by this job" check box. In previous releases, this check box 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.



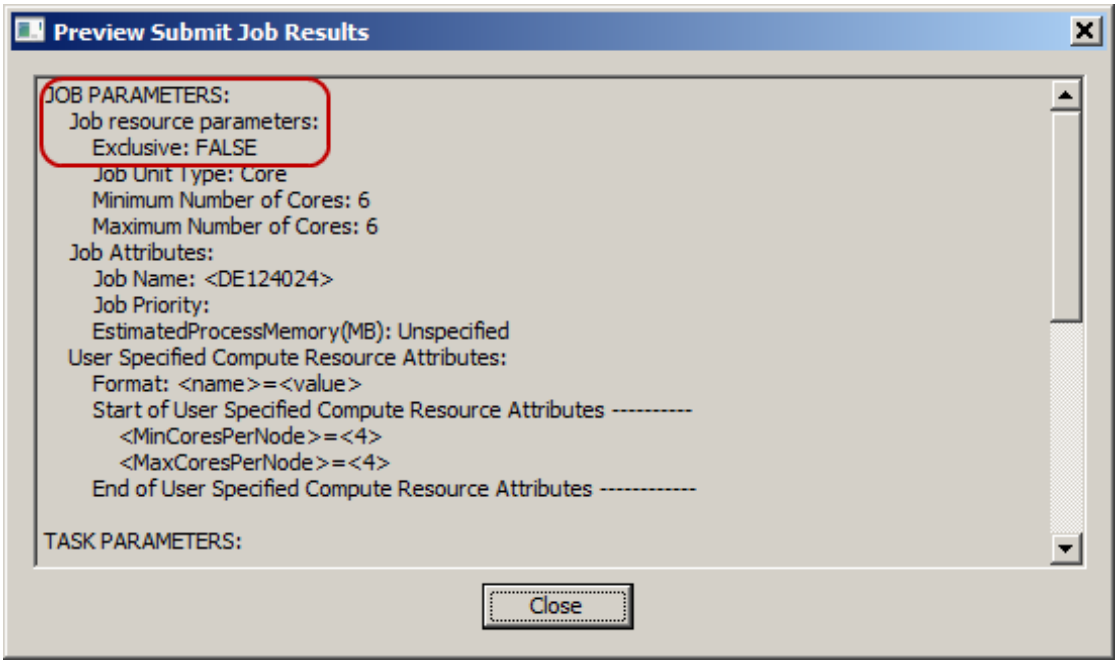
The following figure shows the Job Preview for this case, indicating that the exclusive parameter for the job is true.



If the exclusive check box is unchecked, then the job is submitted with the exclusive parameter set to false. The following figure shows this configuration.

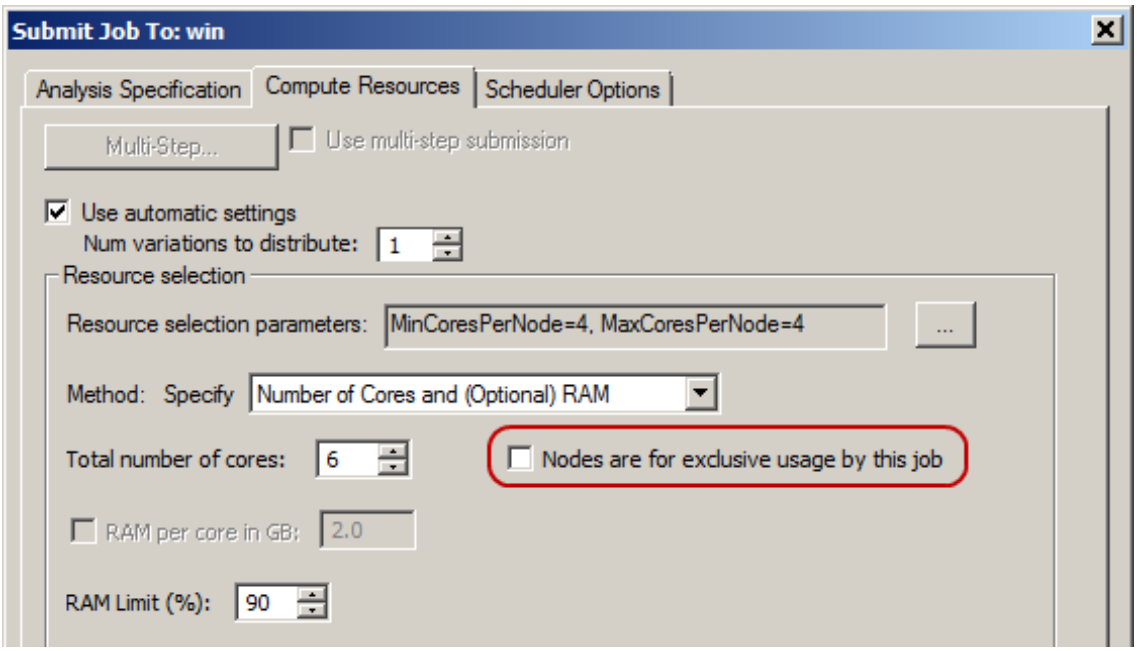


The following figurescreen shot PreviewCoresAndRAMNonEx.png shows the Job Preview for this case, indicating that the exclusive parameter for the job is false.

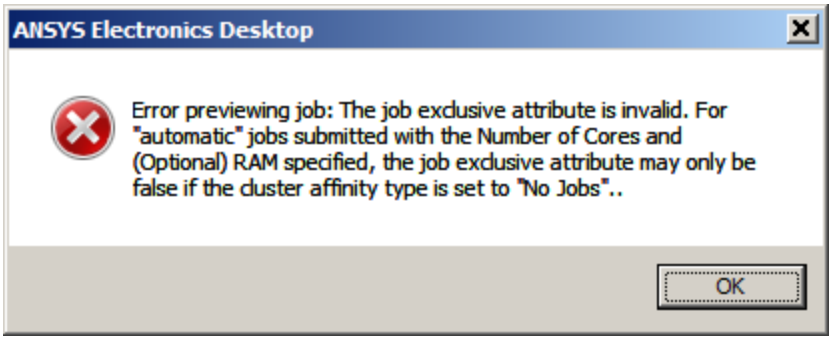


The last example resembles the previous one, except that the Windows HPC cluster AffinityType is set to "Non Exclusive Jobs" in this case. The job is submitted with the "Use

automatic settings" check box 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 check box 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.



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** window to an .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* window appears. If Ansys Cloud Direct or Ansys Cloud Burst Compute are selected then the "Login" button will open the default browser and prompt for an AnsysID login. 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 job submission window. 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 resubmitted 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 Perform 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** window to open the **Submit Job To** window.
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 because 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** > **Q3D ExtractorScripting Help**) for details on the **SelectScheduler**, **SubmitJob**, **LaunchJobMonitor** and **RefreshJobMonitor** commands.

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 Direct Jobs
- Monitoring Ansys Cloud Burst Jobs
- Monitoring Large Scale DSO Jobs
- Web Client for Batch Solve Monitoring and Reporting

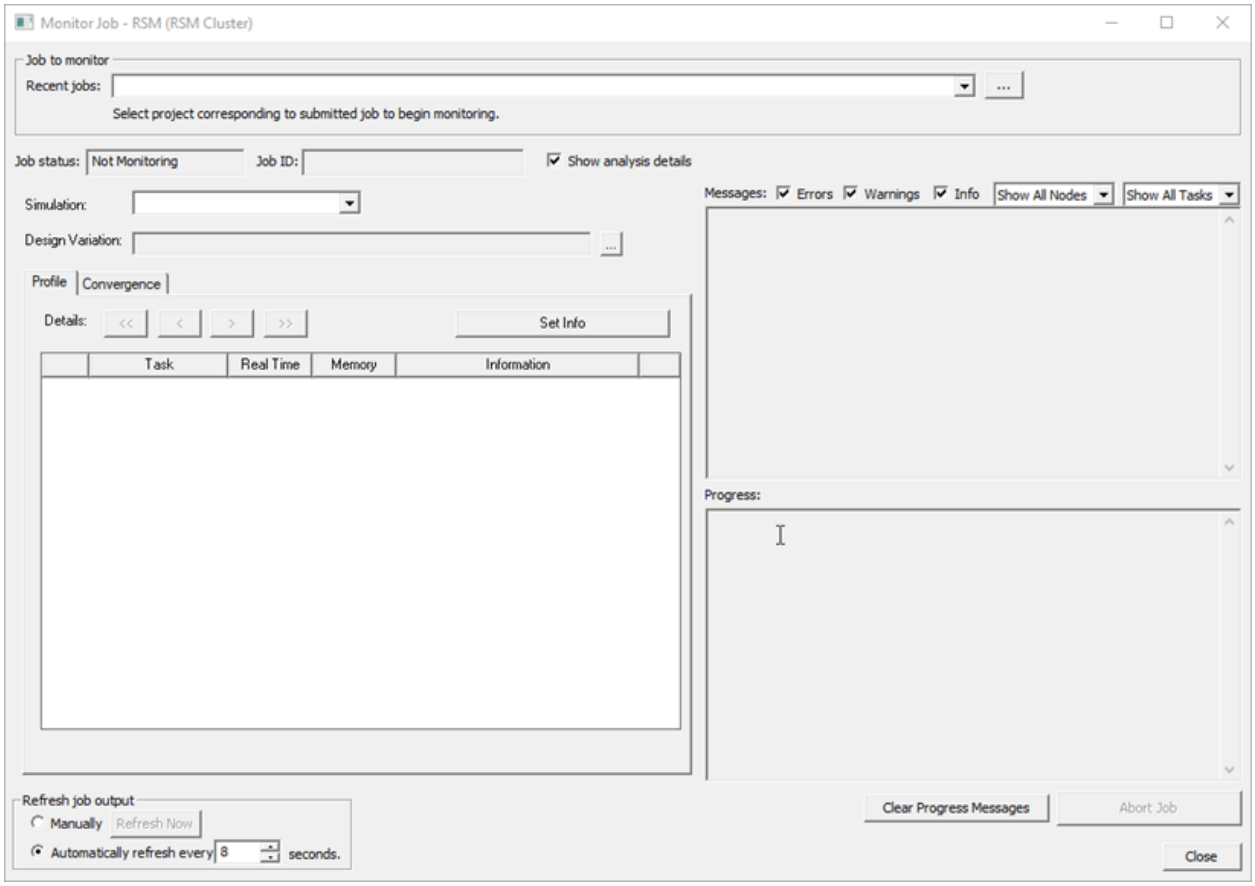
Monitor Job Window

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. Access the Monitor Job window one of the following ways:
   - Click **Tools** > **Job Management** > **Monitor Jobs**.
   - Select the **Simulation** tab and click the **Monitor** icon.
   - 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 and design type.

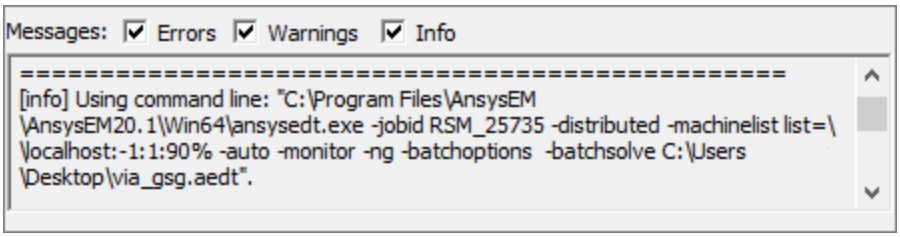
The **Monitor Job** window contains the following areas:

- **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.

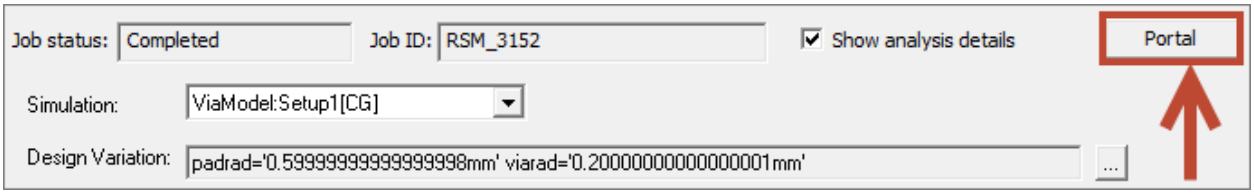
To display less information, deselect **Show analysis details**. This removes the **Simulation**, **Design Variation**, **Profile**, and **Convergence** information, instead displaying only **Messages** and **Progress**.

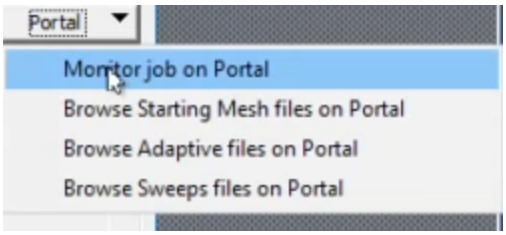
2. Click **Close** when you are done monitoring.

Monitoring Ansys Cloud Direct Jobs

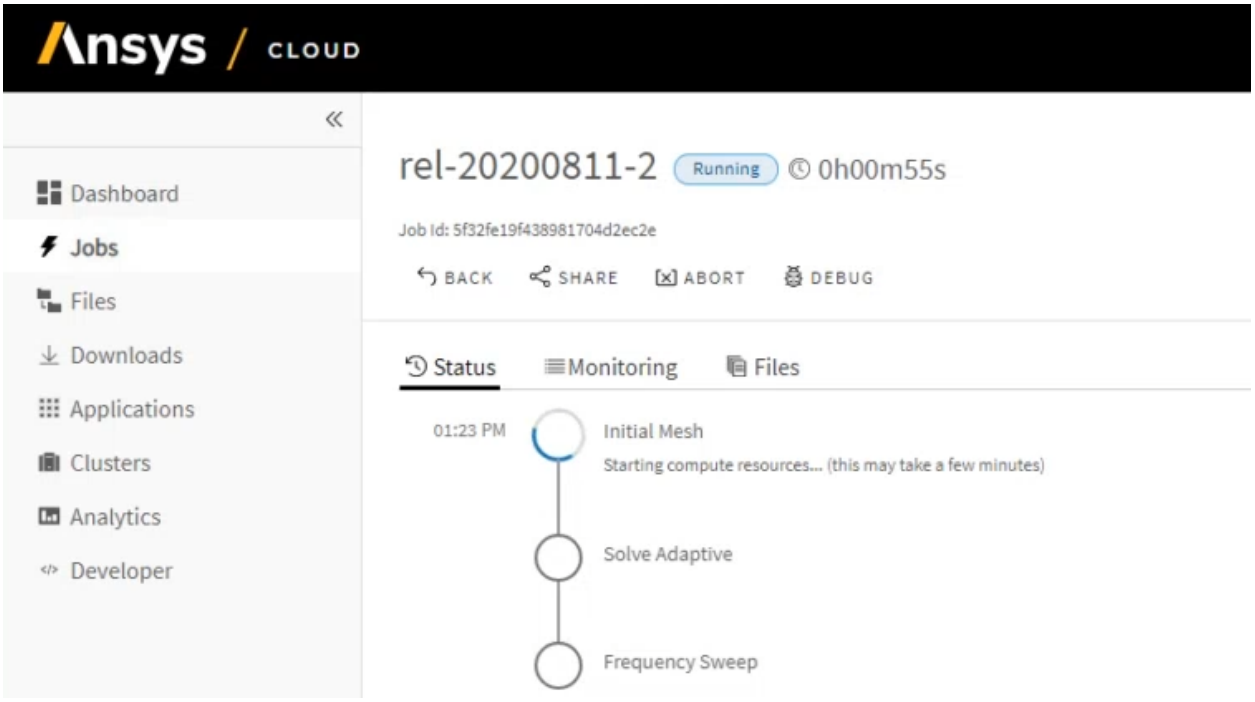
For Ansys Cloud Direct submissions, the **Monitor Job** window includes two additional items:

- **Portal** – link to the Ansys Cloud Direct Portal, where job details and additional monitoring information are available.

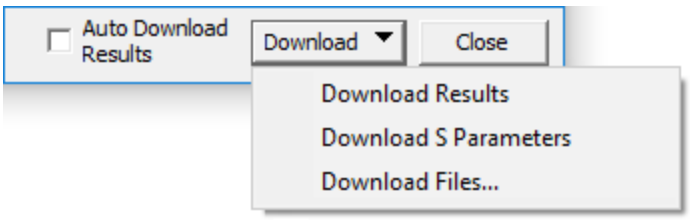




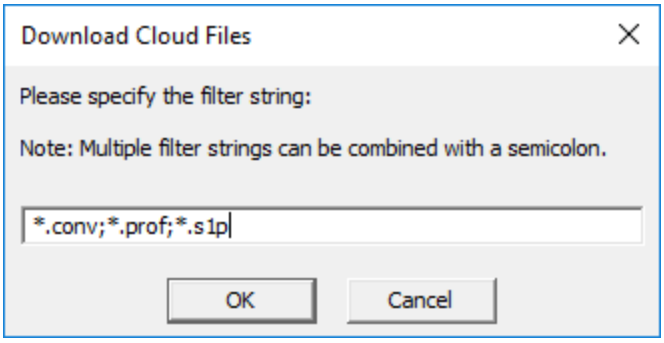
Monitor job on Portal Opens a Portal Web interface that lets you monitor an Ansys Cloud Direct job. The additional menu options for **Portal** are for multi-step submissions, opening the Portal Web interface directly to folders with Starting Mesh Files, Adaptive Files, or Sweeps files.



- **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.



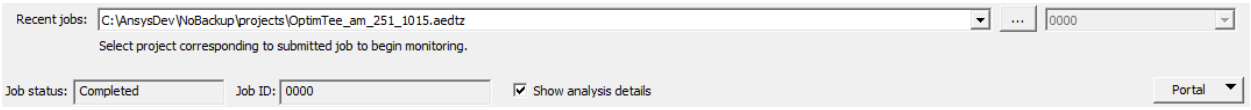
If you choose to **Download Files**, the **Download Cloud Files** dialog box appears:



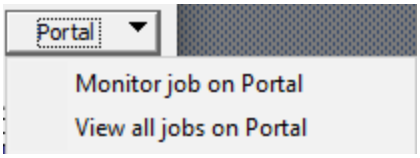
Specify the types of files you want to download (separated by semicolons) and click **OK**.

Monitoring Ansys Cloud Burst Jobs

When you submit a job to Ansys Cloud, the job status is shown in the **Monitor Jobs** window. The Job ID shown here is a short numeric identifier used to distinguish multiple jobs for the same project. Use the drop-down list at the upper right to select a different job for the same project. **The Monitor Job** window includes two additional items, **Portal**, and **Download**.

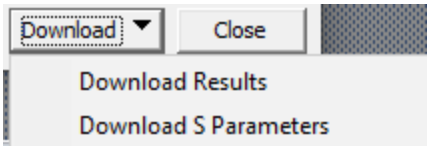


The **Portal** drop-down button provides a link to the Ansys Portal, where job details and additional monitoring information are available. For more information, see the Ansys Portal Guide.



Monitor job on Portal opens a job details view for the current job in Ansys Portal. **View all jobs on Portal** opens a job list view in Ansys Portal, enabling you to select a job to monitor.

Download allows you to download the results or s-parameters. 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.

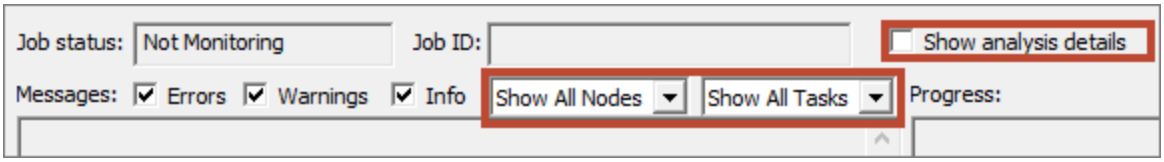


Monitoring Large Scale DSO Jobs

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.

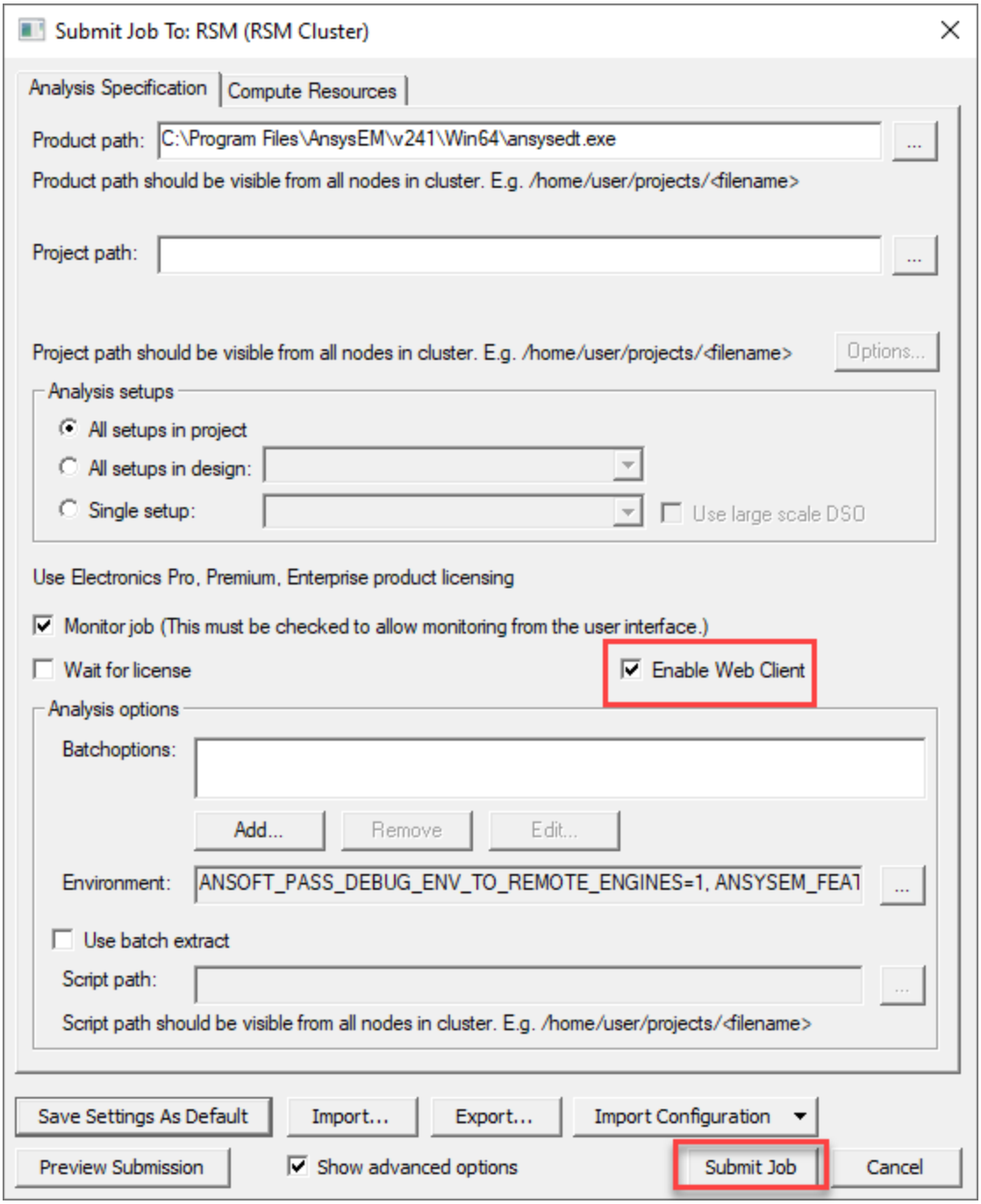


To leave this mode, select **Show analysis details**.

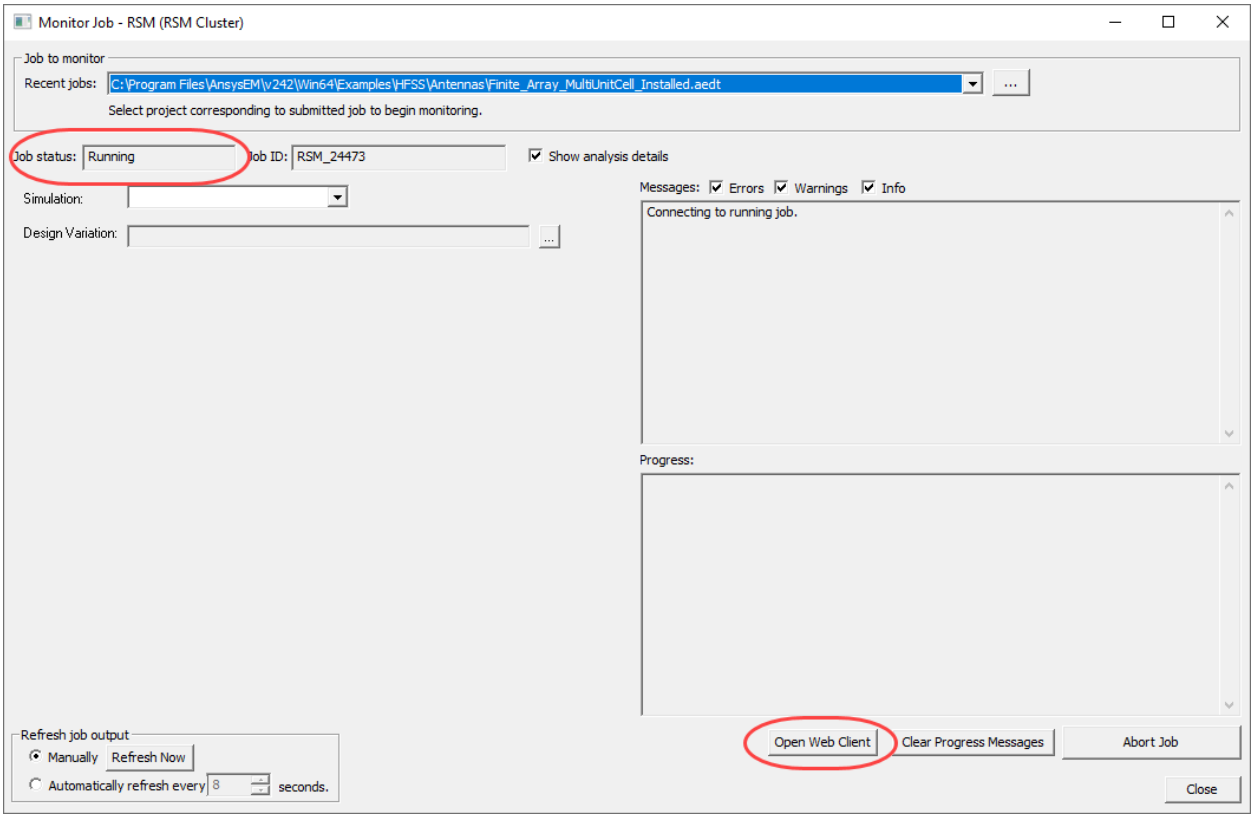
Web Client for Batch Solve Monitoring and Reporting

There is a Beta Option to use a Web client for batch solve monitoring and reporting. If you set the Beta Option f under General for Enable Web Client for Monitoring, you will have access to the feature.

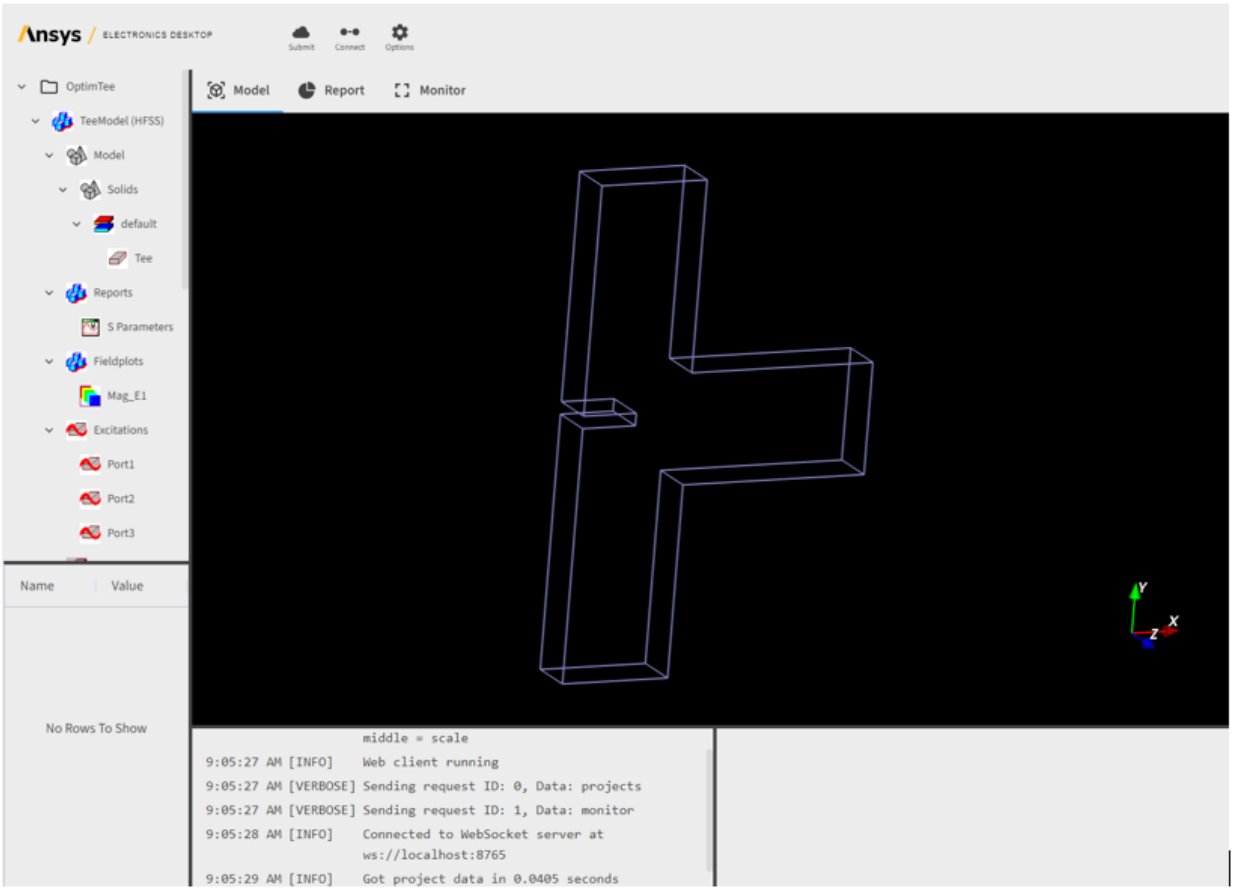
After you have selected the scheduler to use, when you access the **Submit Job To:** dialog, you well see Enable Web Client checked.



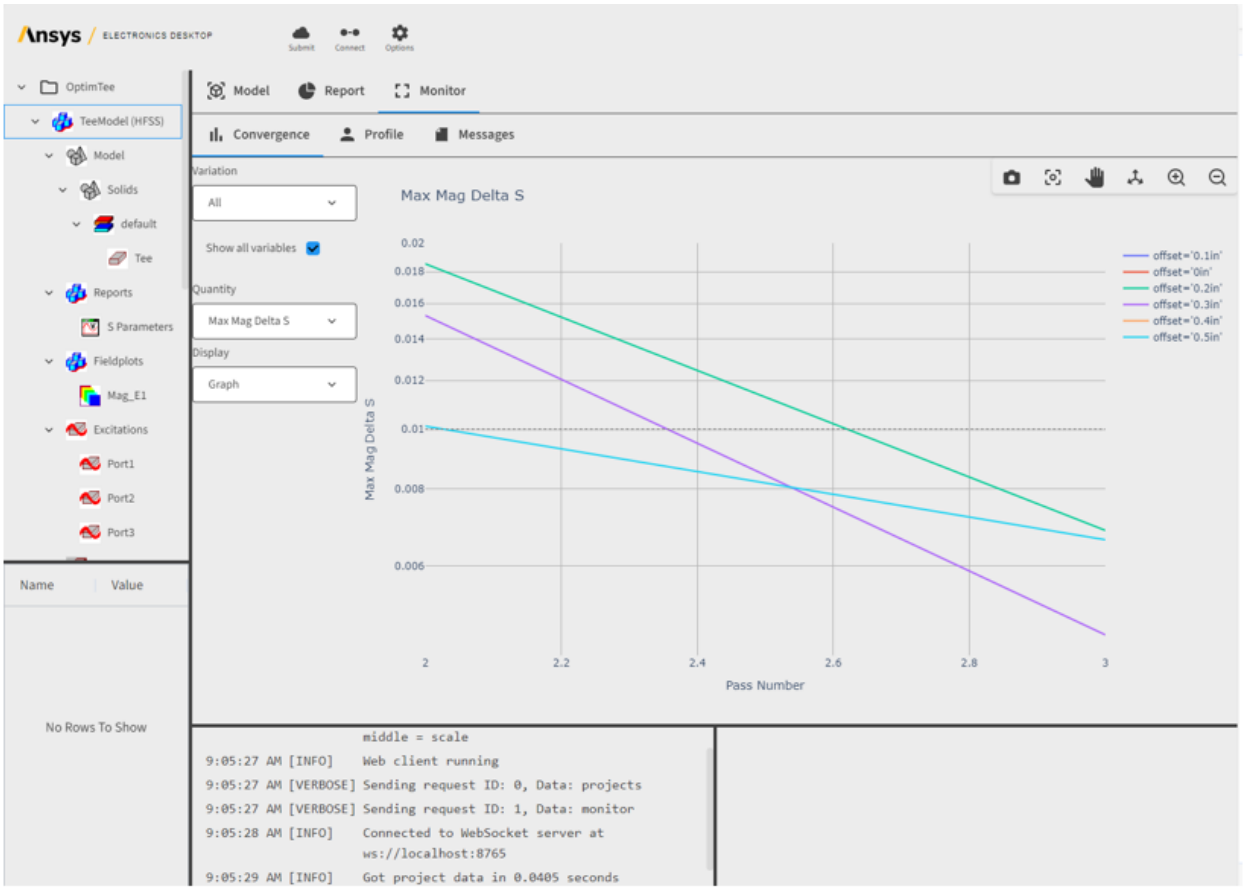
After specifying the project path and setting other parameters, you can then select **Submit Job**. This opens the Monitor Job dialog for your scheduler. Once the **Job Status** changes to Running, press **Open Web Client**.



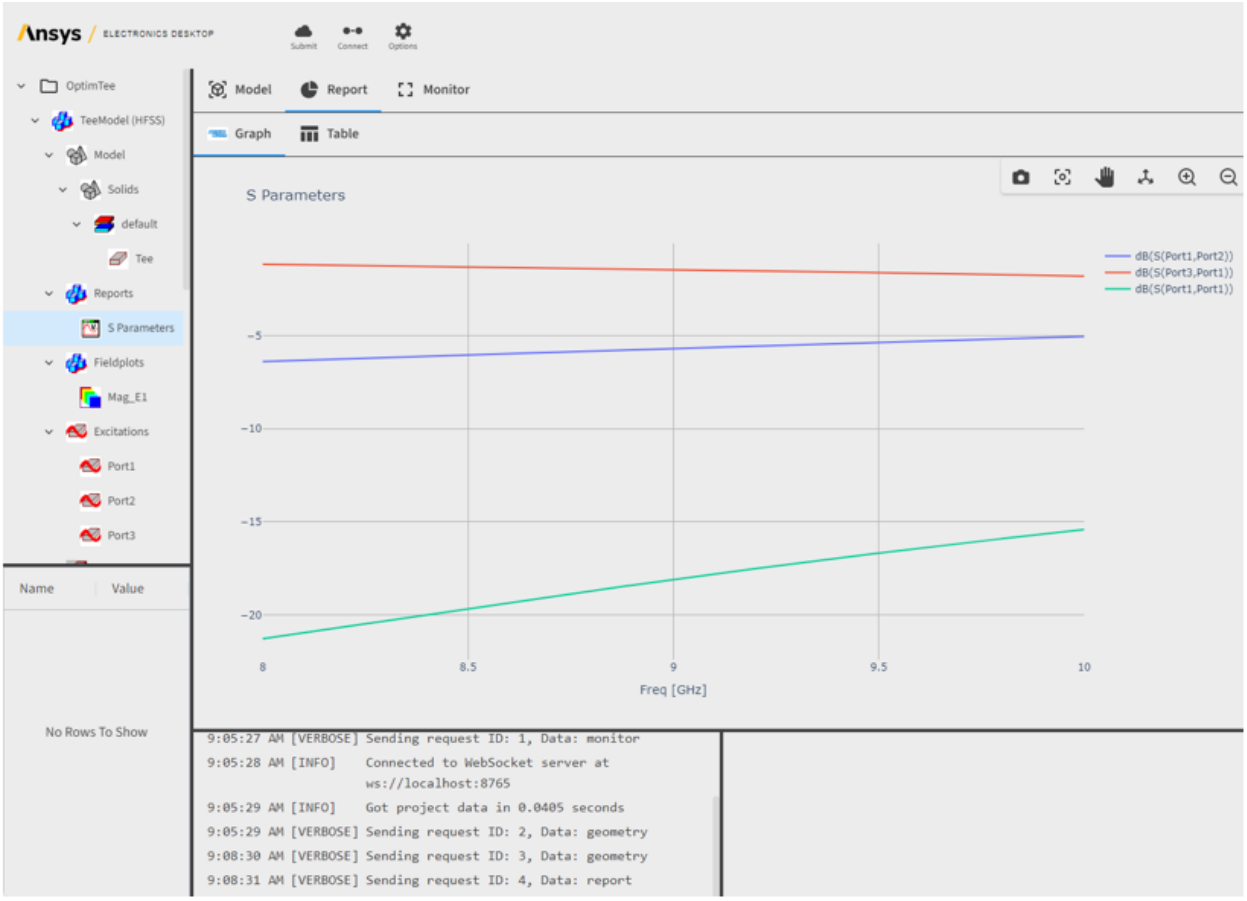
Click **Open Web Client** to open the Web Client in your default browser.



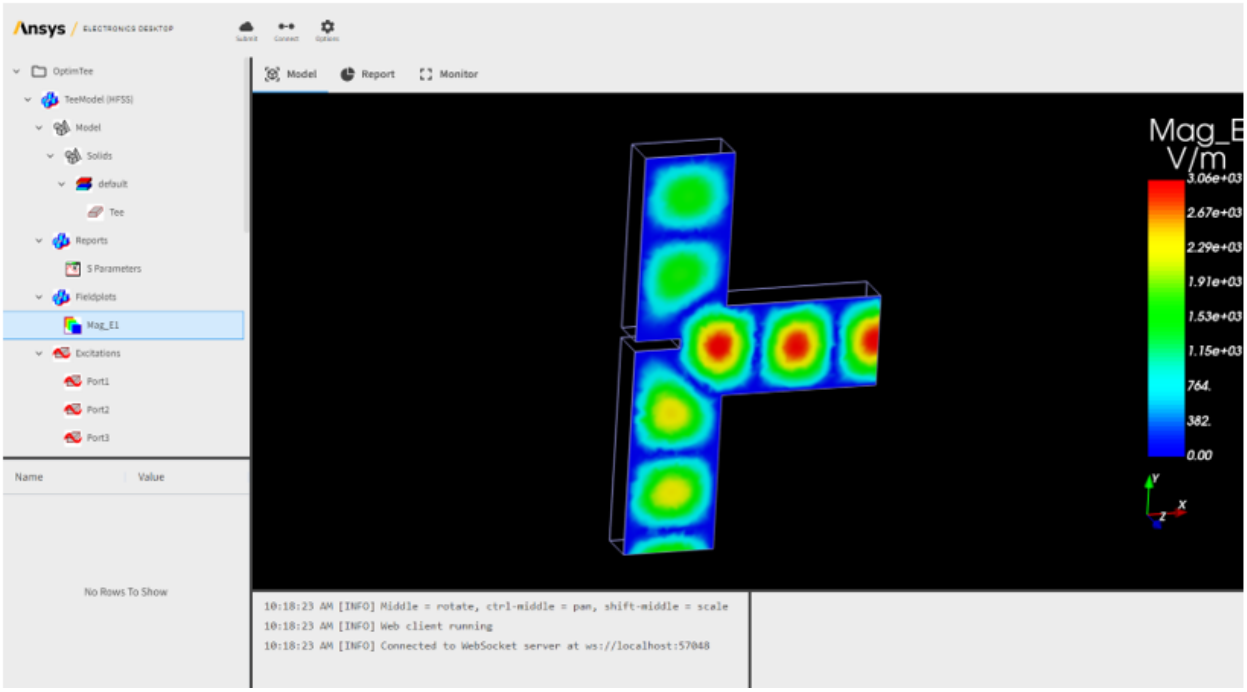
Switch to the **Monitor** tab to gain access to the subtabs for Convergence, Profile and Messages.



Select a Report in the **Project Tree** to view it.



Select a Field Plot in the **Project Tree** to view it.



Aborting an Analysis

To end the solution process before it is complete:

- Right-click In the **Progress** window and click **Abort** or click on drop-down button next to progress bar(s).

  The solver ends the analysis immediately.

The solver ends the analysis immediately. The data for the currently running adaptive pass, frequency point, time step, or variation is lost. Previously completed solutions may or may not be retained (depending on what process was interrupted by the **Abort** command). If you want to ensure that the results of all completed solutions are saved, use **Clean Stop** instead of **Abort**.

To abort the solution process after the current adaptive pass or solved frequency point is complete:

- Right-click the **Progress** window, or click on drop-down button next to progress bar(s) and click **Clean Stop** on the shortcut menu.

  The analysis ends after the currently running adaptive pass, frequency point, time step, or variation has been solved. Solutions completed before the stop request are retained.

If you request a clean stop during the third adaptive pass, the solution for the third pass will be available once the third pass has finished solving, but the fourth pass will not run.

Ansys EM Application as an LSF Job

If you have an Ansys EM application running as an LSF job, you can use the command "bkill -s SIGTERM *jobid*" to terminate that 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.

Linux

For Linux, you can use TERM commands. Sigterm handling is done in the Desktop library. You can abort a running batchsolve by sending a TERM signal to hfss.exe.

desktopproxy Utility to Abort or Cleanstop a BatchSolve

The desktopproxy Utility to abort or cleanstop a batch solve is available in the installation directory: ("...\ANSYS Inc\v251/AnsysEM" on Windows, ".../ansys_inc/v251/AnsysEM" on Linux).

To abort a job, run it on a host which has network access to the first execution host of the job, as well as has access to the directory of the project file. This utility does not abort Large scale DSO jobs for which you need to use desktopjob utility itself.

Usage:

```
desktopproxy -abort <projectpath>
-cleanstop <projectpath>
```

Where the <projectpath> is the full path to the project file.

The behavior (e.g., saving the results computed so far) should be same as if you interactively aborted AnsysEDT solve. if AnsysEDT is 'busy' (e.g., doing Optimetrics calculation, computing a report), abort (using desktopproxy) might take a long time.

The utility exits silently after it requests ansysedt process to abort the job cleanly. After the job ends, the batch log file may contain a message such as:

```
[error] Project:TestProject, Design:HFSSDesign1 (DrivenModal),
Simulation was aborted by user on server: sjohpc-lsf-0.ansys.com.
(3:52:30 PM Oct 11, 2013)
```

The lock file will be automatically deleted.

Example (Linux):

```
desktopproxy -abort /home/user/test/Diff_Via.aedt
```

Example (Windows):

```
desktopproxy -abort \\server\share\projectfolder\Diff_Via.aedt
```

Limitations:

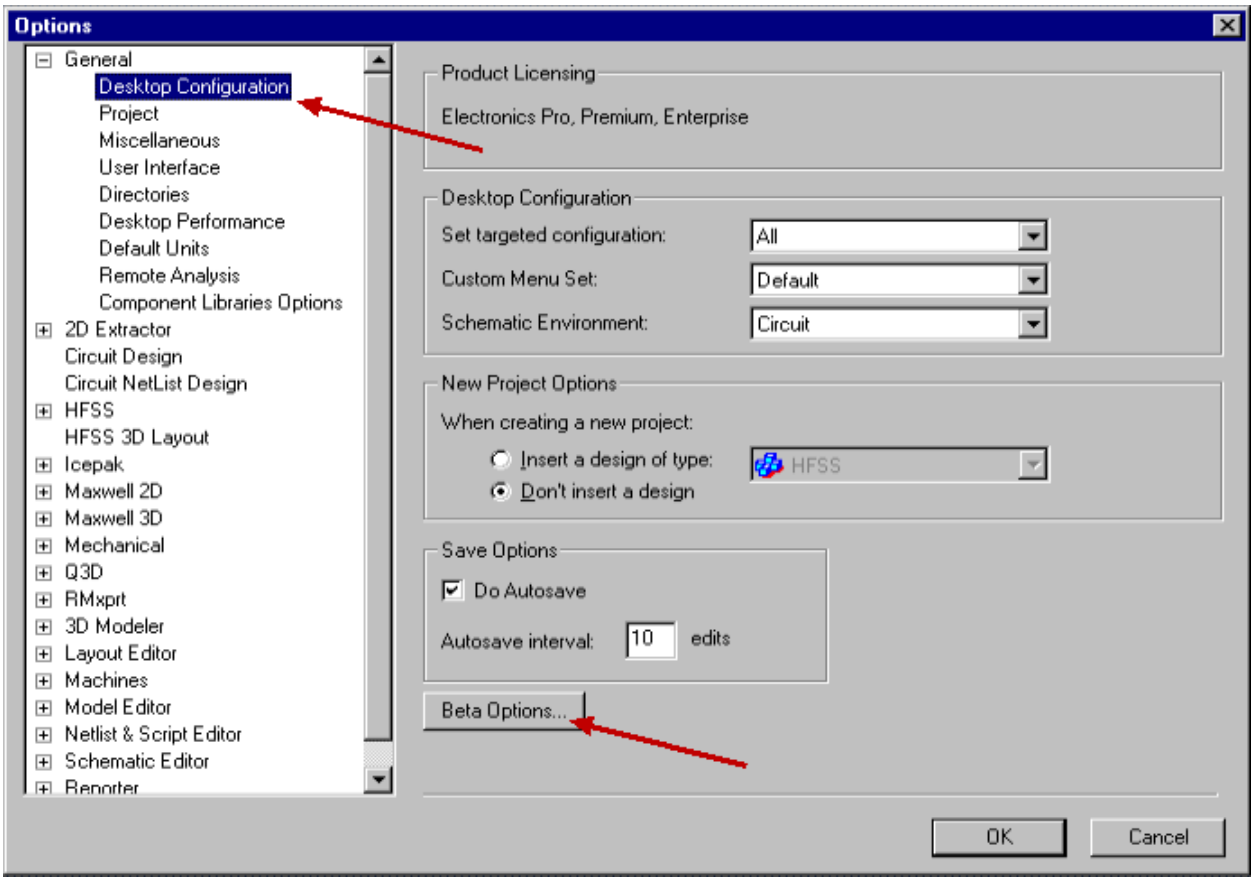
- Aborting a job makes a best effort to delete all the temp directories on all the hosts
- For a hung job, a clean shutdown may not be possible. In that case, the user must use OS/scheduler's facilities to kill the job.

Support for Open MPI on Linux [Beta]

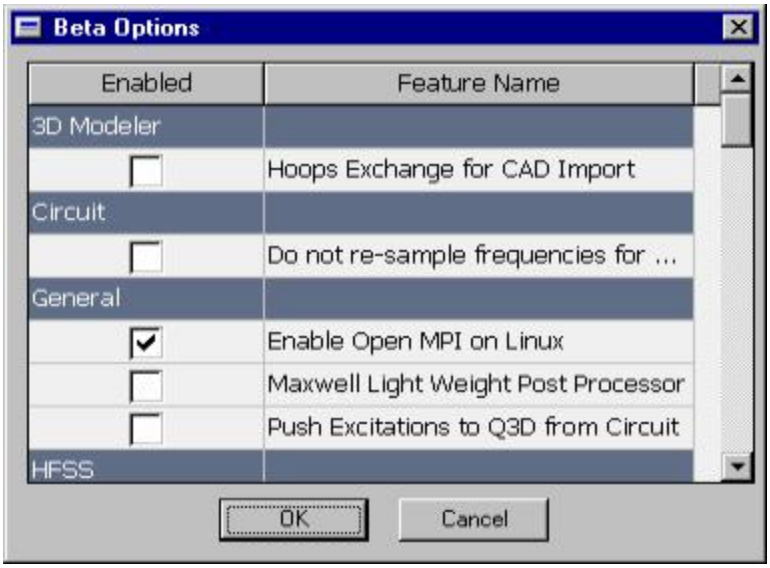
Support for Open MPI on Linux is available as a beta feature for HFSS, HFSS 3D Layout, Icepak, Maxwell (3D and 2D), and Q3D designs.

To enable the beta feature:

1. Access the *Options* dialog box using one of the following two mehtods:
   - From the menu bar, click **Tools** > **Options** > **General Options**.
   - On the **Desktop** ribbon tab, click  **General Options**.
2. In the tree on the left side of the dialog box, expand **General** and select **Desktop Configuration**.
3. Click **Beta Options**:



4. Select **Enable Open MPI on Linux**.



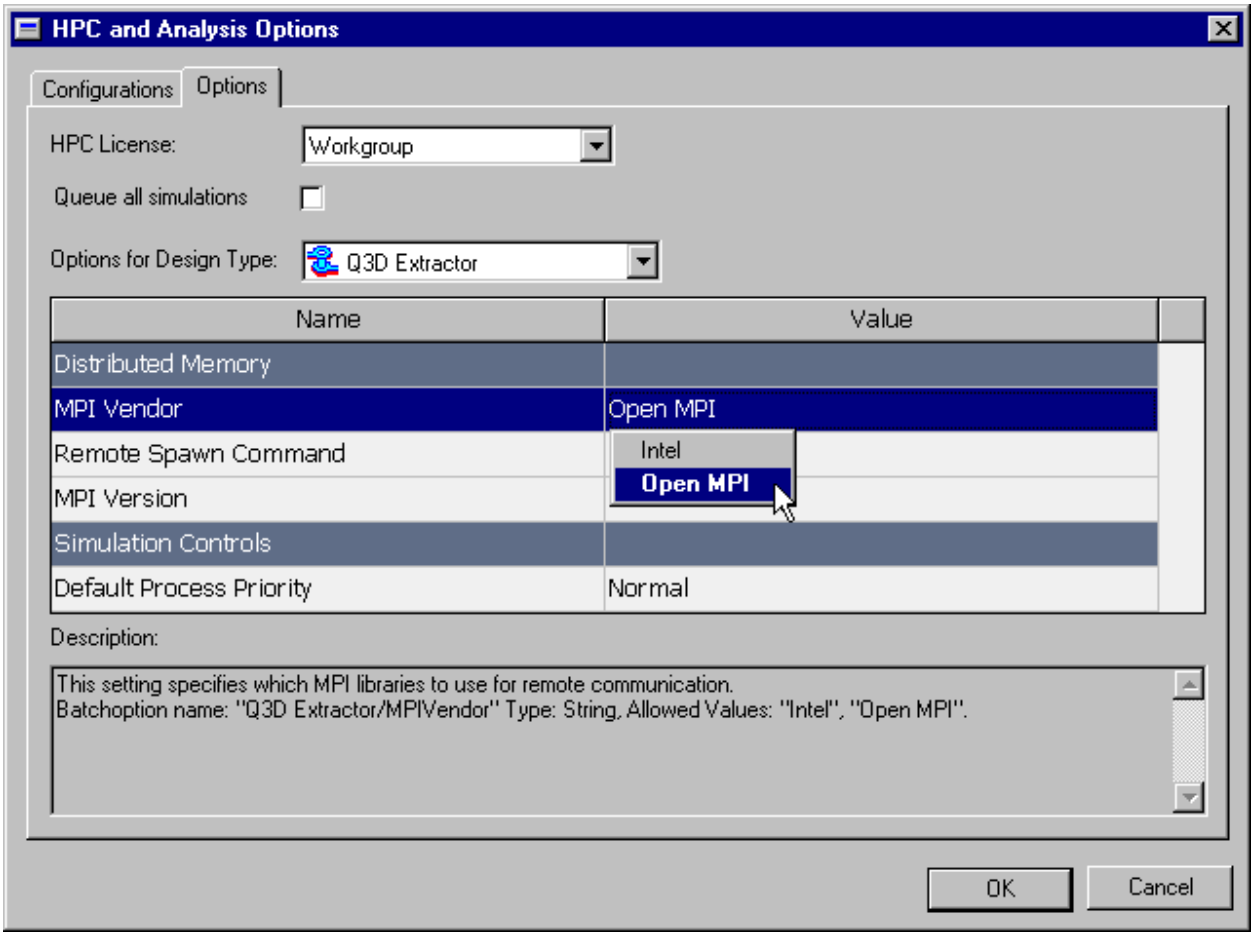
5. When making a change to any beta option, you may be prompted to restart the Ansys Electronics Desktop application. If prompted, click **Yes** to do so.

The **Open MPI** option will now be available in the **MPI Vendor** drop-down menu within the *Options* tab of the *HPC and Analysis Options* dialog box. You only need to perform this procedure once; the Open MPI option will remain available for future sessions and projects.

To choose Open MPI as the MPI Vendor:

1. Open the project in the Ansys Electronics Desktop application (interactive mode).
2. From the **Simulation** ribbon tab, click  **HPC Options**
3. In the *HPC and Analysis Options* dialog box, select the **Options** tab.

   The contents of this tab vary by design type.
4. Select **Open MPI** from the **MPI Vendor** drop-down menu:



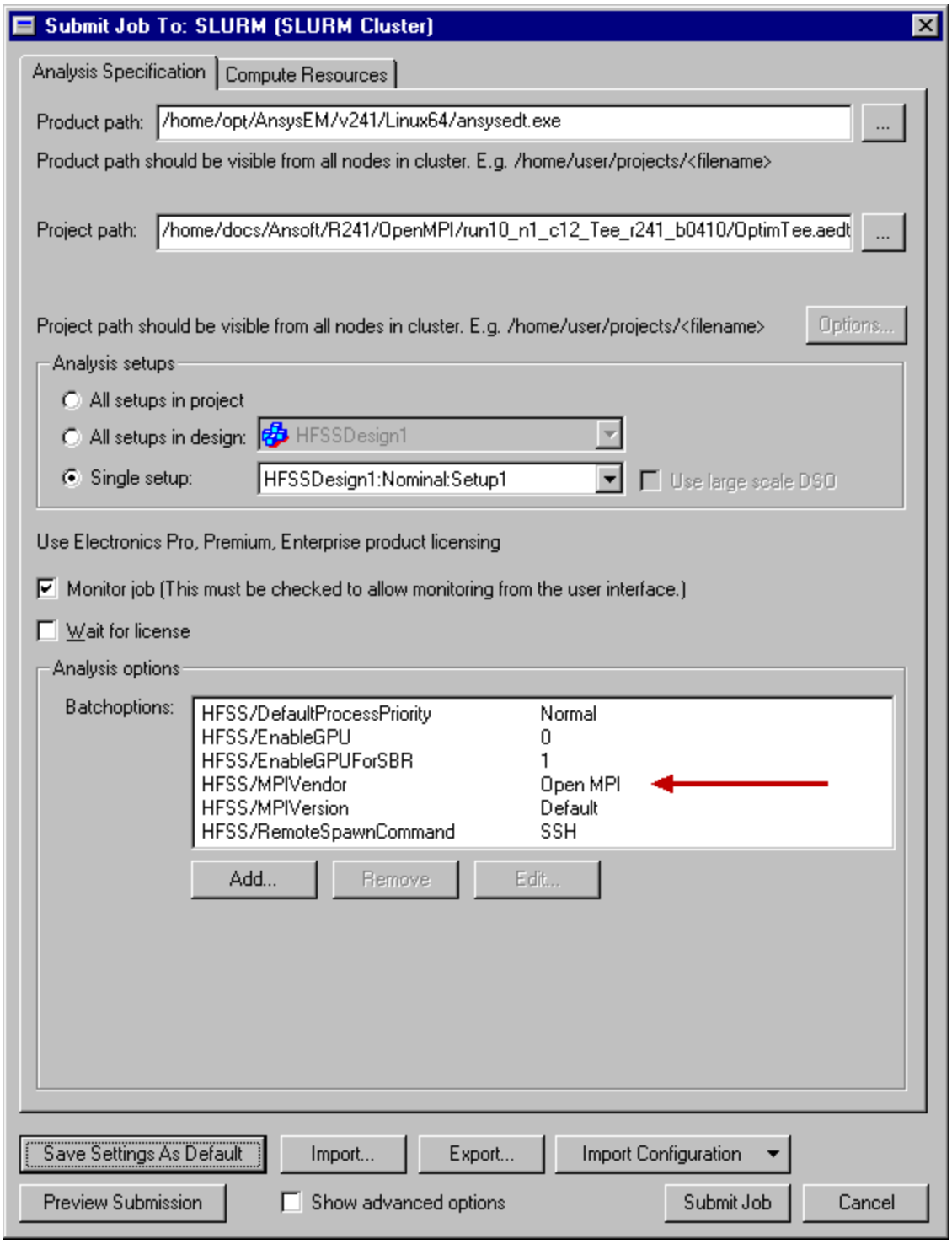
5. Click **OK**.

   You can then run the simulation on a Linux cluster as follows.

6. In the Project Manager, right-click the solution setup to solve and choose **Submit Job** from the shortcut menu.
7. Specify the desired parameters and click **Submit Job**.

   In the *Submit Job* dialog box, *Open MPI* is included as a batch option for each supported design type along with other applicable batch options.

   The following example is for an HFSS design:



Running HPC Diagnostics

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, 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.

This note describes how to use the diagnostics tool.

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 (see High Performance Computing (HPC) Integration), the tool includes basic 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. In case,

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 basic 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 diagnostics subdirectory of schedulers directory.

Linux:

```
opt/ansys_inc/v251/AnsysEM/schedulers/diagnostics
```

Windows:

```
"C:\Program Files\ANSYS Inc\v251\AnsysEM\schedulers\diagnostics
```

Using Diagnostics Scripts on Linux Clusters

The following basic scripts are provided in the diagnostics directory (.../ansys_inc/v251/AnsysEM/schedulers/diagnostics):

These job submission scripts are scheduler specific.

Scheduler	Basic 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:

```
...\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 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: XMLfile name

Diagnostic Report

The diagnostic report is an HTML file that (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 basic diagnostic scripts with the following steps:

1. Locate the relevant basic 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* subfolder of the *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\AnsyEM\v251\AnsysEM"
Linux example	/shared/ansys_inc/v251/AnsysEM
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

Environment variable	**ANSYSEM_DIAG_RESULTS_DIR**
	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_CUSTOM_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, and collect diagnostic information relevant for running HPC jobs. The tool combines the diagnostic information to produce an HTML report. The tool saves 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 user's home directory. On Windows, the user must specify this location using ANSYSEM_DIAG_RESULTS_DIR environment variable.

Changing Solution Priority for System Resources

You can modify the priority of Ansys Electronics Desktop simulations so that system resources are allocated to other computer processes before the solver. If you reduce the priority of Ansys Electronics Desktop simulations, your other software tools will respond as they normally would, but Ansys Electronics Desktop simulations may take longer.

> **Note:**
>
> The Windows Task Manager does not indicate a reduced priority for the Ansys Electronics Desktop solvers. 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 **Tools** > **Options** > **HPC and Analysis** to access the **HPC and Analysis** dialog box, and click the **Options** tab.
2. From the **Change Priority** menu (or the **Default Process Priority** drop-down menu), select one of the following priorities:

Lowest Priority	
Below Normal	
Normal	The default.
Above Normal	
Highest	

3. Click **OK**.

6 - Toolkits in Q3D Extractor

Toolkits can be accessed from the **Q3D Extractor** > **Toolkit** menu.

Available toolkits include:

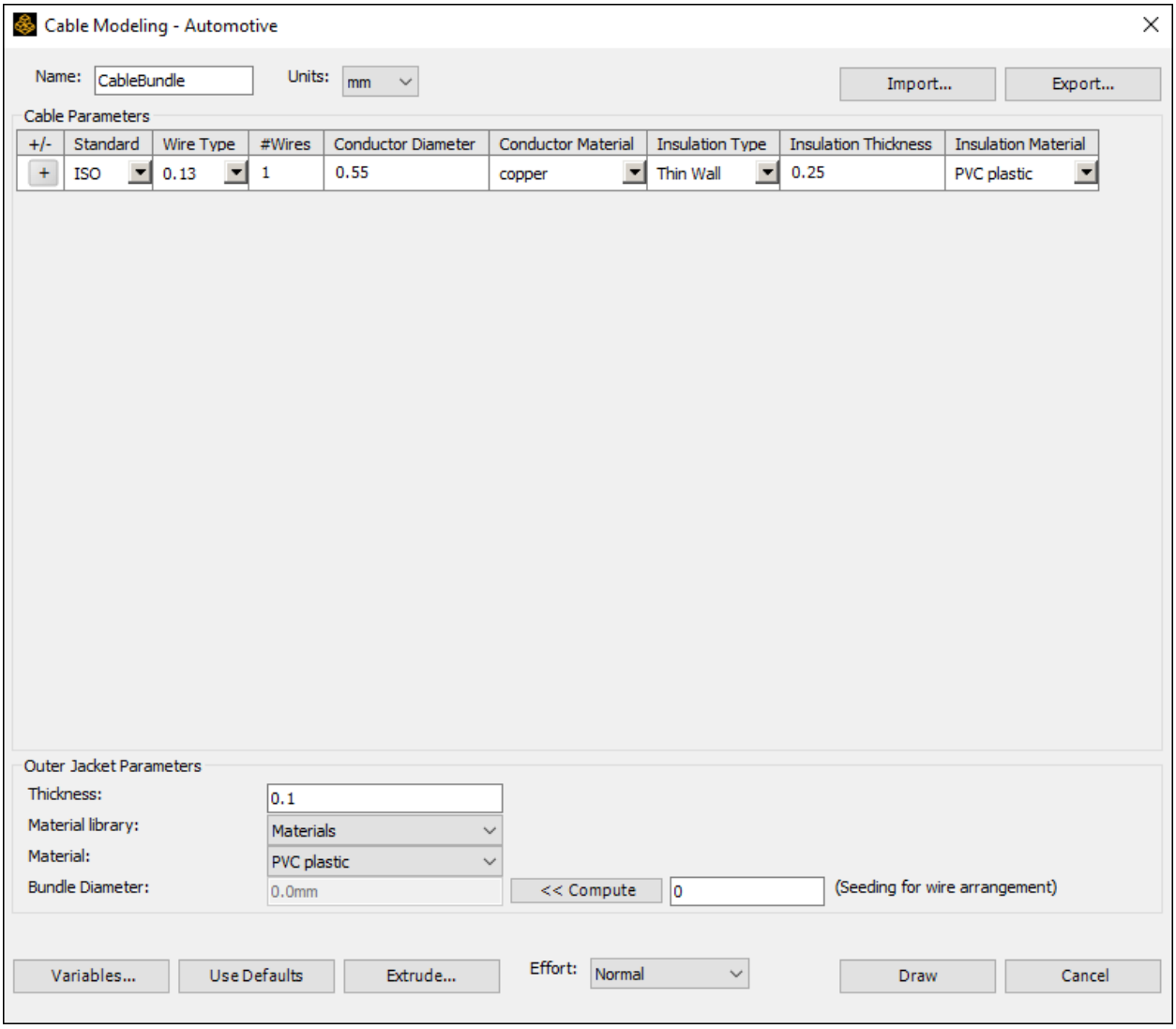
- Cable Modeling - Automotive Toolkit
- Cable Modeling - Oil & Gas Toolkit

Cable Modeling - Automotive Toolkit

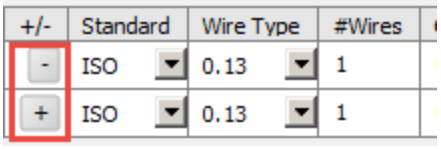
To work with tool kits in the automotive industries:

1. Click **Q3D Extractor** > **Toolkit** > **Cable Modeling** > **Automotive Cable Bundle**.

   The **Cable Modeling - Automotive** dialog box appears with the **Cable Parameters** listed.



2. Enter a **Name** for the toolkit, and select **Units**.
3. If you have a *.csv or *.tab file containing cable and outer jacket parameters, you can **Import** it.
4. To manually modify the **Cable Parameters**:
    - Using the +/- column, add or remove wire types to create a bundle. Clicking the **+** button adds a new row. All rows but the initial row will have a **-** button to remove that row.



- In the **Standard** field, choose between the **ISO** or **AWG** standard types.
- In the **Wire Type** field, use the drop-down menu to select the wire type.
- In the **#Wires** field, adjust the number of wires as necessary.
- In the **Conductor Diameter** field, specify the diameter in the unit you selected.
- In the **Conductor Material** field, use the drop-down menu to select a material.
- In the **Insulation Type** field, use the drop-down menu to select either **Thin Wall** or **Ultra Thin Wall**.
- In the **Insulation Thickness** field, specify a value in the unit you selected.
- In the **Insulation Material** field, use the drop-down menu to select a material.

5. To manually modify the **Outer Jacket Parameters**:
   - In the **Thickness** field, specify a value for the inner diameter in the unit you selected.
   - Use the **Material Library** drop-down menu to select the library containing the material you wish you specify, then use the **Material** drop-down menu to select the material.
   - Populate the **Bundle Diameter** field. After specifying all wires to be included in the bundle, click **Compute** to calculate the minimal bundle diameter of the bundle. The value automatically populates.

> **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.

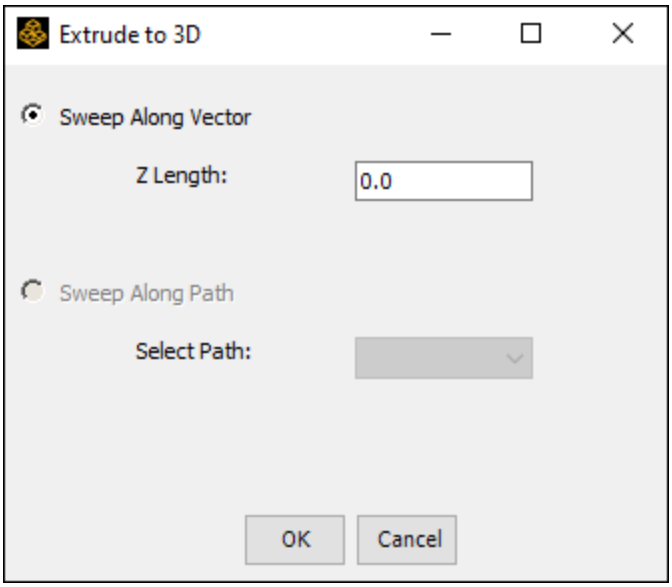
> **Note:**
>
> The Circle Packing algorithm is used to compute the values. It tries to automatically pack the conductors in the bundle tightly with a minimum of empty space between them. It generates and tests a large number of random variations in the conductor positions and attempts to minimize the overall diameter of the bundle.

- You can adjust the **Bundle Diameter** generated by specifying a **Seeding for wire arrangement** value. The value entered here is the seed value for the pseudo-random number generator used in the conductor packing process. Changing this value results in different arrangements of the conductors. This can be useful for performing statistical analysis of the cables.

6. To save your cable and outer jacket parameters for future use, click **Export** to create a *.csv or *.tab file for later import.
7. If desired, click **Use Defaults** to restore the default values. This removes any rows previously added and clears all custom values.
8. To add new variables, click **Variables**. This opens the **Edit Variables** dialog box, listing all variables that are present at the project and design levels. Click **Add** to create new variables. See: Working with Variables.
9. Click **Draw** to draw the 2D cross-section geometry. A validation check is run. This catches errors such as any variable or parameter missing a value.
10. For 3D designs, you can create a 3D geometry:

    The **Effort** drop-down menu contains four options: Low, Normal, High, and Very High. This controls the trade-off between packing quality (how tightly the conductors are packed together) and the time spent generating the model. Low effort runs relatively fast and may be good enough for small numbers of conductors, but may leave significant space in the model when there are many conductors. High effort may be required for models with large numbers of conductors (100+), but could take several minutes to run.

    Click **Extrude**. 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.

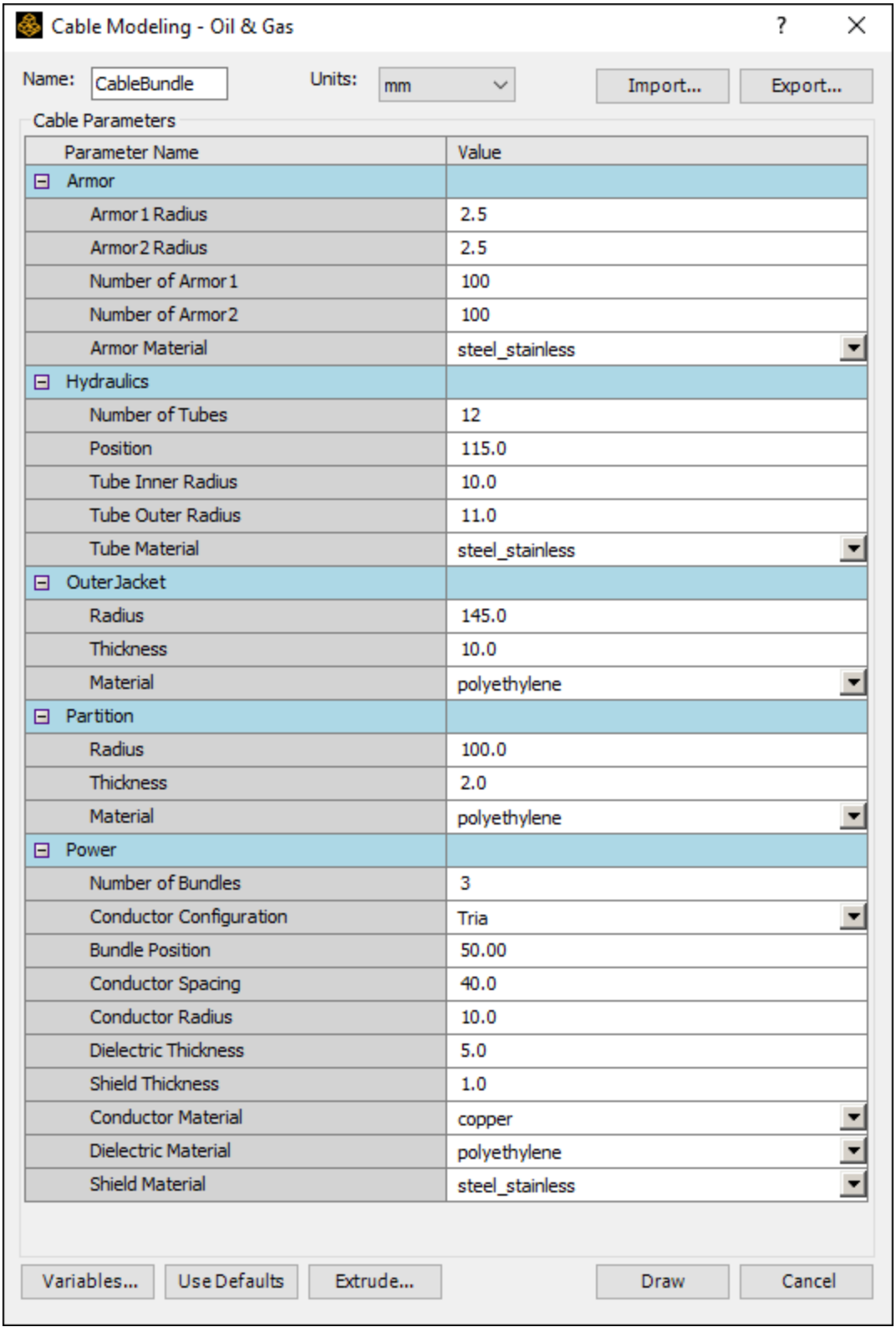
> **Note:**
>
> 3D models of cables are hard to solve if the ratio of extruded length to cross-section diameter is high.

Cable Modeling - Oil and Gas Toolkit

To work with tool kits in the oil and gas industries:

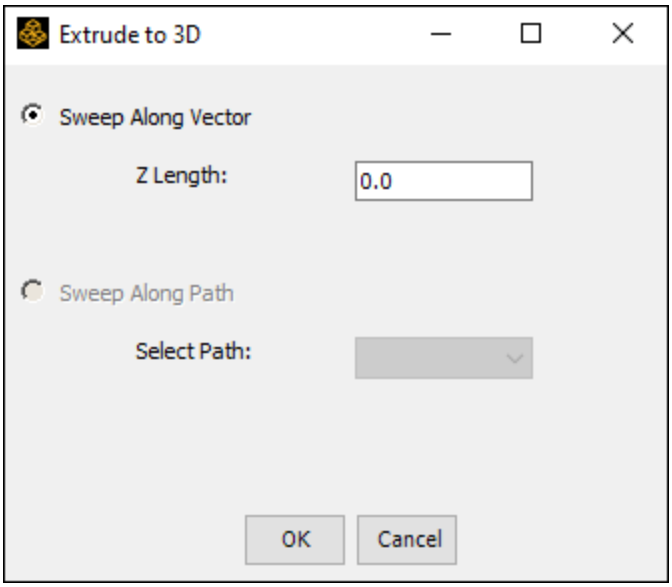
1. Click **Q3D Extractor** > **Toolkit** > **Cable Modeling** > **Oil-Gas Cable Bundle**.

   The **Cable Modeling - Oil & Gas** dialog box appears.



2. Enter a **Name** for the toolkit, and select **Units**.
3. If you have a *.csv or *.tab file containing cable parameters, you can **Import** it.
4. To manually modify the cable parameters:
    - Enter and select values for each of the available fields, in the unit you selected.
5. To save your cable parameters for future use, click **Export** to create a *.csv or *.tab file for later import.
6. 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.
7. Click **Draw** to draw the 2D cross-section geometry. A validation check is run. This catches errors such as any variable or parameter missing a value.
8. For 3D designs, you can create a 3D geometry.

   Click **Extrude**. 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.

7 - Meshing

Once you have finished designing your model, you can generate the computational mesh that is used as the basis of solutions.

A good computational mesh is essential for a successful and accurate solution. If the overall mesh is too coarse, the resulting solution may be inaccurate. If the overall mesh is too fine, the computational cost may become prohibitive. In summary, the cost and accuracy of the solution are directly dependent on the quality of the mesh.

Specifying Initial Mesh Settings in Q3D Extractor

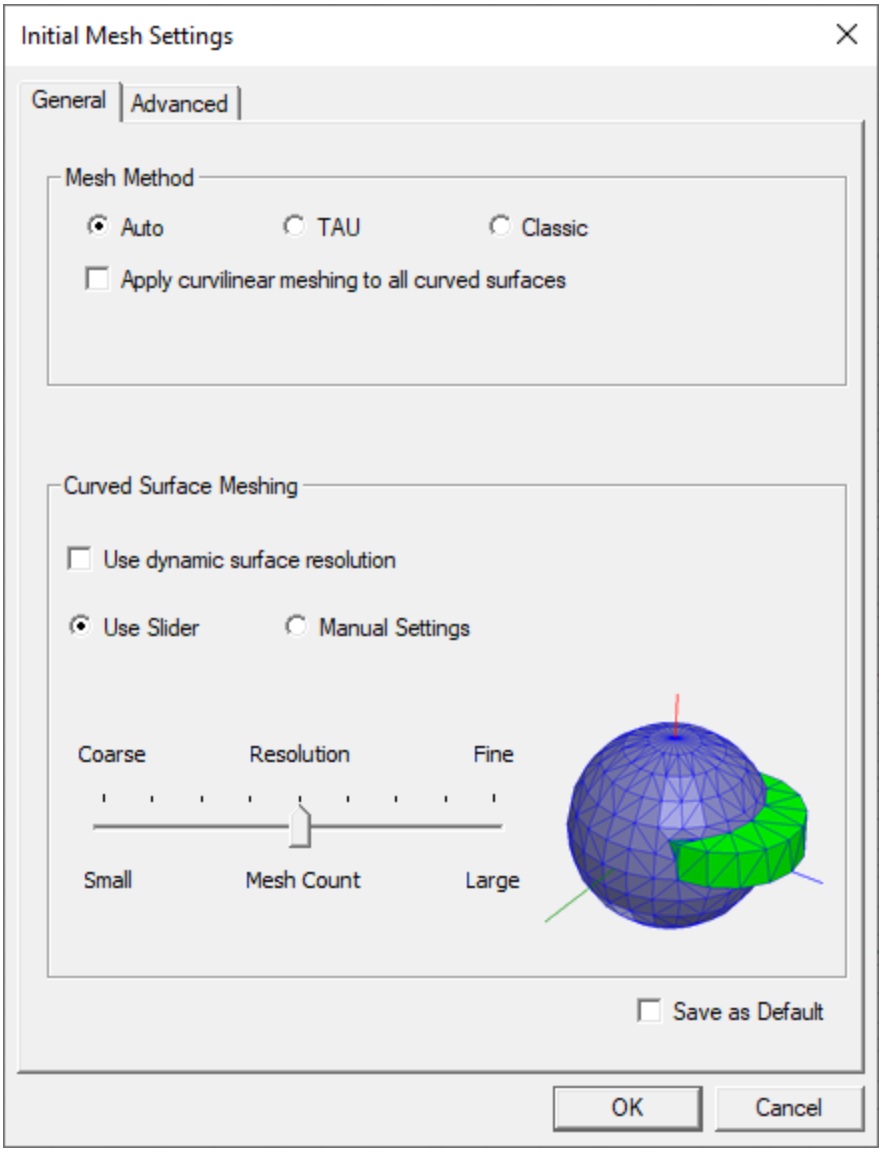
Q3D Extractor automatically chooses which of two meshing approaches to take. Each solver predicts which one gives the best results, balancing mesh reliability, speed, quality, size and design characteristics. In some cases, however, you may wish to specify the initial mesh settings, including surface approximation and the meshing approach, for all objects.

> **Note:**
>
> If you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

To specify initial mesh settings:

1. Access the **Initial Mesh Settings** window one of two ways:
   - Select **Q3D Extractor** > **Mesh** > **Initial Mesh Settings...**
   - In the Project Manager, right-click **Mesh**, and select **Initial Mesh Settings**.
   - From the **Simulation** tab of the ribbon, click **Mesh Settings**.

   The **Initial Mesh Settings** window appears, with the **General** tab selected.



2. The **General** tab contains **Mesh Method** options for:
    - **Auto** – the solver automatically selects the mesher. This is the default setting.
    - **TAU** – only specific curve faces will be remeshed (for example, equation-based axisymmetric faces). If a curve face connects to non-remeshed curve faces, the curve face will not be remeshed. If a curve face 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, it shows in the profile as Mesh TAU (Surface). This can fall back to Mesh TAU (Wrapper), which can fall back to the classic Mesh (Stitch). While the volume mesh (for DC RL) has an automatic fallback to classic, the surface mesh (for CG and AC RL) does not. In general, *use TAU only when Auto/Classic fails*.

- **Classic** – the standard mesher.
- **Enable Ansys Prime** – In order to improve the accuracy of CG or AC RL solutions, Q3D Extractor can make use of Ansys Prime meshing libraries to remesh existing meshes.

  To verify whether Ansys Prime remeshing was used during a simulation, check for lines labeled "Prime Remesh" in the **Solution Profile**:

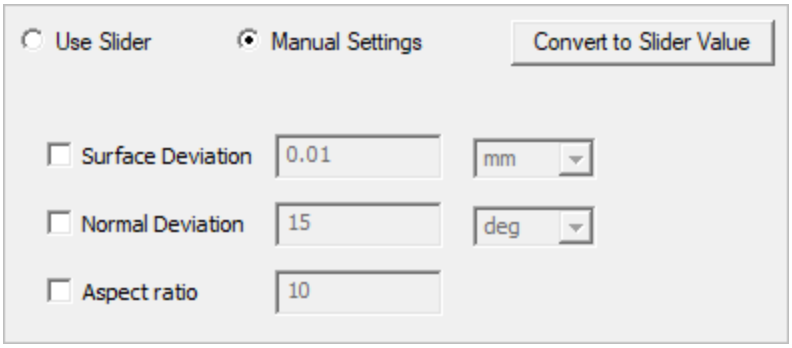
Task	Real Time	CPU Time	Memory
Mesh (phi surface)	00:00:00	00:00:00	37.8 M
Adaptive Pass #1			
Prime Remesh ✔	00:00:00	00:00:00	0 K
Machine Configuratio...			
Solver setup	00:00:03	00:00:12	159 M
Matrix solution	00:00:05	00:00:21	178 M
Adaptive Pass #2			
Mesh (surface, adapt...	00:00:00	00:00:00	31 M
Prime Remesh ✔	00:00:00	00:00:00	0 K
Machine Configuratio...			
Solver setup	00:00:04	00:00:15	206 M

3. **Curved Surface Meshing** options include:
   - **Use Dynamic Surface Resolution** – specifies best-practice mesh operations over geometric models. This mesh operation supports 3D volume mesh and surface mesh in all products. 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 and small curve objects, get relatively coarse triangulations so that overall element count is reduced.

     When dynamic surface resolution is enabled, only **Use Slider** is permitted. **Manual Settings** will be unavailable.

   - **Use Slider** – Drag the slider to select from the range of ***Course*** *Resolution* (with a ***Small*** *Mesh Count*) to ***Fine*** *Resolution* (with a ***Large*** *Mesh Count*. Here, "*Mesh Count*" refers to the number of elements comprising the mesh. The graphic on the right updates to illustrate your selection. See: Modifying Surface Approximation Settings.

- **Manual Settings** – Allows you to manually specify Surface Deviation, Normal Deviation, and Aspect Ratio.

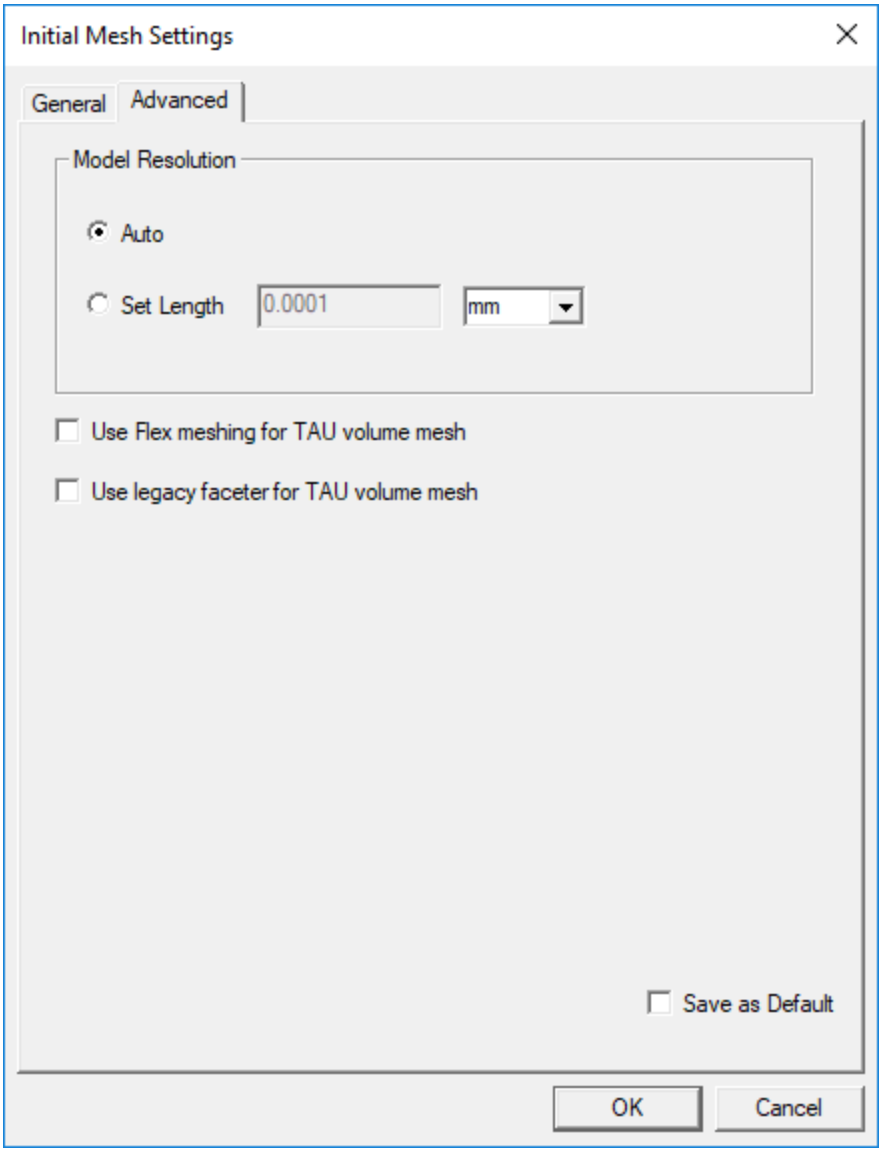


Use the check boxes to enable the fields and specify values:

- **Surface Deviation** – the distance between the true surfaces of the selected faces and the meshed faces.
- **Normal Deviation** – the angular distance between the normal of the true surface and the corresponding mesh surface.
- **Aspect Ratio** – 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.

Clicking **Convert to Slider Value** converts the manually entered values to an equivalent slider setting and returns the panel to the slider view.

4. If desired, select **Save as Default** to keep your settings.
5. If desired, use the **Advanced** tab to specify a **Set Length** for **Model Resolution**. This 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.



Additionally, the Advanced tab contains the following options:

- **Use Flex Meshing for TAU Volume Mesh** – 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. Review the mesh to evaluate whether it is acceptable for simulation. See: TAU Flex Meshing.
- **Use Legacy Faceter for TAU Volume Mesh** – By default, TAU uses the latest faceter to generate geometric models for meshing. Occasionally, the mesher may fail, and TAU will automatically fall back to use a 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. As such, this option should be used only as a last resort.

6. Click **OK** to apply your choices.

   The settings will be applied to the initial mesh generated.

TAU Flex Meshing

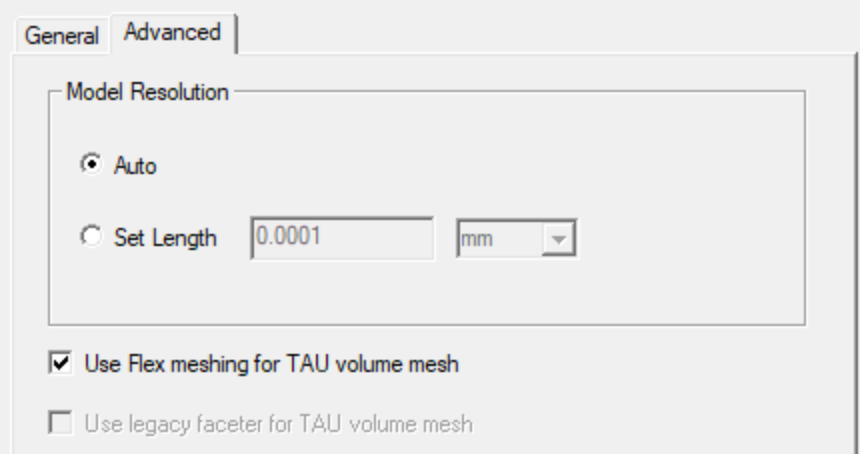
TAU Flex is designed for guaranteed mesh success, and may also reduce meshing time. It provides a mesh even for complex and “dirty” geometry model without any preliminary geometric healing or repair.

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,...), 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.

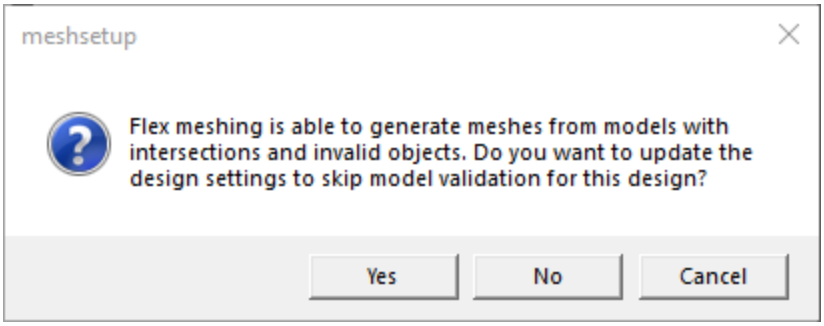
> **Note:**
>
> TAU Flex meshing is not available in 2D Extractor.

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**.

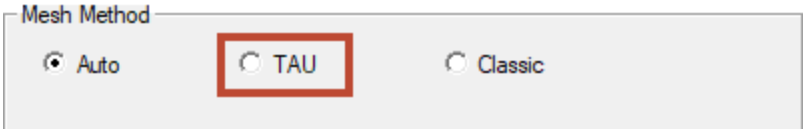


Selecting this option opens a dialog box. Select **Yes** to continue.

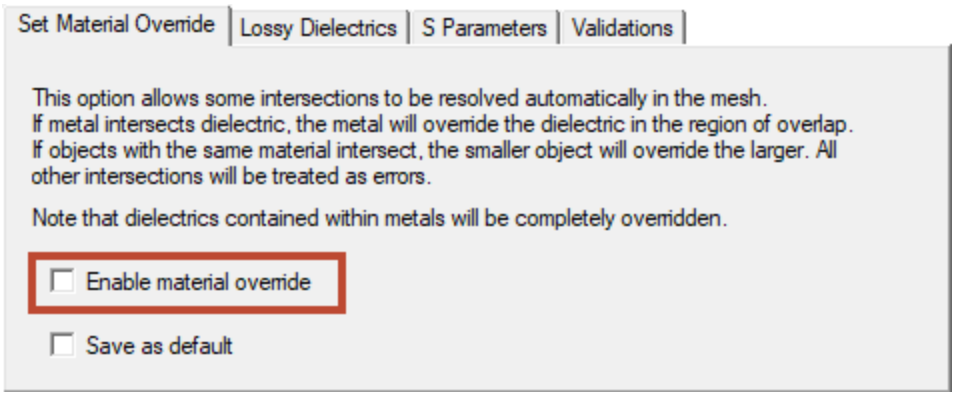


To get the full benefits of TAU Flex, especially for complex or "dirty" models, follow these recommended steps:

1. On the **General** tab of the **Initial Mesh Settings** window, select **TAU**.

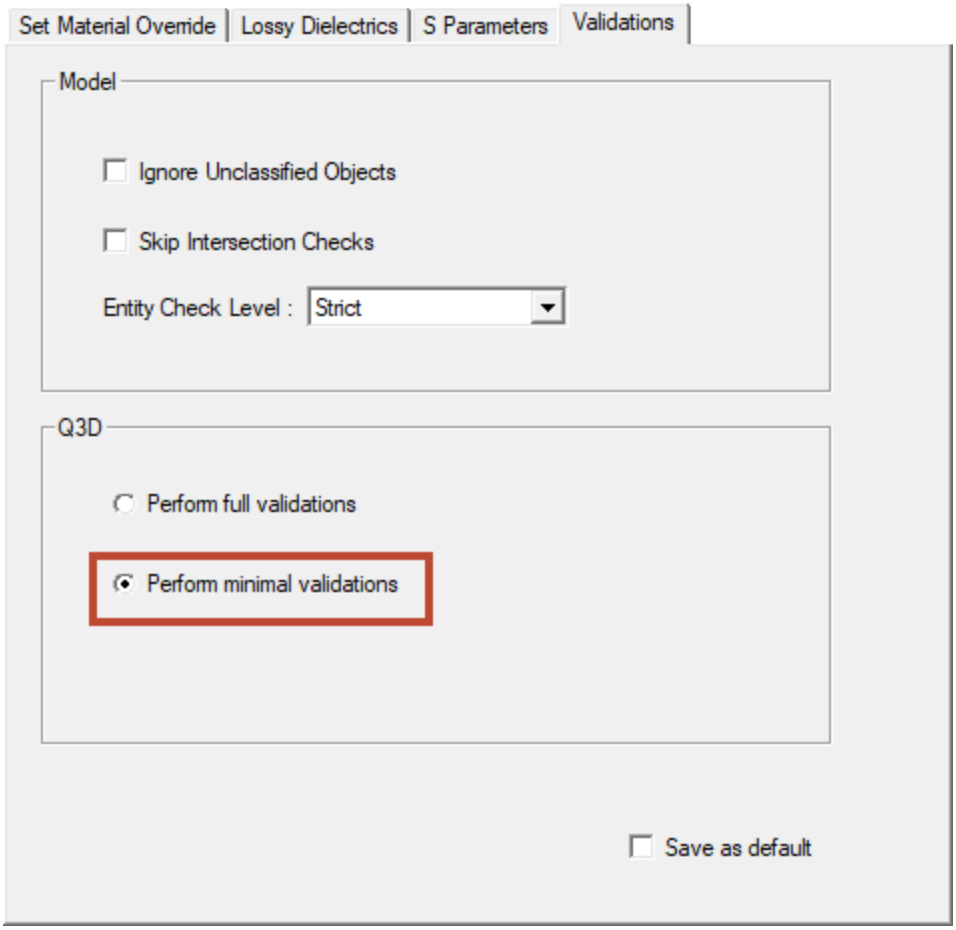


2. Click **Q3D Extractor** > **Design Settings** to open the **Design Settings** window.
3. On the Set Material Override tab, select **Enable material override**.



   This activates TAU's priority meshing method based on metal/non-metal materials.

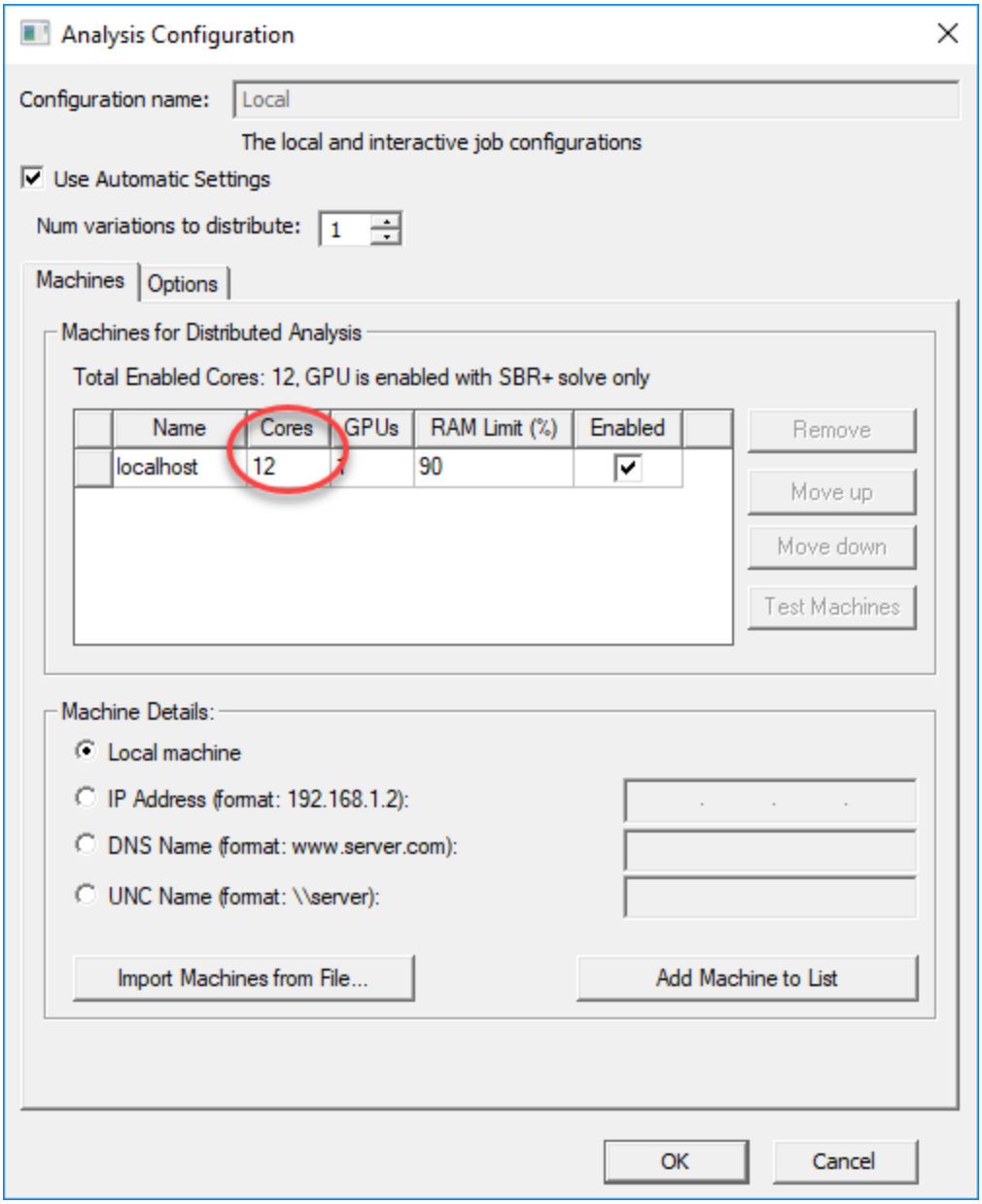
4. Still in the Design Settings window, click the **Validations** tab and select **Perform minimal validations**.



This reduces preprocessing time for complex models, and allows simulation of "dirty" geometry for which TAU Flex can provide a mesh.

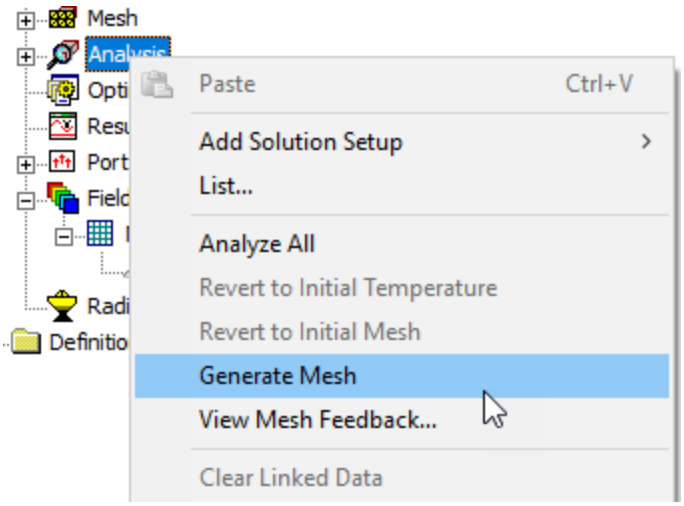
5. Because TAU Flex is a parallel mesher for many processes, you should specify multiple cores in **HPC and Analysis Options** to speed up the meshing process. Access the Analysis Configuration window one of two ways:
   - Select the Simulation tab on the ribbon and click **Analysis Config**.
   - Use **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.

6. In most cases, TAU Flex can provide mesh as accurate as TAU or Classic, but not always. Therefore, we recommend that you **Generate Mesh** without solving, and that you review the mesh before proceeding 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).

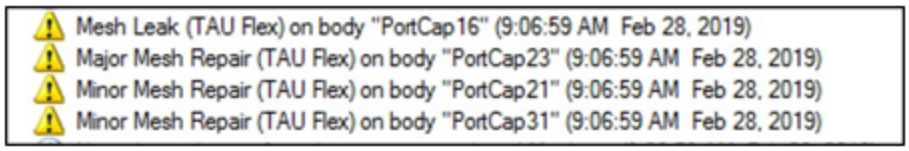
There are three types of warnings:

- **"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

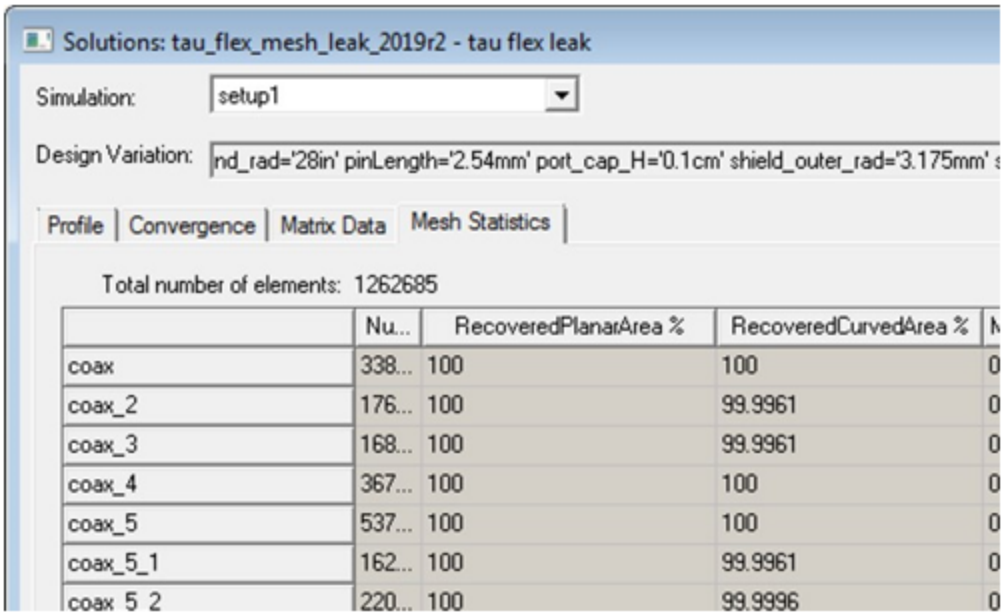
TAU offers three ways to view mesh warnings.

1. **Message Manager** – Immediately after the mesh generation is done, if a tolerant mesh is given, warnings display in the Message Manager. If the window is hidden, click **Show Messages**.



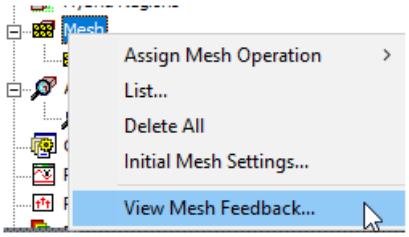
2. **Mesh Statistics** – On the **Results** tab of the ribbon, click **Solution Data** to open a **Solutions** window.

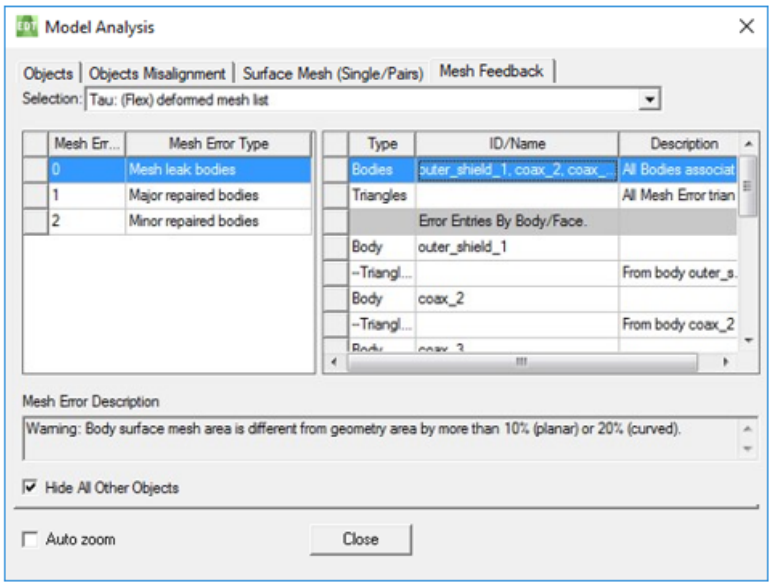
   On the **Mesh Statistics** tab of the **Solutions** window, if a tolerant mesh is generated, the surface area difference between the mesh and geometric model at body level is listed as "RecoveredPlanarArea" and "RecoveredCurveArea". 100% means that the body has a strict mesh. You can click the surface area tab to sort the body list and see the tolerant mesh bodies first.



	Nu...	RecoveredPlanarArea %	RecoveredCurvedArea %	
coax	338...	100	100	0
coax_2	176...	100	99.9961	0
coax_3	168...	100	99.9961	0
coax_4	367...	100	100	0
coax_5	537...	100	100	0
coax_5_1	162...	100	99.9961	0
coax_5_2	220...	100	99.9996	0

3. **Model Analysis** – To view the details of the tolerant mesh, click **Modeler** > **Model Analysis** > **Show Analysis Dialog** > **View Mesh Feedback** to launch the **Model Analysis** window.





Click on the top list to display objects that have tolerant mesh.

Click the "Triangles" to display the mesh on the tolerant objects.

Individual objects are 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 using one of the following techniques:

1. Redraw the geometry in question in the native modeler.
2. Heal the geometry, either with integrated healers or perhaps in Discovery.
3. If the object 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.

Specifying Initial Mesh Settings in 2D Extractor

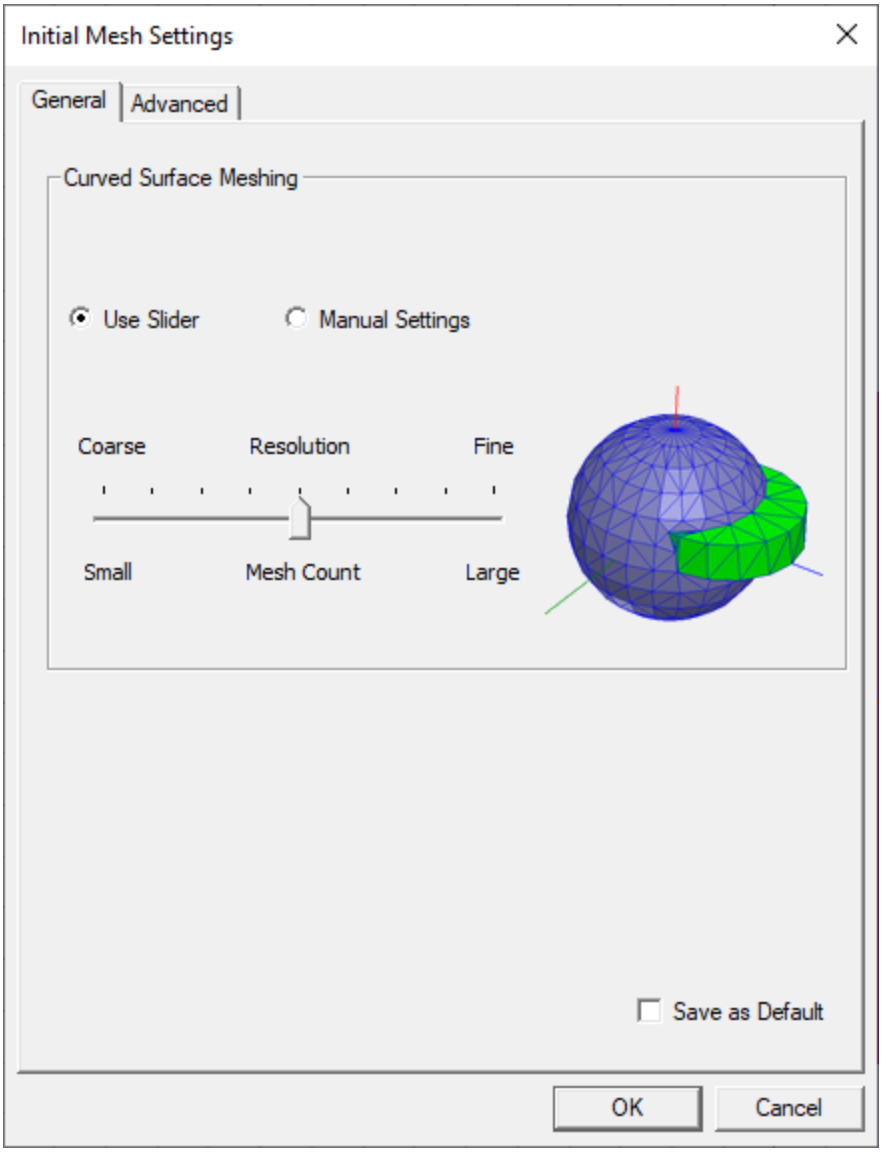
From the **Initial Mesh Settings** window, 2D Extractor allows you to manually change surface approximation and model resolution settings for all objects.

> **Note:**
>
> If you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

To specify initial mesh settings:

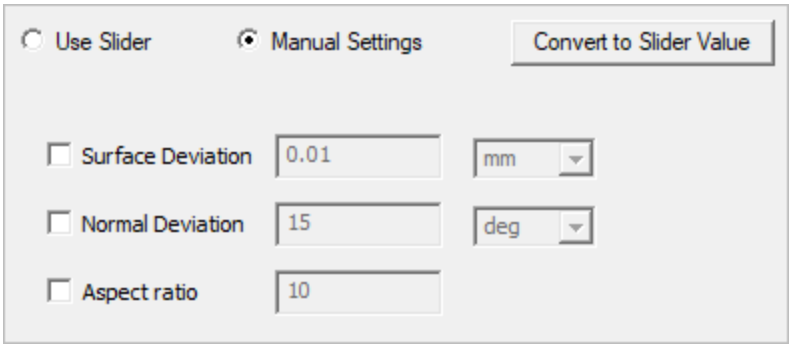
1. Access the **Initial Mesh Settings** window one of two ways:
   - Select **2D Extractor** > **Mesh** > **Initial Mesh Settings...**
   - In the Project Manager, right-click **Mesh**, and select **Initial Mesh Settings**.
   - From the **Simulation** tab of the ribbon, click **Mesh Settings**.

   The **Initial Mesh Settings** window appears, with the **General** tab selected.



2. **Curved Surface Meshing** options include:
   - **Use Slider** – Drag the slider to select from the range of ***Course*** *Resolution* (with a ***Small*** *Mesh Count*) to ***Fine*** *Resolution* (with a ***Large*** *Mesh Count*. Here, "*Mesh Count*" refers to the number of elements comprising the mesh.

     The graphic on the right updates to illustrate your selection. See: Modifying Surface Approximation Settings.
   - **Manual Settings** – Allows you to manually specify Surface Deviation, Normal Deviation, and Aspect Ratio.

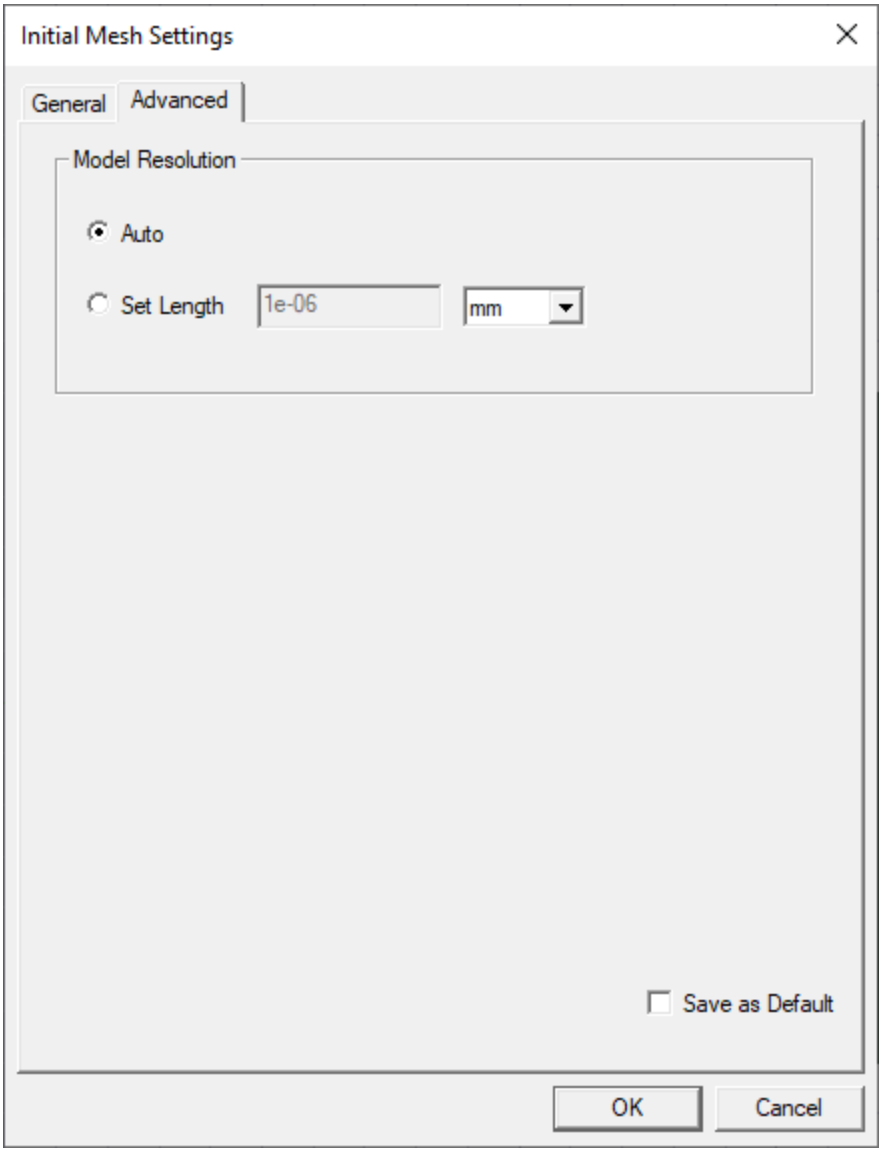


Use the check boxes to enable the fields and specify values:

- **Surface Deviation** – the distance between the true surfaces of the selected faces and the meshed faces.
- **Normal Deviation** – the angular distance between the normal of the true surface and the corresponding mesh surface.
- **Aspect Ratio** – 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.

Clicking **Convert to Slider Value** converts the manually entered values to an equivalent slider setting and returns the panel to the slider view.

3. If desired, select **Save as Default** to keep your settings.
4. If desired, use the **Advanced** tab to specify a **Set Length** for **Model Resolution**. This 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.



5. Click **OK** to apply your choices.

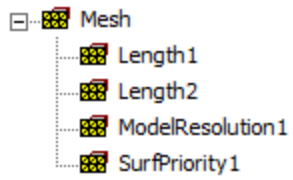
   The settings will be applied to the initial mesh generated.

Defining Mesh Operations in Q3D Extractor

Mesh operations are optional mesh refinement settings that provide the mesh construction guidance. This technique of guiding mesh construction is referred to as "seeding" the mesh. Seeding is performed using **Mesh** commands.

Mesh refinement is the process of subdividing or splitting up of a selected subset of an existing mesh. This subset is usually identified by the solver from physical or mathematical characteristics of the underlying electromagnetic problem. Mesh refinement almost always increases the mesh size.

Each mesh operation you define appears in the Project tree:



Note:

There are no restrictions against assigning multiple mesh operations to the same entities. The *type* of mesh operation can differ between overlapping assignments. For example, length-based refinement could be assigned inside a body, and skin-depth seeding on one or more of its faces. Additionally, overlapping mesh operations of the *same type* can have differing, or even redundant, parameter values (such as element length) without causing an error condition.

Typically, the operation that dictates the finest local meshing results will control the outcome in overlapping assignment areas or volumes.

Mesh Generation Process

An initial mesh (which includes surface approximation settings) is generated . If necessary, the mesher automatically performs any repairs needed to recover an accurate mesh representation of a model. The solution profile indicates when mesh repairs have been made, and the results of these repairs are displayed per object in the mesh statistics panel.

Adding a Mesh Operation

To add a mesh operation:

1. In the project tree, right-click **Mesh**.
2. Select the operation you want to add.

Valid operations in Q3D Extractor are:

- Length-Based Mesh Refinement – reduce the maximum size of tetrahedral elements and refine the surface length of all tetrahedral elements on a surface.
- Surface Approximation Settings – adjust settings for curved surface meshing.

- Surface Priority for TAU Mesher – designate high priority for critical surfaces.
- Specify Model Resolution – change settings to define large and small objects.

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. This command also allows you to override the automatic choice of mesher used.

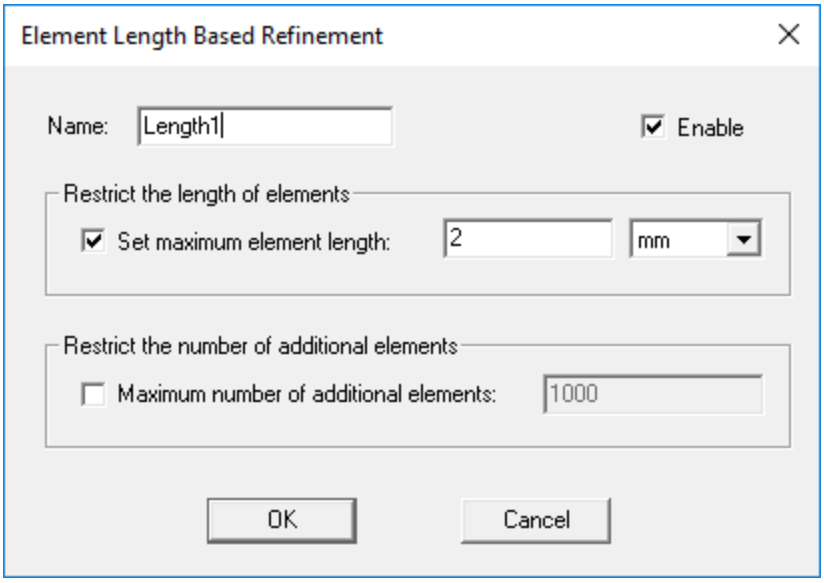
Length Based Mesh Refinement

Length-based mesh refinement settings help control the length of the tetrahedral elements and the growth of the whole mesh. If you are not sure how big a refinement is needed, by way of mesh refinement you can set a ceiling on the mesh growth with the option **Restrict the Number of Additional Elements**.

To apply a length based mesh operation either to an object face or inside an object:

1. Select the object in the modeling workspace.
2. In the Project Manager, right-click **Mesh** > **Assign Mesh Operation** > **[On Selection / Inside Selection]** > **Length Based**.

   The **Element Length Based Refinement** dialog box appears.



3. Enter a **Name** for the mesh operation, or leave the default.
4. Select the **Enable** check box to set the operation as active. You can later deselect this to deactivate it.
5. To restrict the length of tetrahedra edges, select the **Set maximum element length** check box and specify values and units using the drop-down menus.

6. To restrict the number of elements added during refinement, select the **Maximum number of additional elements** check box and specify a value.
7. Click **OK**.

When you set a mesh operation on a selection of objects (more than one) and set a maximum length, it applies to the whole mesh. Even if you selected just one object, when you refine its boundary, elements are added both inside and outside of that object.

When there are no more elements to refine or when no more elements can be added, the refinement process stops because it has either met the growth limit or the element length target. However, there are more quality improvement swaps, smoothing, matching boundary issues performed at the finishing stage that might slightly alter the mesh size.

If you restrict the number of elements to say 10000, it will add 10000 elements. If you have multiple mesh operations each for 10000 elements, each mesh operation will add 10000 elements (assuming there was no element length targets).

If you set a restriction of say 2000 elements on an object which creates 1100 elements after lambda refinement without any mesh operation, the starting mesh will contain more than 3000 elements.

Modifying Surface Approximation Settings

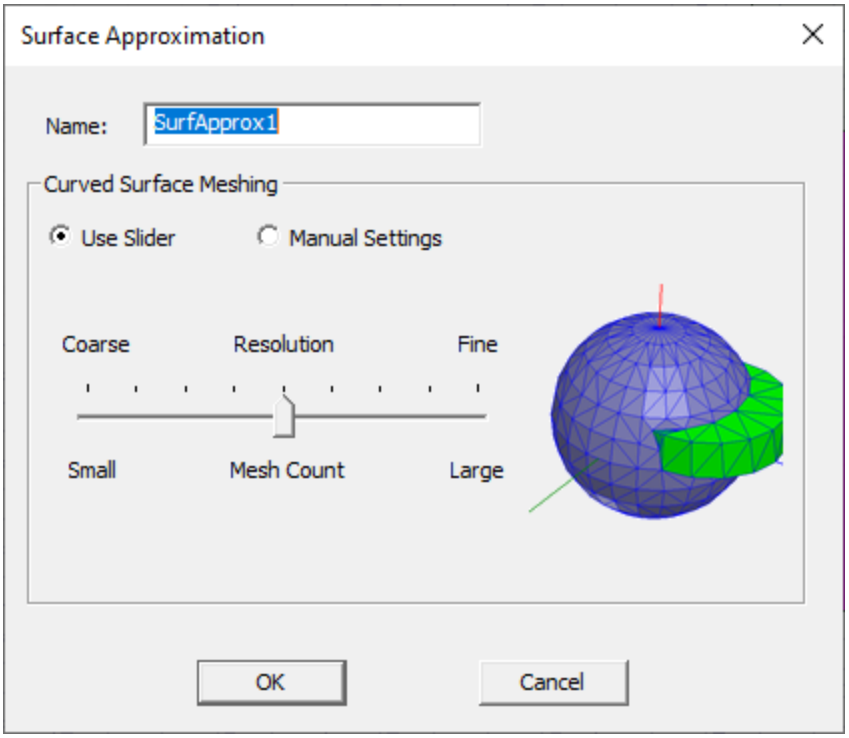
Each solver applies surface approximation settings for selected objects when it generates the initial mesh. If you modify a solver's default settings after the initial mesh has been generated, they will not affect the mesh for that design variation. Surface approximation makes sense for curved surfaces, 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. For a given surface approximation slider setting, the approximation is more refined for IE and SBR+ solvers than for FEM solvers.

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 **Q3D Extractor** > **Mesh** > **Assign Mesh Operation** > **Surface Approximation** or right click on **Mesh** in the Project Manager and select **Assign Mesh Operation** > **Surface Approximation**.

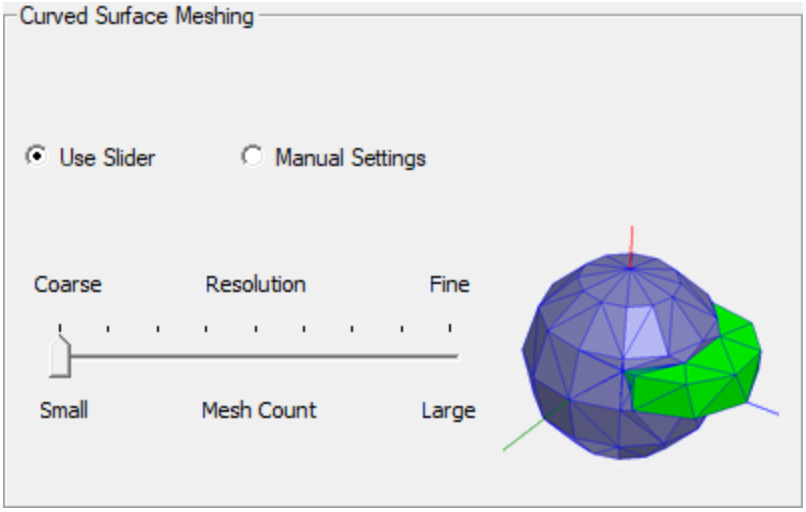
   The *Surface Approximation* dialog box appears.



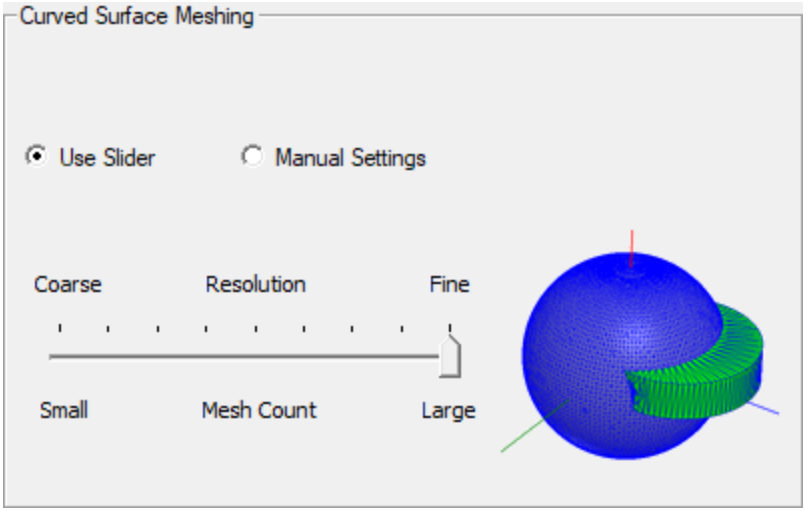
3. Type a name for the group of settings in the **Name** text box or accept the default name.
4. Under **Curved Surface Meshing**, you can select **Use Slider** or specify **Manual Settings**.

   The slider includes a visual representation of your choice, ranging from a ***Coarse*** *Resolution* (with a ***Small*** *Mesh Count*) through a nine position scale to a ***Fine*** *Resolution* (with a ***Large*** *Mesh Count*). Here, "*Mesh Count*" refers to the number of elements comprising the mesh.

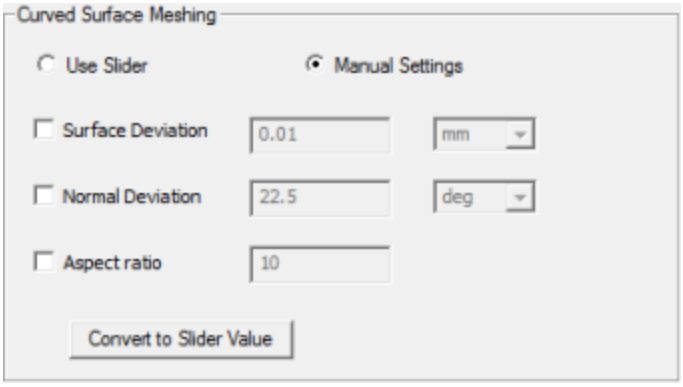
   Coarse Resolution, Small Mesh Count:



Fine Resolution, Large Mesh Count:



5. If you choose **Manual Settings**, the dialog box changes to show text fields.



> **Note:**
>
> If you selected **Use dynamic surface resolution** in Initial Mesh Settings, use of **Manual Settings** is not permitted. You must use slider bar to specify one of three levels of surface representation: coarse (1-3), normal (4-6), and fine (7-9).

6. Select **Surface Deviation**. Use the drop-down menu to select a unit of measure and then enter the distance between the true surfaces of the selected faces and the meshed faces.
7. Select **Normal Deviation**. Use the drop-down menu to select a unit of measure and then enter the angular distance between the normal of the true surface and the corresponding mesh surface.
8. Select **Aspect Ratio** and then type 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.
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:

Selecting **Use dynamic surface resolution** 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 optimized mesh operations with model analysis. In general, large curve faces, curve faces with small gaps, or skewed cables get smoother curvature representation while small curve faces, such as fillets and small curve objects, get relatively coarse triangulations to reduce overall mesh size.

This dynamic surface resolution mesh operation supports 3D volume mesh and surface mesh in all products.

Important:

The consolidation of surfaces into a conformal mesh is skipped for dynamic and tolerant meshing, including for light weight geometries. This can lead to overlapping surfaces in SBR+ simulations. The user should carefully avoid overlapping surfaces or objects as SBR+ can produce unexpected results.

Specifying Surface Priority for TAU Mesher

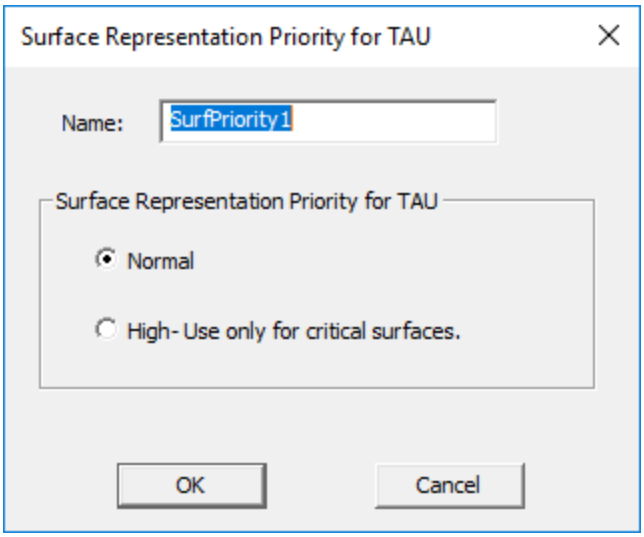
You can choose to specify the surface representation priority for the TAU mesher. For most designs, you can let the solver automatically choose which of two meshing approaches to take. The solver predicts which one gives the best results, balancing mesh reliability, speed, quality, size and design characteristics. 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.

Specifying the Model Resolution

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.

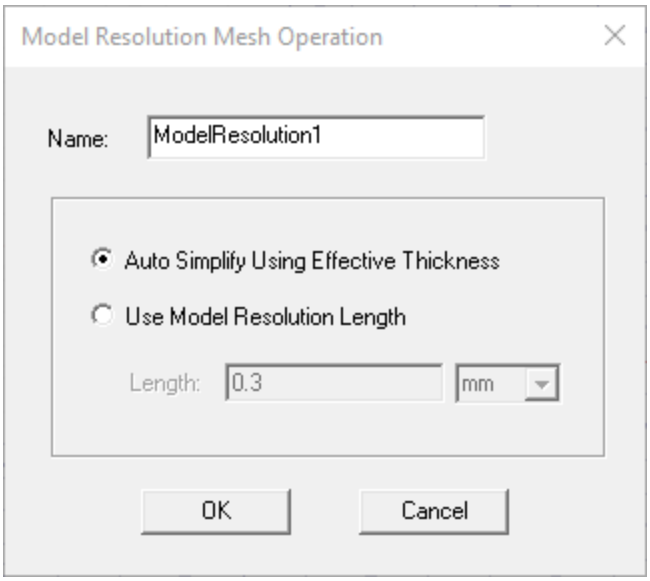
While removing small details, if the meshmaker finds that the representation of the model is not accurate enough, it returns an error condition. The meshmaker then starts with the most accurate representation and prunes away the details smaller than the model resolution length. It returns an error if 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.

Important:

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.

1. Select the object or objects on which to specify a Model Resolution operation.
2. Click on **Q3D Extractor** or **2D Extractor** > **Mesh** > **Assign Mesh Operation** > **Model Resolution** or right click on the **Mesh** icon in the **Project** tree and select **Assign Mesh Operation** > **Model Resolution**.

   The **Model Resolution Mesh Operation** dialog box displays.



   Alternatively, you can display the same dialog box if you:

   a. Right-click either **Mesh** in the Project Tree, or right-click in the **3D Modeler** window to display the respective shortcut menu.
   b. Click on **Assign > Model Resolution** in the **Project Tree** menu or click on **Assign Mesh Operation > Model Resolution** on the shortcut menu.

The **Model Resolution Mesh Operation** dialog box contains text fields for the mesh operation *Name* and radio buttons with choices for the following:

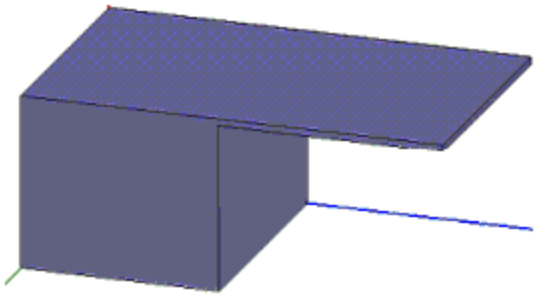
- **Auto Simplify Using Effective Thickness**

  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**:



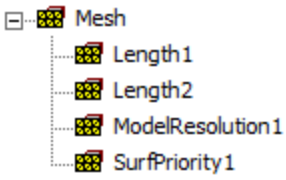
3. After defining the operation, click **OK**.

   This adds the named Model Resolution operation under the **Mesh** icon in the Project Manager.

> **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.

Defining Mesh Operations in 2D Extractor

Mesh operations are optional mesh refinement settings that provide the mesh construction guidance. This technique of guiding mesh construction is referred to as "seeding" the mesh. Seeding is performed using **Mesh** commands. Each mesh operation you define appears in the Project tree:



Note:

There are no restrictions against assigning multiple mesh operations to the same entities. The *type* of mesh operation can differ between overlapping assignments. For example, length-based refinement could be assigned inside a body, and skin-depth seeding on one or more of its faces. Additionally, overlapping mesh operations of the *same type* can have differing, or even redundant, parameter values (such as element length) without causing an error condition.

Typically, the operation that dictates the finest local meshing results will control the outcome in overlapping assignment areas or volumes.

Mesh Generation Process

An initial mesh (which includes surface approximation settings) is generated . If necessary, the mesher automatically performs any repairs needed to recover an accurate mesh representation of a model. The solution profile indicates when mesh repairs have been made, and the results of these repairs are displayed per object in the mesh statistics panel.

Adding a Mesh Operation

To add a mesh operation:

1. In the project tree, right-click **Mesh**.
2. Select the operation you want to add.

Valid operations in 2D Extractor are:

- Length-Based Mesh Refinement – reduce the maximum size of tetrahedral elements and refine the surface length of all tetrahedral elements on a surface.
- Skin Depth Based Mesh Refinement – specify skin depth parameters.
- Surface Approximation Settings – adjust settings for curved surface meshing.

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. This command also allows you to override the automatic choice of mesher used.

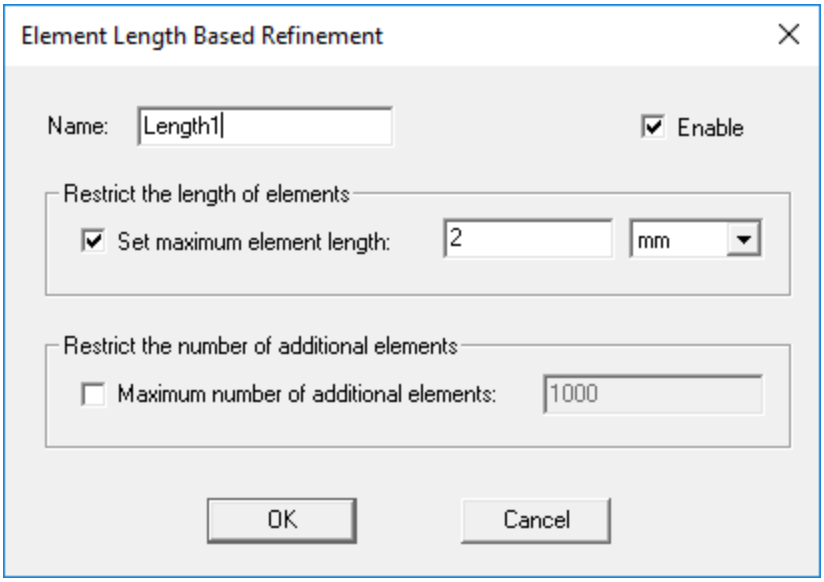
Length Based Mesh Refinement

Length-based mesh refinement settings help control the length of the tetrahedral elements and the growth of the whole mesh. If you are not sure how big a refinement is needed, by way of mesh refinement you can set a ceiling on the mesh growth with the option **Restrict the Number of Additional Elements**.

To apply a length based mesh operation either to an object face or inside an object:

1. Select the object in the modeling workspace.
2. In the Project Manager, right-click **Mesh** > **Assign Mesh Operation** > **[On Selection / Inside Selection]** > **Length Based**.

   The **Element Length Based Refinement** dialog box appears.



3. Enter a **Name** for the mesh operation, or leave the default.
4. Select the **Enable** check box to set the operation as active. You can later deselect this to deactivate it.
5. To restrict the length of tetrahedra edges, select the **Set maximum element length** check box and specify values and units using the drop-down menus.
6. To restrict the number of elements added during refinement, select the **Maximum number of additional elements** check box and specify a value.
7. Click **OK**.

When you set a mesh operation on a selection of objects (more than one) and set a maximum length, it applies to the whole mesh. Even if you selected just one object, when you refine its boundary, elements are added both inside and outside of that object.

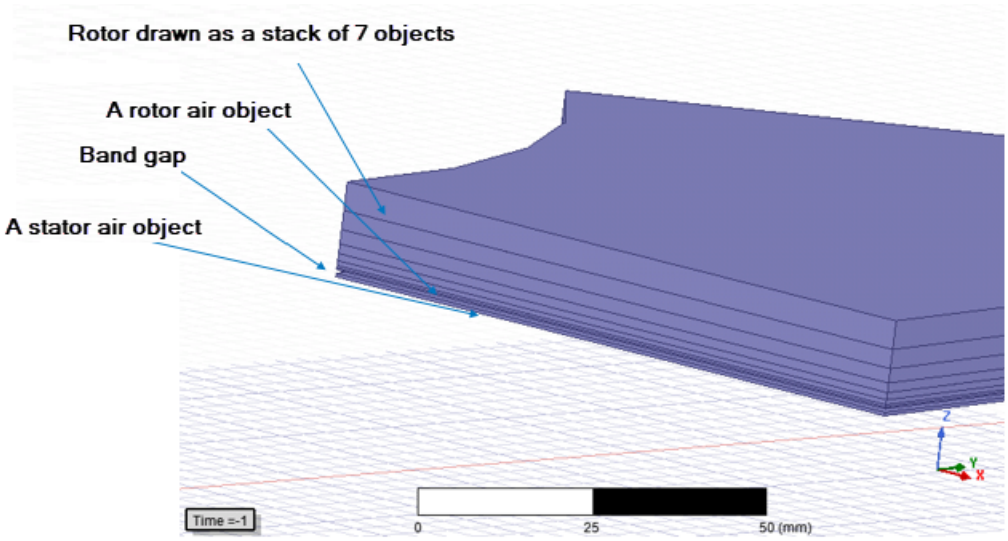
When there are no more elements to refine or when no more elements can be added, the refinement process stops because it has either met the growth limit or the element length target. However, there are more quality improvement swaps, smoothing, matching boundary issues performed at the finishing stage that might slightly alter the mesh size.

If you restrict the number of elements to say 10000, it will add 10000 elements. If you have multiple mesh operations each for 10000 elements, each mesh operation will add 10000 elements (assuming there was no element length targets).

If you set a restriction of say 2000 elements on an object which creates 1100 elements after lambda refinement without any mesh operation, the starting mesh will contain more than 3000 elements.

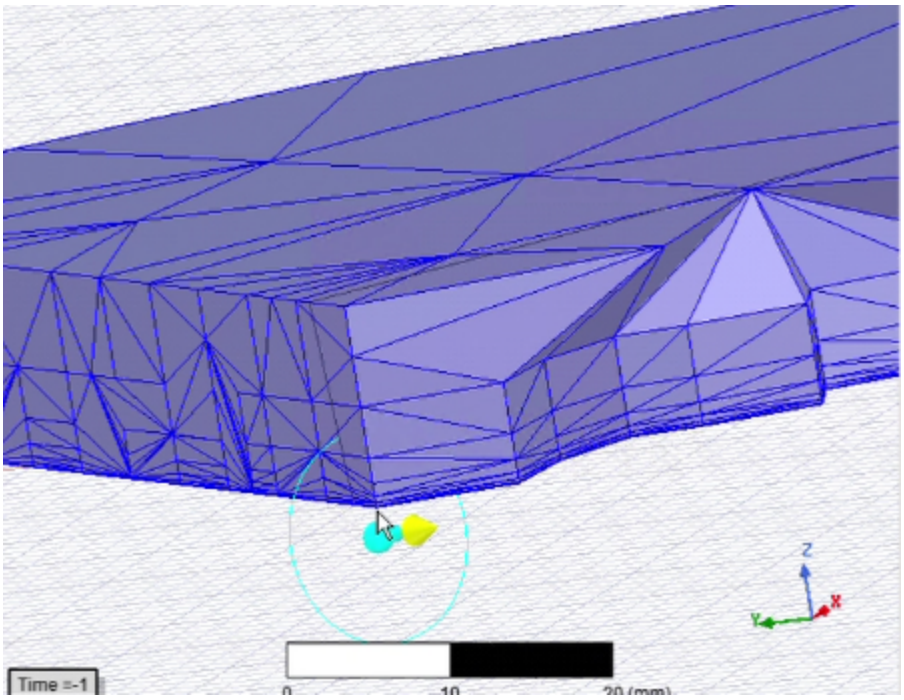
Assigning Skin Depth-Based Mesh Refinement on Object Faces

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:

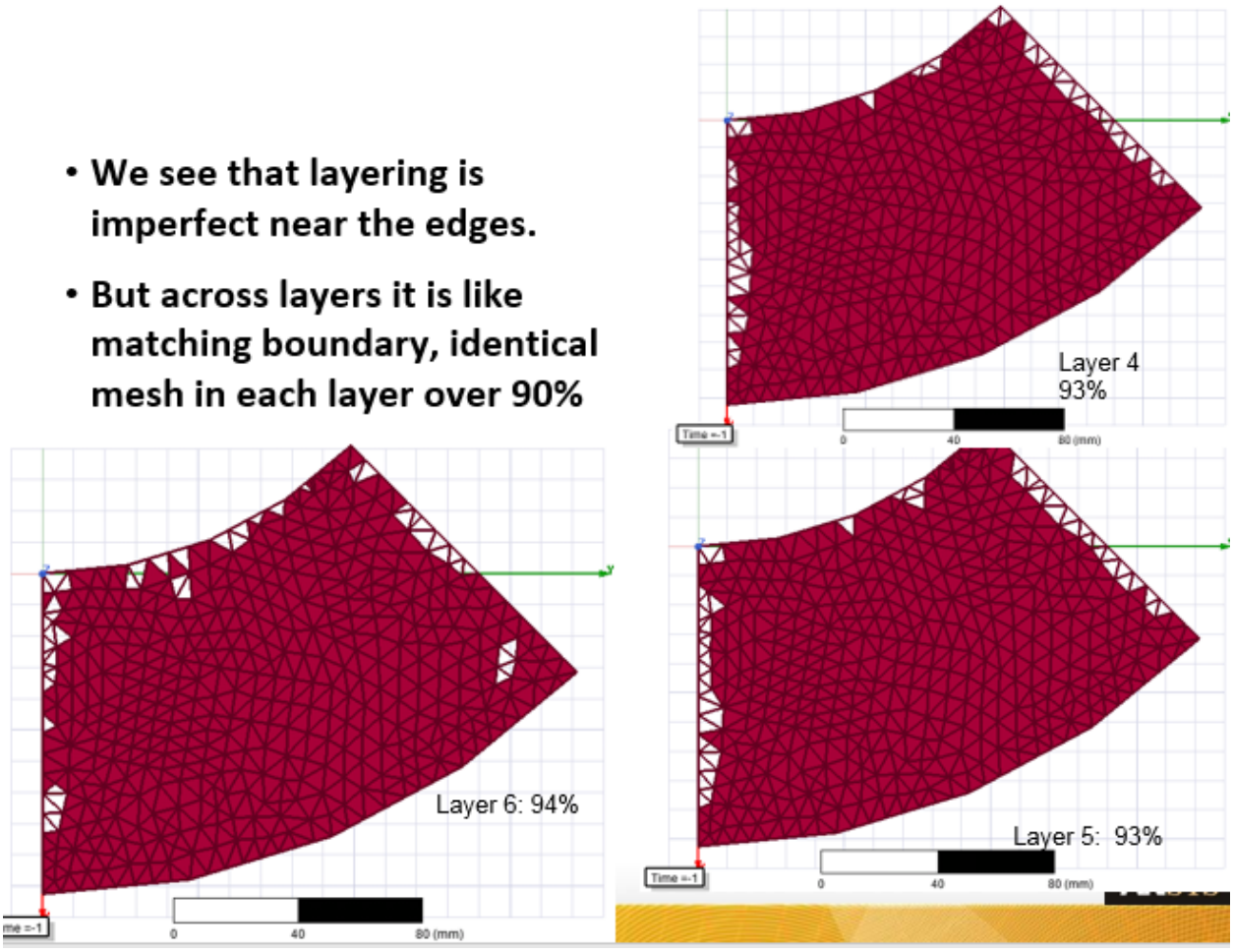


To test various stacking models with pseudo bodies involves recreating the model with different thicknesses and remaking the entire initial mesh. The Tau mesher may not be available for all layer heights. You can achieve comparable results by using skin depth based refinement by reverting to the initial 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.

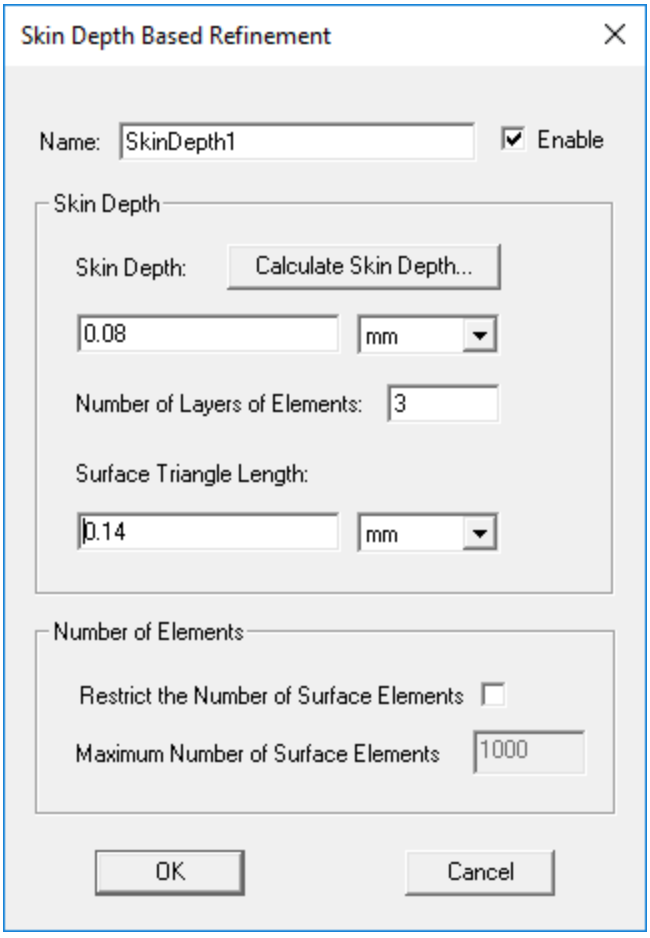
To use Skin Depth-Based Refinement:

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 **2D Extractor** > **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 HFSS 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, unavailable, the number in the box provides an estimate of mesh growth.

When the mesh is generated, the refinement criteria you specified will be 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.

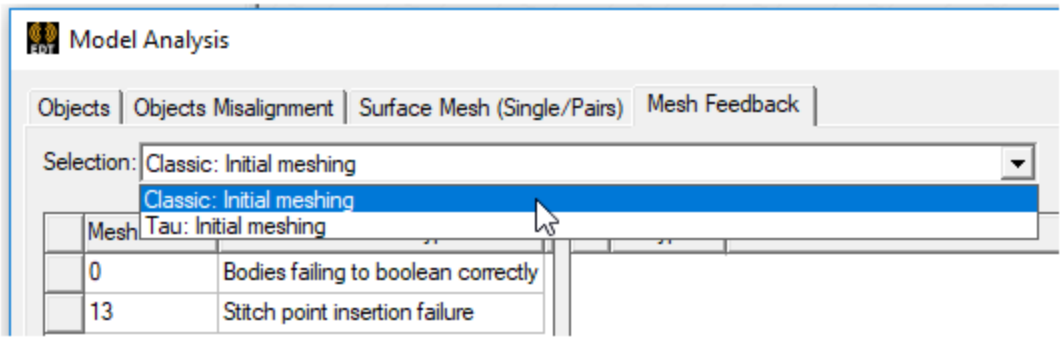
You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

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 box as follows:

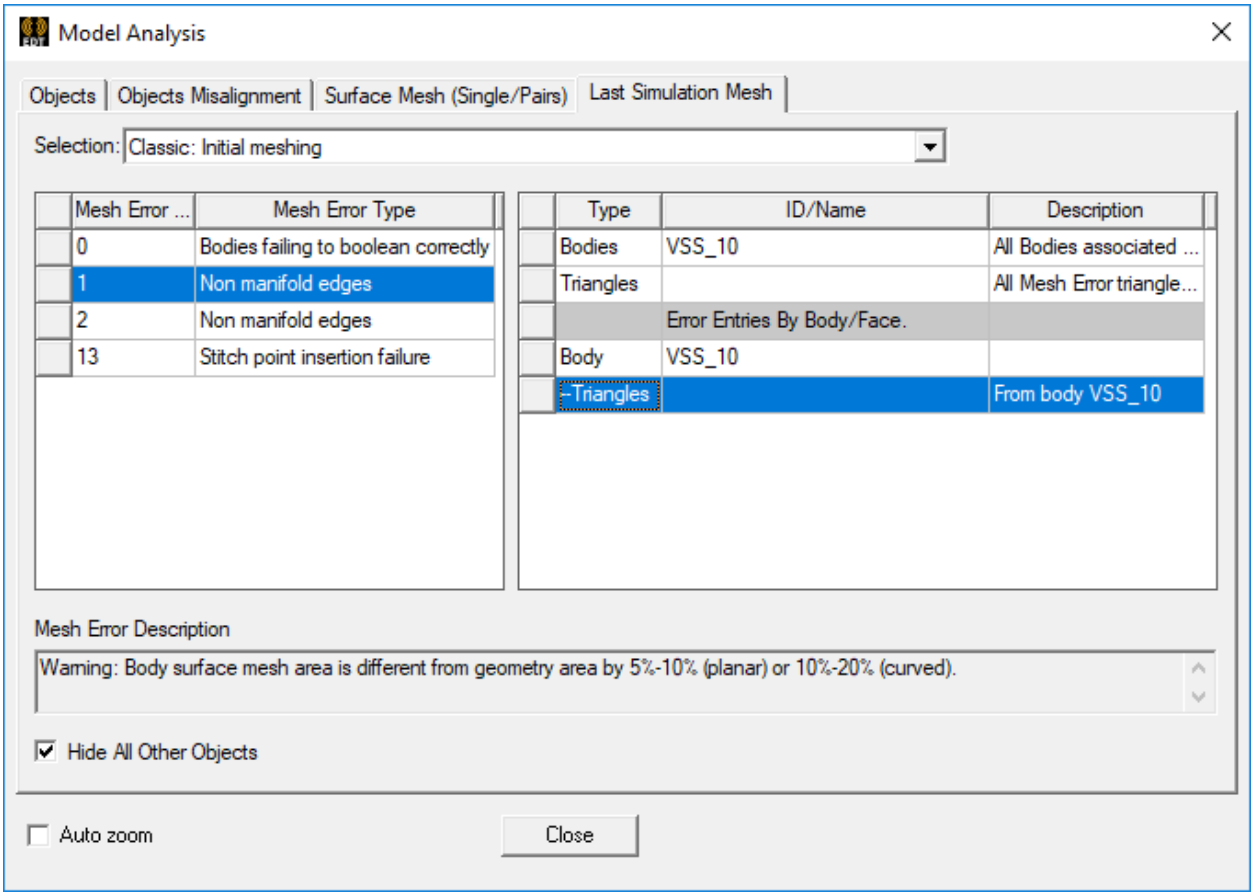
1. Select the object on which you have generated a meshplot, and click **Modeler** > **Model Analysis** > **Show Analysis Dialog** > **View Mesh Feedback**.

   The **Model Analysis** dialog box appears. If the project has solutions by different solvers, the **Selection** drop-down menu lets you 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.



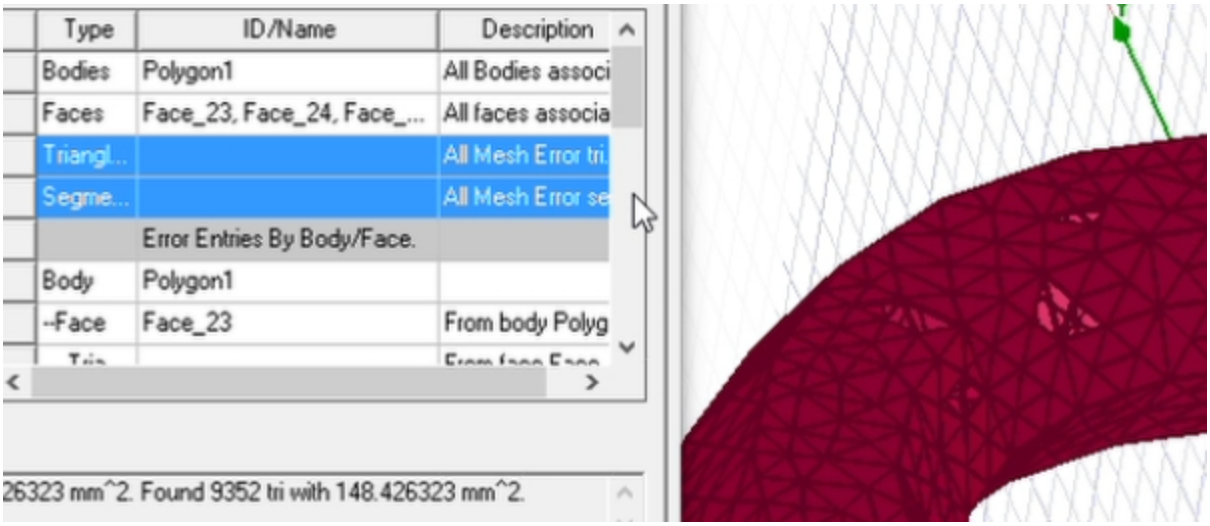
3. On the upper right table, select the body, face, or triangle desired, scrolling the list to find any object, face, triangle, or area of interest.

   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.



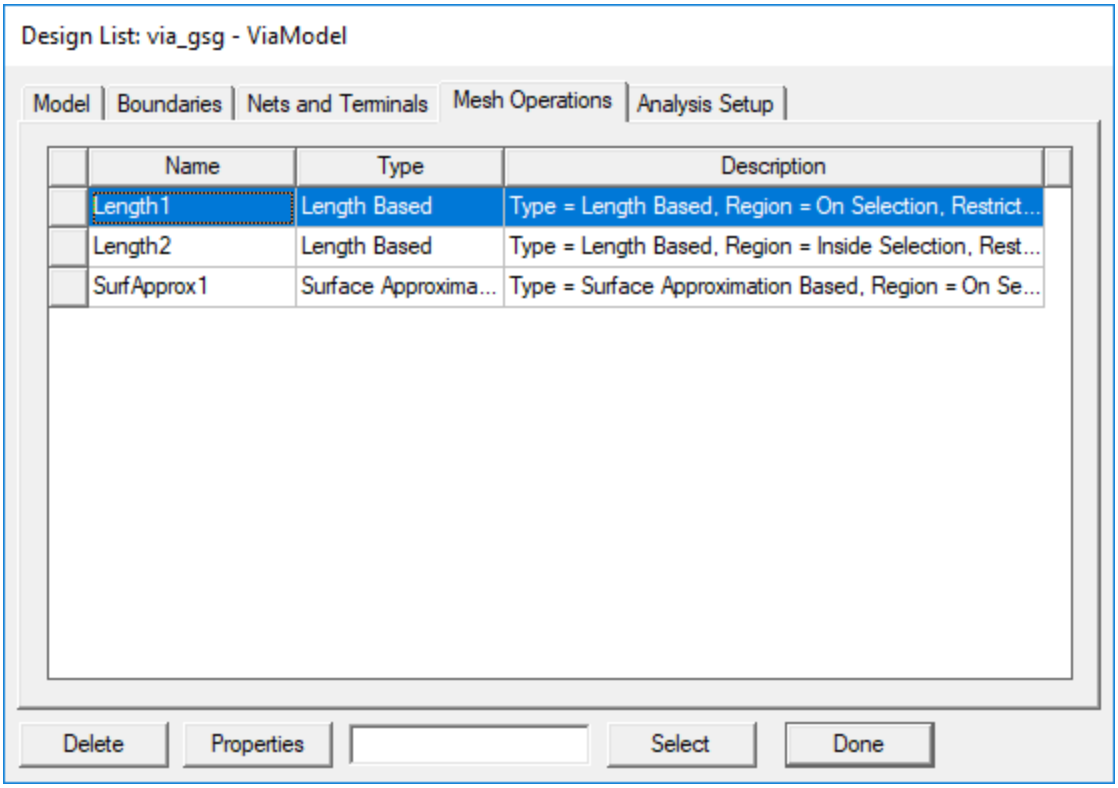
4. Check boxes allow you to **Hide All Other Objects** checked by default, and **Auto zoom to selection**, unchecked by default.

Viewing List of Mesh Operations

To list all mesh operations for the project:

1. In the Project Tree, right-click **Mesh** and select **List**.

   The **Design List** window appears, on the **Mesh Operations** tab.

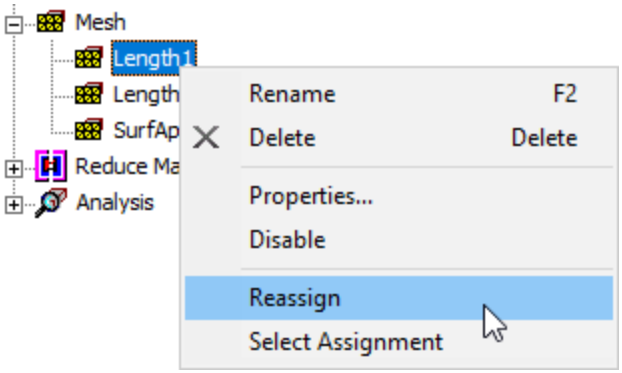


2. Click **Done** to close the window.

Reassigning a Mesh Operation

To reassign a previously assigned mesh setup:

1. Select the object(s) to which you want to reassign the mesh operation.
2. In the Project Tree, under **Mesh**, right-click the mesh refinement you had previously defined and select **Reassign**.

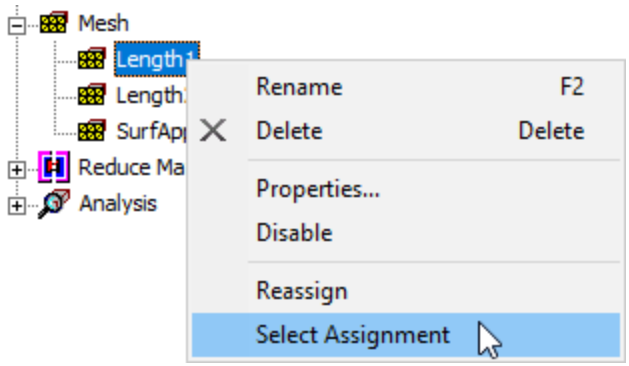


The mesh operation is reassigned to the currently selected objects.

> **Note:**
>
> This pertains to mesh setup operations only. If you want to see information about a mesh that has been generated, see Viewing Mesh Statistics.

To view the objects to which a mesh operation is assigned:

1. In the Project Tree, select the mesh operation.
2. Right-click the previously completed mesh refinement and select **Select Assignment**.



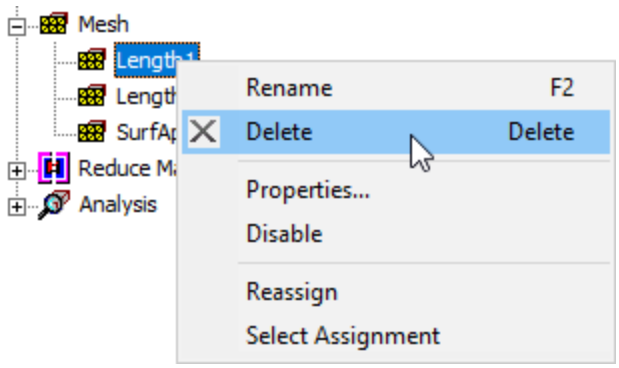
The objects to which the operation is assigned are selected in the modeling workspace.

Deleting Previously Assigned Mesh Refinements

You can delete an individual mesh refinement, or delete all previously assigned mesh refinements.

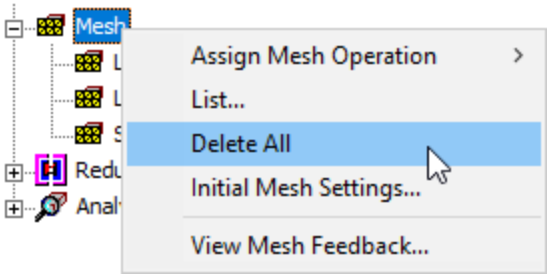
To delete a single refinement:

- In the Project Manager, right-click the mesh refinement and click **Delete**, or press the **Delete** key.



To delete all previously assigned mesh refinement data:

- Click **Q3D Extractor** or **2D Extractor** > **Mesh** > **Delete All**.
- In the Project Manager, right-click **Mesh** and select **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.

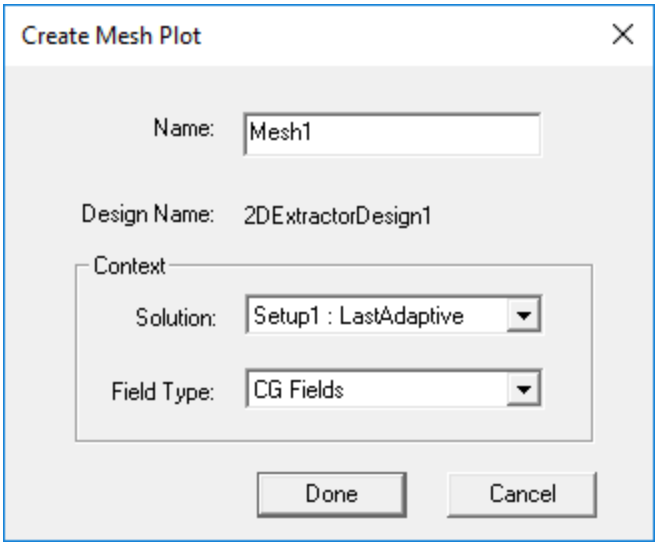
Plotting the Mesh

Before or after the solution is complete, you can plot the finite element mesh on surfaces or within 3D objects. For an HFSS or Maxwell design that includes Layout Components, you can also select nets and layers from the layout components to plot.

1. Select one or more surfaces or objects to create the mesh plot on or within. If the design is HFSS and includes Layout Components, you can select the surfaces either before or after selecting **Plot Mesh**.

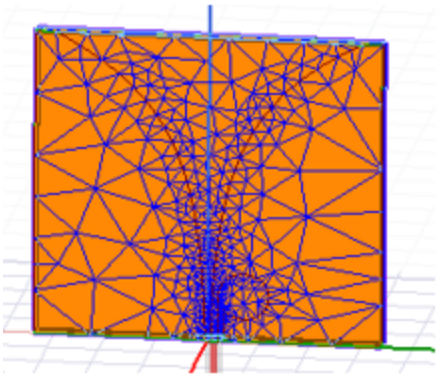
2. Select **Plot Mesh** in one of two ways:
   - Click **[Q3D Extractor / 2D Extractor]** > **Fields**> **Plot Mesh**.
   - In the Project Manager, right-click **Field Overlays** and select **Plot Mesh...**

   The *Create Mesh Plot* dialog box appears.

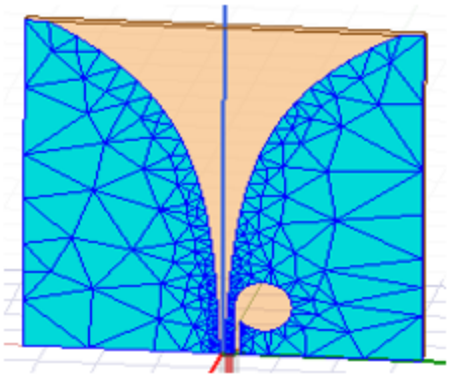


3. Select the **Solution** to use from the drop-down menu.
4. If **Field Type** is enabled, you can select from the drop-down menu for a mesh plot.
5. Click **Done**.

   The mesh appears on the surface or object you selected. For example:

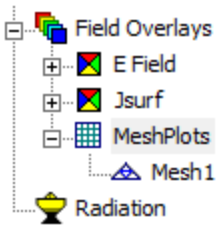


Mesh Plot of All Objects



Mesh Plot of Selected Surfaces

An icon for the mesh also appears in the Project Manager 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 select **Update Plots**. This updates the mesh plot to include the latest data available. After the last adaptive pass, the mesh plot is automatically updated.

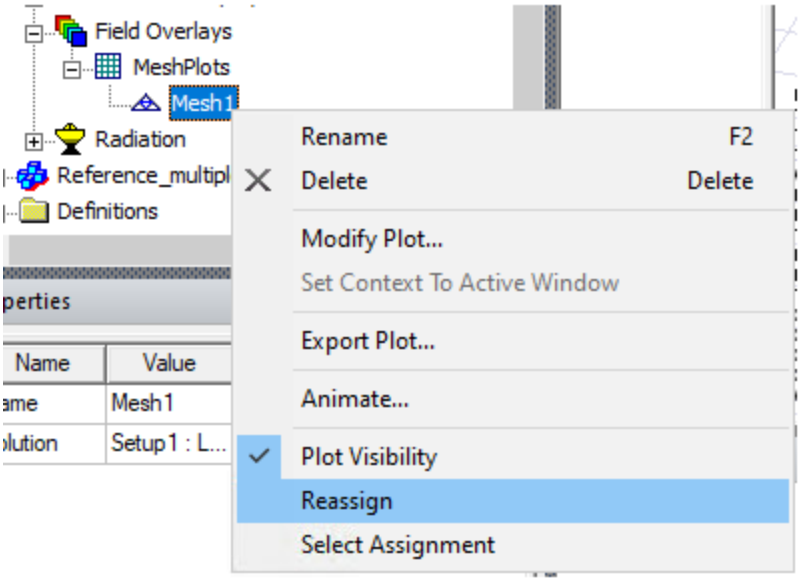
If a mesh includes seeding, these effects appear at the start of the adaptive passes. Any mesh adaptation at the start of a subsequent pass is not plotted until after that pass is completed. This delay ensures that mesh plots and actual solutions remain consistent with each other.

You can modify an existing plot by selecting the plot and changing the properties.

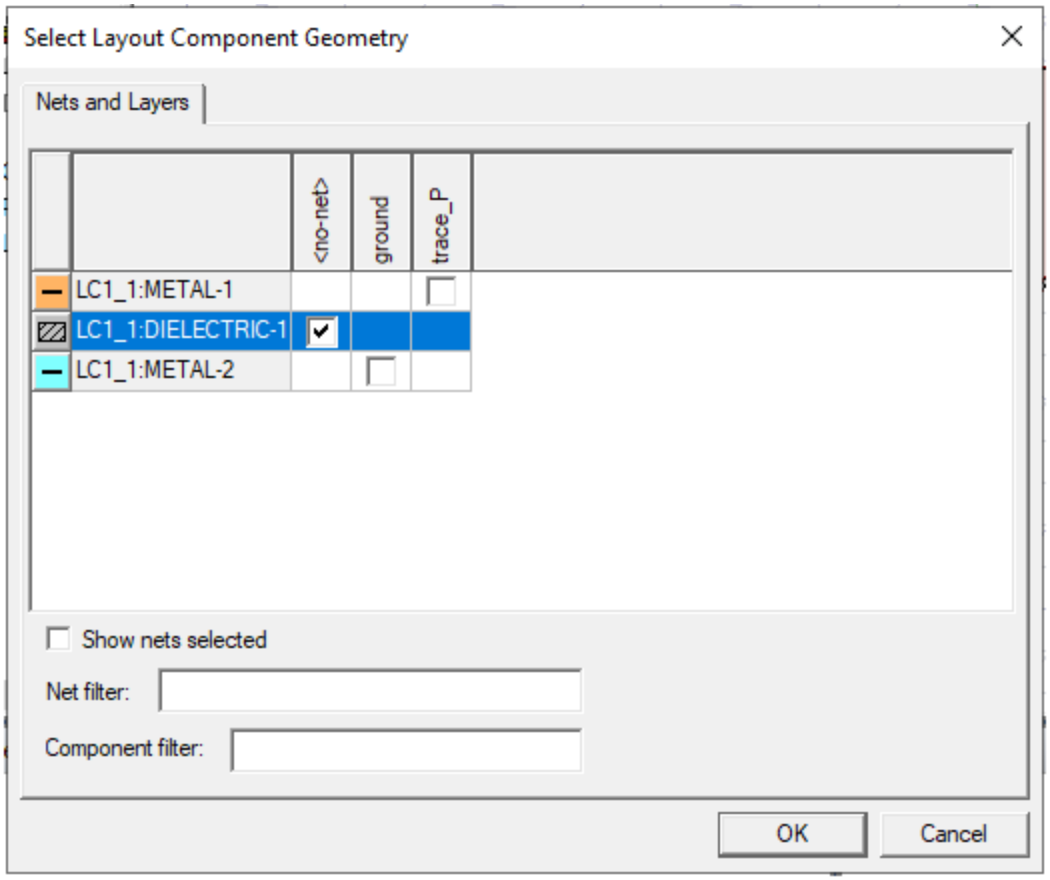
If you modify the geometry of an object, the solution is invalidated and the mesh plot is removed.

After making selections click **Done**. The mesh plot will be generated on the objects belonging to the selected layer/net combinations.

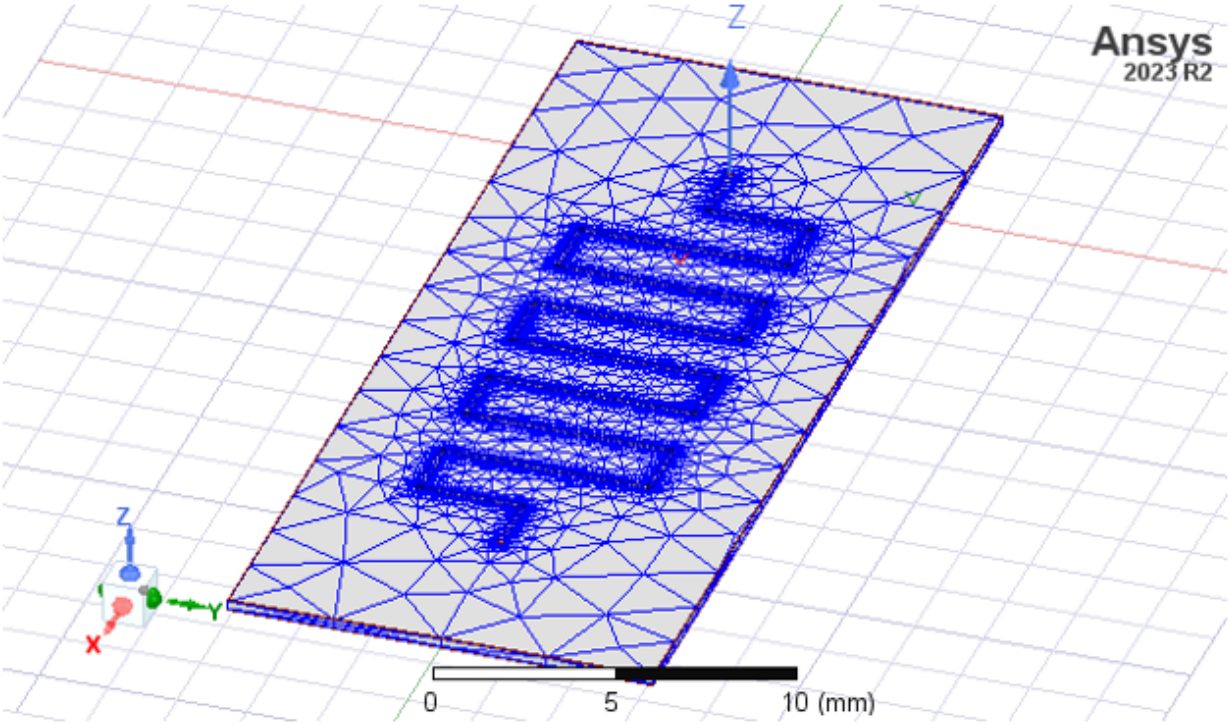
If you want to reassign a mesh or field plot, either select 3D geometries from layout components in the modeler window before selecting **Reassign** in the right-click menu, or select **Reassign**.



For HFSS designs with layout components, the **Select Layout Component Geometry** dialog appears. This has the same functionality and menus as the this **Nets and Layers** section of the **Create Mesh Plot** dialog.

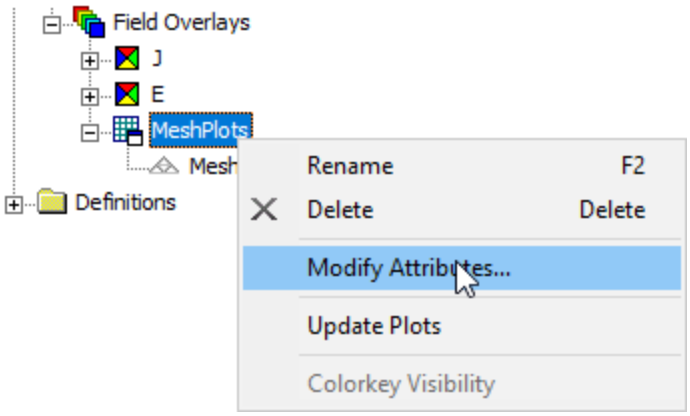


Change the selection in the dialog and click **OK**. The mesh plot will be re-assigned to the newly selected objects, in addition to any 3D geometries previously selected from the modeler window. If you click cancel, the plot is only re-assigned on 3D geometries previously selected. If there are no previously selected 3D geometries, the program issues an error message. Change the selection in the dialog and click **OK** to display the reassigned mesh plot.

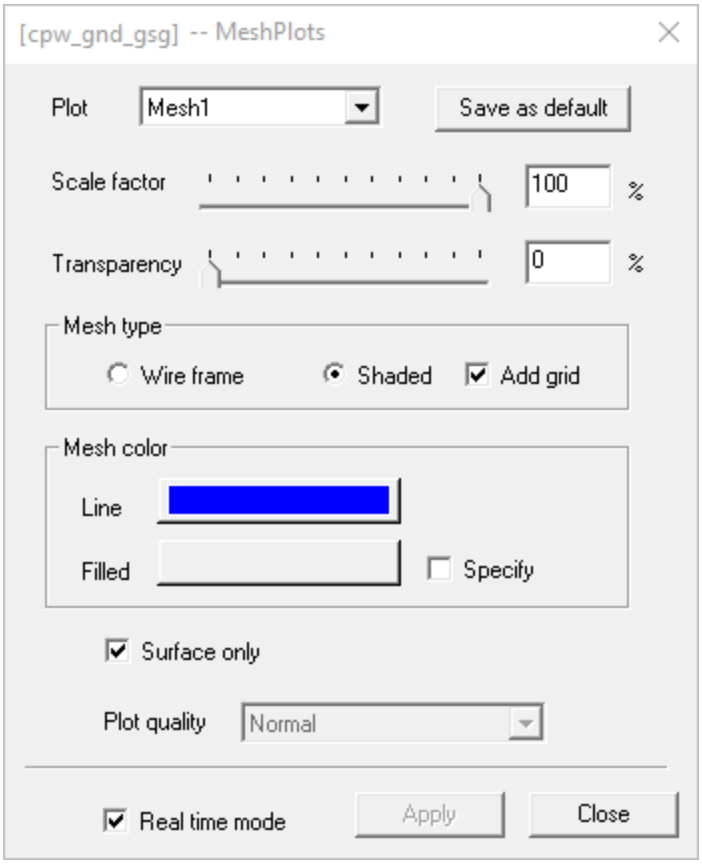


Setting Mesh Plot Attributes

1. In the Project Tree, expand **Field Overlays**.
2. Right-click **Mesh Plots** and select **Modify Attributes**.



The *MeshPlots* window appears.



Note:

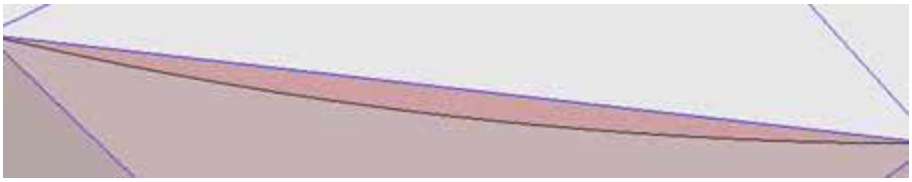
You can also open the MeshPlots window from **Q3D Extractor** > **Fields** > **Modify Plot Attributes**. You will be prompted to select the folder containing the mesh you want to modify.

For Mesh plots, the following attributes can be modified:

Plot	A drop-down menu of available plots.
Scale Factor	The size at which the tets 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 or decrease the size of the tetrahedra.

Transparency	The degree of transparency for the tets. This is useful for viewing objects or plots behind the current plot. Use the **Transparency** slider to increase or decrease plot transparency.
Mesh Type	Whether to display the tets as wire frame or shaded, and whether to Add Grid.
Mesh Color for Line and Fill	The color for the tet edge lines and fill. Clicking the button for each displays a color selection dialog box. You must select **Specify** to enable fill color.
Surface Only	Whether to plot the surface only, or all tets inside selected objects.
Plot Quality	The resolution quality of the plot. Normal by default. Use the drop-down menu to select Coarse, Normal, Fine, or Very Fine. The higher the resolution, the more memory used.
Real Time	Whether to show changes to a mesh immediately in the view window. If this option is cleared, click **Apply** when you want to see the changes.

For **Plot Quality**, higher quality selections permit visualization of curvilinear mesh elements. With the **Coarse** setting, facets do not snap to the midpoint edge:



The **Normal** setting displays snapping to the midpoint edge:

The **Fine** setting displays more of the curvilinear tets:

The **Very fine** setting gives the most accurate display, but uses the most memory:

3. Click **Save as default** button if you want the tab's settings to apply to mesh plots created after this point.
4. Click **Close**.

Generating Mesh without Solving

If you want to refine the mesh on a face or volume but do not want to generate a solution, define mesh operations. Then do one of the following:

- Select [**Q3D Extractor**] > **Analysis Setup** > **Generate Mesh**.
- Right-click the **Analysis** or **Setup** icon in the **Project** window 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. If you specified "Prompt for analysis machine when launching analysis" under **Tools** > **Options** > **General Options** > **Analysis Options**, the **Server Setup** dialog box appears and you can specify a solve machine.

- If a current mesh has been generated, the solver will refine it using defined mesh operations.

- If a current mesh has not been generated, the solver will apply the mesh operations to the initial mesh.
- If an initial mesh has not been generated, the solver will generate it and apply the mesh operations to the initial mesh.
- If the defined mesh operations have been applied to the selected face or object, the current mesh will not be altered.

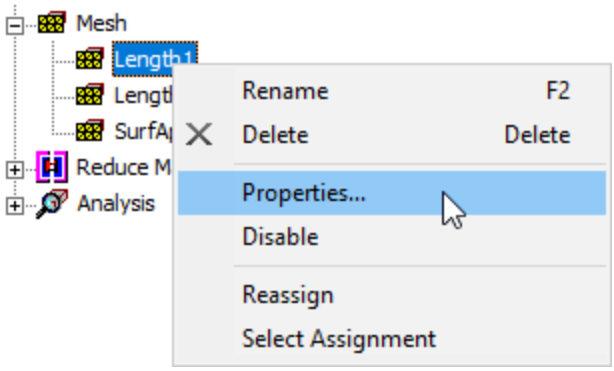
> **Tip:**
>
> Define a new mesh operation rather than modify an existing mesh operation. The solver will not re-apply a modified mesh operation.

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 them or by reverting to the initial mesh.

Viewing Mesh Properties

To view mesh properties:

- In the Project Tree under **Mesh**, right-click a previously completed mesh refinement and select **Properties**.



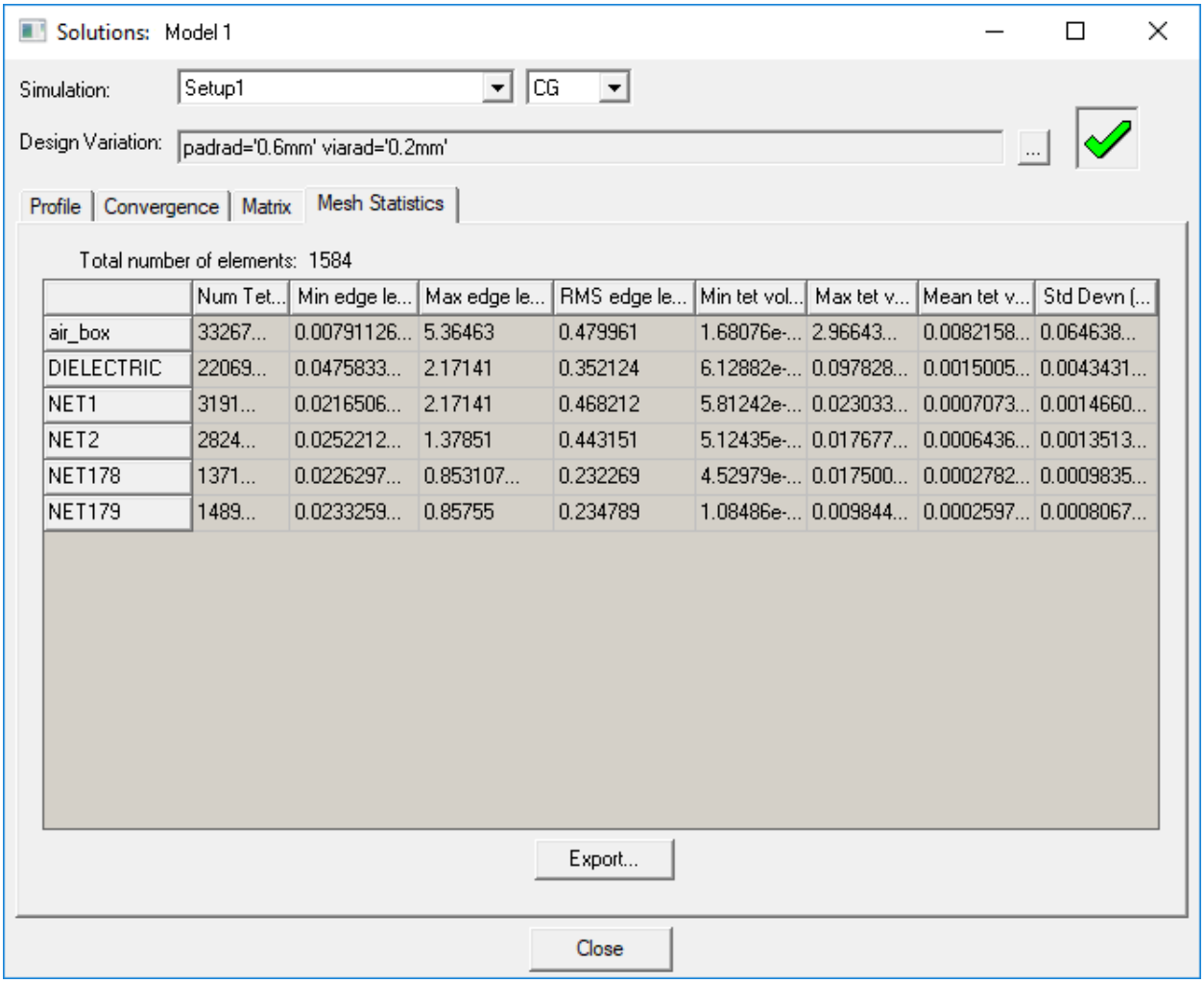
A dialog box opens with properties specific to that mesh (length-based, surface approximation, etc.). You can edit the mesh settings as appropriate.

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**. Alternately, click the **Solution Data** icon on **Results** tab of the ribbon.

   The **Solutions** window appears with the **Mesh Statistics** tab displayed.



The table lists design elements and for each includes: Num Tets, 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 **Repaired %**. These columns indicate the fraction of an object that was successfully recovered, and the fraction that needed some repair.

To sort mesh statistics by column from low to high values or vice-versa, click a 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 order.

Click on a blank cell above the object list to invert the order of objects. In this case, the cell does not display a directional triangle.

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, use the 3D component libraries, the Antenna Design Toolkit, the SBR+ Parametric Antenna feature, as well as the Cable Modeling toolkits.

To open a new **3D Modeler** window, do one of the following:

- Insert a new design into the current project.
- Double-click an Ansys Electronics Desktop design that uses the 3D Modeler (HFSS, Q3D, Maxwell (2D and 3D), or Icepak) in the project tree.

If a **3D Modeler** window for an existing design is not open, do one of the following:

- Click **[solverName]** > **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 Ansys Electronics Desktop is running and one or more modeler windows are open, those modeler windows automatically close. The message manager window displays a message indicating that Ansys Electronics Desktop 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 **[solverName]** > **3D Model Editor** to focus the modeling window on the selected design.

If the menu command is unavailable, then the selected design is already in the modeler window.

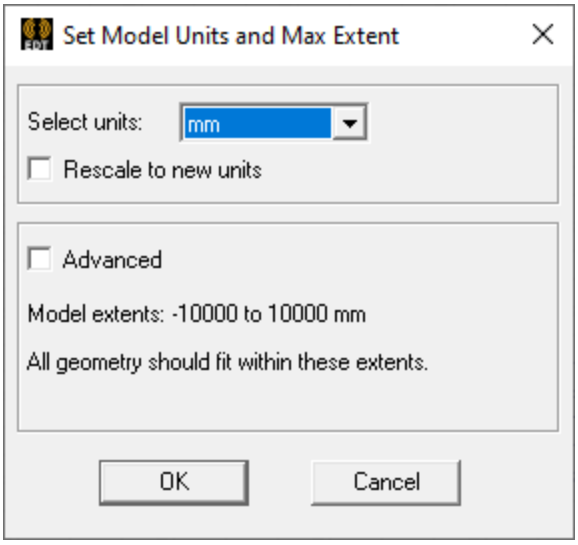
Setting Units of Measurement for the Model

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:

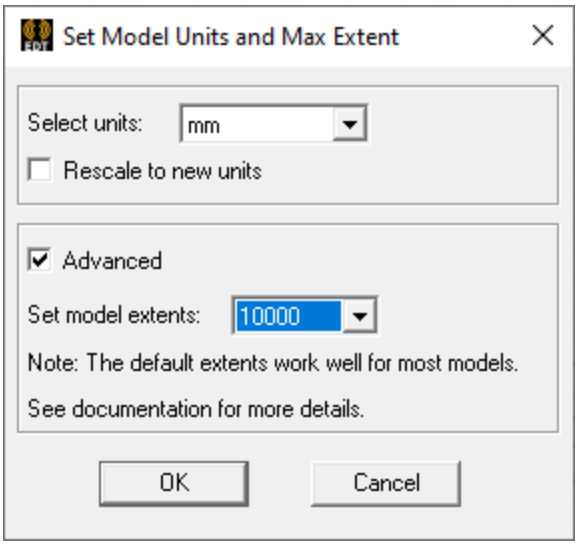
1. Click **Modeler** > **Units**.

   The **Set Model Units** dialog box appears.



2. Select the new units for the model from the **Select units** 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.

4. While most models should fit the default, you can check Advanced to enable Max model extent (E) 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.

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 curve, equation based surface, 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.

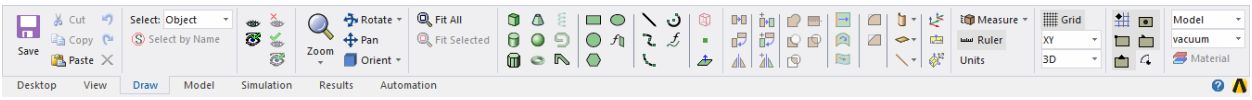
You can also edit the properties of a polyline from the history tree to assign it a Cross Section property as line or rectangular. If you then assign it either a height or a width, the polyline becomes a sheet object.

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.

You can access the **Draw** commands via icons on the **Draw** tab ribbon:



By default, the history tree organizes 3D objects by material. To change this, select the **Objects** icon, and right-click to display the shortcut menu to see **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 the Drawing Plane
- Select the Coordinate System and choose Cartesian, Cylindrical or Spherical coordinates.
- Change Snap Settings

- Set Reference Point for the movement mode
- Adjust the View

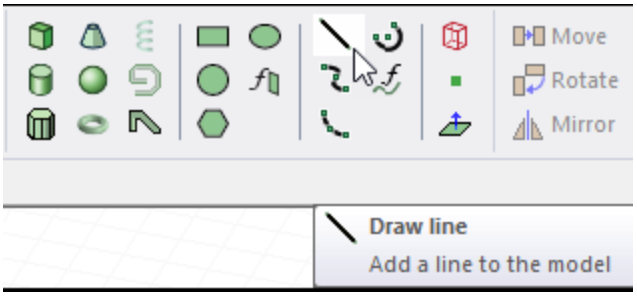
After you draw an object in the **3D Modeler** window, you can modify the object's properties, such as its position, dimensions, or color, in the **Properties** dialog box. Most model object properties can be assigned as Design variables when can then be manipulated during the solve to test their effect on the solution. For non-model objects, you can use Post Processing variables (default and Design variables,.

> **Note:**
>
> If you access your machine via Remote Desktop, if Ansys Electronics Desktop is running and one or more modeler windows are open, those modeler windows automatically close. The message manager window displays a message indicating that Ansys Electronics Desktop closed the modeler windows.

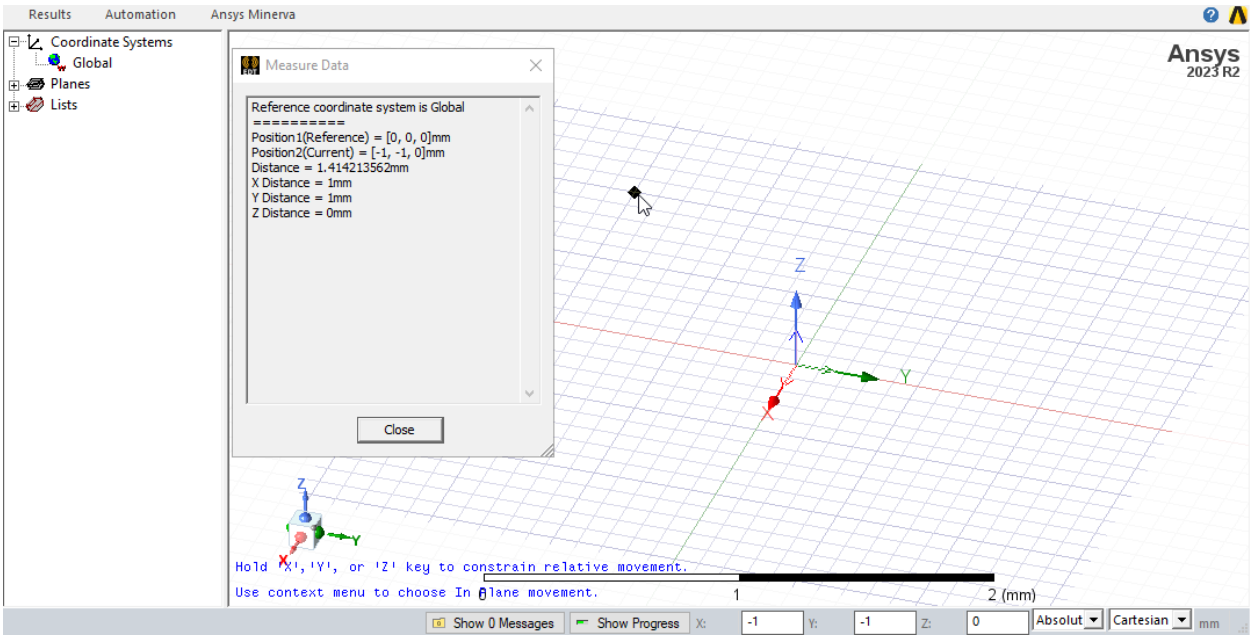
Drawing a Line Segment

To create an object with one or more straight line segments, use the **Draw** > **Line** command.

1. From the menu bar, click **Draw**> **Line** or, on the **Draw** ribbon tab, click the **Draw line** icon:



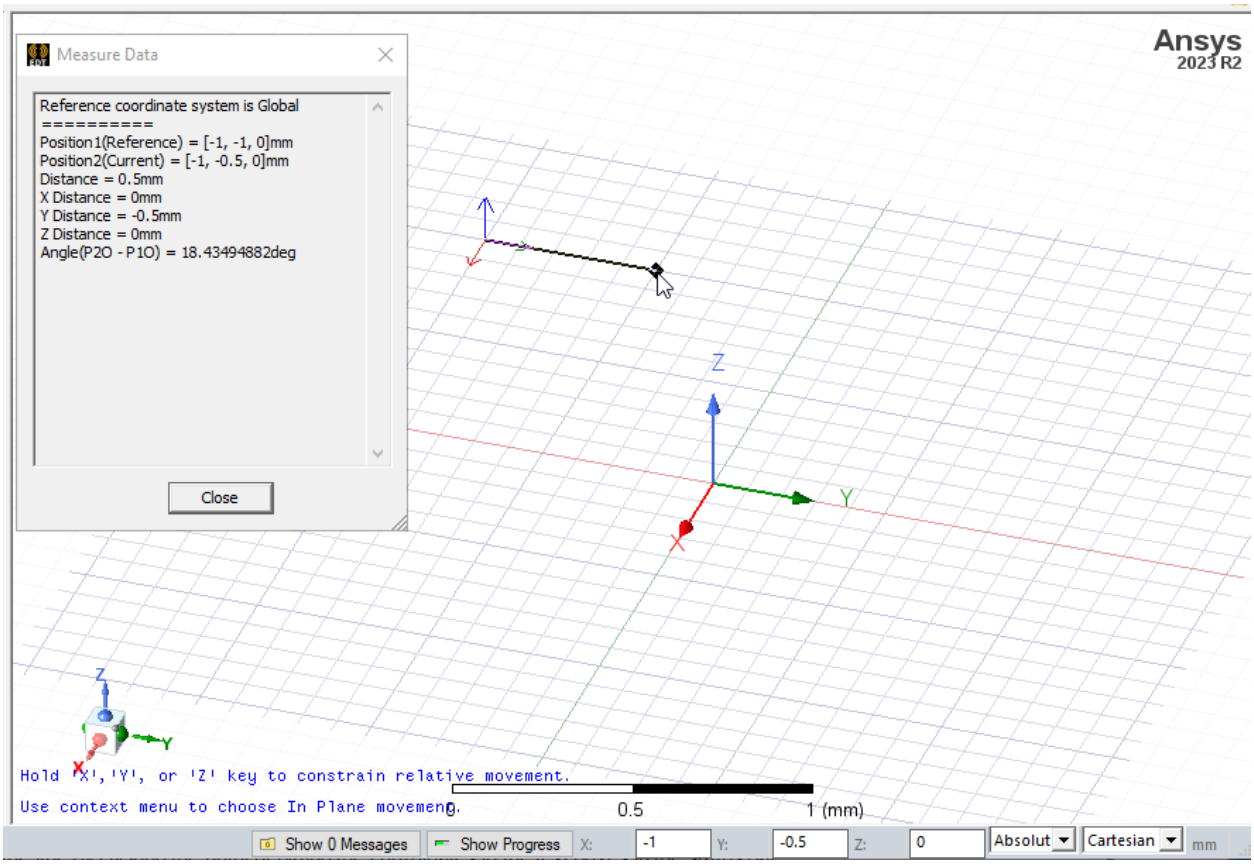
2. Select the first point of the line in one of the following ways:
   - Click the point in the Modeler window, checking the Measure Data dialogue for the precise coordinates.



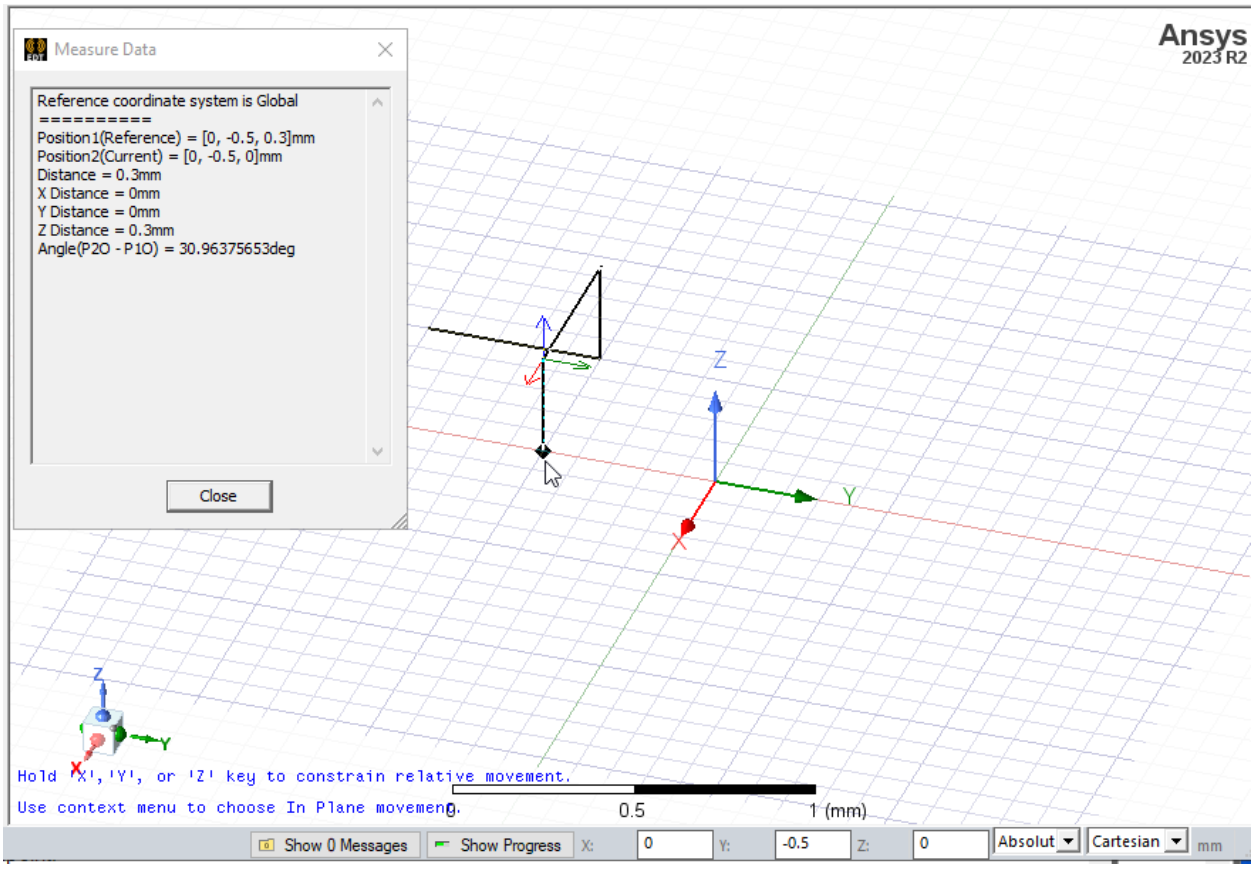
- You can accept the first point or change it by editing its coordinates in the X, Y, and Z boxes in the status bar. Note that the Status bar includes selections for Absolute or Relative Coordinates, as well as for Cartesian, Cylindrical and Spherical Coordinates.

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 next point of the line by clicking the point or typing the coordinates in the text boxes in the status bar.



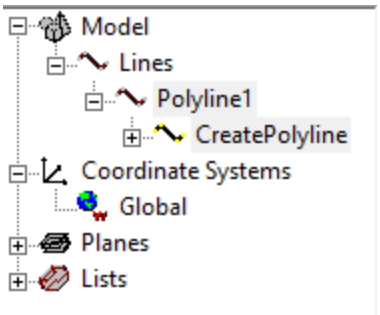
The endpoint serves as the start point for a subsequent line segment. You can specify the plane through which the line passes by selecting the X, Y, or Z keys to constrain movement. For example, the following figure shows a line with five points that includes X, Y, and Z movements.



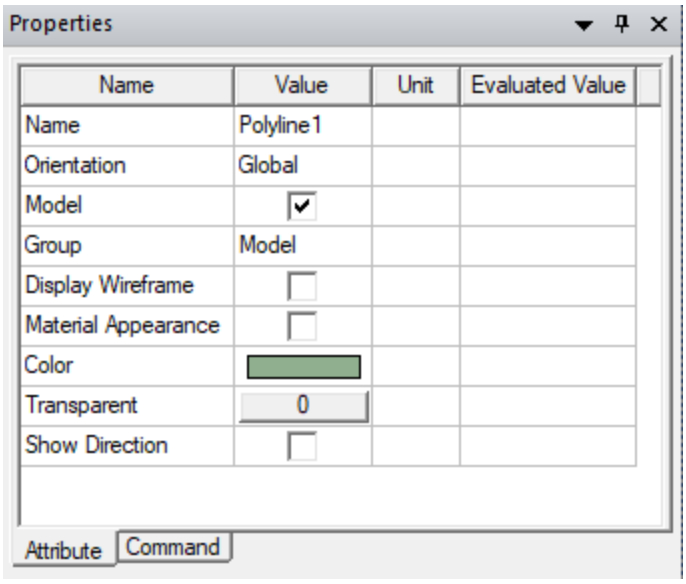
To delete all points and start over, press **Esc** or click **Escape Draw Mode** on the shortcut menu.

4. Complete the line in one of the following ways:
   - Double-click the endpoint.
   - Click **Done** on the context (right-click) menu.
   - Press **Enter**.

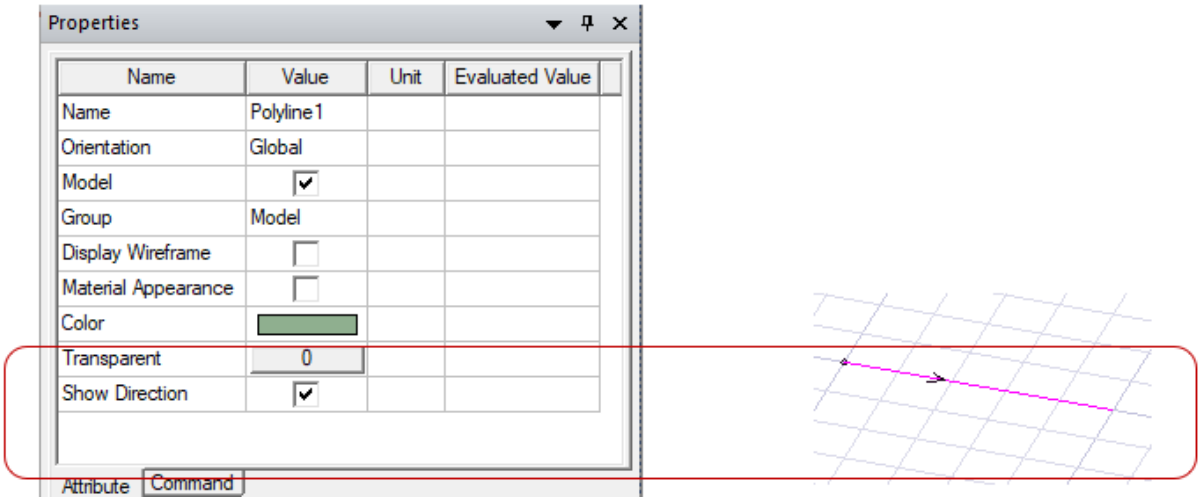
The new line appears selected in the Modeler window and shown in the History tree.



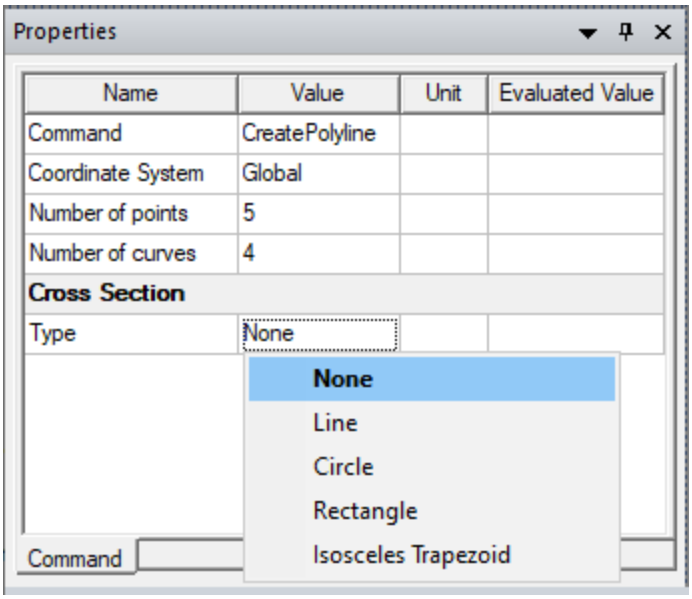
If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the object's properties. You can also do this through the docked Properties window.



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. The Command tab **Properties** includes the Coordinate System that applies to the line, the number of points and curves, as well as the ability to assign a specific Cross Section to the line.



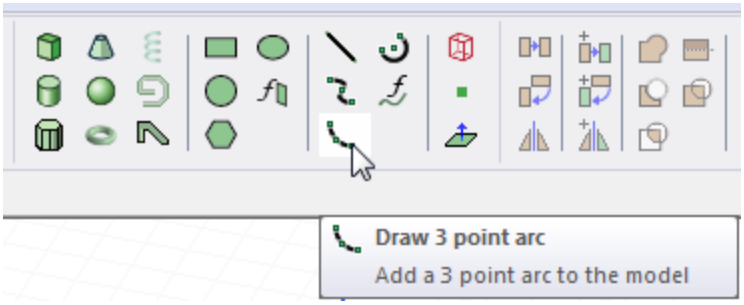
6. If you created the line with the Modeler option for editing properties of new primitives, click **OK** to close the **Properties** dialog box.

> **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.

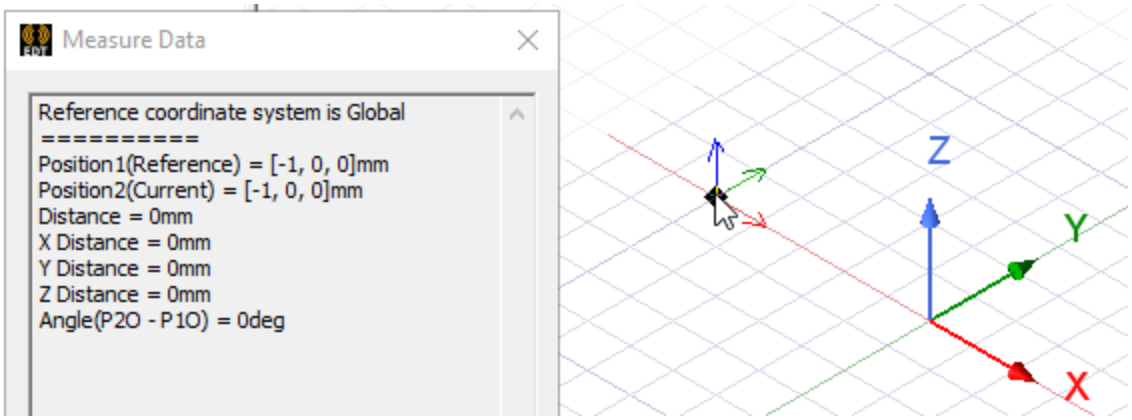
Drawing a Three-Point Arc Line

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. Before you draw a three-point line arc, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

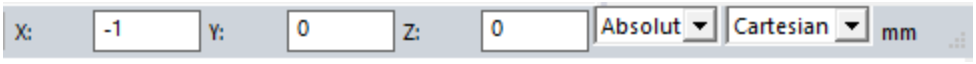
1. From the menu bar, click **Draw**> **Arc**> **3 Point** or, on the **Draw** ribbon tab, click the **Draw 3 point arc** icon:



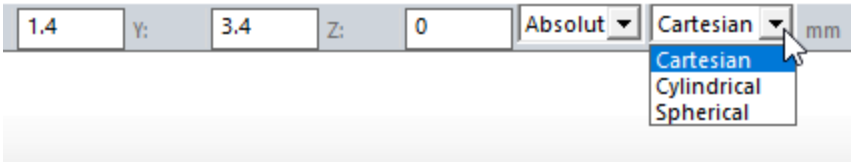
2. Select the start point of the arc in one of the following ways:
   - Click the point.



   - Type the point's coordinates in the **X, Y,** and **Z** text boxes.



     The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the line in Cartesian, Cylindrical, or Spherical coordinates.



3. Select the midpoint of the arc by clicking the point or typing the coordinates in the 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 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 by editing the Properties. 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**.

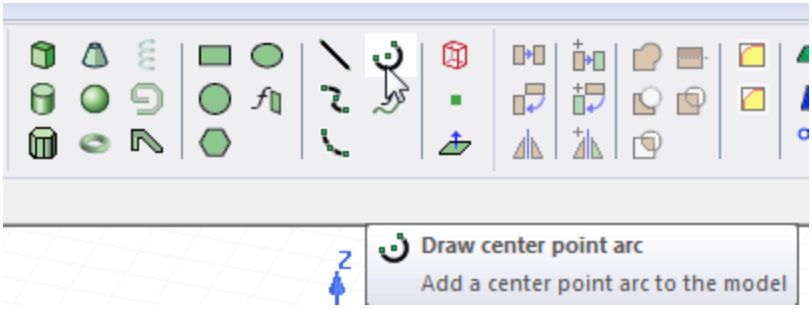
   Based on the three points you specified, the modeler calculates the center point and radius of the arc and draws an arced line through the three points.

> **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.

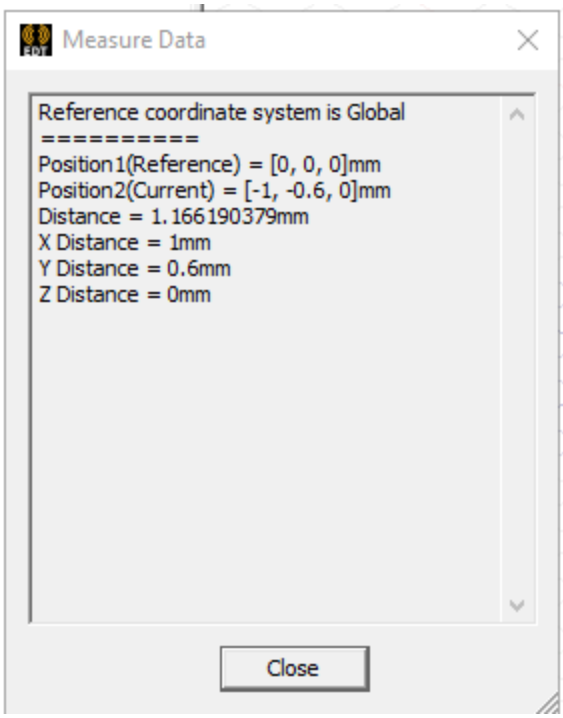
Drawing a Center-Point Arc Line

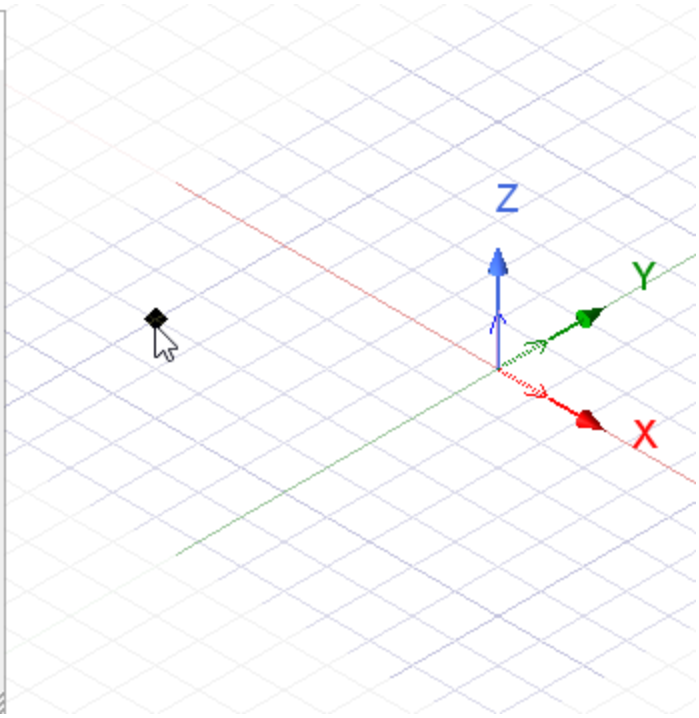
In the modeler, a center-point arc line segment is an arced line defined by a center point, start point and angle. Use the **Draw** > **Arc** > **Center Point** command to create a polyline object with one or more center-point arc line segments. Before you draw a center-pint arc line, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Arc**> **Center Point** or, on the **Draw** ribbon tab, click the **Draw center point arc** icon:



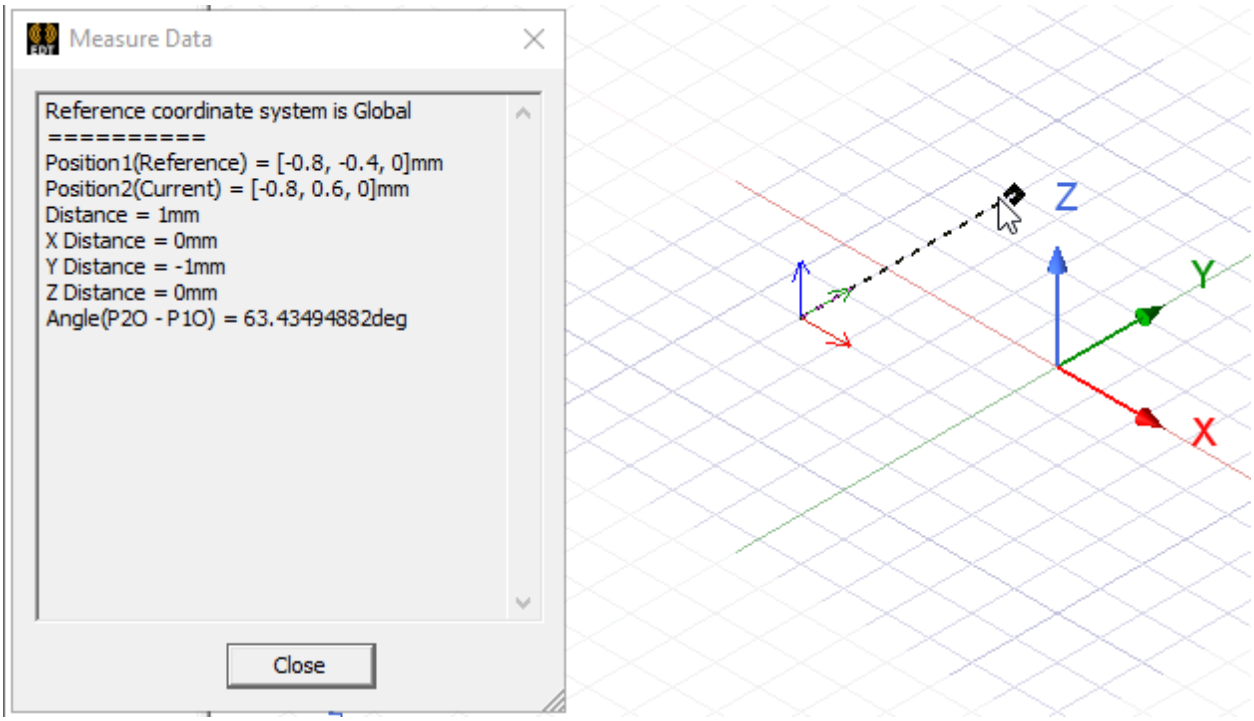
2. Select the center point of the arc in one of the following ways:
   - Click the point.





   - Type the point's coordinates in the text boxes in the status bar.

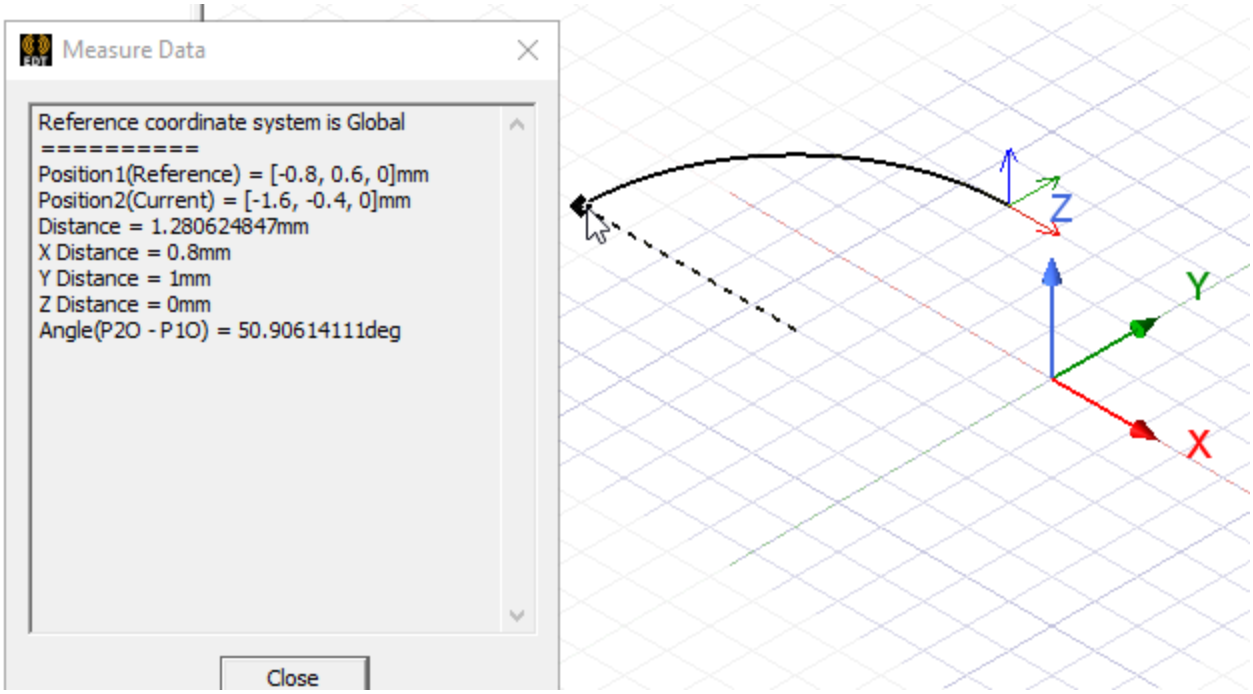
3. Select the start point, or radius, of the arc by clicking the point or typing the coordinates in the text boxes in the status bar.



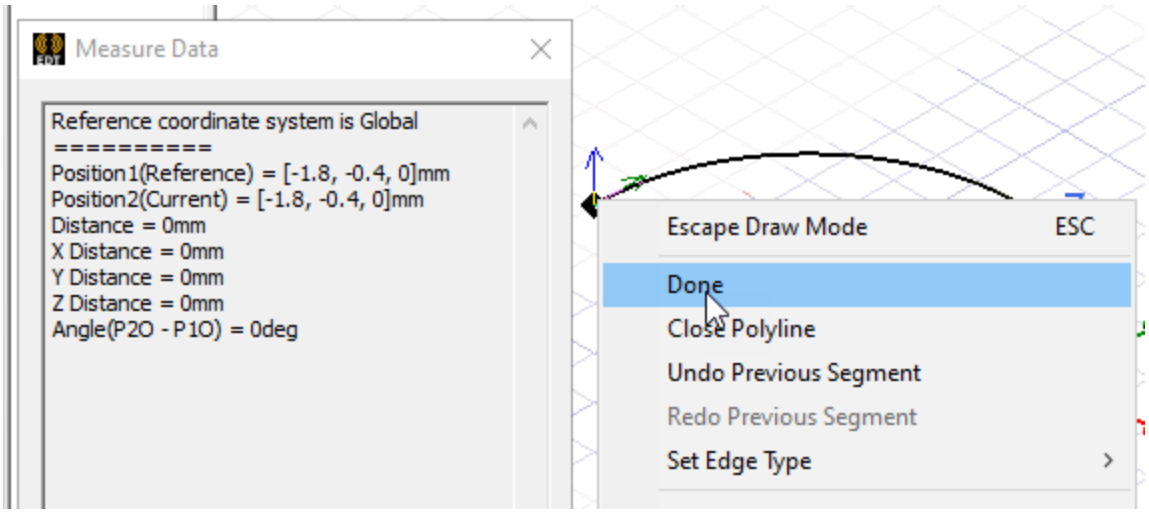
To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

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 text boxes in the status bar.

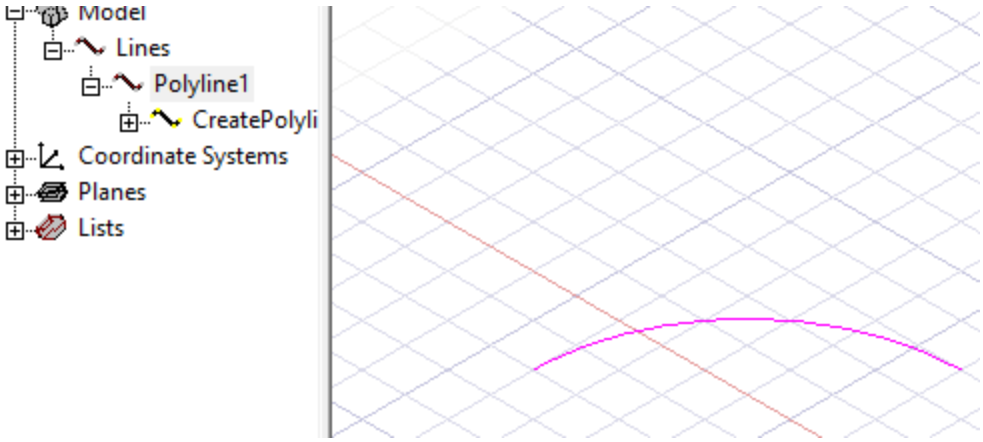


5. If the endpoint is the last point of the polyline object, double-click the point to complete the polyline or click **Done** on the shortcut menu.



If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's attributes. The new polyline

appears in the modeler window and in the History tree.



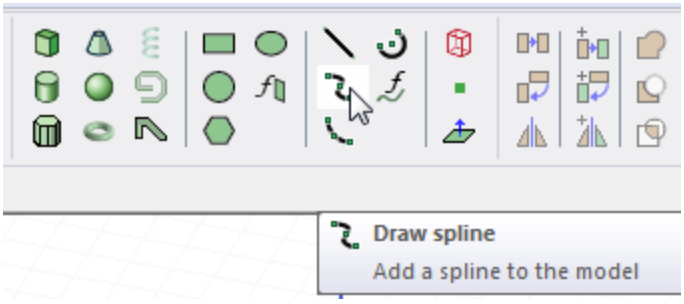
6. Click **OK**.

> **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.

Drawing a Spline

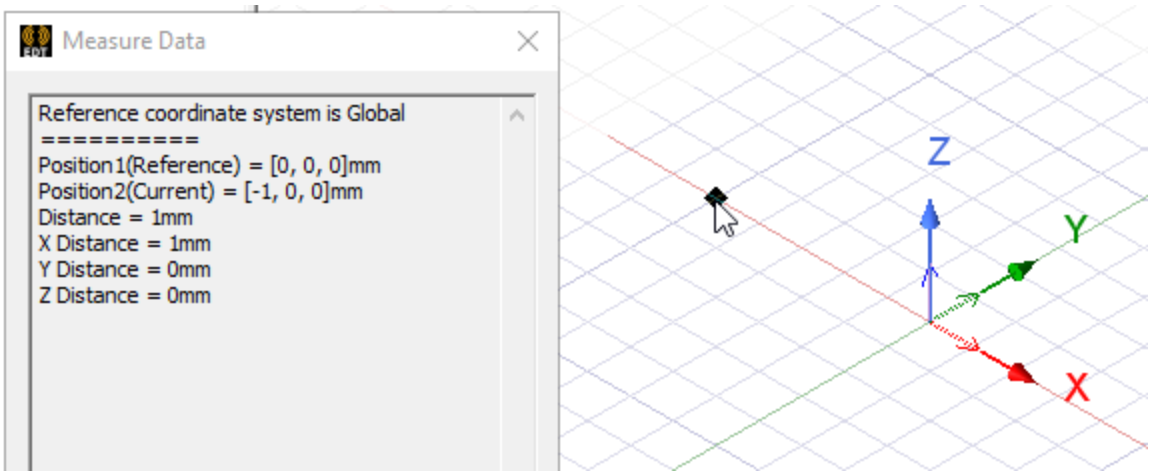
A spline is a curved line defined by three points. The modeler uses a natural spline type: a piece wise cubic spline with an end condition that has a derivative of zero. Use the **Draw** > **Spline** command to create a polyline object with one or more spline segments. Before you draw a spline, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Spline** or, on the **Draw** ribbon tab, click the **Draw spline** icon:

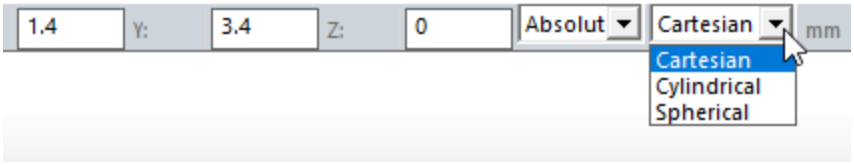


2. Select the spline's start point in one of the following ways:

- Click the point.



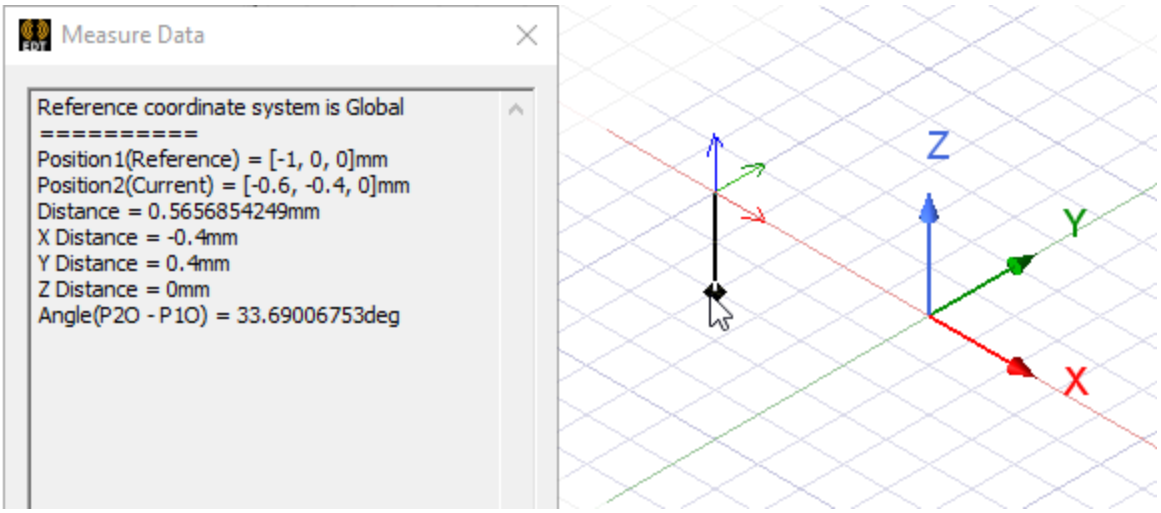
- Type the point's coordinates in the text boxes in the Status bar, and then press **Enter**.
  The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the spline in Cartesian, Cylindrical, or Spherical coordinates.



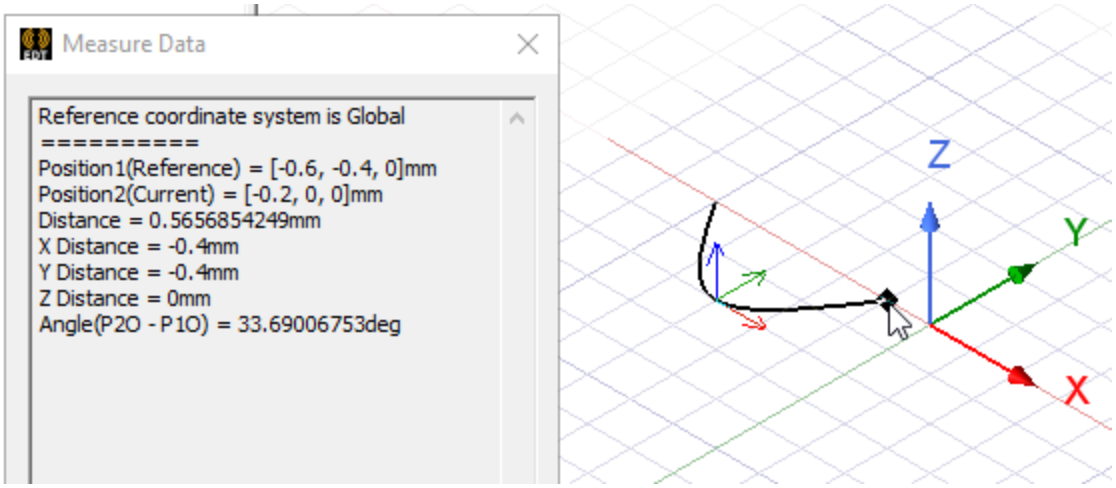
To delete the last point entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

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 text boxes in the status bar.



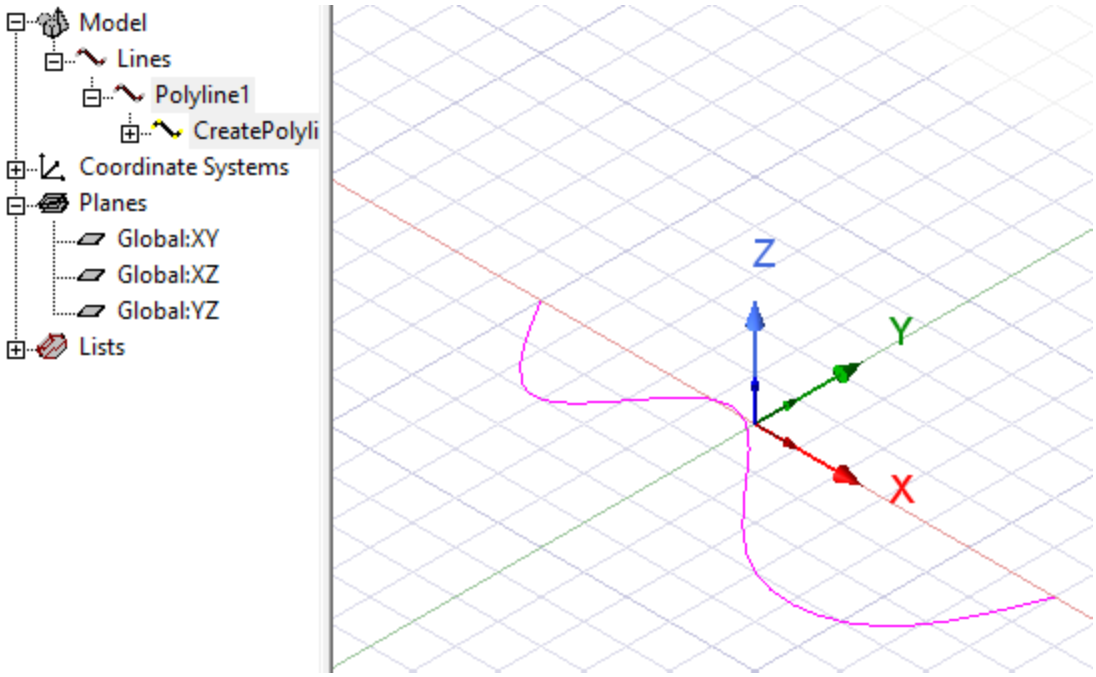
4. Select the endpoint of the spline by clicking the point or typing the coordinates in the text boxes in the status bar.



The endpoint serves as the start point for a subsequent spline segment. You can complete the spline at this point, or add additional segments.

5. Complete the spline in one of the following ways:
   - Double-click the endpoint.
   - Click **Done** on the shortcut menu.

- Press **Enter**.
  The spline appears in the Modeler window and the History tree.



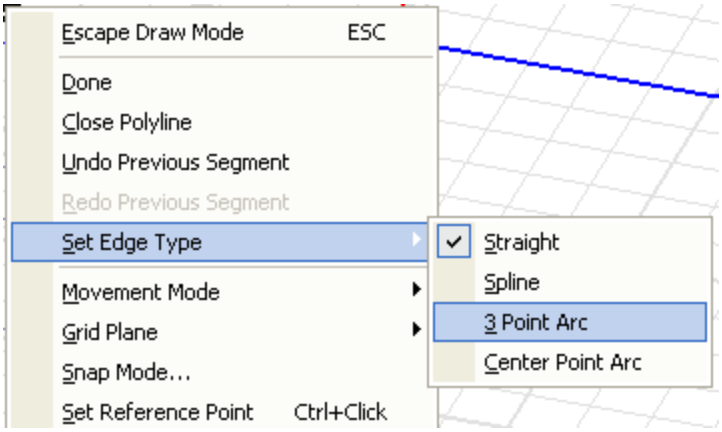
If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes by editing the Properties. Those listed under the Command tab describe the commands used to create the object. These commands also appear in 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. If you have edited Properties, 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.

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.

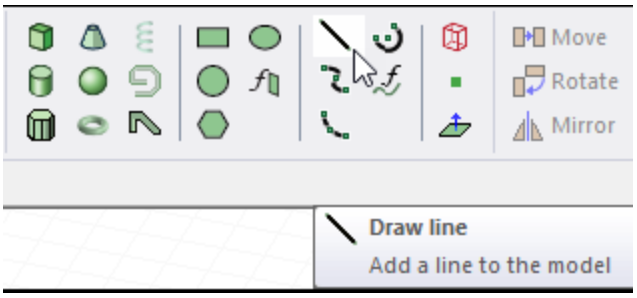


Before drawing a polyline, determine how closed polylines will be handled in the 3D Modeler options, as follows:

Using the menu bar, click **Tools** > **Options** > **General Options**. In the *Options* dialog box that appears, choose **3D Modeler** > **Operation** from the tree on the left side. Then, in the **Polyline Creation** section of the operation options, select or clear the **Automatically cover closed polylines** check box.

If checked, closed polylines become sheet objects and are listed under *Sheets* in the History Tree. If unchecked, closed polylines are listed under *Lines* in the History Tree.

1. From the menu bar, click **Draw**> **Line** or, on the **Draw** ribbon tab, click the **Draw line** icon:



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. Depending on your selection in the previous step, do one of the following:
   - 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 by editing the Properties. 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 plotting fields along a line. Notice that by going to the History Tree and selecting Create Polyline for that object, you can assign a cross section and dimensions to a polyline.

7. Click **OK**.

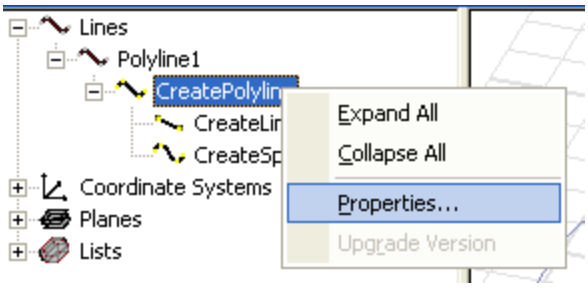
   If you select a polyline in the History Tree, you can use the Measure mode to see the total length.

Assigning a Cross Section and Dimensions to a Polyline

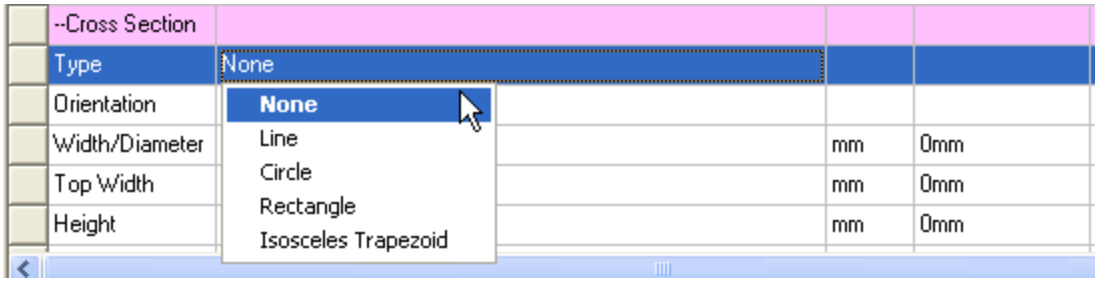
By viewing the properties of a polyline from the History Tree, you can assign either a Line, Circle, Rectangle or Isosceles Trapezoid cross section to a polyline. This assignment enables editable dimension properties of width for a line, diameter for a circle, and height and width for a rectangle or trapezoid. To assign a cross section to a polyline:

1. In the History Tree of the Modeler window, select the **CreatePolyline** entry for the polyline that you want to give a cross section.

This action displays the polyline properties in the docked **Properties** window (if you have it displayed). Alternatively, right-click the **CreatePolyline** entry and choose **Properties...** from the shortcut menu to display the **Properties** dialog box.



2. In the **Cross Section** section of the properties, in the second column of the **Type** row, click **None** to display the cross section choices.



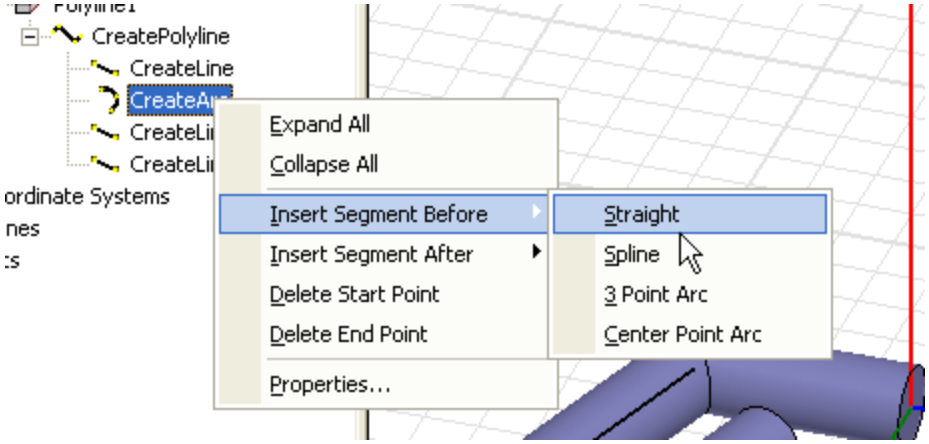
3. Choose a **Line, Circle, Rectangle,** or **Isosceles Trapezoid** cross section.
   - Selecting **Line** causes the Cross Section area of the polyline properties to display editable fields for **Orientation** and **Width**.
   - Selecting **Circle** causes the Cross Section area of the polyline properties 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 setting specifies the direction in which the **Height** 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 cross section) or as a Solid object (Circle or two dimensional Rectangle cross section).

6. You can modify the new polyline either by editing the properties, or by using the History Tree to select one of the segments that make up the polyline and right-clicking to display the shortcut menu showing commands to **Insert**, **Delete**, or display the segment **Properties**.



A segment that you select in the History Tree is indicated in the Modeler window by black highlighting. If you insert a new segment, it adopts the dimensions you specified for the polyline object.

Inserting Line Segments

You can insert line segments of various kinds for existing line, arc, or spline objects.

1. Select a line segment in the History Tree (not the modeler window). The individual segments entries are **CreateLine**, **CreateArc**, **CreateAngularArc**, or **CreateSpline**.

   This action highlights the object and enables the **Insert Line Segment** subcommands in the **Draw** menu.

2. Using the menu bar, go to the **Draw** > **Line Segment** submenu and then choose either the **Insert Segment Before** or **Insert Segment After** submenu.

   Alternatively, right-click a polyline segment in the History Tree and, from the shortcut menu, choose either the **Insert Segment Before** or **Insert Segment After** submenu

3. Use this submenu to specify the kind of segment to add. The inserted segment can be: **Straight, Spline. 3 Point Arc,** or **Center Point Arc**.
4. Depending on your selection in the previous step, do one of the following:
   - If you clicked **Straight**, follow the procedure for drawing a straight line.
   - 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.
   - If you clicked **Spline**, follow the procedure for drawing a spline.

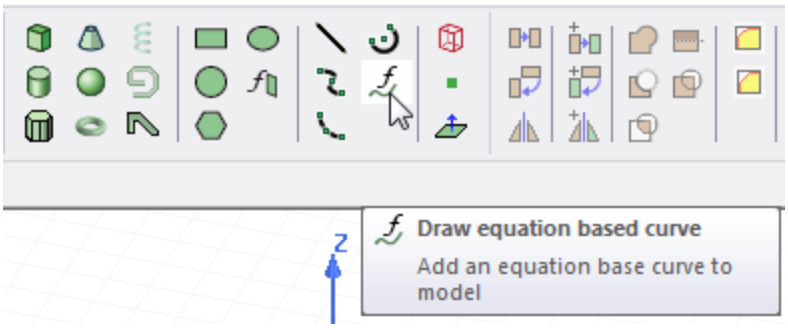
   > **Note:**
   >
   > When inserting straight segments or arcs, the insertion operation will terminate as soon as you complete the definition of a single new segment. For splines, you can insert a spline with any number of defining points, and the insertion operation terminates as soon as you terminate the spline.

5. Repeat steps 1 through 4 for each segment of the polyline before or after which you want to insert a new segment. The endpoint of the previous segment serves as the start point for the next segment.

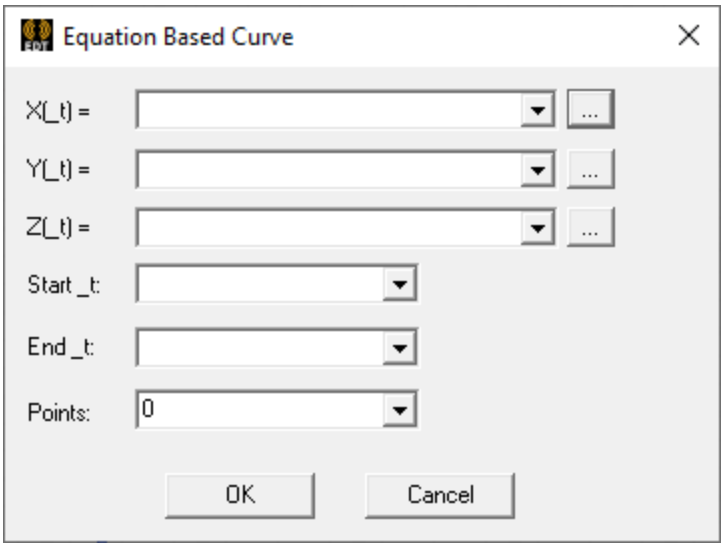
Drawing an Equation-Based Curve

Any line that can be described by an equation in three dimensions can be drawn. Before you draw am equation-based curve, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Equation Based Curve** or, on the **Draw** ribbon tab, click the **Draw equation based curve** icon:



   The *Equation Based Curve* dialog box opens:

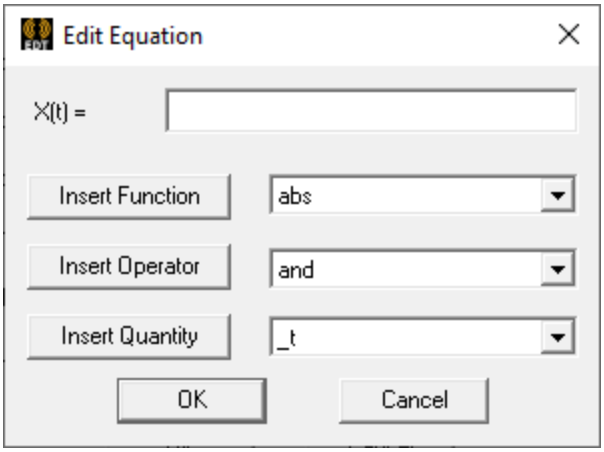


2. Type equations for **X(_t)**, **Y(_t)**, and **Z(_t)**

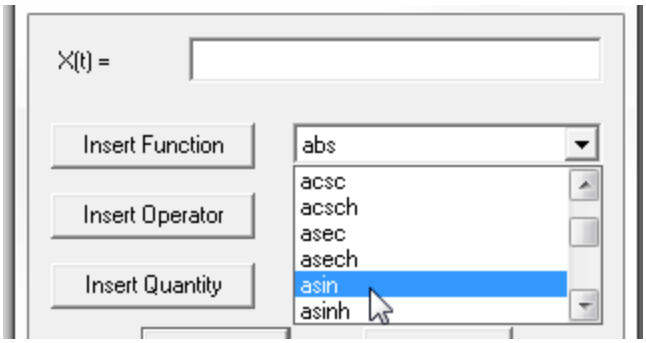
   You can also define an equation by doing the following:

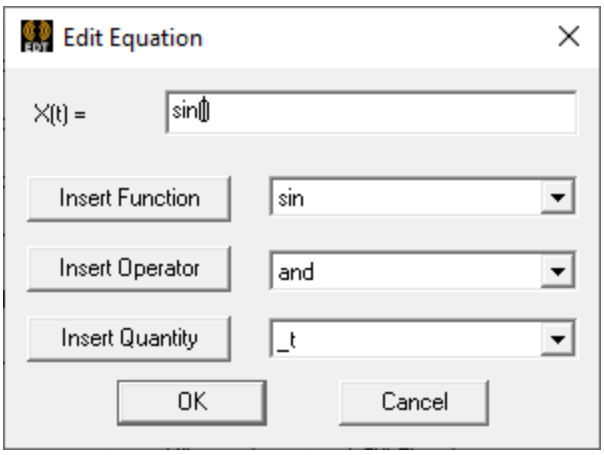
   a. Click an ellipsis button **[...]**.

      The **Edit Equation** dialog box appears:



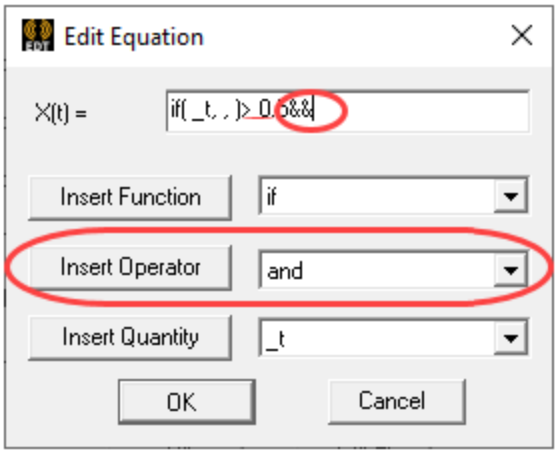
   b. Do one or more of the following to define the equation:
      - Type a numerical value, variable, or expression directly in the text box.
      - Build an equation for X(_t), Y(_t), or Z(_t) by specifying appropriate function, operator, and quantity to insert into the expression from the three drop-down menus:

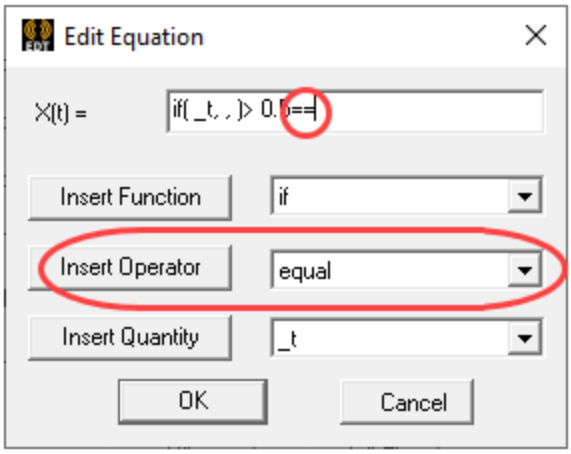




Click the corresponding **Insert Function, Insert Operator,** or **Insert Quantity** buttons to insert the selections into the equation text box at the current cursor location. The if(, ,) function has been added under **Insert Function**.

- The **Insert Operator** button inserts corresponding operator:



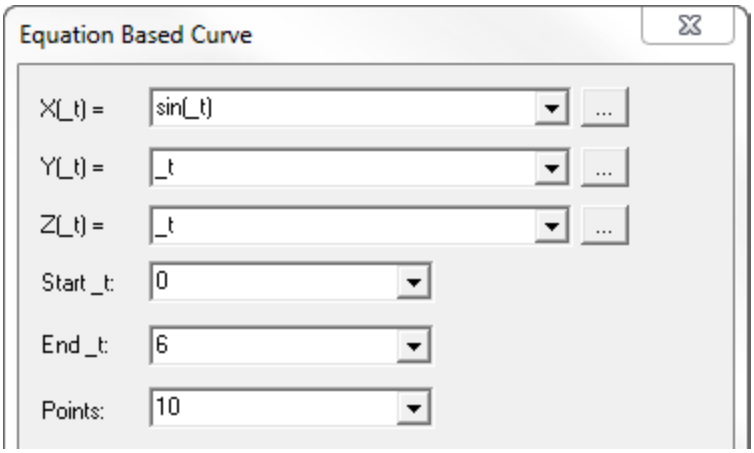


- Use a combination of the preceding two methods to build complex equations.

c. 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.

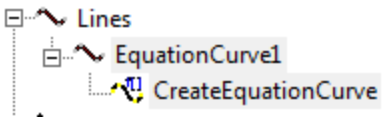
> **Note:**
>
> - The equations are taken to be meters. Use of units in equations is not properly supported.
> - While parsing expressions, equation based curves convert each variable separately to model units and assume that the resulting expression is in model units.
> - Equation based curves depend on the variable value library to correctly evaluate the units of expression.
> - Functions that do not support intrinsic (_t for curve, _u and _v for surface) will give an error if not supported.

3. Select a start value from the **Start_t** drop-down menu.
4. Select an end value from the **End_t** drop-down menu.
5. Type in or select the number of points in the curve from the **Points** drop-down menu.



6. Click **OK** on dialog to close it and create the curve.

   The curve appears in the modeler window, and the History tree shows the curve object and the command.



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 by editing the Properties.

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.

Functions, Operators and Quantities for the Edit Equation Dialog Box

This table lists the functions, operators and Quantities for the Edit Equation dialog box.

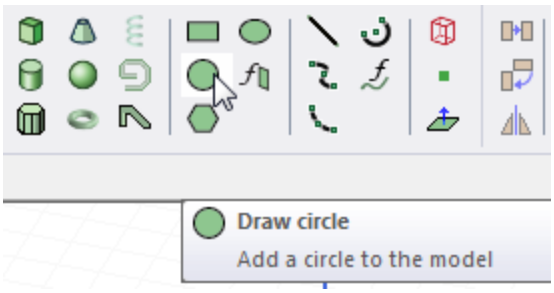
Item	Definition
Functions	
abs	Absolute value of the simulation quantity which results in a number that is always positive.
acos	Arc cosine (the inverse function of a cosine).
acosh	Inverse hyperbolic arc cosine.
acot	Inverse cotangent
acoth	Inverse hyperbolic cotangent
acsc	Inverse cosecant
acsch	Inverse hyperbolic cosecant
asec	Inverse secant
asech	Inverse hyperbolic secant
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
cos	Cosine
cosh	Hyperbolic cosine
cot	Cotangent
coth	Hyperbolic cotangent
csc	Cosecant
csch	Hyperbolic cosecant
ln	Natural logarithm.
log	Natural logarithm (same as ln).
sin	Sine.
sinh	Hyperbolic sine.
sqrt	Square root of the selected simulation quantity.
tan	Tangent.

Item	Definition
tanh	Hyperbolic tangent.
Operators	
and	
cross	
division	
dot	
equal	
exp	
exponent	
greater_than	
greater_than_ or_equal	
if	if(, ,)
not	
not_equal	
or	
Quantities	
_t	For equation based lines
_u,_v	For equation based surfaces.
PI	

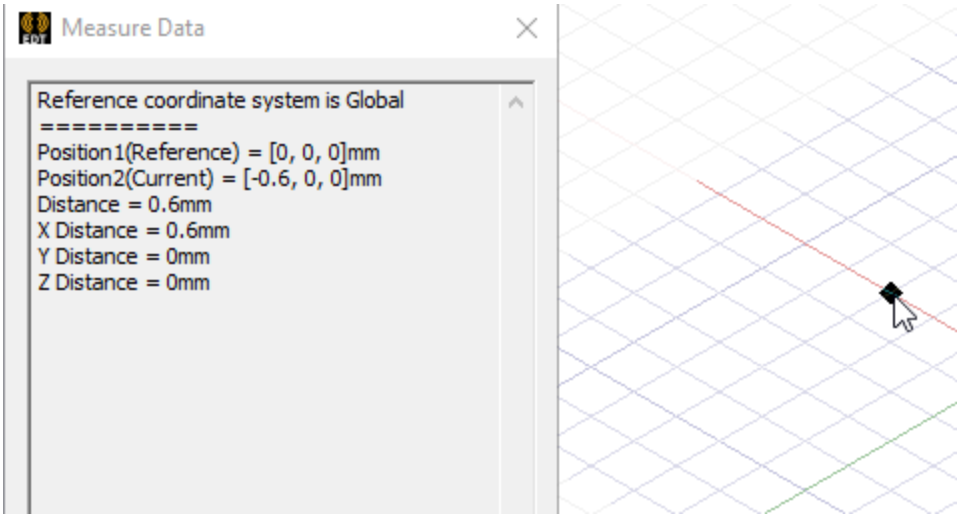
Drawing a Circle

Draw a circle by selecting a center point and a radius. Circles are drawn as true surfaces in the modeler. Before you draw a circle, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Circle** or on the **Draw** ribbon tab, click the **Draw circle** icon:

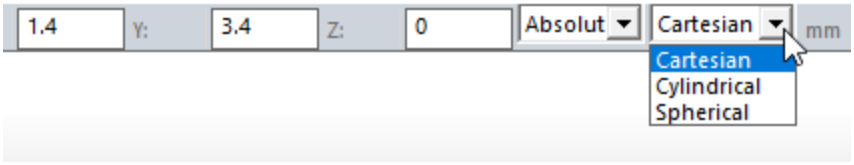


2. Select the center point of the circle in one of the following ways:
   - Click the point.



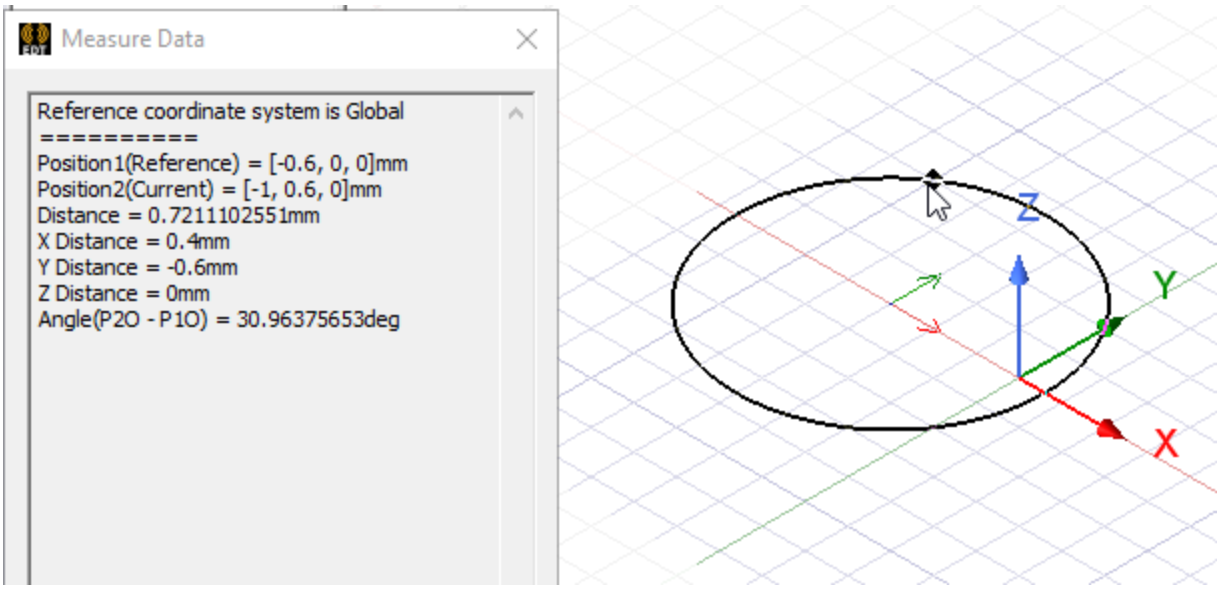
   - Type the point's coordinates in the text boxes in the status bar.

     The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the circle in Cartesian, Cylindrical, or Spherical coordinates.



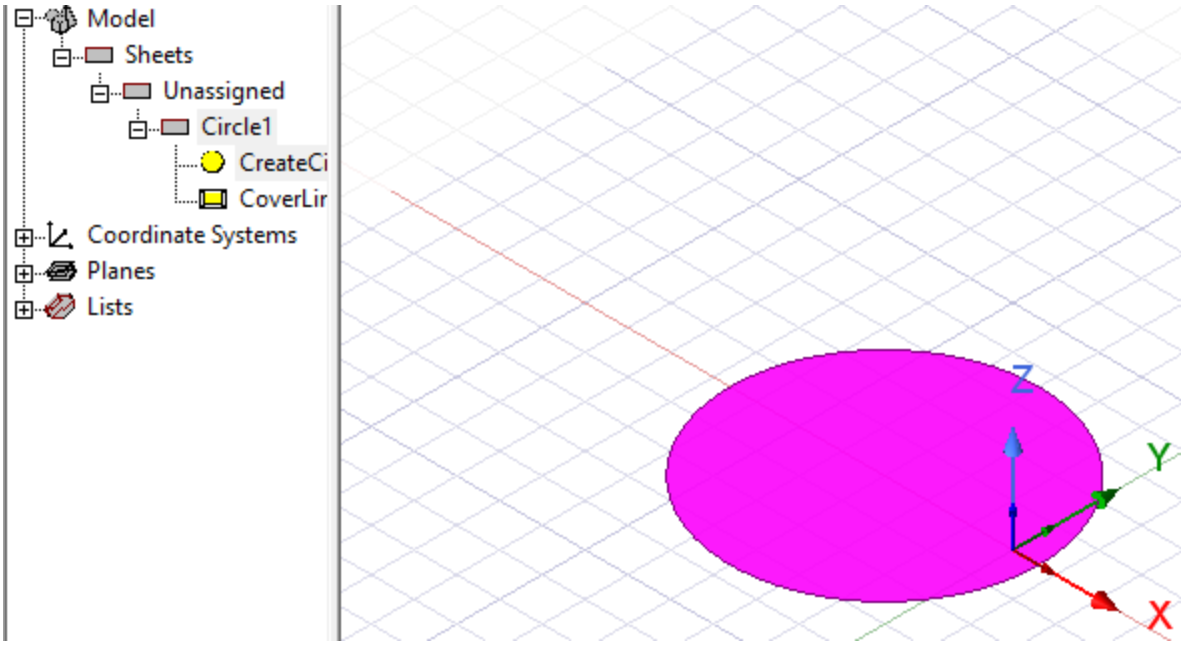
3. Specify the radius by selecting a point on the circle's circumference in one of the following ways:

- 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.

4. Double-click or right-click and select Done from the shortcut menu. 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. 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. The new circle appears in the Modeler window and History tree.



Note:

The 3D Modeler permits drawing true curved objects. However, the solution is obtained using 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:

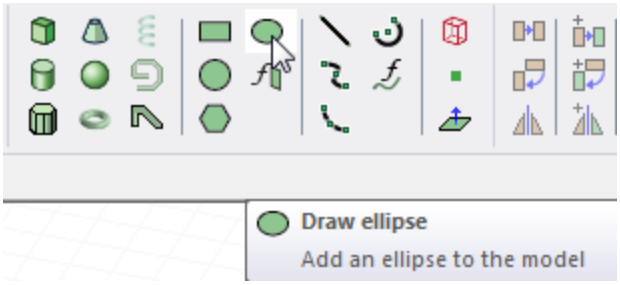
Modifying Surface Approximation Settings and related sections:

Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

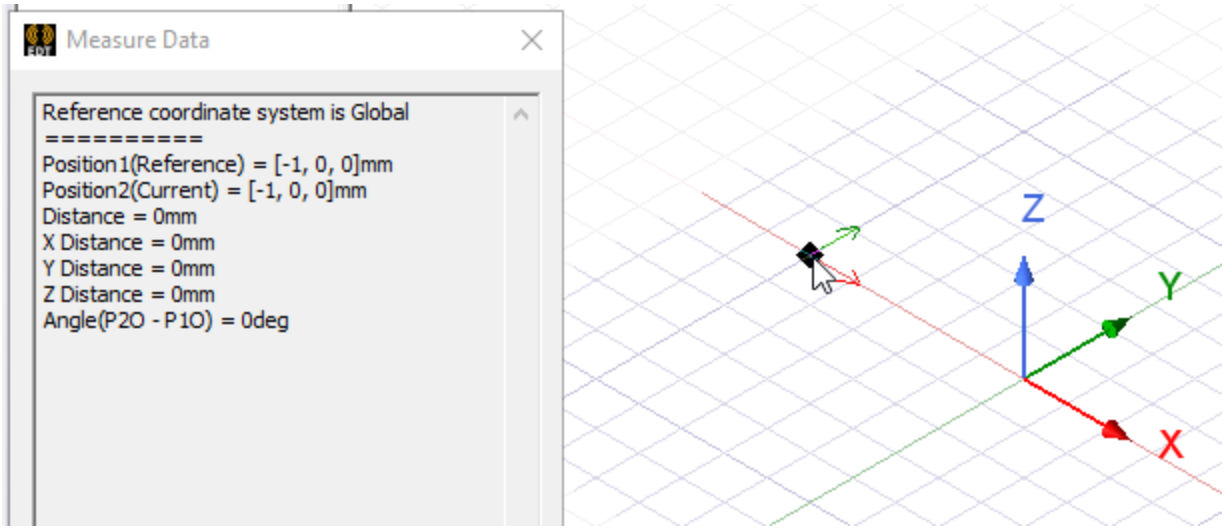
Drawing an Ellipse

Draw an ellipse by specifying a center point, base radius, and secondary radius. Before you draw an ellipse, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

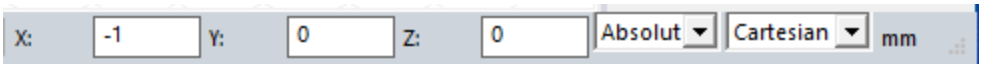
1. From the menu bar, click **Draw>** **Ellipse** or, on the **Draw** ribbon tab, click the **Draw ellipse** icon:



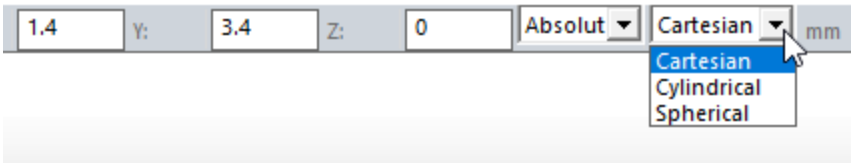
2. Select the center point of the ellipse in one of the following ways:
   - Click the point.



   - Type the point's coordinates in the text boxes in the status bar.

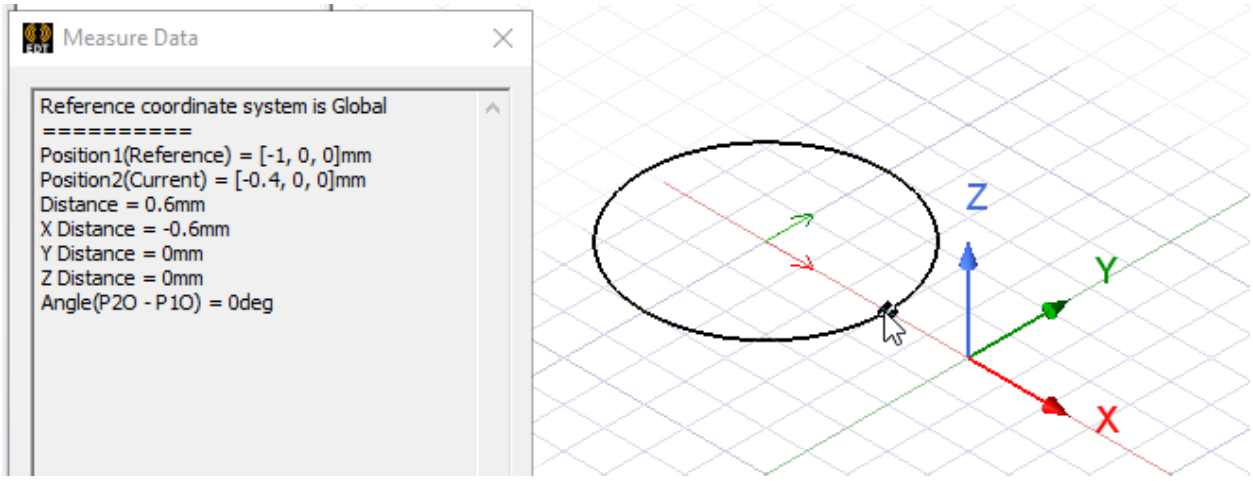


     The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the ellipse in Cartesian, Cylindrical, or Spherical coordinates.



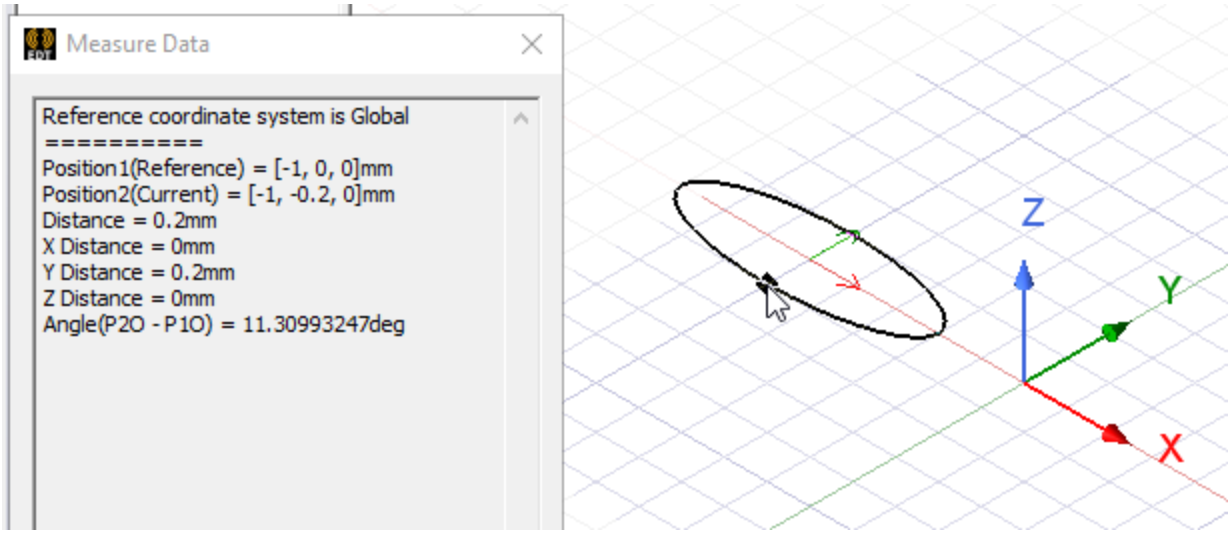
3. Specify the base radius of the ellipse. If the current drawing plane is xy, then x is the base radius direction. If the drawing plane is yz, then y is the base radius direction. If the drawing plane is xz, then z is the base radius direction. Select the point in one of the following ways:

- Click the point. Ansys Electronics Desktop constrains mouse movement to the base radius direction.



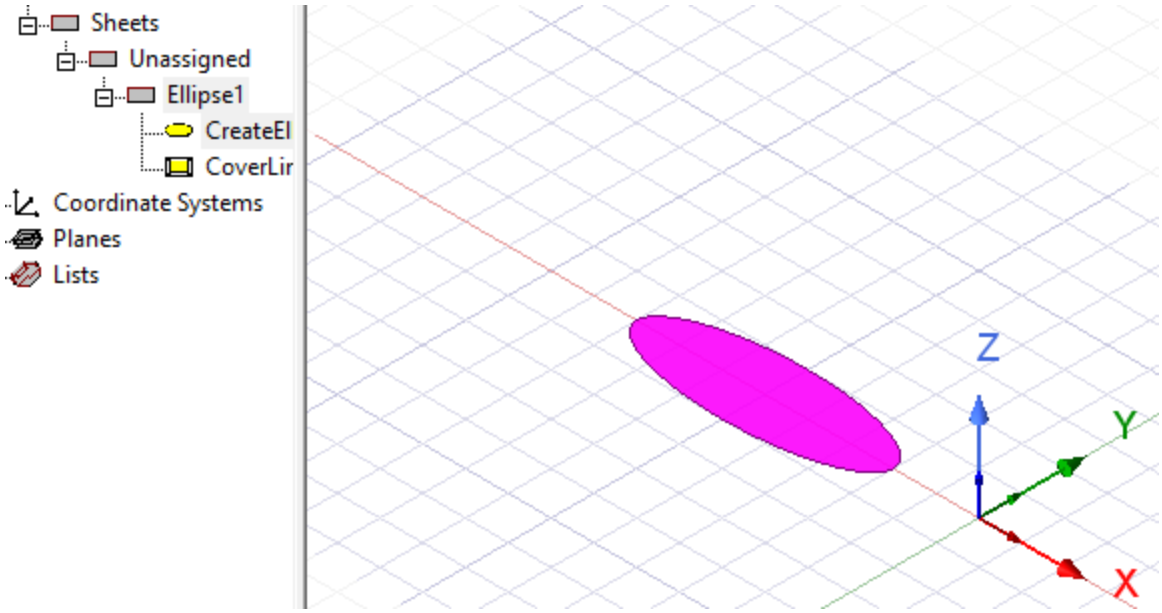
- Type the coordinates of a point relative to the center point in the **dX, dY,** or **dZ** box, where **d** is the distance from the previously selected point.

4. Specify the secondary radius of the ellipse. Select the point in one of the following ways:
   - Click the point. Ansys Electronics Desktop 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 **dXm dY,** or **dZ** box.

5. Double click, or right click and select **Done** from the short cut menu. The ellipse appears in the Modeler window and in the History Tree.



6. 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. The **Ratio** value represents the aspect ratio of the secondary radius to the base radius.

   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 Modeler permits drawing true curved objects. However, the solution is obtained using 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:

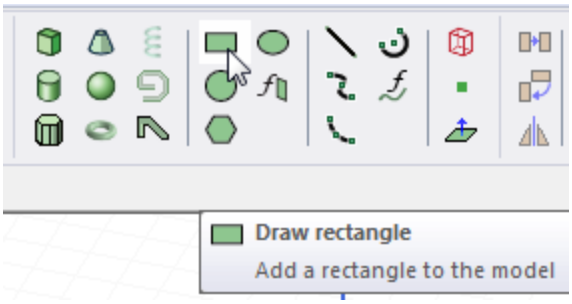
Modifying Surface Approximation Settings and related sections:

Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

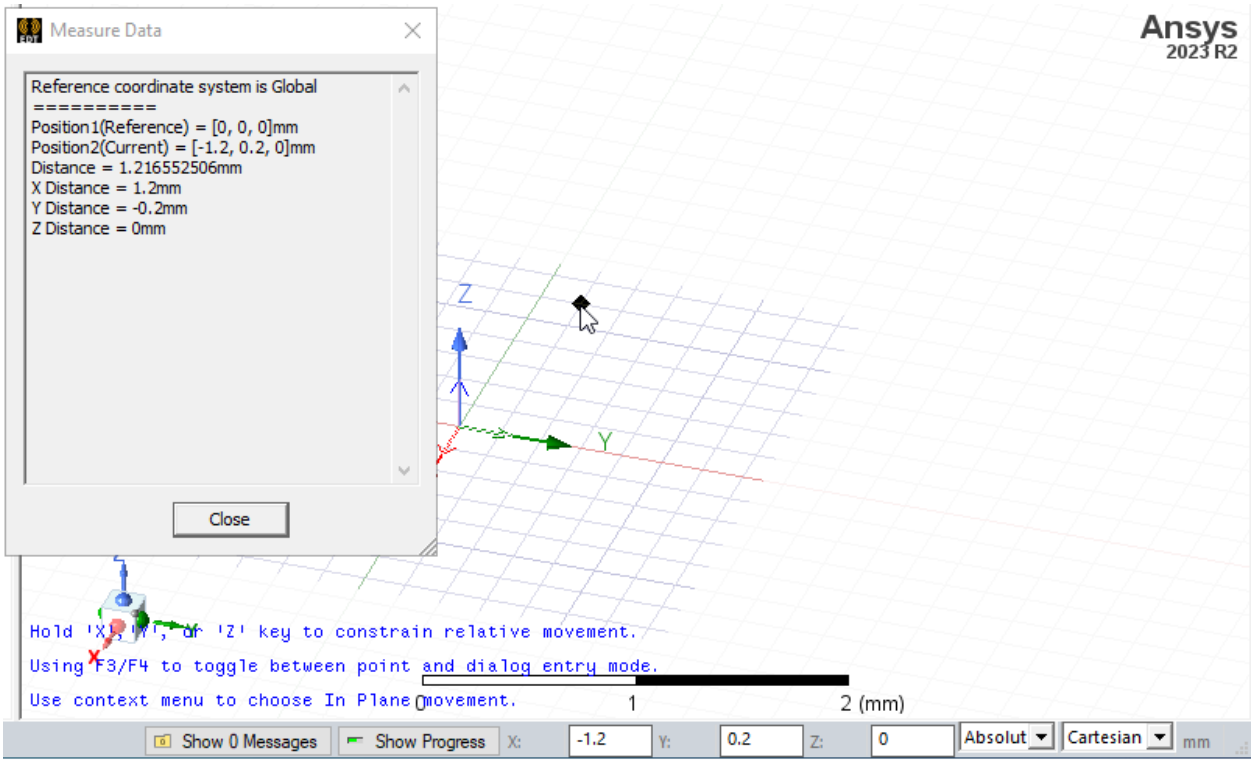
Drawing a Rectangle

Draw a rectangle (or square) by selecting two diagonally opposite corners. Before you draw a rectangle, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

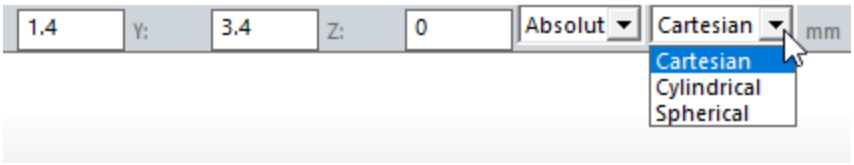
1. From the menu bar, click **Draw**> **Rectangle** or, on the **Draw** ribbon tab, click the **Draw rectangle** icon:



2. Select the first diagonal corner in one of the following ways:
   - Click the point. The Measure Data dialog shows the coordinates and other information.

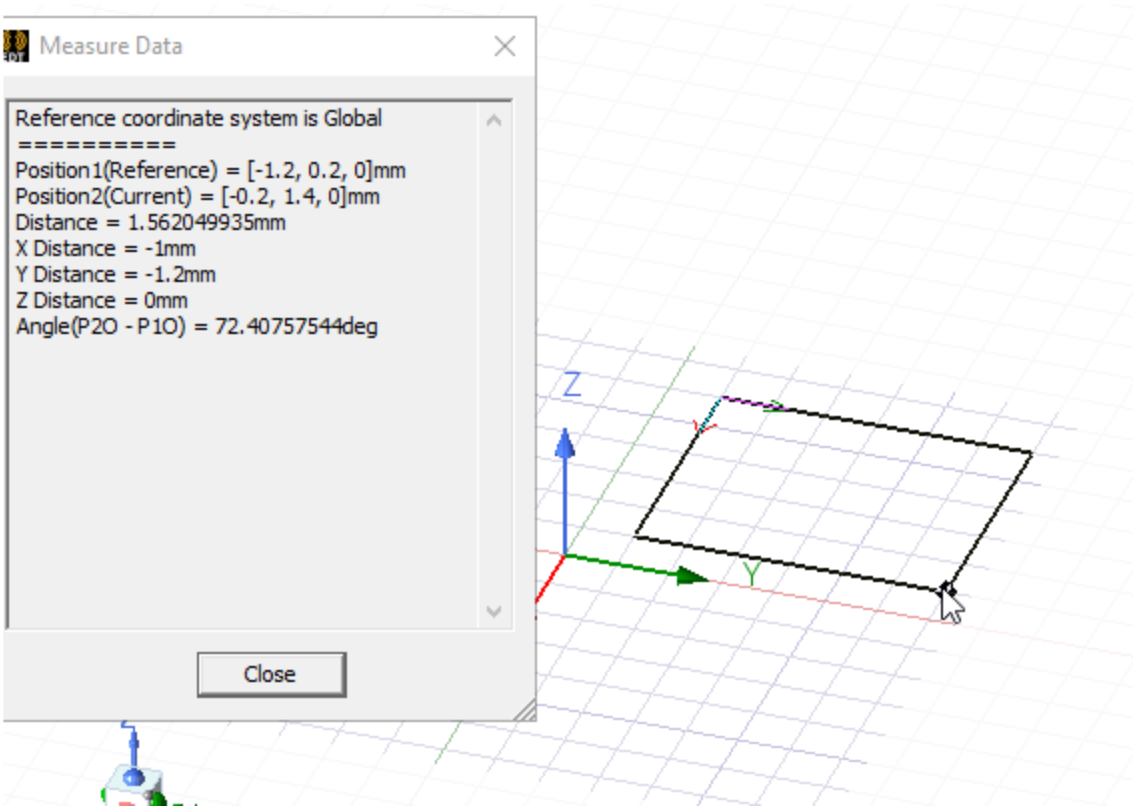


- Type the point's coordinates in the text boxes in the status bar. The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the cylinder in Cartesian, Cylindrical, or Spherical coordinates.



To delete the selected point and start over, press **Esc** or click **Escape Draw Mode** on the shortcut menu.

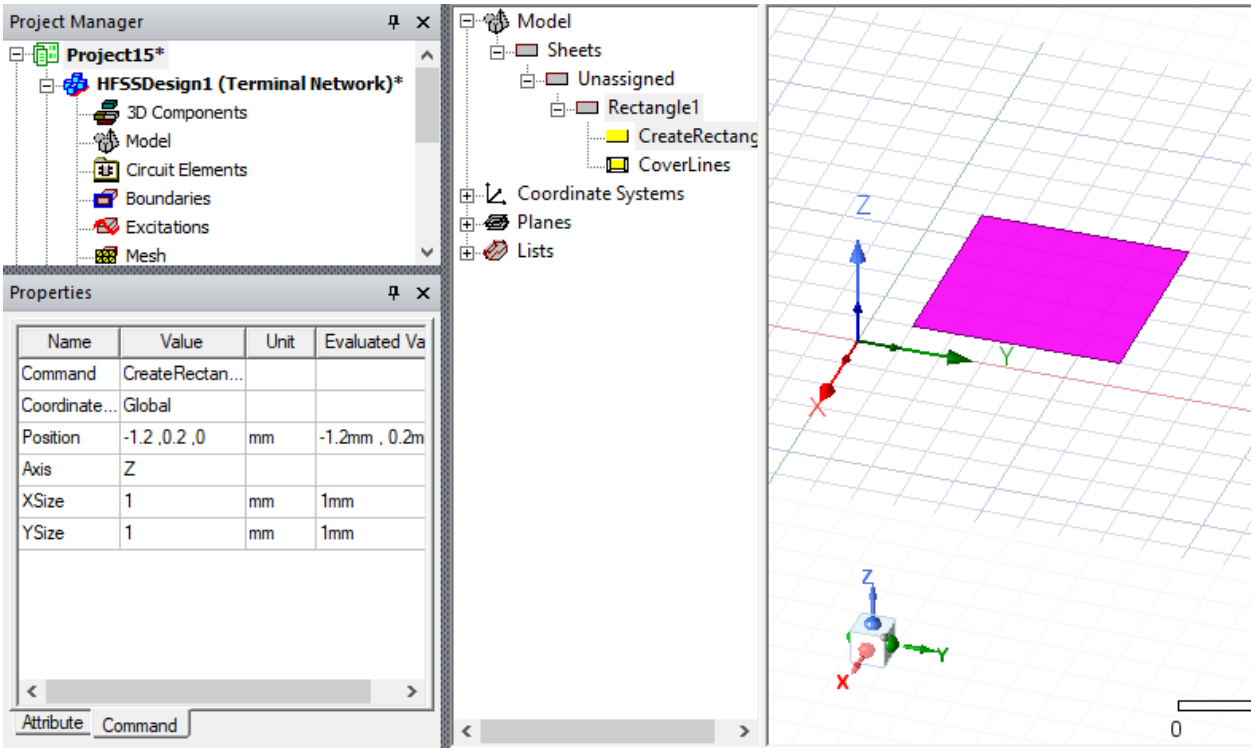
3. Select the second corner of the rectangle in one of the following ways:
   - Click the point.



   - 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, enabling you to modify the object's properties. You can also use the docked Properties window.

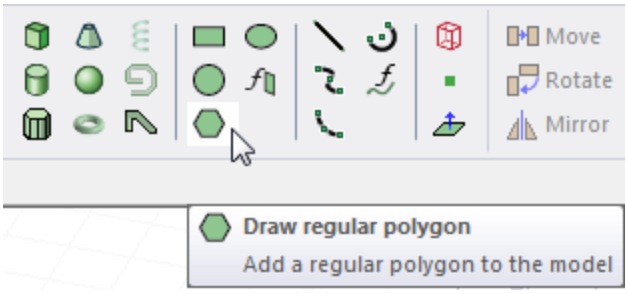
4. Click **OK**. If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the rectangle will be covered, resulting in a 2D sheet object, as shown in the History tree in the figure below. Otherwise it will be a closed 1D polyline object.



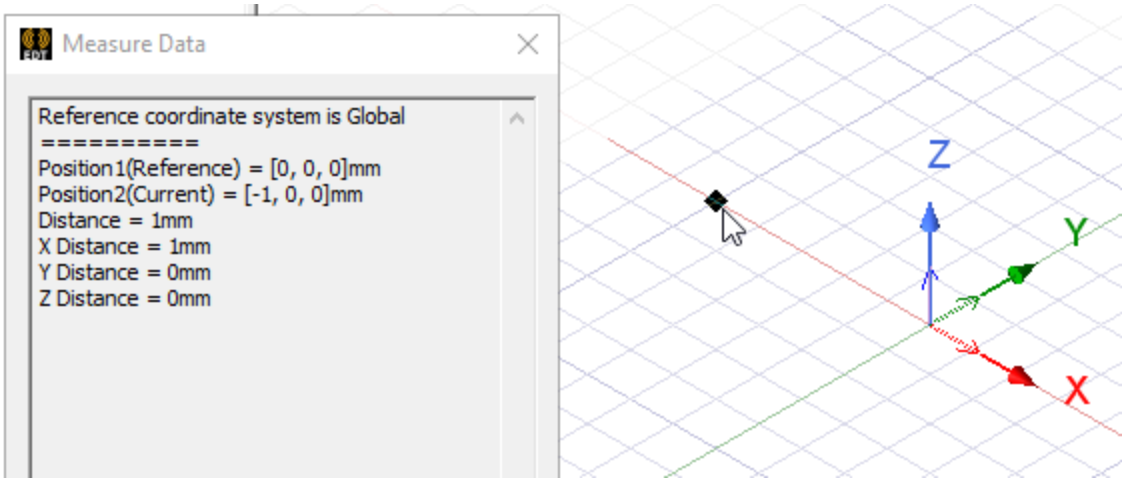
Drawing a Regular Polygon

A regular polygon is a 2D object with three or more equal sides. Regular polygons are useful for drawing faceted 2D objects. Before you draw a regular polygon, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. Click **Draw>** **Regular Polygon** or, on the **Draw** ribbon tab, click the **Draw regular polygon** icon:

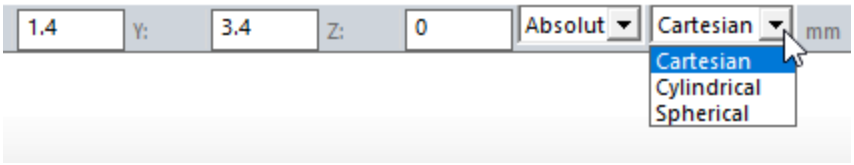


2. Select the center point of the polygon in one of the following ways:
   - Click the point.



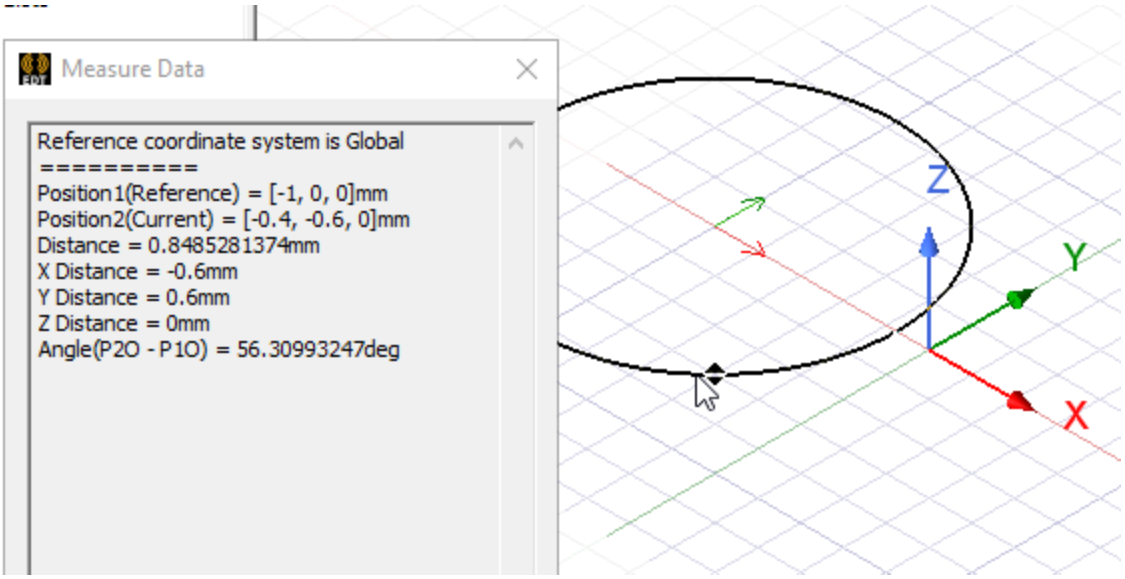
   - Type the point's coordinates in the text boxes in the status bar.

     The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the regular polygon in Cartesian, Cylindrical, or Spherical coordinates.



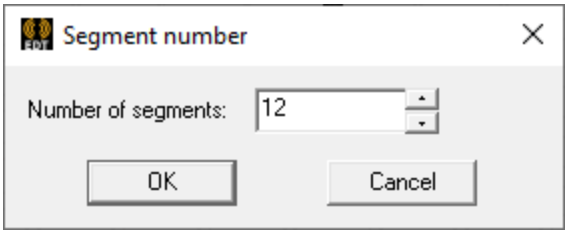
3. Specify the polygon's radius, the distance from the center point to one of the polygon's vertices, in one of the following ways:

- Click the point.

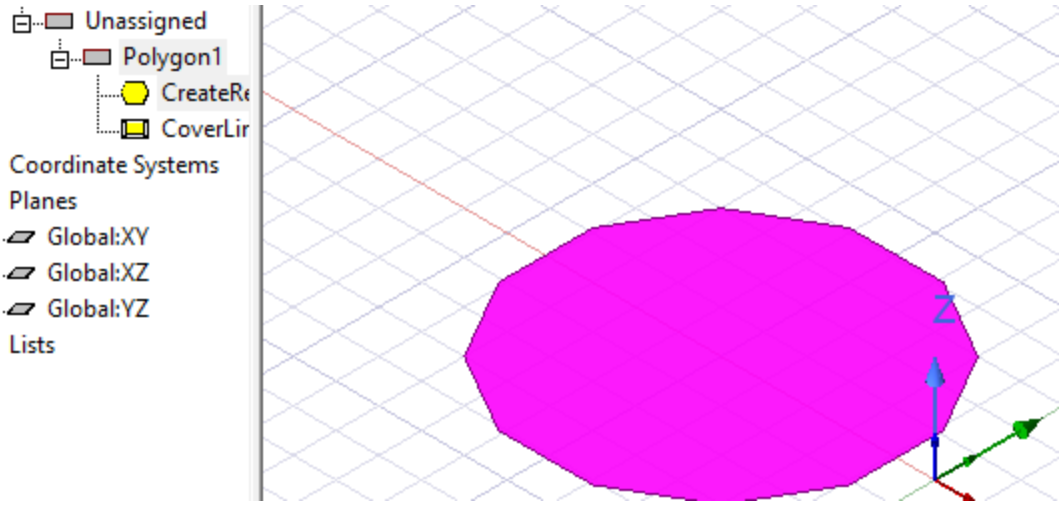


- Type the coordinates of the point relative to the center point in the **dX, dY,** and **dZ** boxes, where **d** is the distance from the previously selected point.

4. In the **Segment number** dialog box, enter the **Number of segments** in the polygon, and then click **OK**.



Click OK. The polygon appears in the Modeler window and in the History tree.



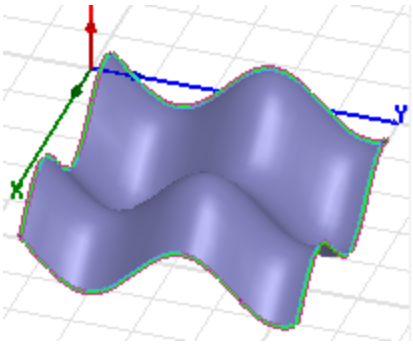
5. If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties. If you have edited the Properties dialog, click **OK**.

> **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.

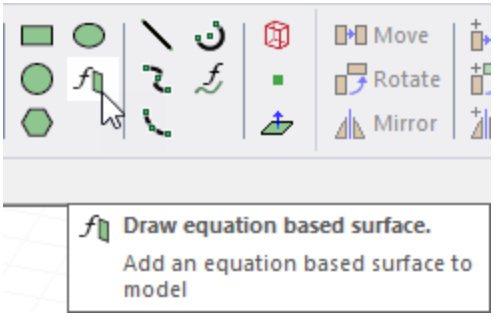
If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the polygon will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Drawing an Equation-Based Surface

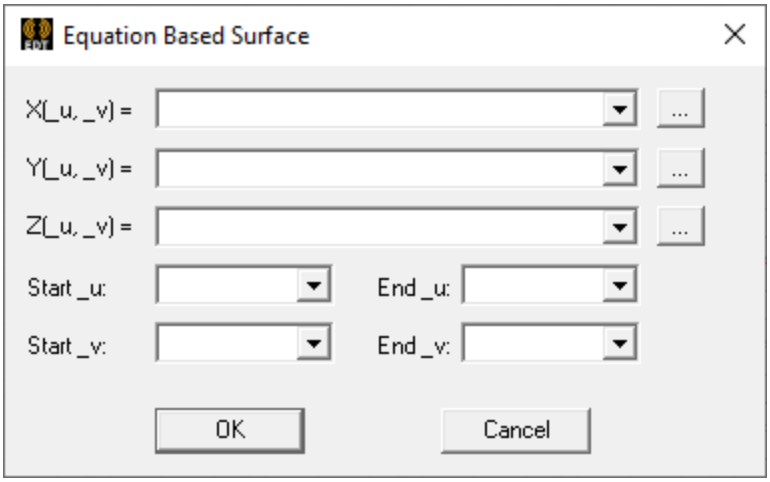
Any surface that can be described by an equation in three dimensions can be drawn. Before you draw an equation-based surface, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.



1. From the menu bar, click **Draw**> **Equation Based Surface** or, on the **Draw** ribbon tab, click the **Draw equation based surface** icon:



The **Equation Based Surface** dialog box opens.

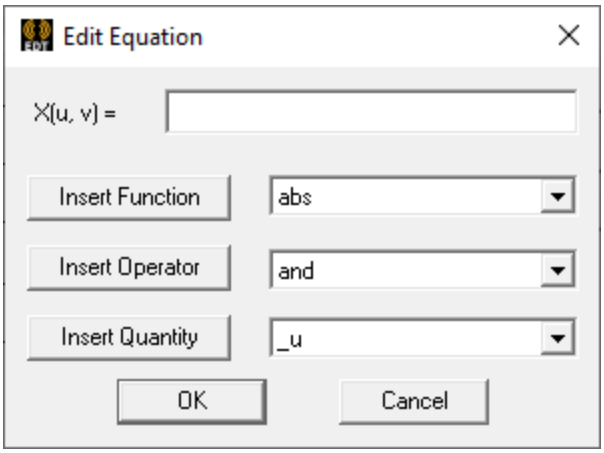


2. Type equations for **X(_u, _v)**, **Y(_u, _v)**, and **Z(_u, _v)**.

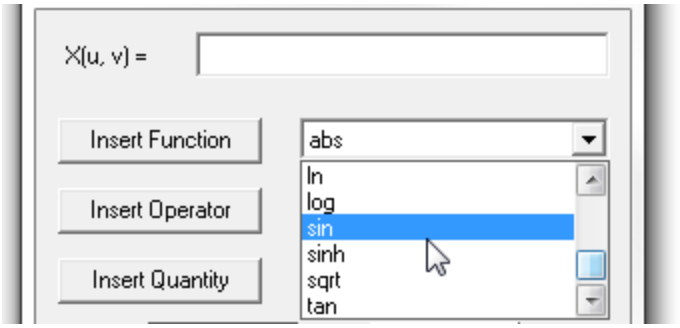
   You can also define an equation by doing the following:

   a. Click the ellipsis button (**...**).

      The **Edit Equation** dialog box appears:

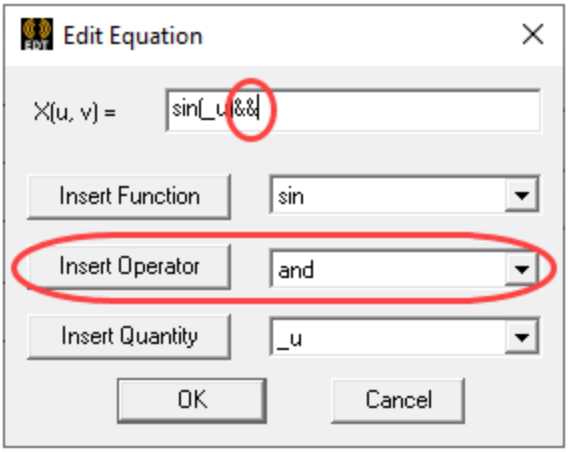


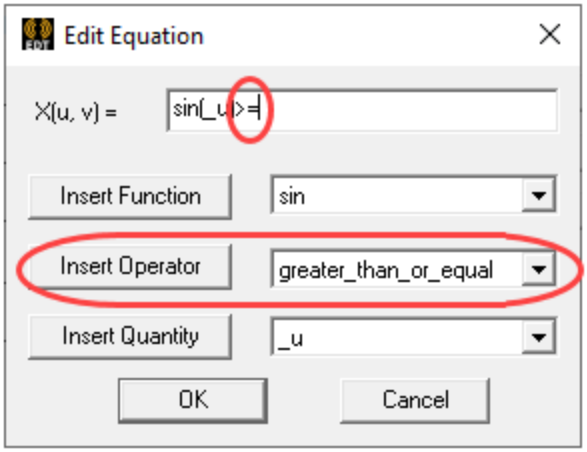
b. Do one or more of the following to define the equation:
   - Type a numerical value, variable, or expression directly in the text box
   - Build an equation for X(u, v), Y(u, v), or Z(u, v), by selecting the function, operator, and quantity to insert into the equation from the three drop-down menus:



     Click the corresponding **Insert Function, Insert Operator,** or **Insert Quantity** buttons to insert the selections into the equation text box at the current cursor location. The if(, ,) function has been added under **Insert Function**.

   - The **Insert Operator** button inserts corresponding operator:



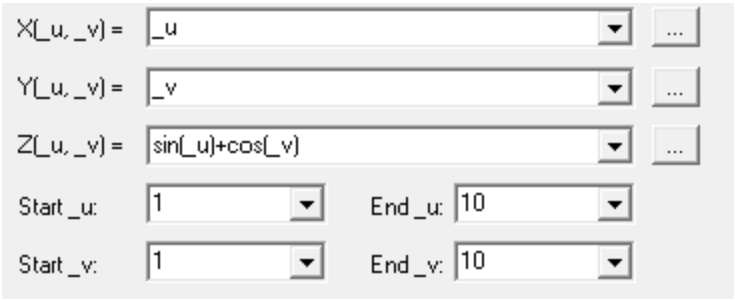


- You can combine the preceding two methods to build complex equations.

c. 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.
   - The equations are taken to be meters. Use of units in equations is not properly supported..
   - 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.
   - Functions that do not support intrinsic (_t for curve, _u and _v for surface) will give an error if not supported.

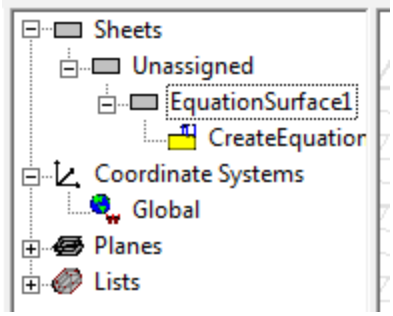
3. Specify start values for the **Start_u** and **Start_v** fields.

   Select end values for **End_u** and **End_v** fields.



4. Click **OK** on the **Properties** dialog box.

The surface is drawn in the Modeler window. The History tree contains the Equation Surface object and the command properties.

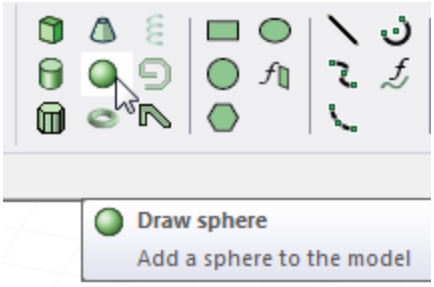


If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's attributes.

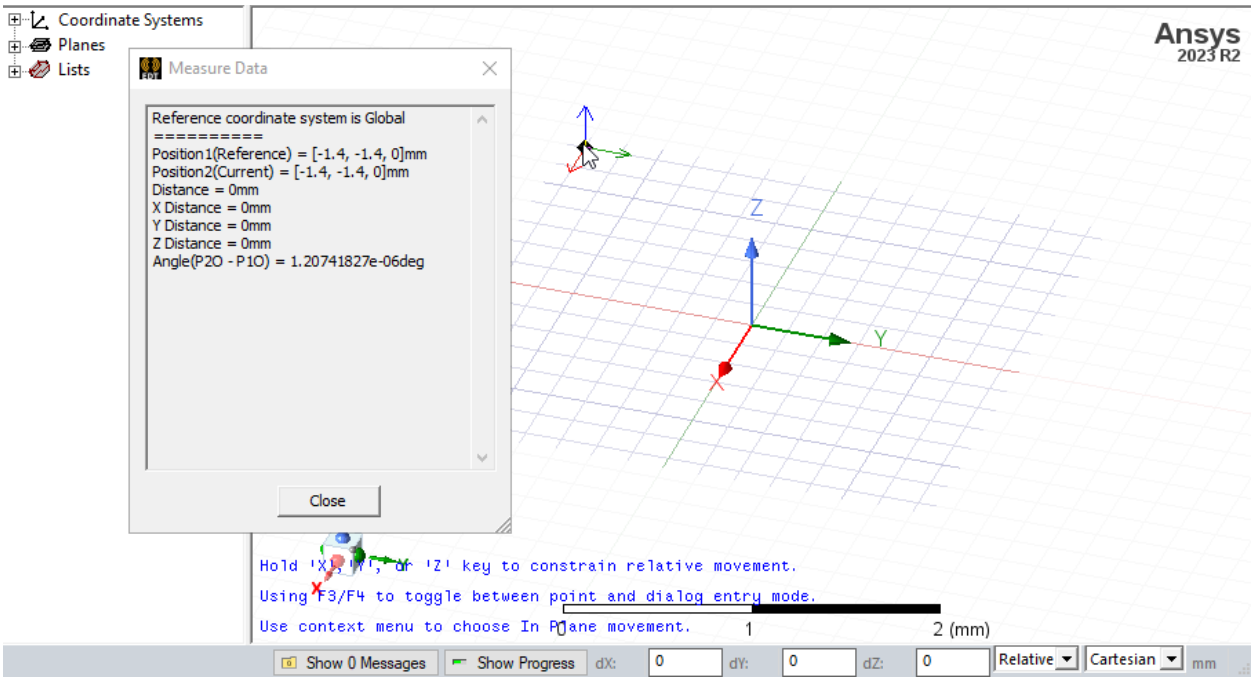
Drawing a Sphere

Draw a sphere, a 3D circle, by selecting a center point and a radius. Spheres are drawn as true surfaces in the modeler. Before you draw a sphere, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Sphere** or, on the **Draw** ribbon tab, click the **Draw sphere** icon:



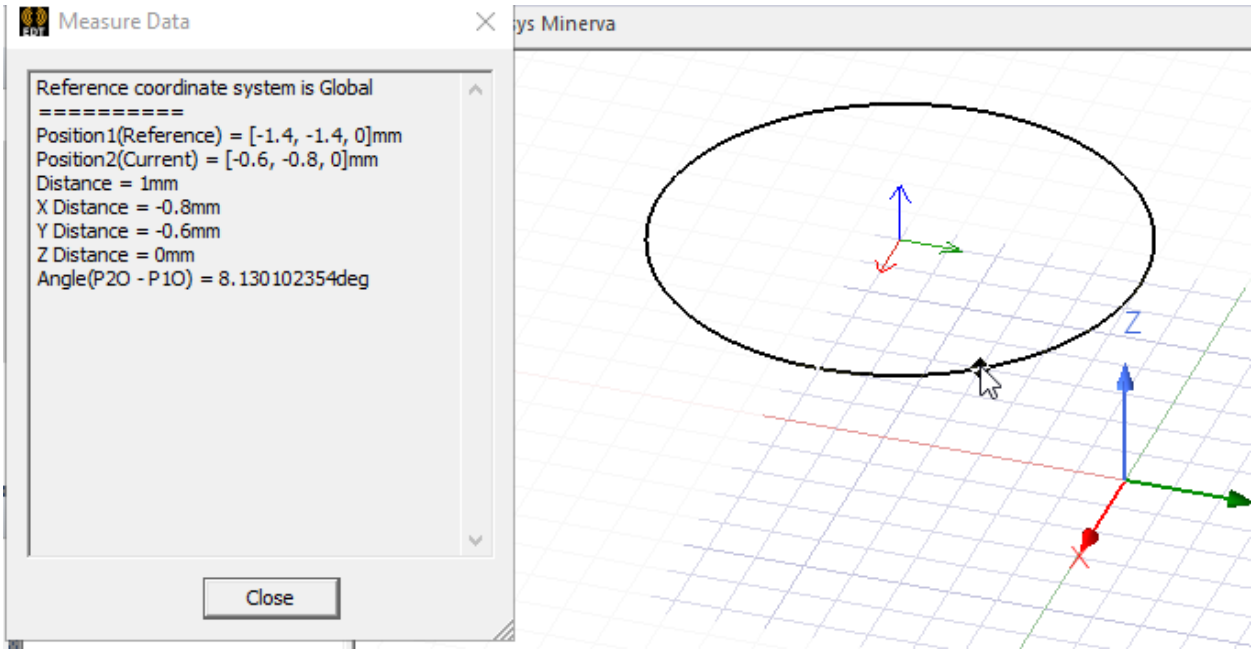
2. Select the center point of the sphere in one of the following ways:
   - Click the point. The Measure Data dialog shows the coordinates and other information.



- Type the point's coordinates in the text boxes in the status bar. The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the sphere in Cartesian, Cylindrical, or Spherical coordinates.

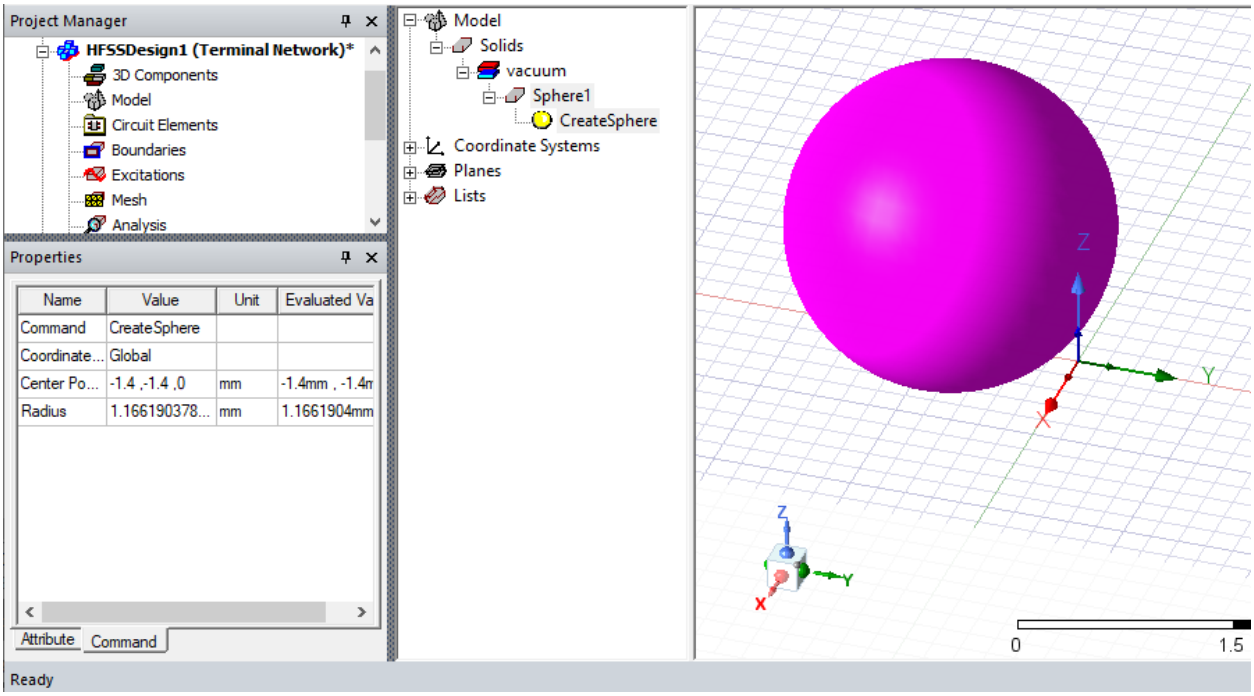
3. Specify the radius by selecting a point on the sphere's circumference in one of the following ways:

- 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, enabling you to modify the object's properties. You can also use the docked Properties window.



4. Click **OK**.

Note:

The 3D Modeler permits drawing true curved objects. However, the solution is obtained using 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:

Modifying Surface Approximation Settings and related sections:

Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

Related Topics

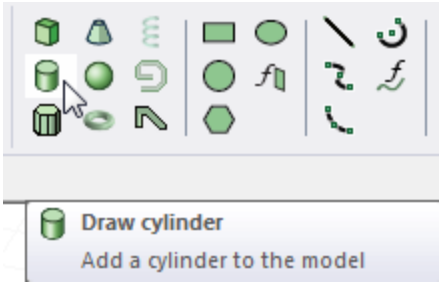
Modifying Objects

Modifying Surface Approximation Settings

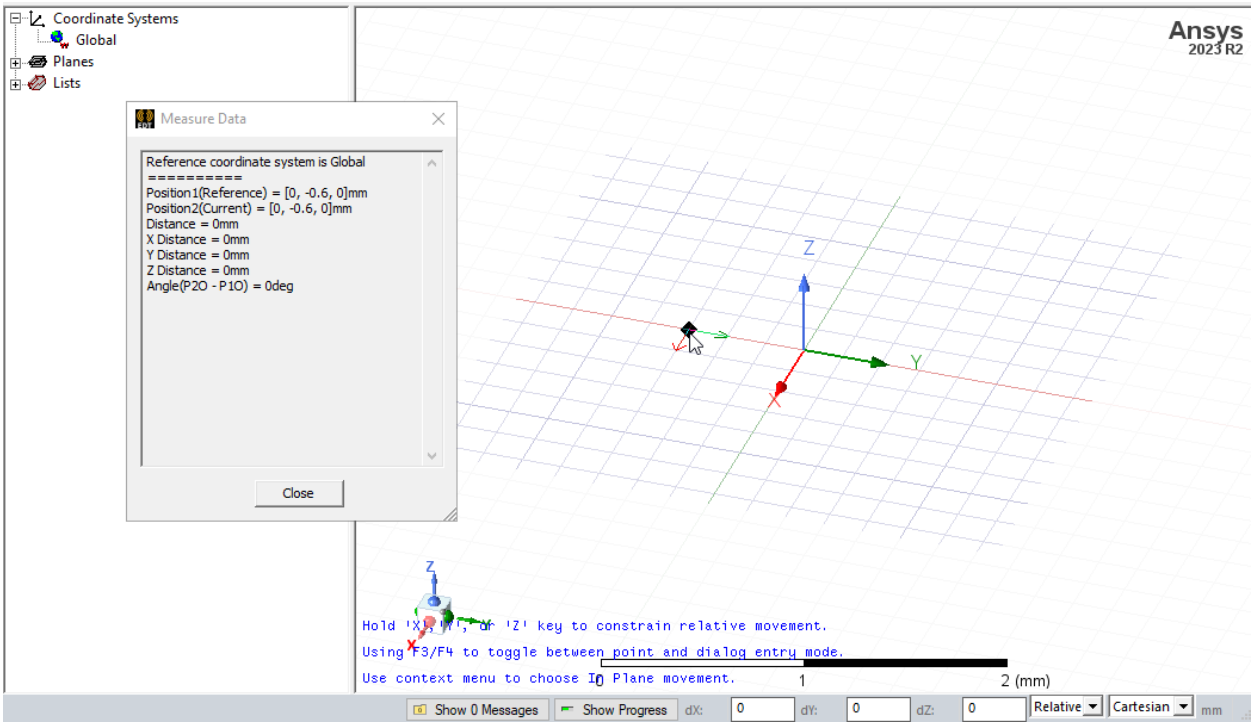
Drawing a Cylinder

Draw a cylinder by selecting a center point, radius, and height. Cylinders are drawn as true surfaces in the modeler. Before you draw a cylinder, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Cylinder** or, on the **Draw** ribbon tab, click the **Draw cylinder** icon:

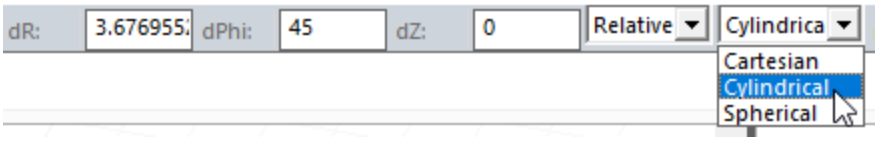


2. Select the center point of the cylinder's base circle in one of the following ways:
   - Click the point. The Measure Data dialog shows the coordinates and other information.

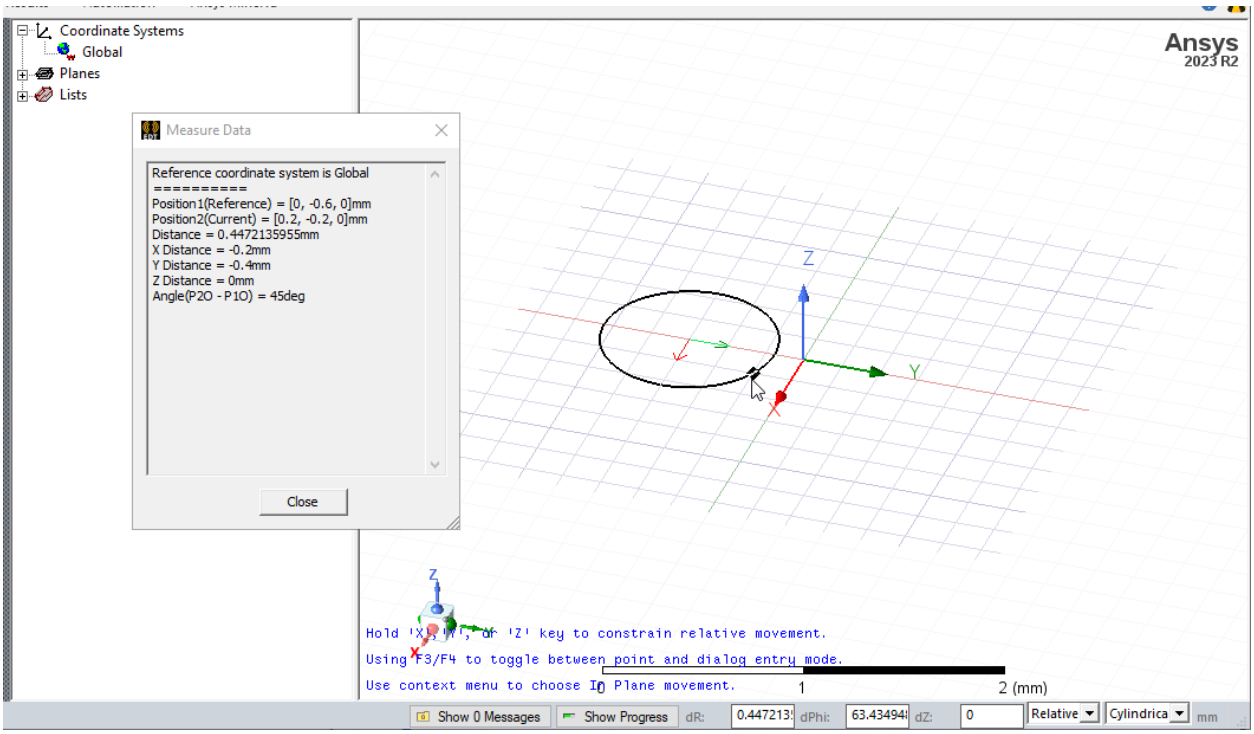


   - Type the point's coordinates in the text boxes in the status bar. The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the cylinder in Cartesian, Cylindrical, or

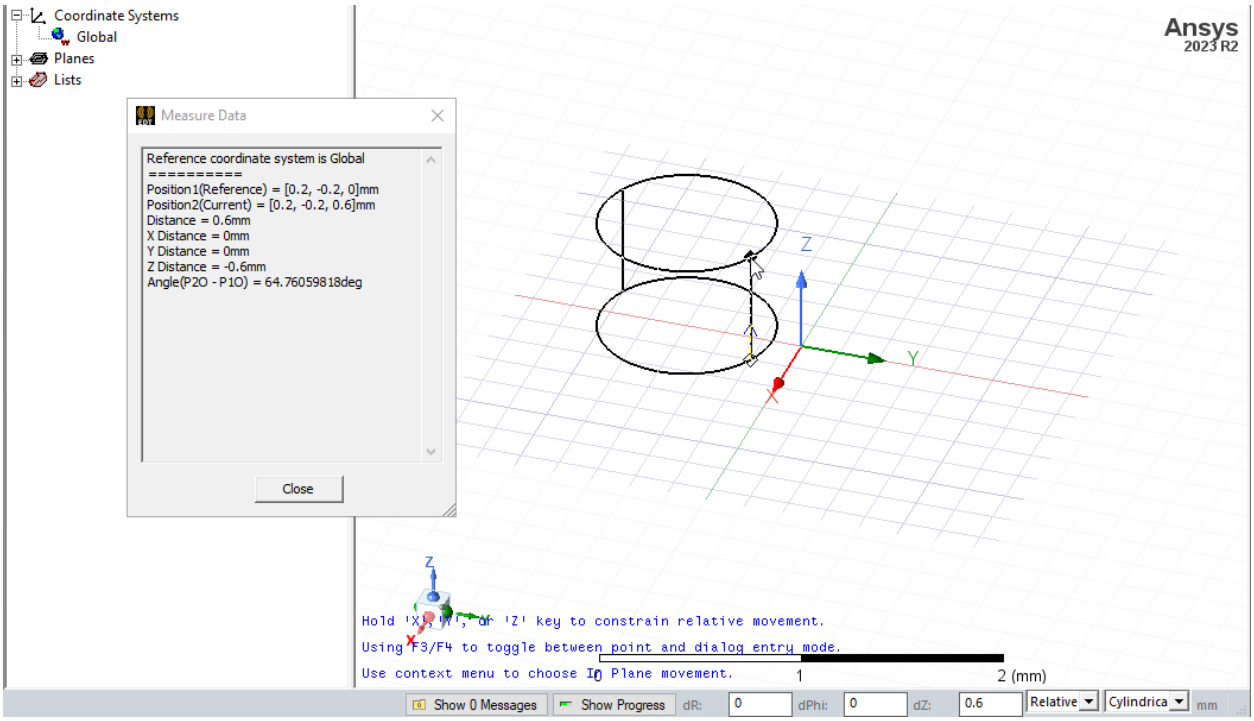
Spherical coordinates.



3. Specify the radius by selecting a point on the base circle's circumference in one of the following ways:
   - Click the point.



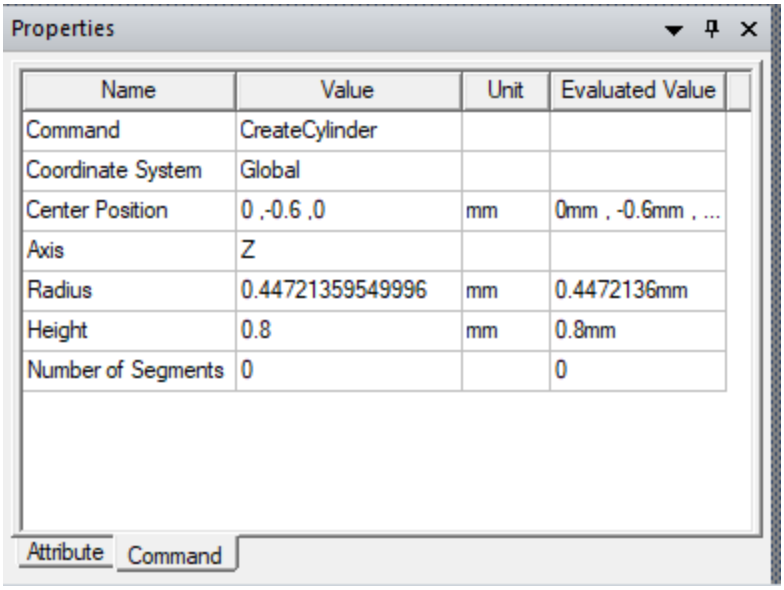
   - Type the coordinates of the point relative to the center point in the **dX, dY,** and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the cylinder's height by selecting a point on the axis perpendicular to the base circle's plane. Select the point by clicking the point or typing the coordinates in the text box.

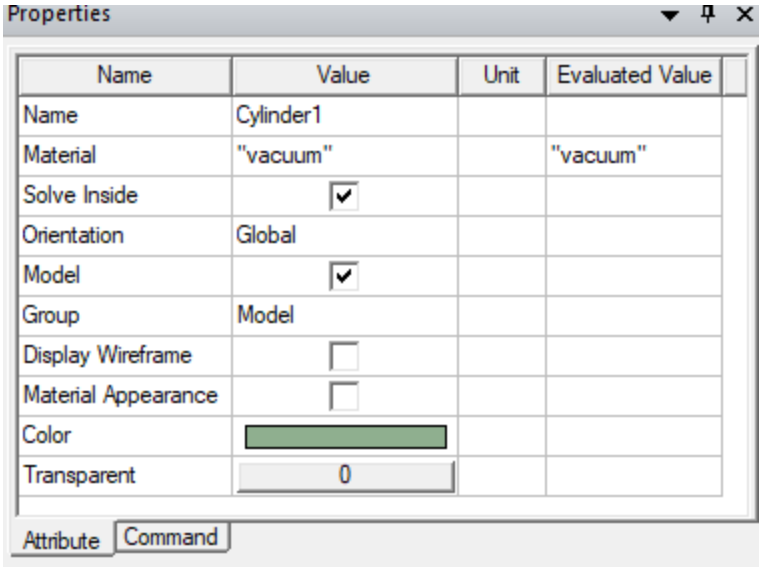


> **Note:**
>
> If you create a cylinder with a height of zero, Ansys Electronics Desktop draws a circular sheet object.

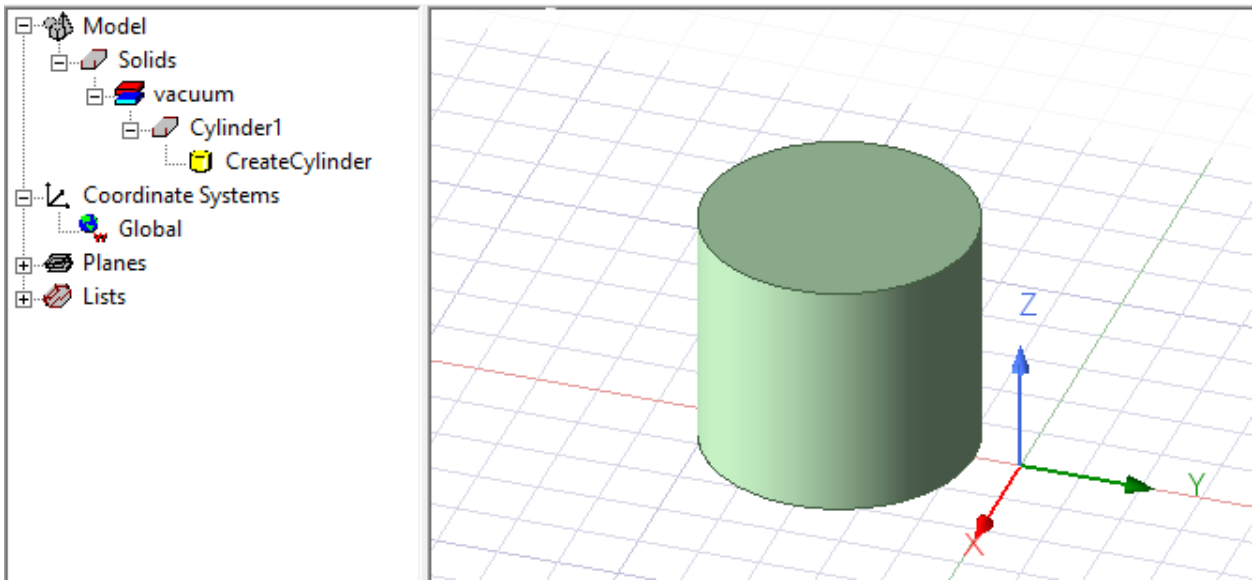
Double click, or right click and select **Done** from the short cut menu. If the Modeler option for editing properties of new primitives is checked, the *Properties* dialog box appears, enabling you to modify the object's Command and Attribute properties. You can also use

the docked Properties window.





5. If you used the Properties window, click **OK**. The cylinder appears in the Modeler window and is listed in the History tree.



Note:

The 3D Modeler permits drawing true curved objects. However, the solution is obtained using 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:

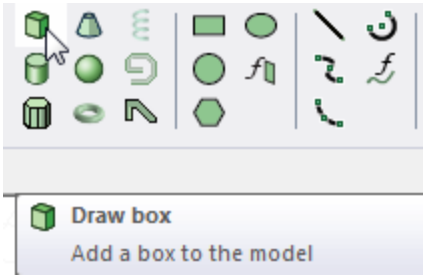
Modifying Surface Approximation Settings and related sections:

Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

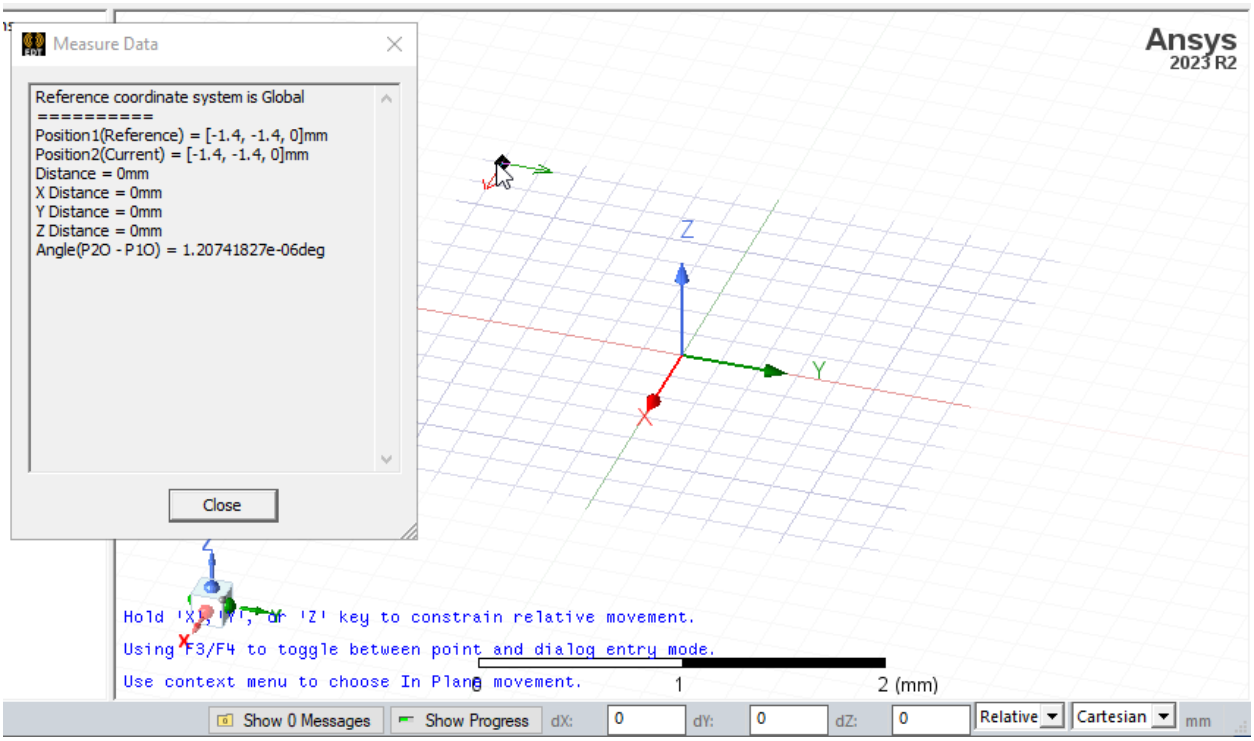
Drawing a Box

Draw a box by selecting two diagonally opposite corners of the base rectangle, then specifying the height. Before you draw a box, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

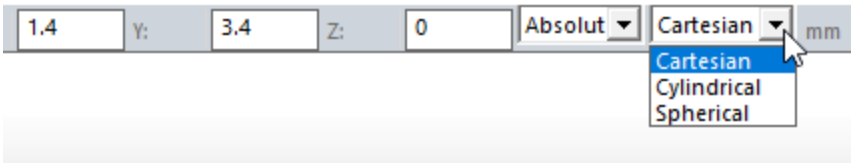
1. From the menu bar, click **Draw**> **Box** or, on the **Draw** tab of the ribbon, click the **Draw box** icon:



2. Select the first diagonal corner of the base rectangle in one of the following ways:
   - Click the point. The Measure Data dialog shows the coordinates and other information.

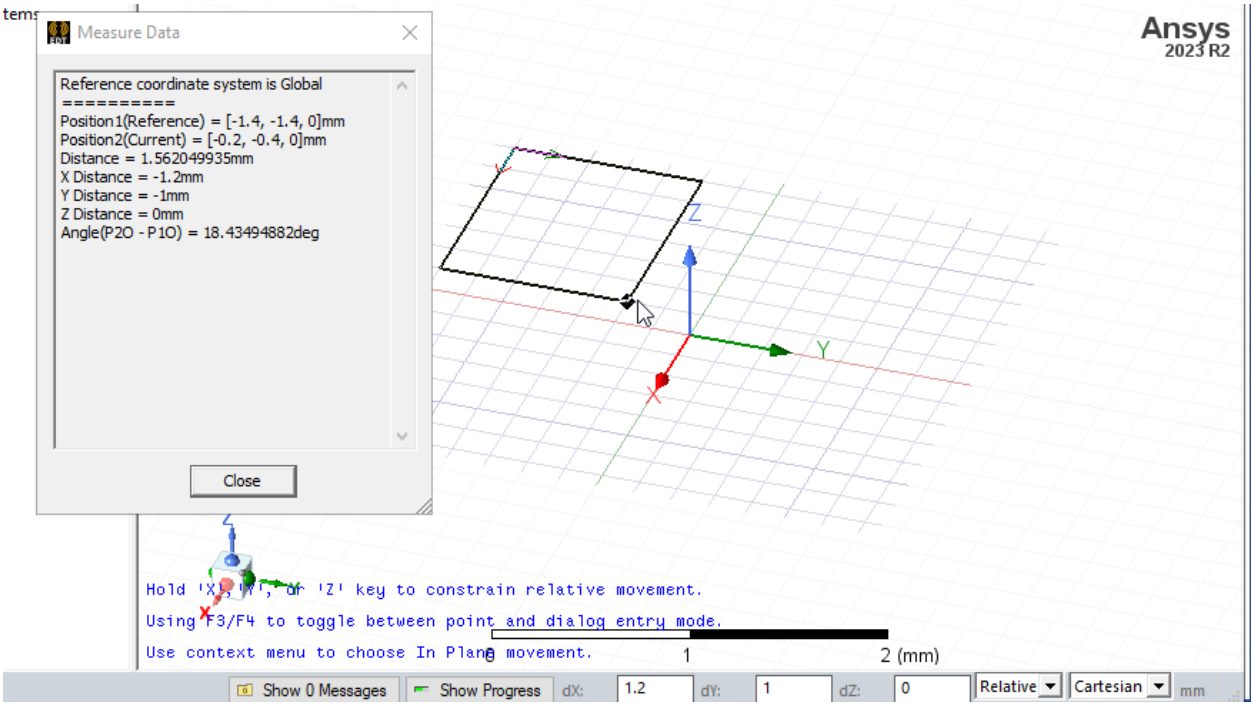


   - Type the point's coordinates in the text boxes in the status bar. The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the box in Cartesian, Cylindrical, or Spherical coordinates.

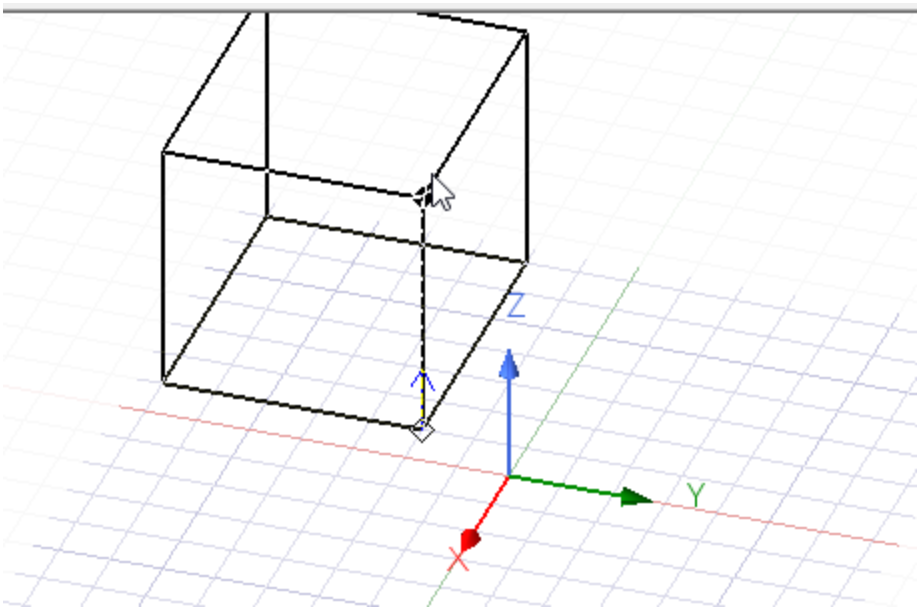


To delete the selected point and start over, press **Esc** or click **Escape Draw Mode** on the shortcut menu.

3. Select the second corner of the base rectangle in one of the following ways:
   - Click the point.

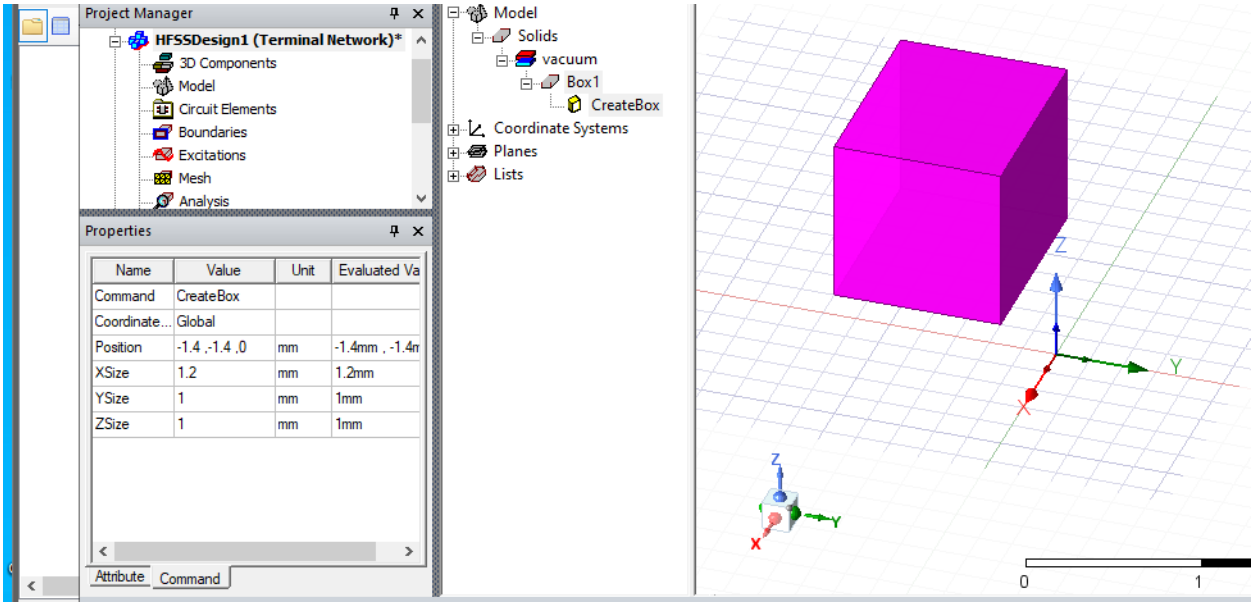


   - Type the coordinates of the point relative to the first diagonal corner in the **dX, dY,** and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the height of the box by selecting a point on the axis perpendicular to the base rectangle. Select the point by clicking the point or typing the coordinates in the **dX, dY,** and **dZ** boxes.



5. Double -click or right-click and select **Done** from the shortcut menu.

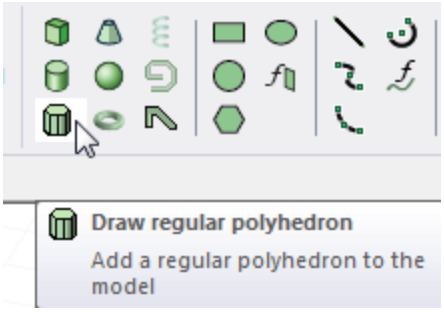
   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties. You can also use the docked Properties window.



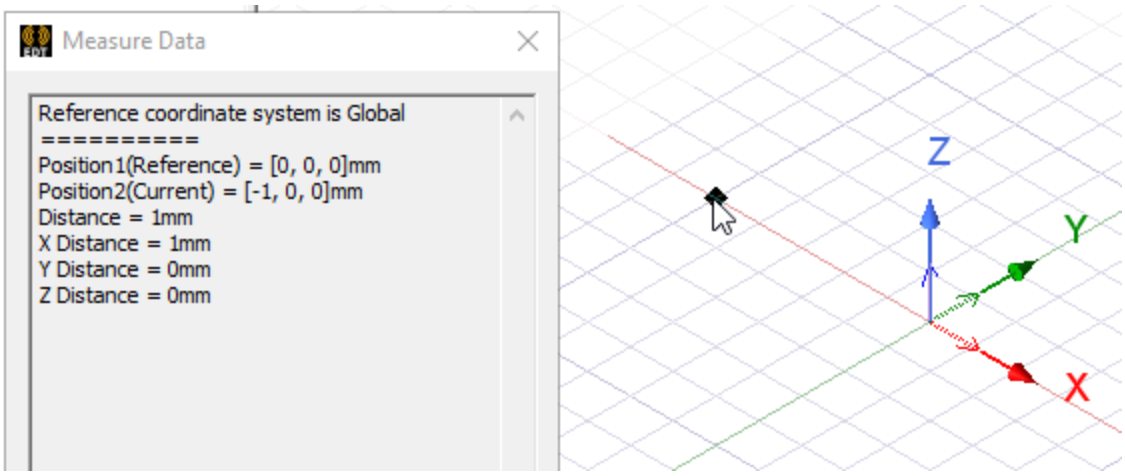
Drawing a Regular Polyhedron

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. Before you draw a regular polyhedron, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

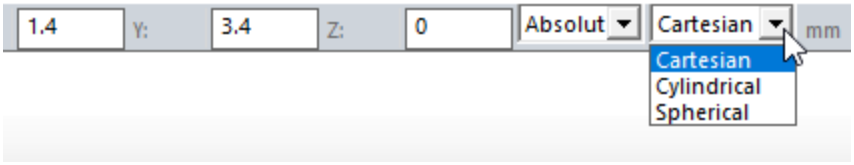
1. From the menu bar, click **Draw**> **Regular Polyhedron** or, on the **Draw** ribbon tab, click the **Draw regular polyhedron** icon:



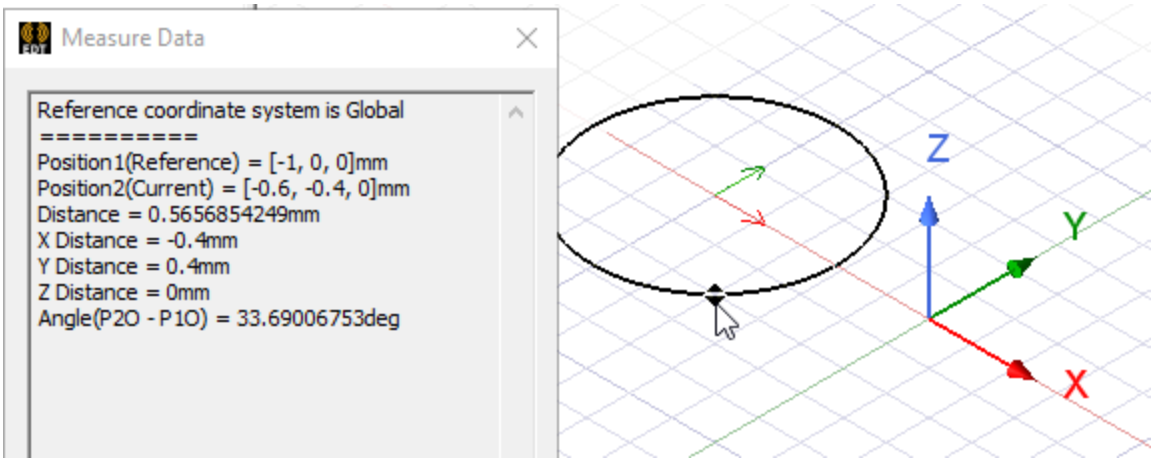
2. Select the center point of the polyhedron in one of the following ways:
   - Click the point.



   - Type the point's coordinates in the text boxes in the status bar.
     The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the regular polygon in Cartesian, Cylindrical, or Spherical coordinates.

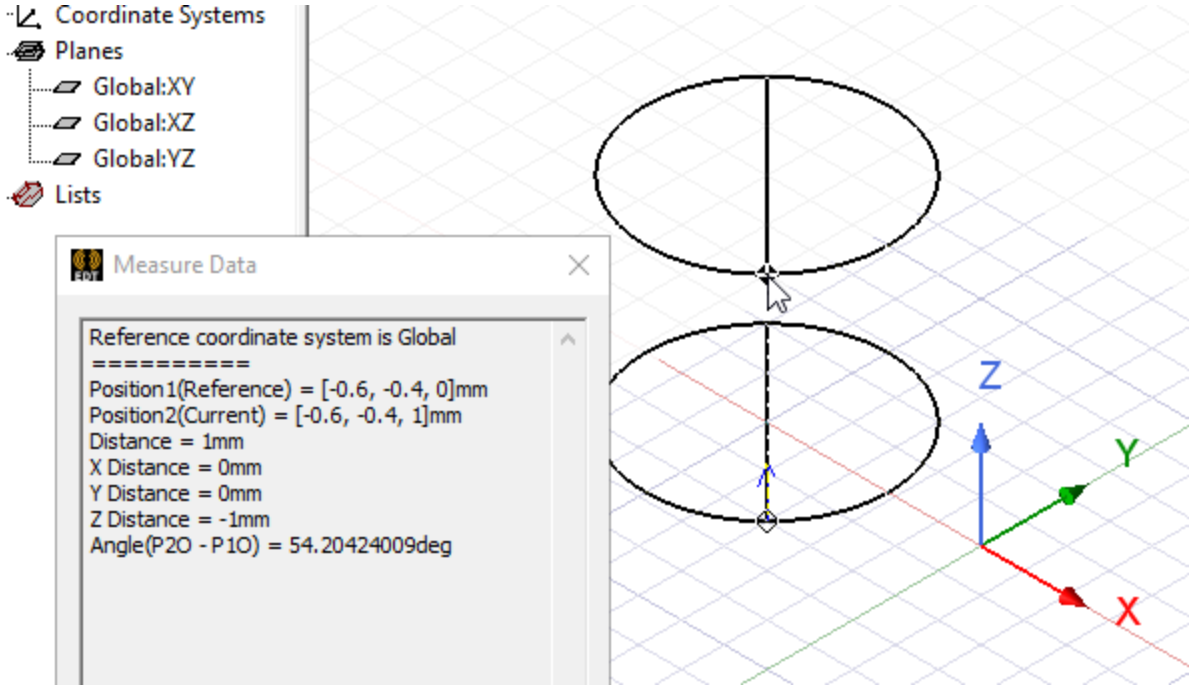


3. Select the radius of the polyhedron, the distance from the center point to one of the polyhedron's vertices, in one of the following ways:
   - Click the point.

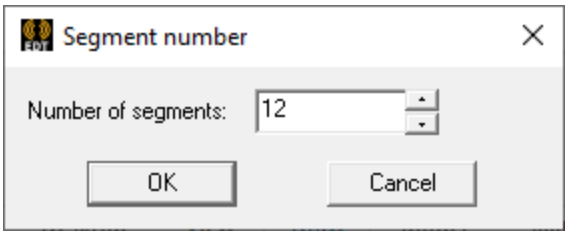


   - Type the coordinates of the point relative to the center point in the **dX, dY,** and **dZ** boxes, where **d** is the distance from the previously selected point.

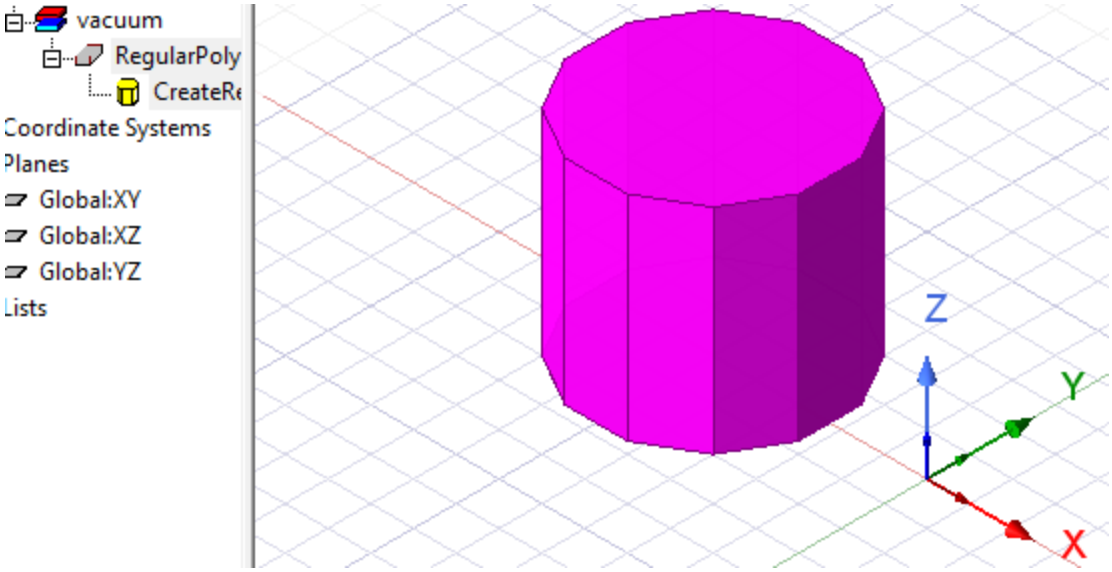
4. Select the height of the polyhedron, the distance from the origin plane along the normal from the radius point.



5. In the **Segment number** dialog box, enter the **Number of segments** in the polyhedron, and then click **OK**.



Click **OK**. The **Segment number** dialog closes and the regular polyhedron appears in the Modeler window and in the History Tree.



6. 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. Click **OK**.

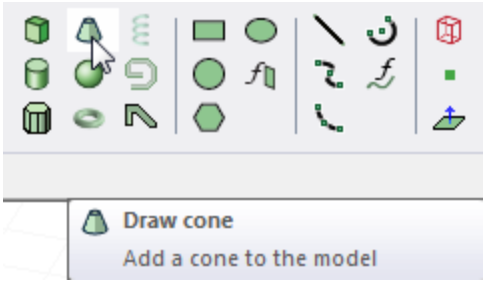
> **Note:**
>
> The radius is measured from the center point to a corner of the polygon, or the intersection of two edges. It is *not* measured from the center point to the midpoint of an edge.

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.

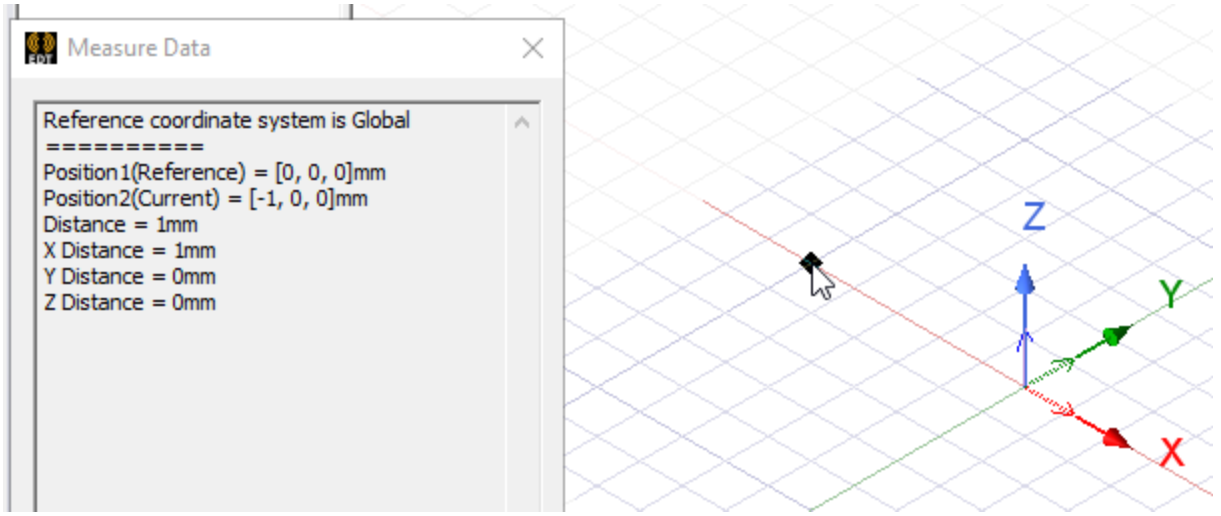
Drawing a Cone

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. Before you draw a cone, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

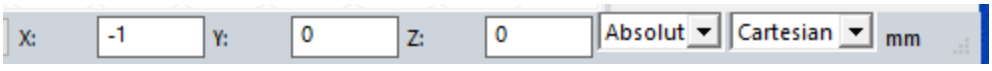
1. From the menu bar, click **Draw**> **Cone** or, on the **Draw** ribbon tab, click the **Draw cone** icon:



2. Select the center point of the cone's base circle in one of the following ways:
   - Click the point.

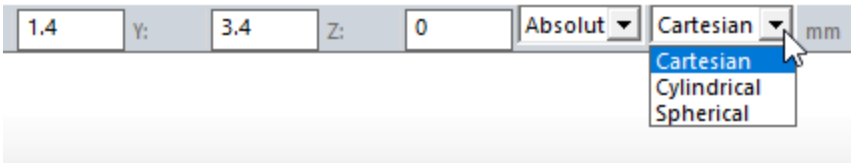


   - Type the point's coordinates in the text boxes in the status bar.

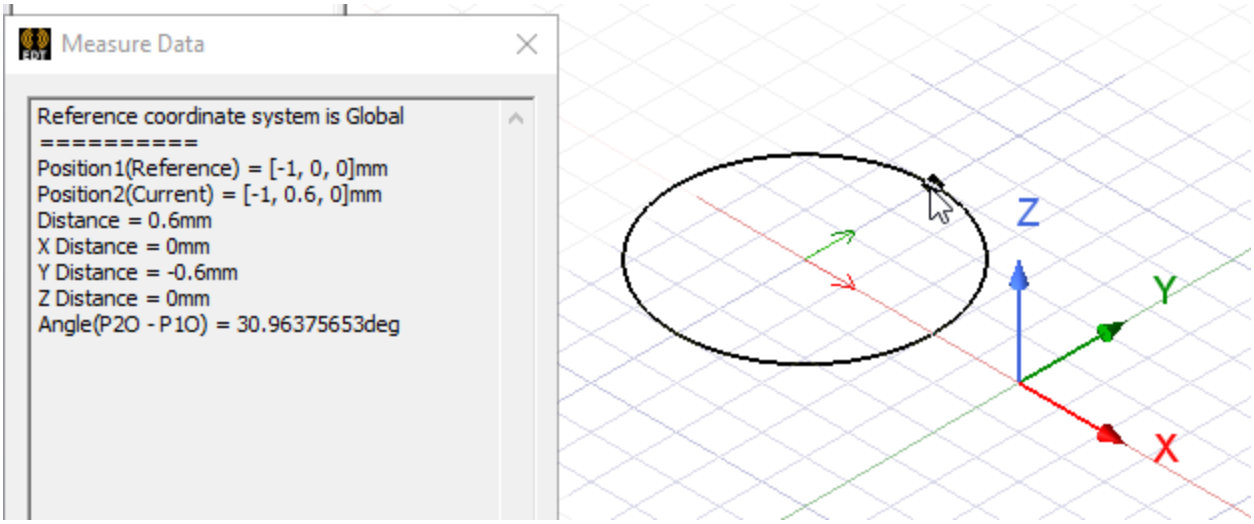


   The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the cone in Cartesian,

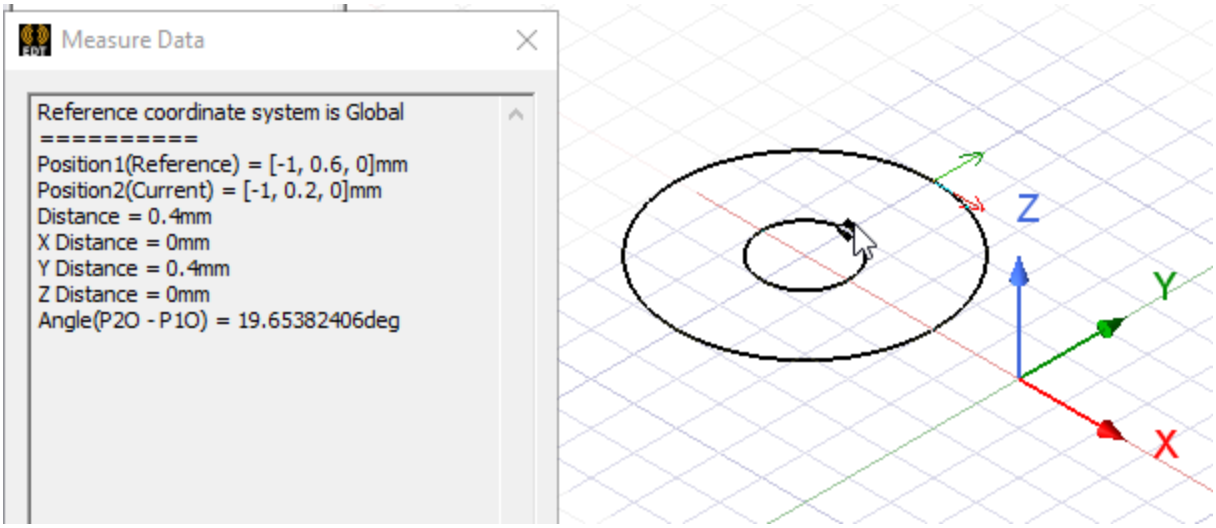
Cylindrical, or Spherical coordinates.



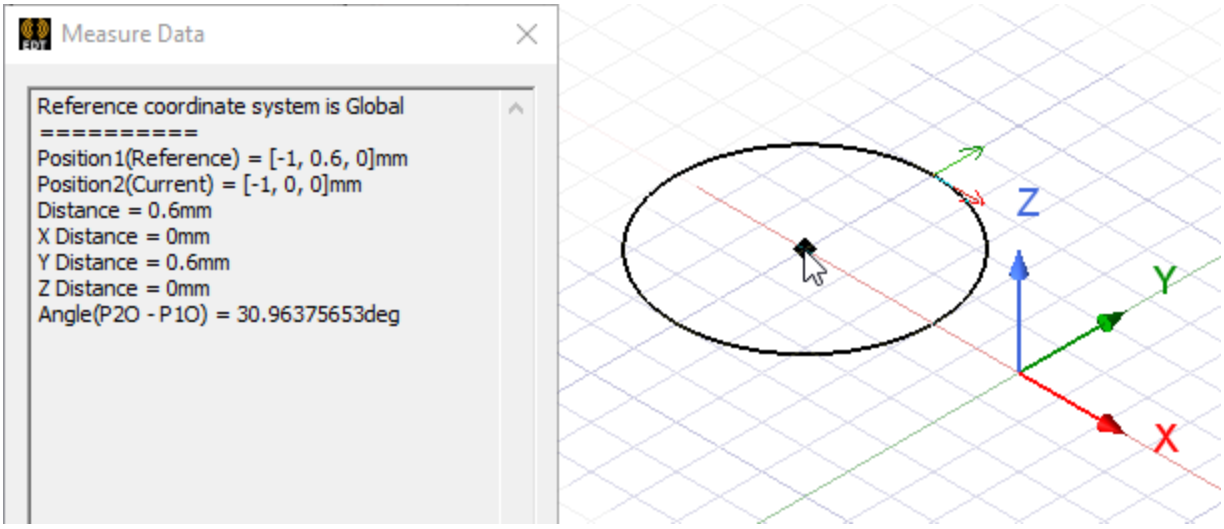
3. Specify the radius of the cone's base circle by selecting a point on the base circle's circumference. Select the point in one of the following ways:
   - Click the point.



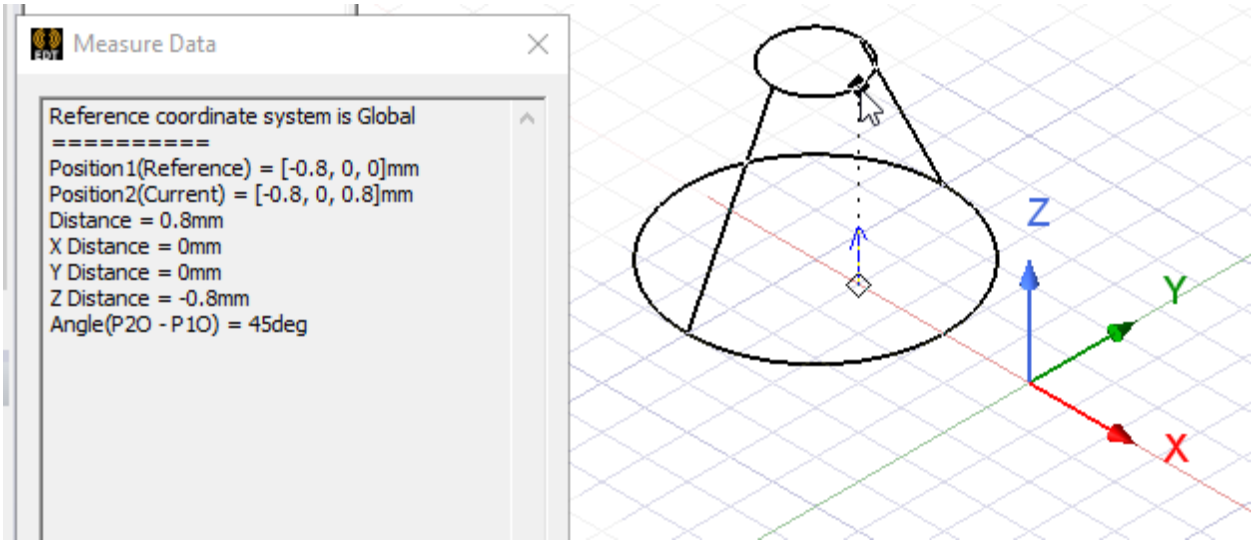
   - Type the coordinates of the point relative to the center point in the **dX, dY,** and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the radius of the cone's top circle by selecting a point on its circumference. Select the point by clicking it or typing its coordinates in the **dX, dY,** and **dZ** boxes.



To create an apex, select the same center point as the cone's base circle.

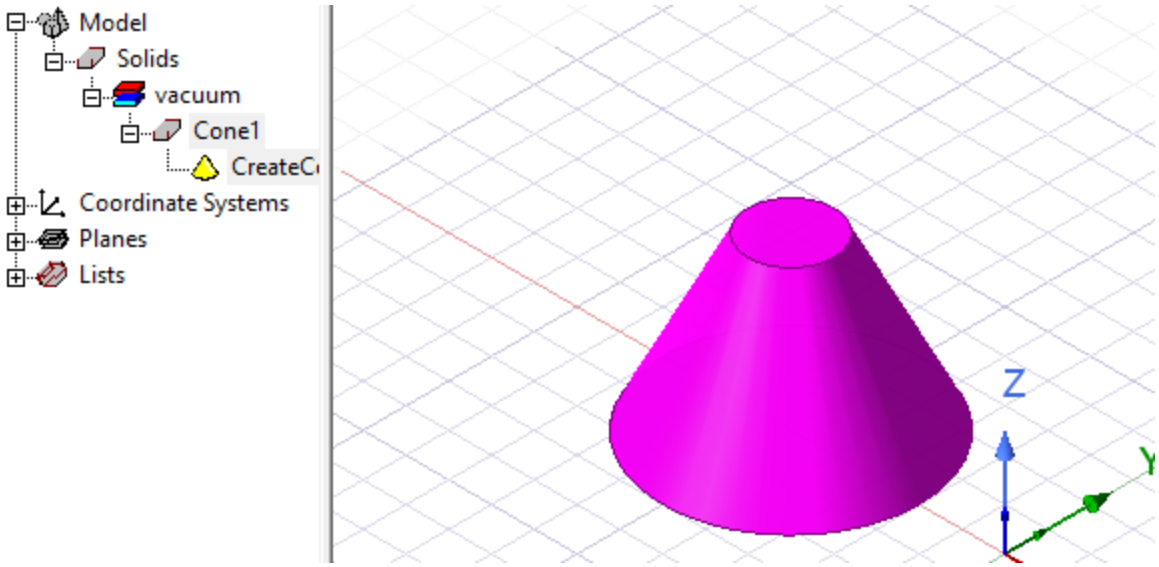


5. Specify the height of the cone by selecting a point on the axis perpendicular to the base circle's plane. Select the point by clicking the point or typing the coordinates in the **dX, dY,** and **dZ** boxes.



6. Double click, or right click and select **Done** from the short cut menu. 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. The cone appears in the Modeler window

and in the History Tree.



Note:

The 3D Modeler permits drawing true curved objects. However, the solution is obtained using 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:

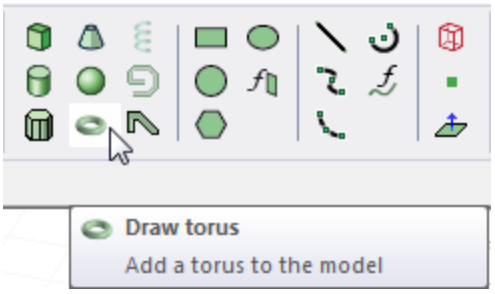
Modifying Surface Approximation Settings and related sections:

Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

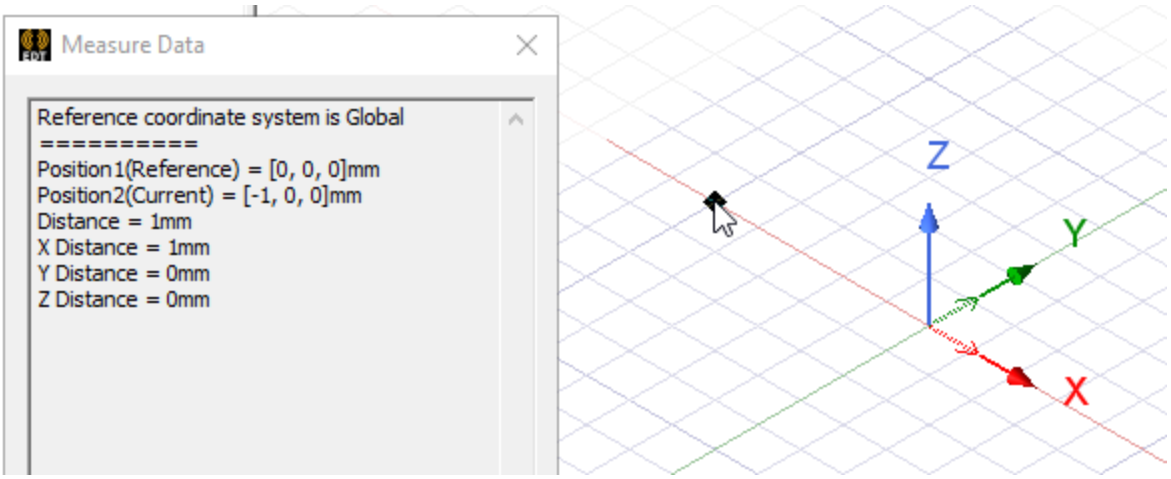
Drawing a Torus

Draw a torus by selecting its center point, major radius, and minor radius. The modeler then sweeps a circle around a circular path. A torus is drawn as true surfaces in the modeler. Before you draw a torus, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Torus** or, on the **Draw** ribbon tab, click the **Draw torus** icon:

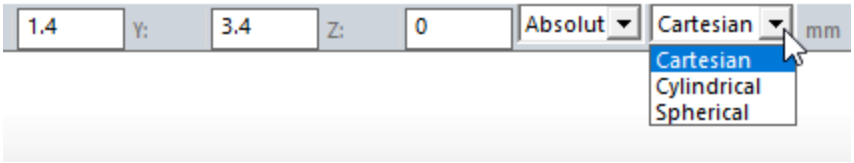


2. Select the center point of the torus in one of the following ways:
   - Click the point.

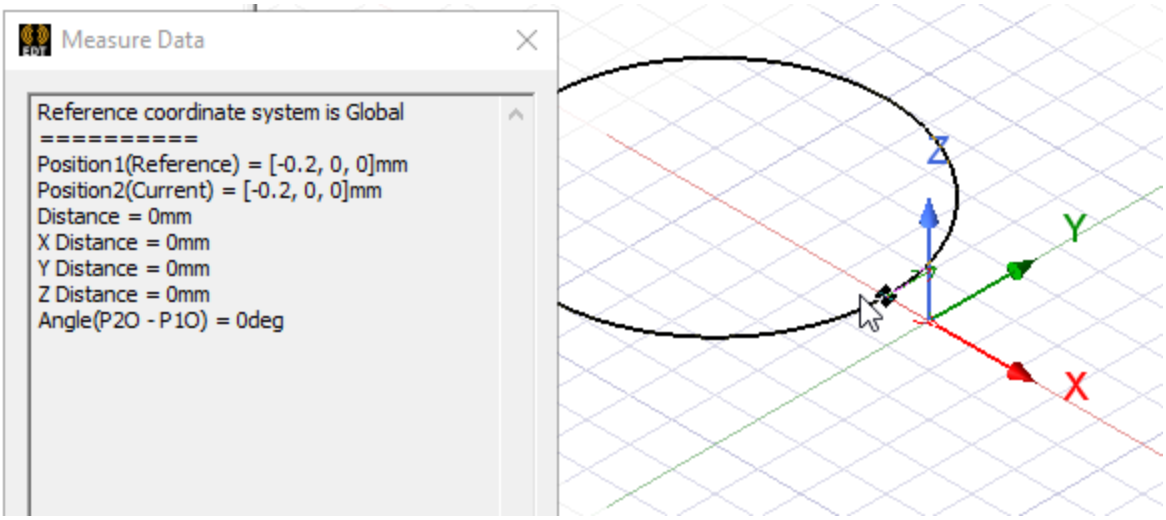


   - Type the point's coordinates in the text boxes in the status bar.

     The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the torus in Cartesian, Cylindrical, or Spherical coordinates.



3. Specify the major radius by selecting a point in one of the following ways:
   - Click the point.

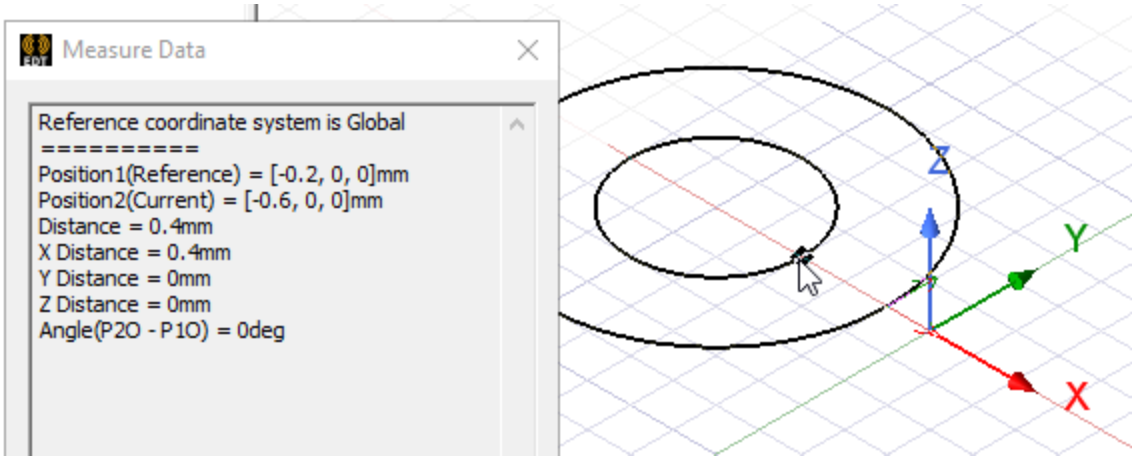


   - Type the coordinates of the point relative to the center point in the **dX, dY,** and **dZ** boxes, where **d** is the distance from the previously selected point.

   The major radius determines the diameter of the torus.

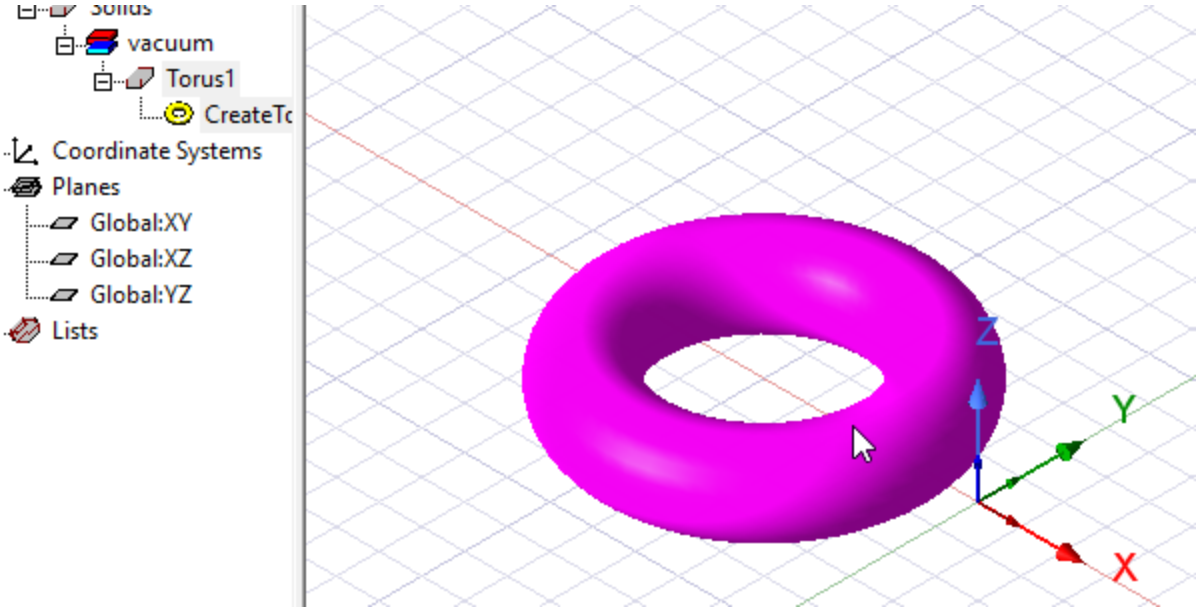
4. Specify the minor radius by selecting a point relative to the major radius point.

   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, enabling you to modify the object's properties.

5. Double-click or right-click to open the shortcut menu and click **Done**. The torus appears in the Modeler window and in the History Tree.



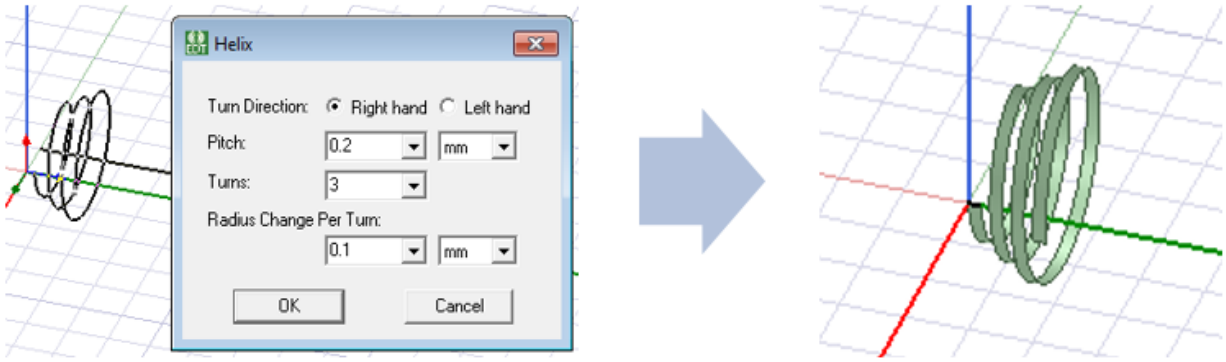
6. If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties. Click **OK**.

> **Note:**
>
> The 3D Modeler permits drawing true curved objects. However, the solution is obtained using 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:
>
> Modifying Surface Approximation Settings and related sections:
>
> Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

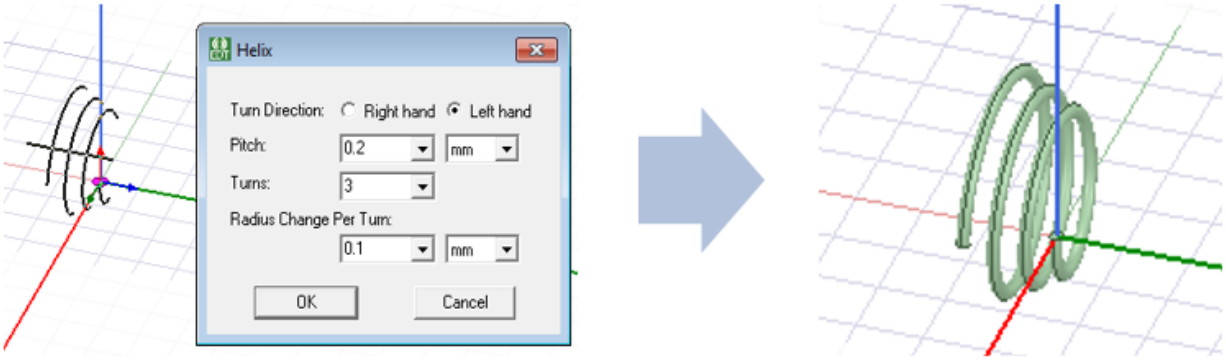
Drawing a Helix

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 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

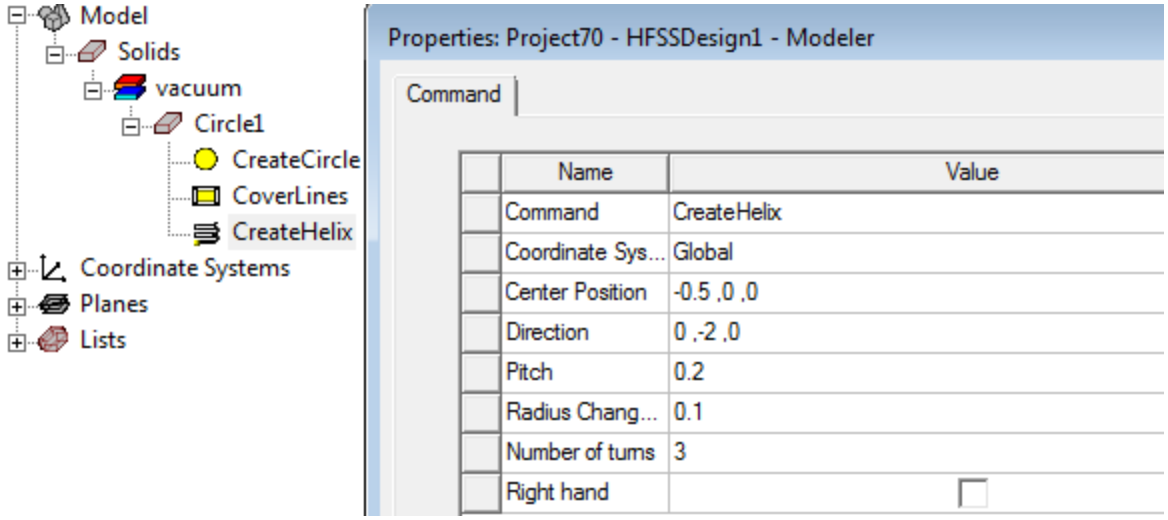
create the vector, the **Helix** dialog opens, where you then specify the turn direction, pitch, number of turns, and radius of the turn to define the helix.



Sweeping a 2D object results in a hollow 3D object. Sweeping a 2D sheet object results in a 3D solid object.

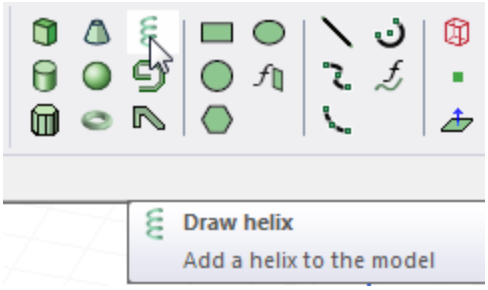


Once you have created a helix, you can changing it by edit the properties of the original object, and by editing the helix command properties for coordinate system, center position (which corresponds to the vector placement), direction, pitch, turns, radius change, and direction.



To create a helix:

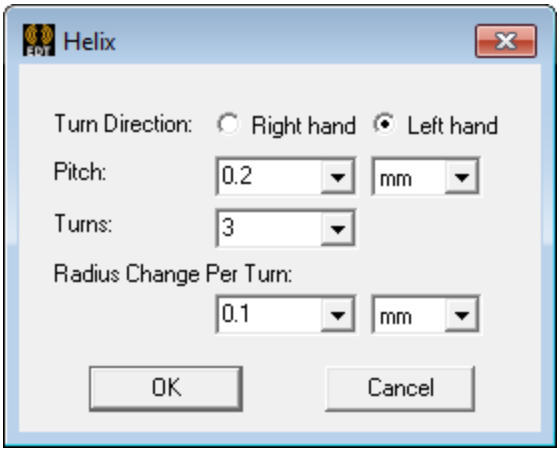
1. Create and select the line or 2D object you want to sweep to form a helix. Selecting a valid object enables the **Helix** icon on the **Draw** tab of the ribbon.
2. From the menu bar, click **Draw**> **Helix** or, on the **Draw** ribbon tab, click the **Draw helix** icon:



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 of 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 drop-down menu.
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 drop-down menu.
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 Modeler permits drawing true curved objects. However, the solution is obtained using 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:

Modifying Surface Approximation Settings and related sections:

Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

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. Before you draw a helix, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. Click **Draw** > **User Defined Primitive** > **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**.

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. Before you draw a segmented helix, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

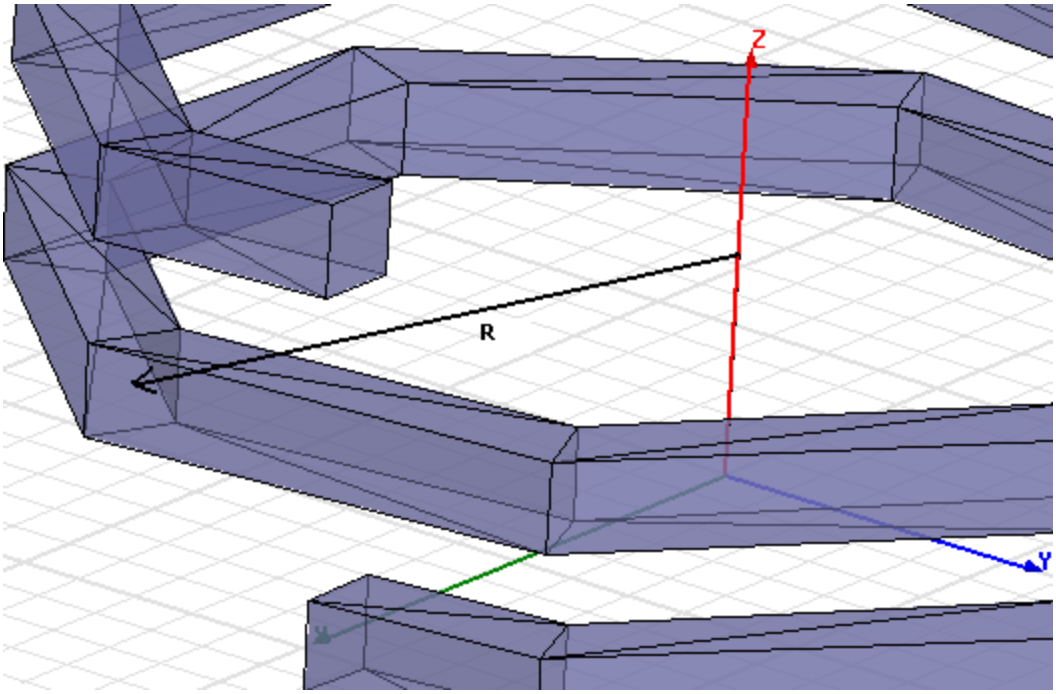
1. Click **Draw** > **User Defined Primitive** > **SegmentedHelix** > **RectHelix**.

   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.

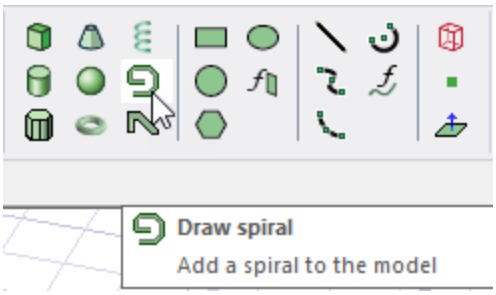
3. Click **OK**.



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. Before you draw a spiral, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. Select the 1D or 2D object you want to sweep to form a spiral.
2. From the menu bar, click **Draw**> **Spiral** or, on the **Draw** ribbon tab, click the **Draw spiral** icon:



3. 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.
   The Status bar also includes options to specify the Coordinate System as Absolute or Relative, and drop down menu options to specify the sphere in Cartesian, Cylindrical, or Spherical coordinates.

   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.

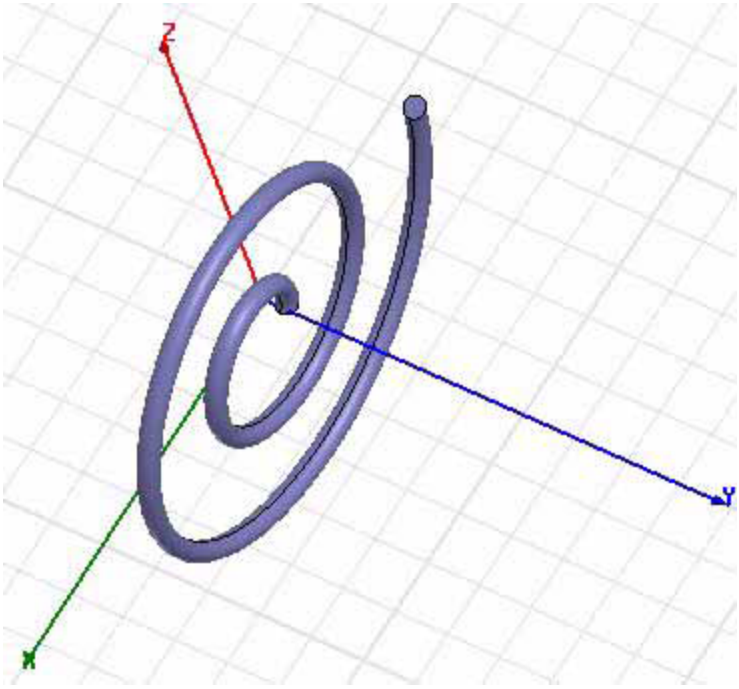
4. Select **Right hand** if the turn direction is clockwise and **Left hand** if the turn direction is counter-clockwise.
5. In the **Radius Change** text box, type the difference in radius between each turn of the spiral.

   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.

6. Click a unit for the radius in the pull-down list.
7. 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, enabling you to modify the object's properties.

8. Click **OK**.



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 Modeler permits drawing true curved objects. However, the solution is obtained using 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:
>
> Modifying Surface Approximation Settings and related sections:
>
> Rectilinear Elements and Curvilinear Elements and Guidelines for Modifying Surface Approximations.

Drawing a Spiral Using User Defined Primitives

Ansys provides you with a DLL and a Python script to define the parameters of a rectangular spiral. Before you draw a spiral, you can specify the coordinate system, and you can set the

drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw** > **User Defined Primitive** > **Examples** > **RectangularSpiral (DLL)** or **Rectangular Spiral (Python)**.

   The **User Defined Primitive Operation** dialog box appears. 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 following parameters:

Xpos	X location of the starting point.
Ypos	Y location of the starting point.
Dist	The separation distance between turns.
Turns	The number of complete revolutions the object will make around the vector
Width	The width of the spiral.
Thickness	The thickness/height of the spiral. If you specify the thickness as zero, the modeler draws a sheetobject.

3. Click **OK**.

   This creates the primitive and displays the **Properties** dialog box for the new object.

Tip:

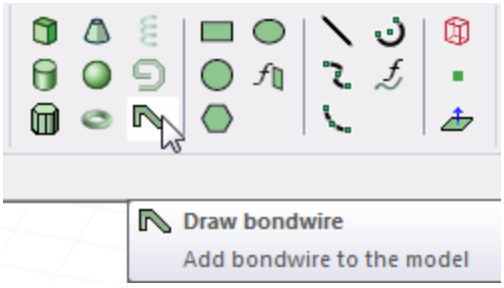
To see newly created DLLs, click **Draw** > **User Defined Primitive** > **Update Menu**.

To see the primitives that you have created, click **Draw** > **User Defined Primitive** > **UserLib.**

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 *HFSS: Technical Notes* before drawing a bondwire.

1. From the menu bar, click **Draw**> **Bondwire** or, on the **Draw** ribbon tab, click the **Draw bondwire** icon:



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.

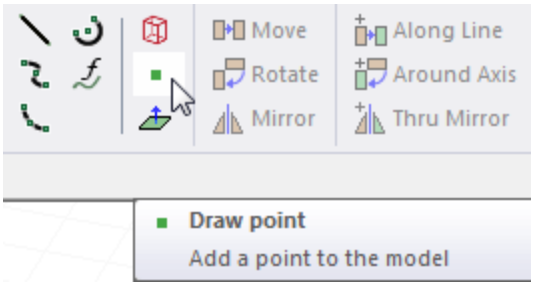
   The minimum value is 3. The value describes the number of faces that make up the circumference of the bondwire.
6. In the diameter field, specify a diameter value and select the units from the drop-down menu.
7. Enter the height between the bond pad and the top of the loop in the **h1** text box. Include 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 Ansys Electronics Desktop 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**.

Drawing a Point

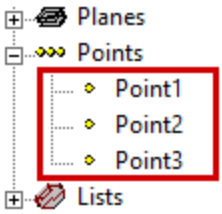
Drawing a point object within the problem region enables you to plot fields or perform field computations at that point. Points are always considered non-model objects by the modeler. Before you draw a point, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Point** or, on the **Draw** ribbon tab, click the **Draw point** icon:



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:

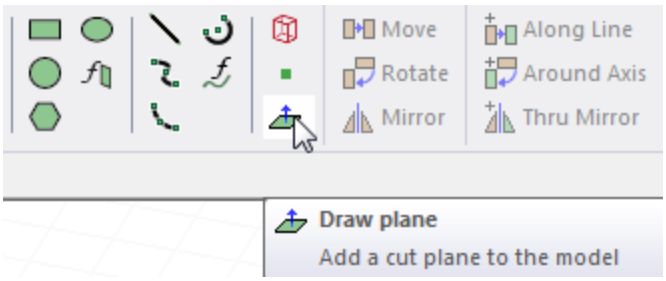


> **Note:**
>
> There is no visible indication of the point in the drawing area unless it is selected in the History Tree. Also, the point cannot be selected graphically once added to the model. It must be selected from the History Tree or from the **Geometry** window within the **Fields Calculator**.

Drawing a Plane

A plane object is a cut plane through the problem region. You can use plot fields to perform field computations on its surface. Planes are always considered non-model objects by the modeler. Before you draw a plane, you can specify the coordinate system, and you can set the drawing plane as Z, Y, or Z, or you can edit the plane in the properties.

1. From the menu bar, click **Draw**> **Plane** or, on the **Draw** ribbon tab, click the **Draw plane** icon:

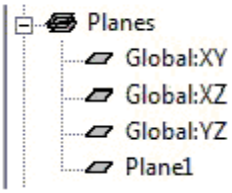


2. Select the origin in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   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.

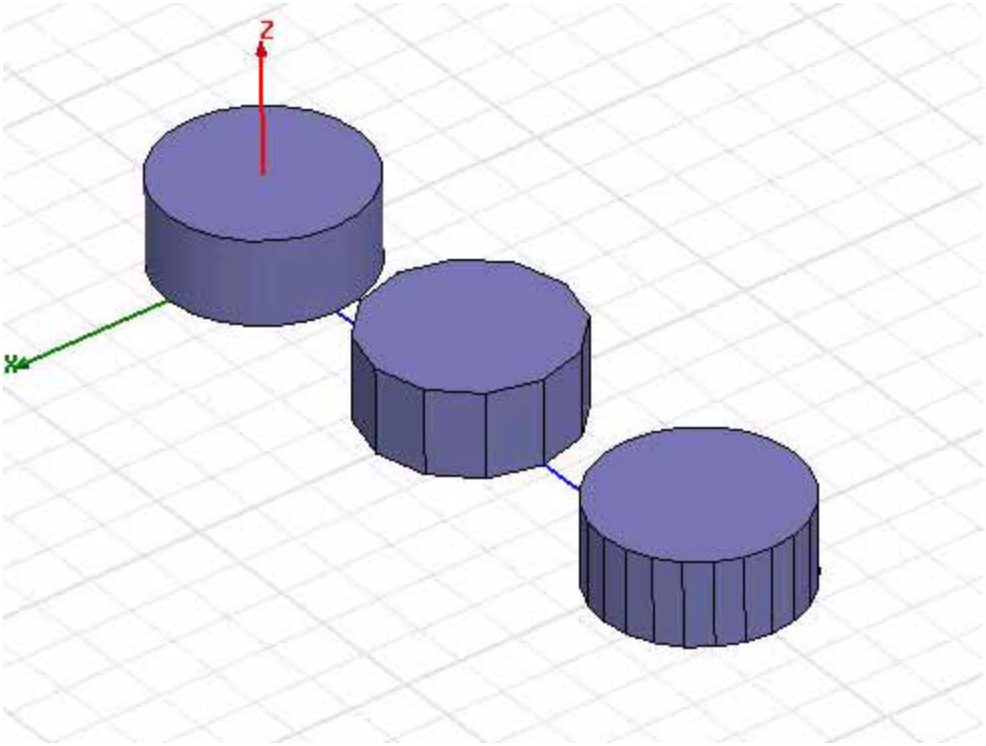
   The plane is created. Its center point is located at the origin you specified and oriented perpendicular to the normal point you specified. The new plane is listed under **Planes** in the History Tree.



> **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.

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.



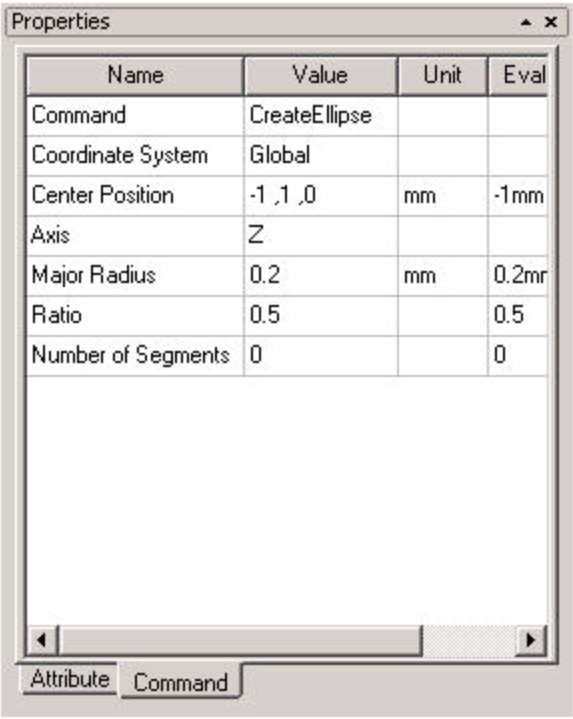
The following model objects can be created as segmented structures:

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.

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.

You can assign output variables (default) and design variable as property values for non-model objects.

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 in the reporter 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, which is always a non-model object, in order to plot fields in the Reporter 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.

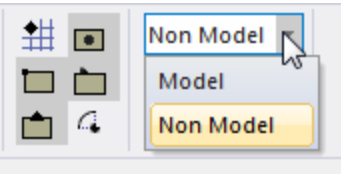
Modify an existing model object to be a non-model object.

Selecting Non-Model Drawing Mode

To switch to non-model drawing mode:

1. Click **Modeler**> **New Object Type**> **Non Model**.

   Alternatively, on the **Draw** ribbon tab, choose **Non Model** from the **Set to model non model type** drop-down menu:

2. Draw the object.

Changing an Object to Non Model

To modify an existing object to be a non-model object:

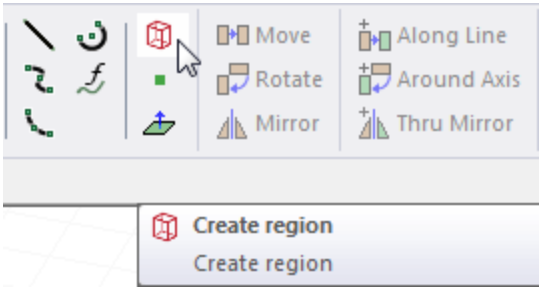
1. Select the object you want to modify.
2. In the docked **Properties** window, clear the **Model** option.

   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.

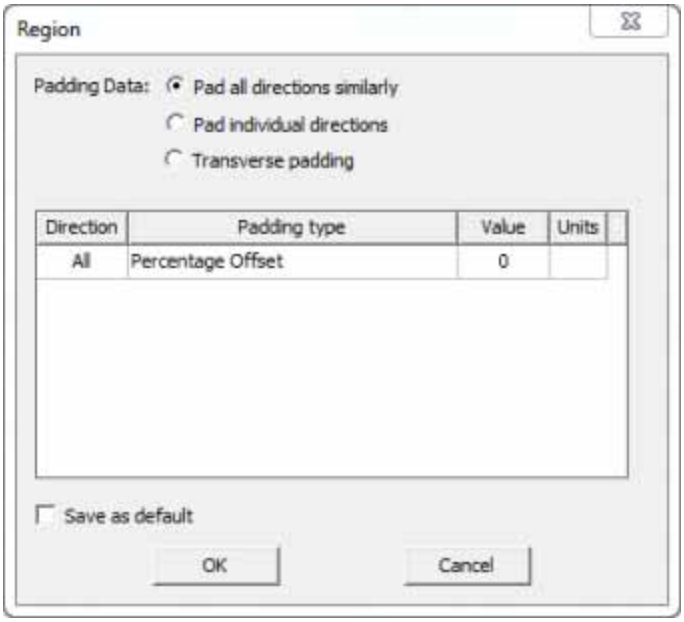
Drawing a Region

To draw a region encompassing the objects in the current project:

1. From the menu bar, click **Draw** > **Region** or, on the **Draw** ribbon tab, click the **Create region** icon:

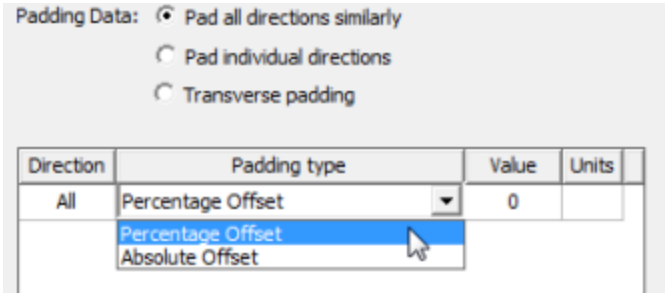


This displays the **Region** dialog box.

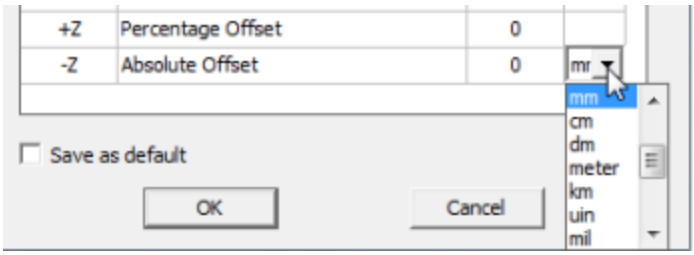


2. For the Padding data, click the Padding Data radio button as **Pad all directions similarly, Pad individual directions**, or **Transverse padding**.

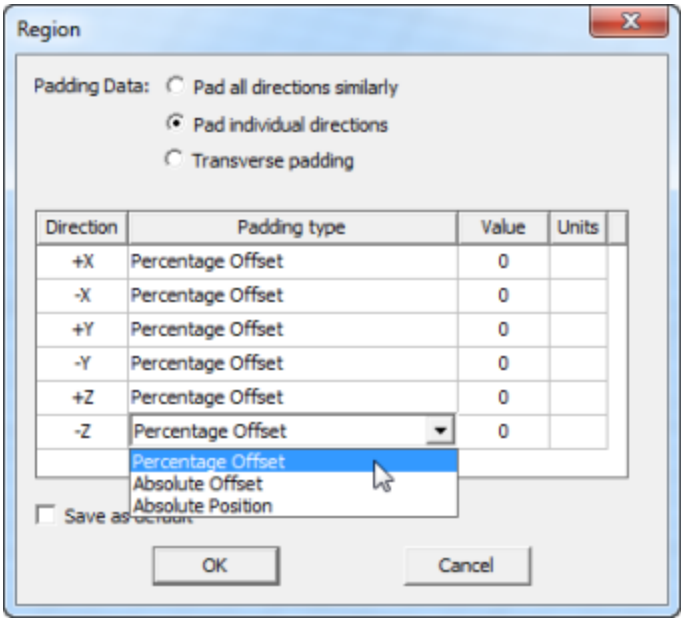
   Selecting **Pad all directions similarly** leaves the Padding type field as requiring a single value that affects all directions. In this case, you can specify the Padding type by selecting **Percentage Offset** or **Absolute Offset** from the drop-down menu.



If you select **Absolute Offset**, you also specify the units by selecting from a drop-down menu.



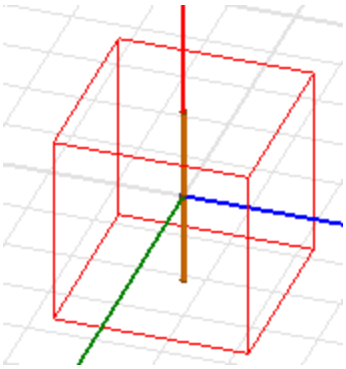
Selecting **Pad individual directions** displays the Padding Percentage as a table of Positive and Negative X,Y, and Z coordinates, permitting you to specify padding for each direction. In this case, you can specify the Padding type by selecting **Percentage Offset** or **Absolute Offset** or **Absolute Position** from the drop-down menu.



Selecting **Transverse padding** means that padding in one direction is controlled by dimensions in other two directions. Each direction (for example, X) is padded with user-specified percentage of diagonal length of the other two directions (Y and Z).

For a single antenna element, something that is radiating does not radiate with reasonable efficiency until it is lambda/2 in length, L, and it radiates in the direction transverse to the dimension L. So we get the dimension L from the region command and pad it 50% which would then correspond to the lambda/4 of the frequency at which the device is lambda/2 but only in the direction in which energy radiates, the transverse direction.

A basic example is a thin resonate dipole oriented in z. There is zero radiation in the z-direction so no need to pad the abc in that direction since no energy goes in that direction. But in X and Y transverse to the length L we do need to pad lambda/4 (the 50% setting) because it is that direction that the device radiates.



3. Specify the Padding values in the fields and select the units from the drop-down menu.
4. If desired, select 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 Create Region icon in the History tree shows the coordinate system and Padding values. The Properties dialog for the selected model lists attributes tab including Name, Material (Default, vacuum), 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 modeler automatically expands the region 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.

Related Topics

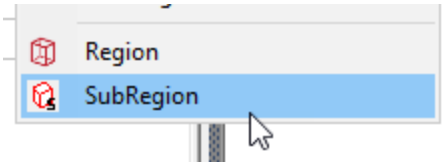
Drawing a Sub-Region

Drawing a SubRegion

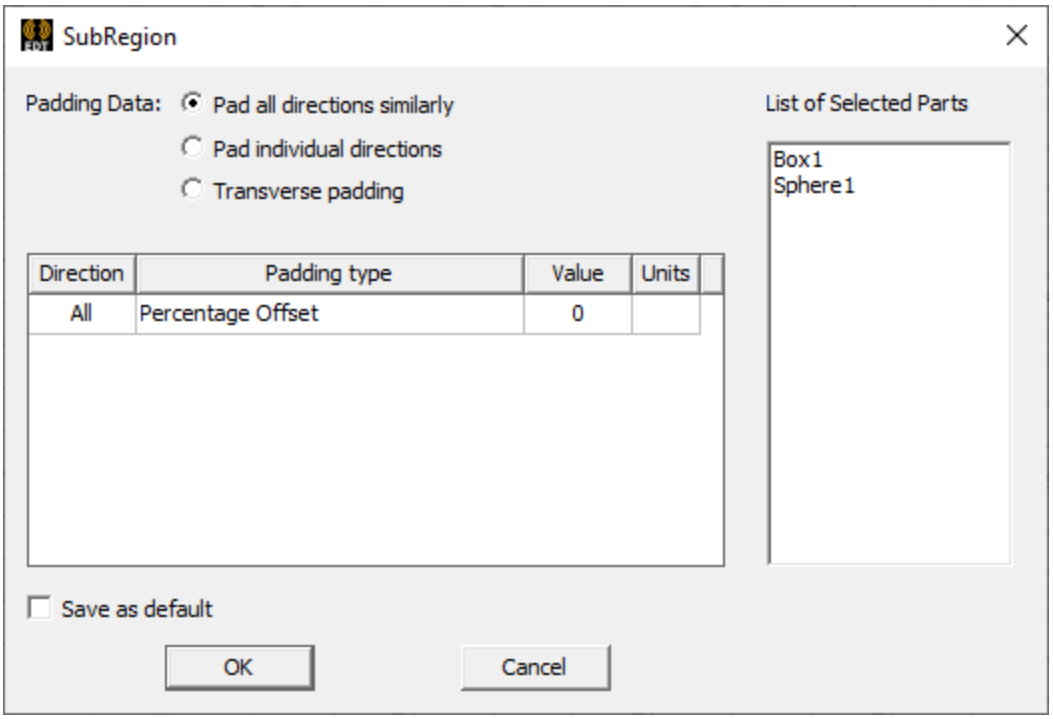
The Modeler allows you to define a subregion that encompasses a set of objects inside previously defined region. Like regions, subregions automatically resize if the objects contained in these subregions change. You can create nested subregions, and reassign the content of a subregion.

To draw a sub-region in the current project:

1. Select the parts of components that you want to include in a subregion.
2. From the menu bar, click **Draw** > **SubRegion**:

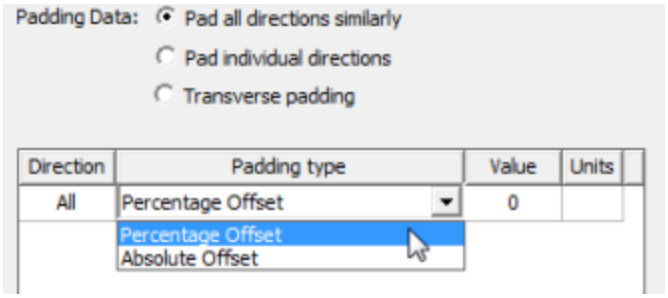


This displays the **SubRegion** dialog box.

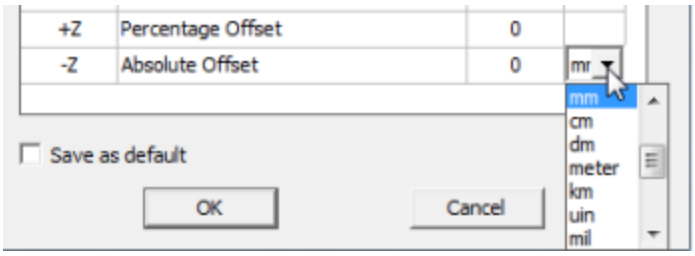


3. For the Padding data, click the Padding Data radio button as **Pad all directions similarly, Pad individual directions**, or **Transverse padding**.

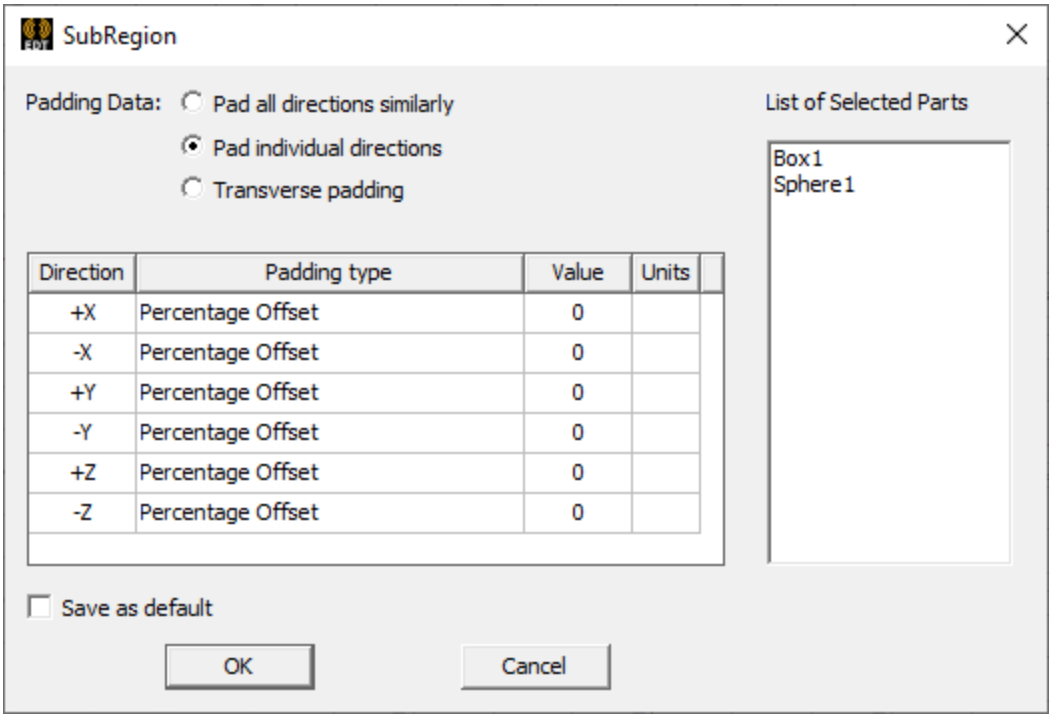
   Selecting **Pad all directions similarly** leaves the Padding type field as requiring a single value that affects all directions. In this case, you can specify the Padding type by selecting **Percentage Offset** or **Absolute Offset** from the drop-down menu.



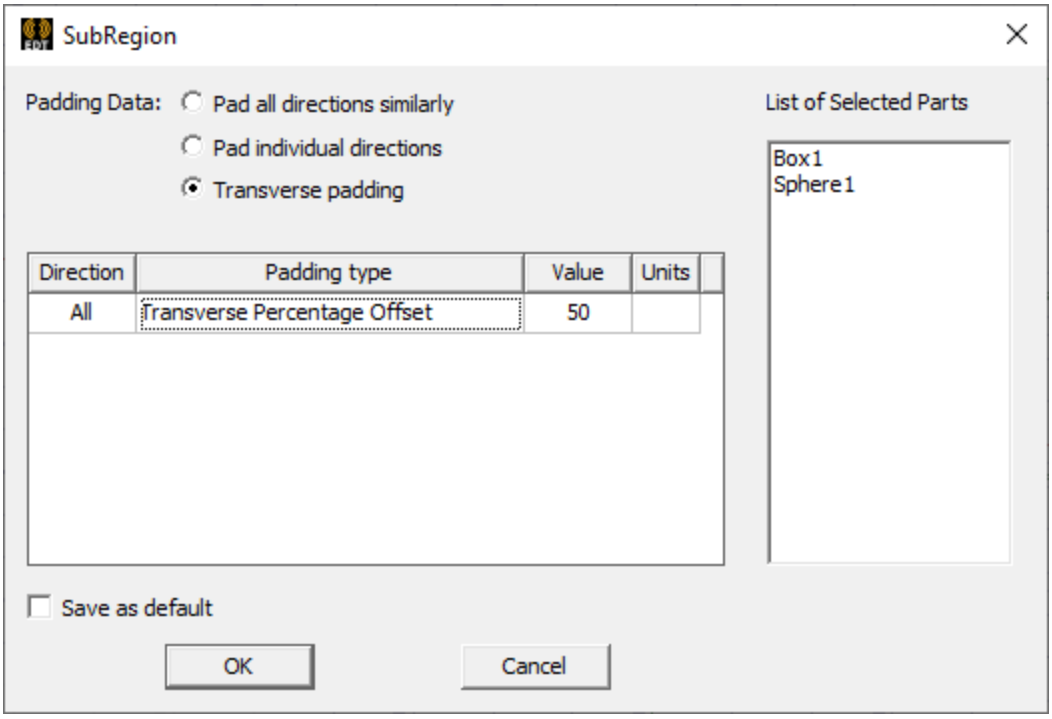
If you select **Absolute Offset**, you also specify the units by selecting from a drop-down menu.



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** or **Absolute Offset** or **Absolute Position** from the drop-down menu.

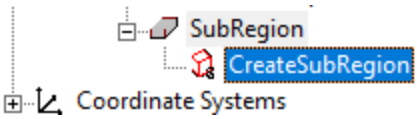


Selecting **Transverse padding** means that padding in one direction is controlled by dimensions in other two directions. Each direction (for example, X) is padded with user-specified percentage of diagonal length of the other two directions (Y and Z).



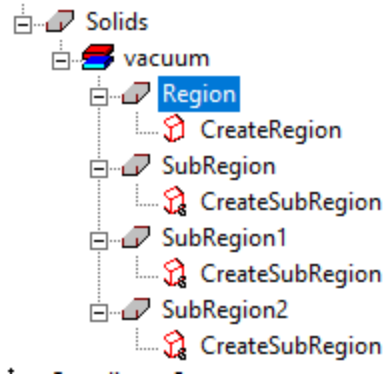
4. Specify the Padding values in the fields and select the units from the drop-down menu.
5. If desired, select the check box to save the values as Default.
6. Click **OK** to close the dialog and create the sub-region.

The sub-region is drawn, selected, and displayed in the History tree. It is created using the current coordinate system.

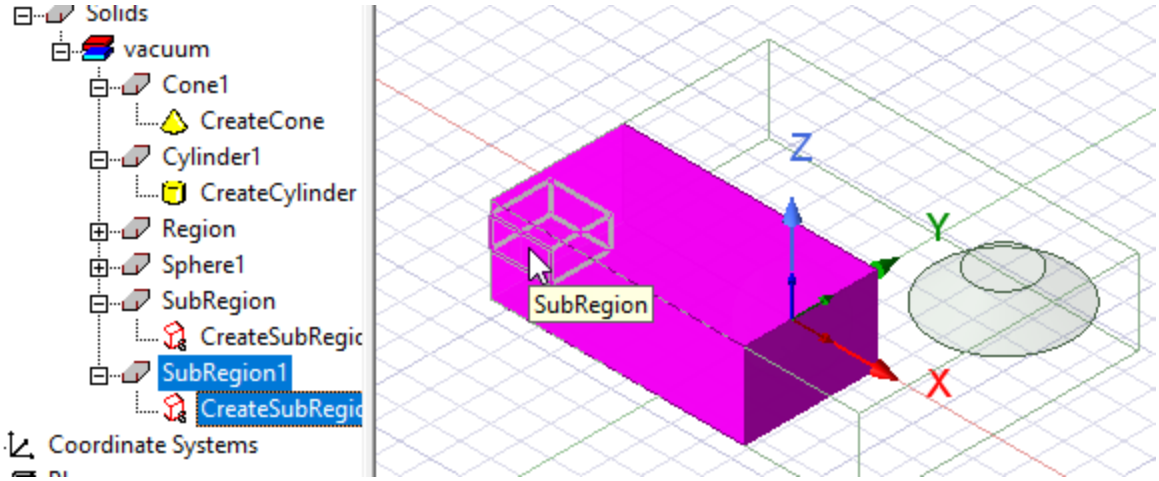


Additional SubRegions and Nested SubRegions.

If you create additional sub-regions, these appear in the History Tree.

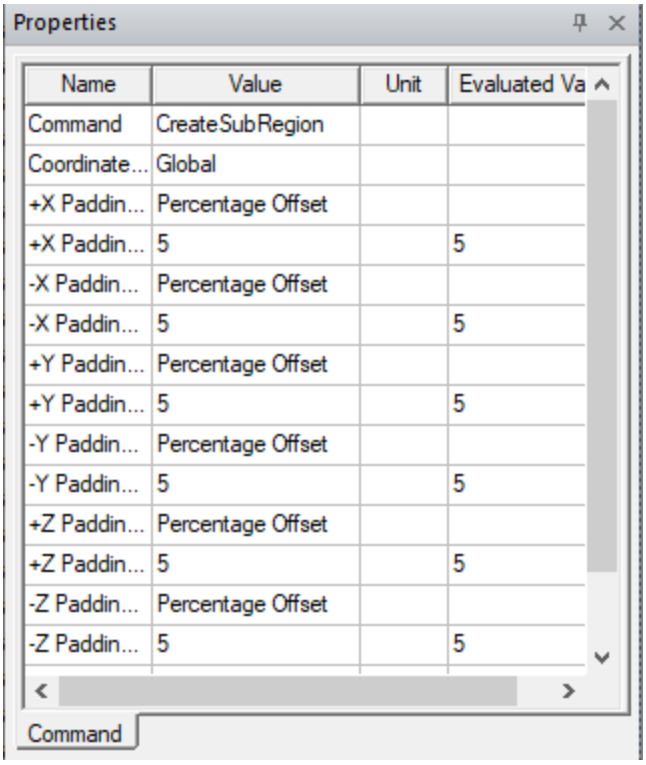


SubRegions can contain other SubRegions, that is, SubRegions can be nested.



Properties for SubRegions

The Properties dialog for the Create SubRegion icon in the History tree shows the coordinate system and Padding values.



The Properties dialog for the selected model lists attributes tab including Name, Material (Default, vacuum), Orientation, Model, Color, Display Wireframe, and Transparency. You can edit all of these values.

You can use modeling operations on these sub region parts with following rules

- Geometry of non-subregion parts should not depend on sub-region parts. That is, sub-region parts should not be allowed to be used as a tool, not even after cloning them
- Simple local operations like face move, fillet, and arrange operations should be allowed on sub-region parts.

Global region should take the sub-regions into account in its computation.

Sub-Regions can be included in components.

Model Analysis

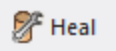
For some models it may be beneficial to remove unnecessary small entities and to fix object misalignments to avoid potential mesh issues. The Modeler window includes Model Analysis functions to help you evaluate models you have imported or created. Select **Modeler** > **Model Analysis** to see the menu 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**
- **Show Analysis dialog**

> **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.

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.

1. Select the objects and invoke **Modeler** > **Model Analysis** > **Analyze Objects**; or, on the **Model** ribbon, click **Heal**.

   This displays the **Analysis Options** dialog box, which allows 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 box is displayed.

2. Choose the objects that have "Invalid Entities Found" and **Perform** > **Heal Objects**.

   In most cases, the objects will be healed and the errors fixed.

3. 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: (See Error Detection.)

- Non-manifold topology. These are non-manifold edges and vertices that are present in the model.
- Object pair intersection. This detects whether pairs of objects intersect.
- Small feature detection - small edge length, small face area and sliver face detection.
- Misaligned entities detection - detects pairs of faces from objects that can be aligned to remove object intersections. This improves the probability of mesh success.

- 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.

Small feature errors and misaligned entities should be resolved before you invoke the meshing for the model.

By default, the **Heal** command is automatically applied to imported objects.

Analysis Options Dialog Box

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 box, 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**, you also see this dialog box.

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**. See: Modeler Validation Settings.

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 box, which contains the results of the analysis.

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** drop-down menu on the **Objects** tab.

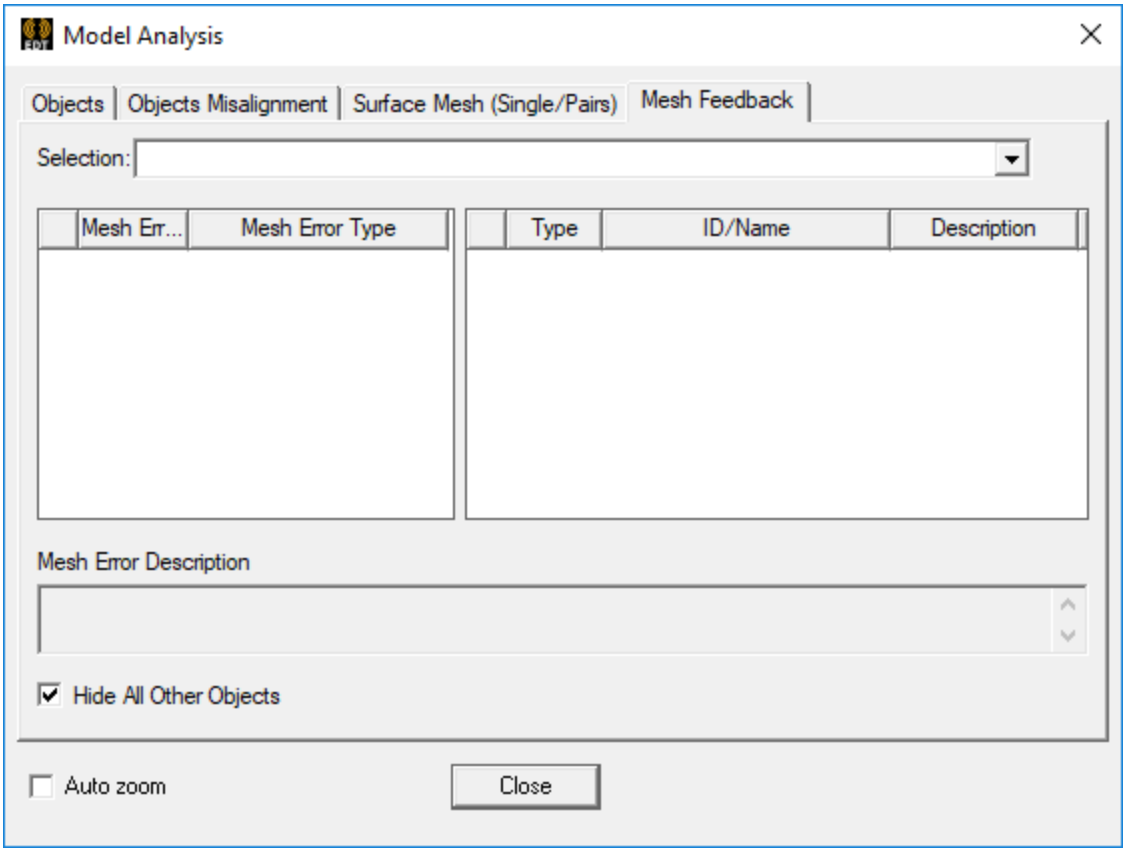
3. Select or clear the **Perform Object Pairs Analysis** check box. Selecting this option evaluates the mesh for all combinations of the selected objects.
4. Select or clear the **Ignore Objects Separated by greater than** check box, and enter a value in the text box. Selecting this option means that 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.

Model Analysis Dialog Box

This dialog box 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
6. Click **Close** to close the **Model Analysis** dialog box.

Objects Tab

All results relating to model analysis of specific objects are presented under the **Objects** tab.

1. The results table contains the following information.
    - **Object** – 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 box.
        - **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 Ansys EM recommends that you fix before 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 box.
2. Select any object name in the table which contains errors to display a table with a list of Error types, corresponding Entity List IDS for associated faces, edges and vertices, and error descriptions. Selecting the row for each error causes the Modeler window to highlight the associated entity.

    > **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 box 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 box to display a healing log which includes information about operations performed on an object during the healing process.

Object Misalignment Tab

The table in this panel displays results of an **Interobject Misalignment** analysis. It contains a list of Alignable Faces, described in a list of Object Sets, and corresponding Misaligned Faces.

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 **Align Faces** to align selected faces.
- **Clear All Analysis Data** – this button removes all information from the tables.
- **Display Log for Object Set** – checking this causes the **Model Analysis** dialog box to display a healing log which includes information about operations performed on an object set during the healing process.
- **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 box 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.

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.

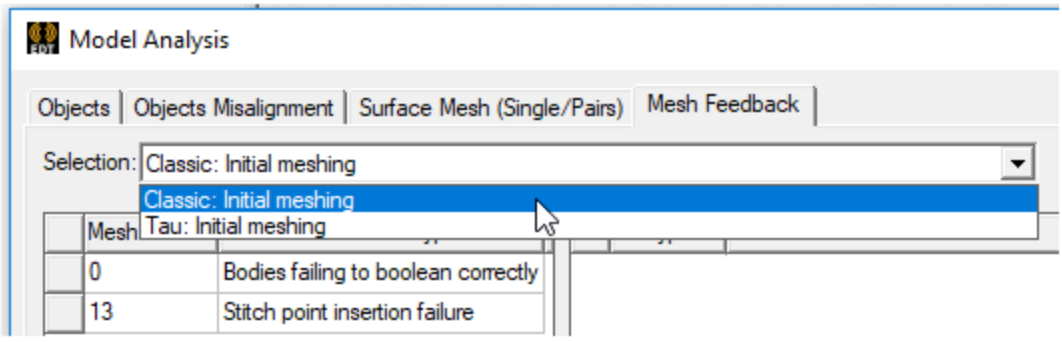
2. The results table contains the following information:
   - **Object** – column listing object name or a pair of object names.
   - **Last Analysis Status** – column stating the meshing status of the object or object pair.
     - Mesh Success
     - Mesh Failure
   - **Error Type** – this column gives 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** – 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.

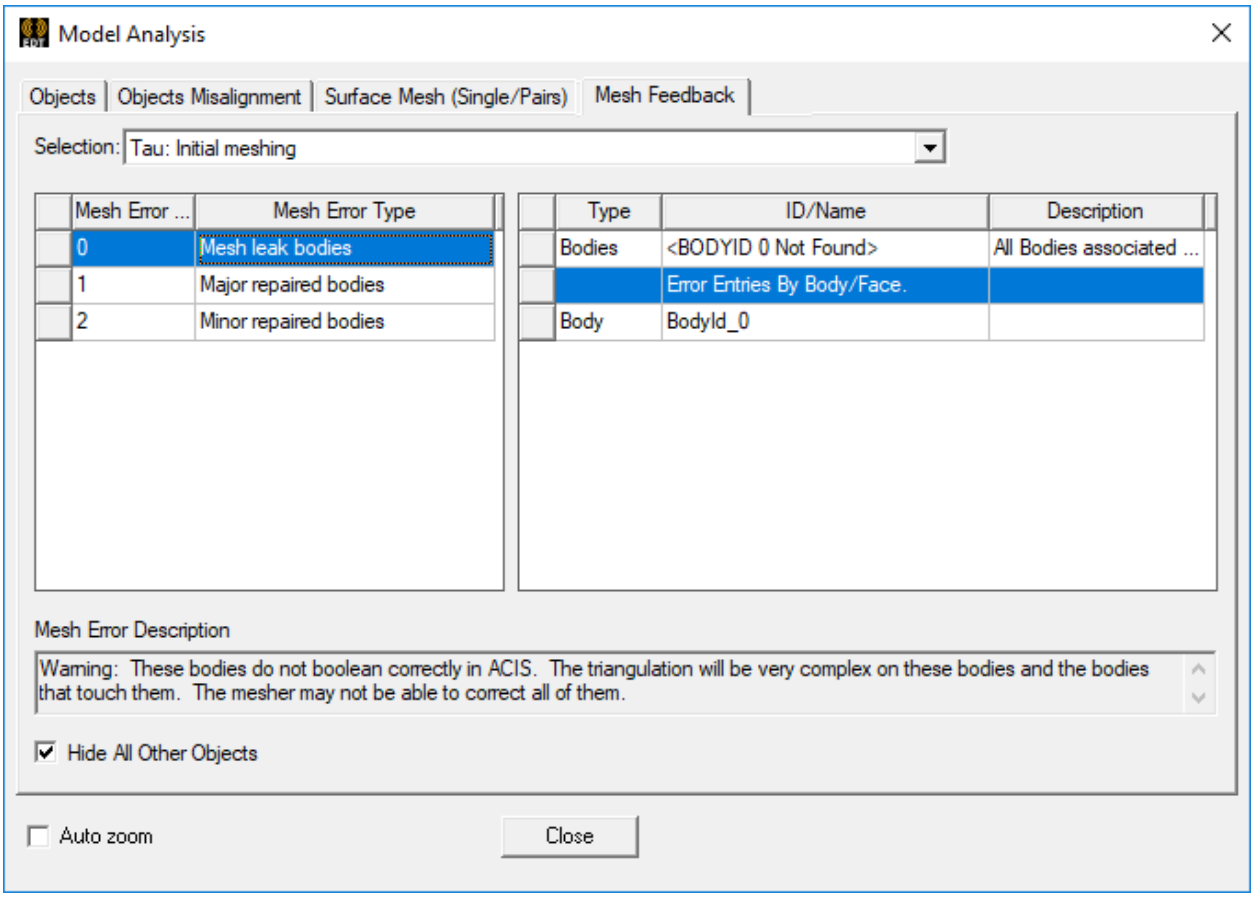
Mesh Feedback Tab

The tables in this panel list model errors as viewed by the mesher. If the project has solutions by different solvers, the **Selection** drop down lets you 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.
- **Mesh 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** window 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:

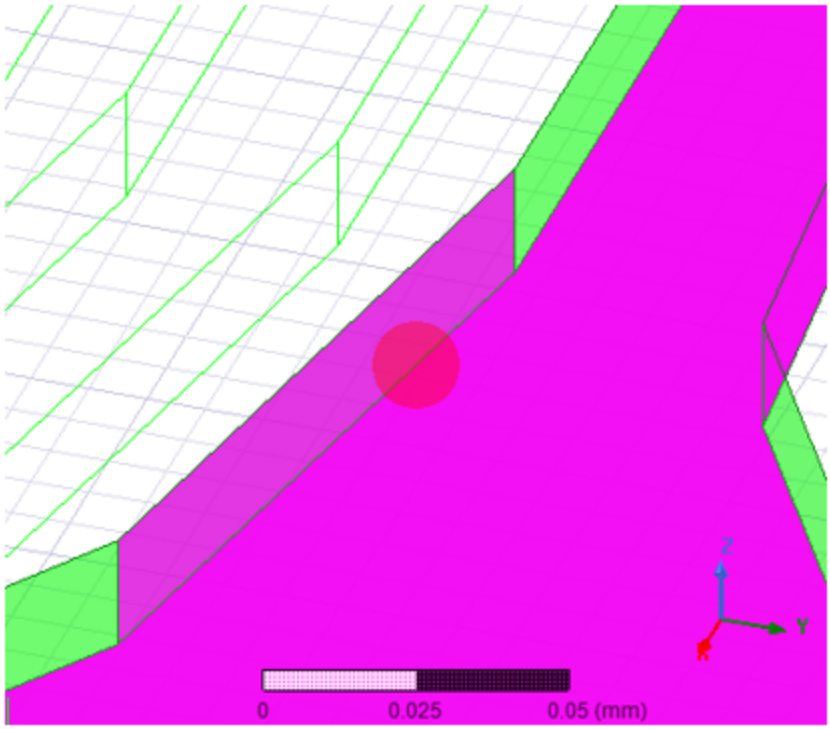
- **Type** – whether the error applies to Bodies, Faces, or Triangles. If you select a particular error type, the columns for Type, ID/Name and Description are filled in.
- **ID/Name** – the object name or object ID for the error type.
- **Description** – the first three rows are for Bodies, Faces, and 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:

- **Hide All Other Objects** – hides objects in the Modeler window not associated with the selected Bodies, Faces, or Triangles.
- **Auto Zoom to Selection** – checking this causes the modeler to automatically zoom to objects or faces corresponding to the selected Mesh Error Serial ID (on the left side table) or the ID/Name (on the right side table).

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.



The error shown in the figure above is a contact error between the faces. 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.

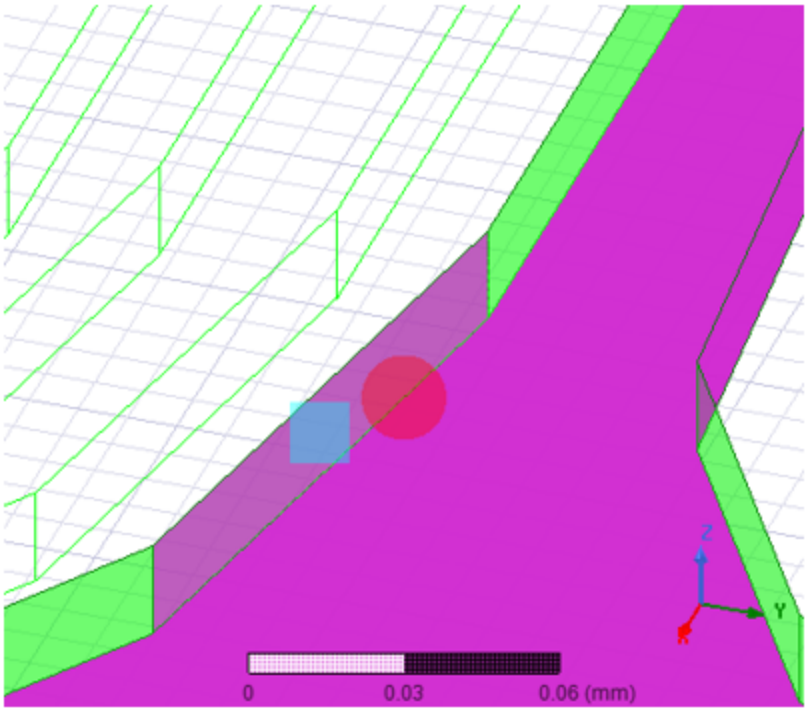
Mesh Error Markers

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 you select an individual face in the right panel, then the face is marked using a blue square as shown below:



If many individual faces are selected in the right panel, each face is marked by a blue square. In order to find which blue square corresponds to which face in the right panel in the dialog box,

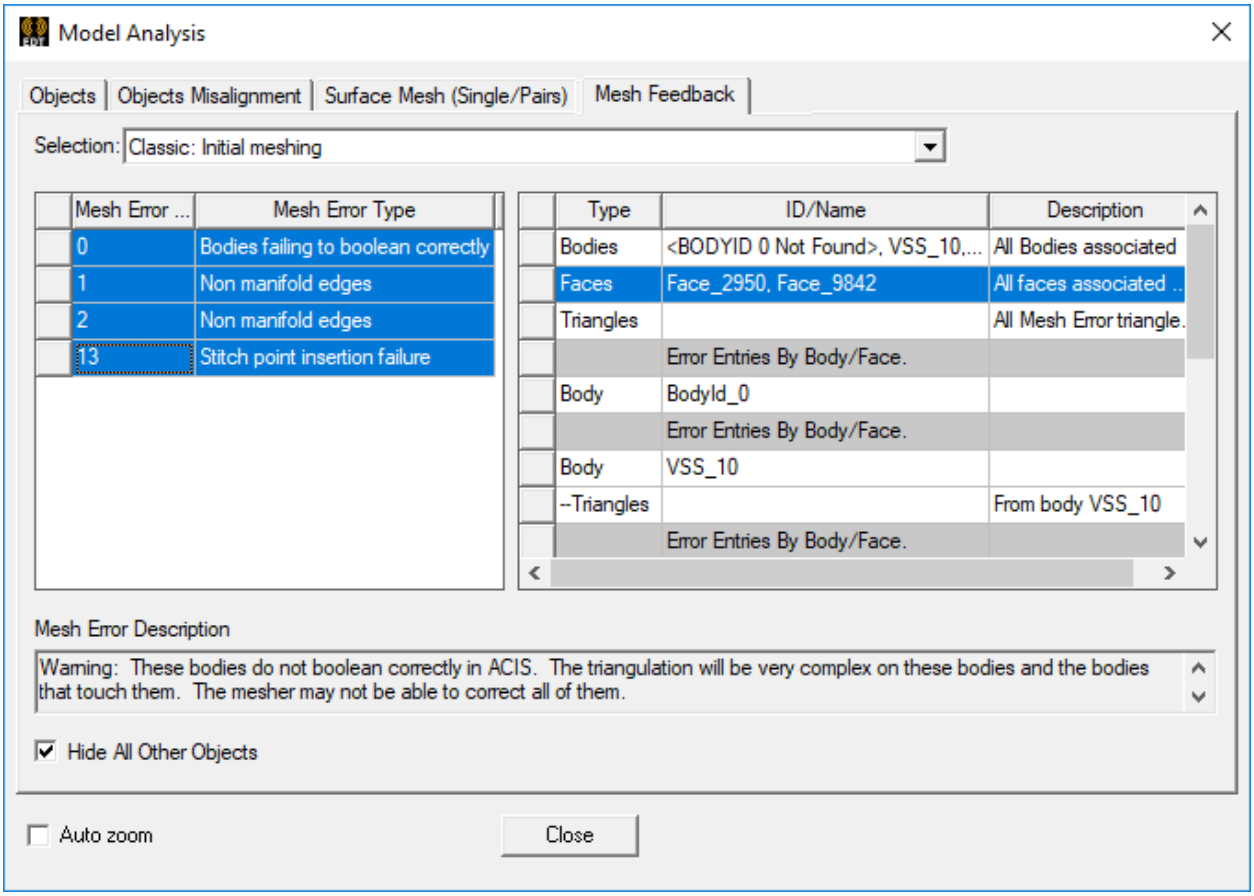
click on the blue square. This action selects the corresponding face row in the right panel of the dialog.

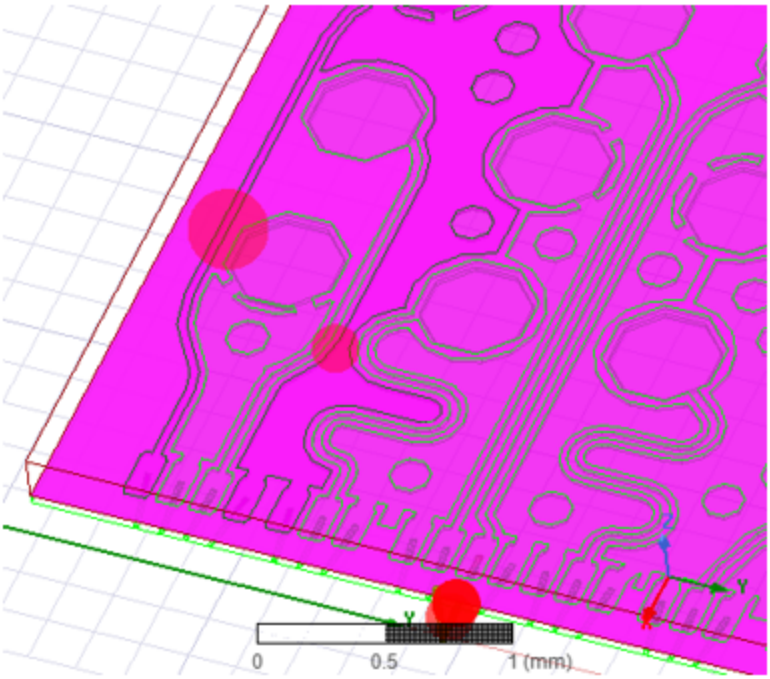
Face Selection

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**, as shown in figures below:





With **Auto Zoom** enabled, the view fits to all the errors, providing an overview of all the mesh errors, as shown in above figure.

Also, when multiple errors are selected, right panel shows the merged information about all bodies, faces, error triangles etc. And Mesh Error Description text box shows description of all the errors.

Design Settings

The tabs and settings within the *Design Settings* dialog box are dependent on the design type, and there is a product-specific topic for each applicable design type. Please click the appropriate link below for the product of interest:

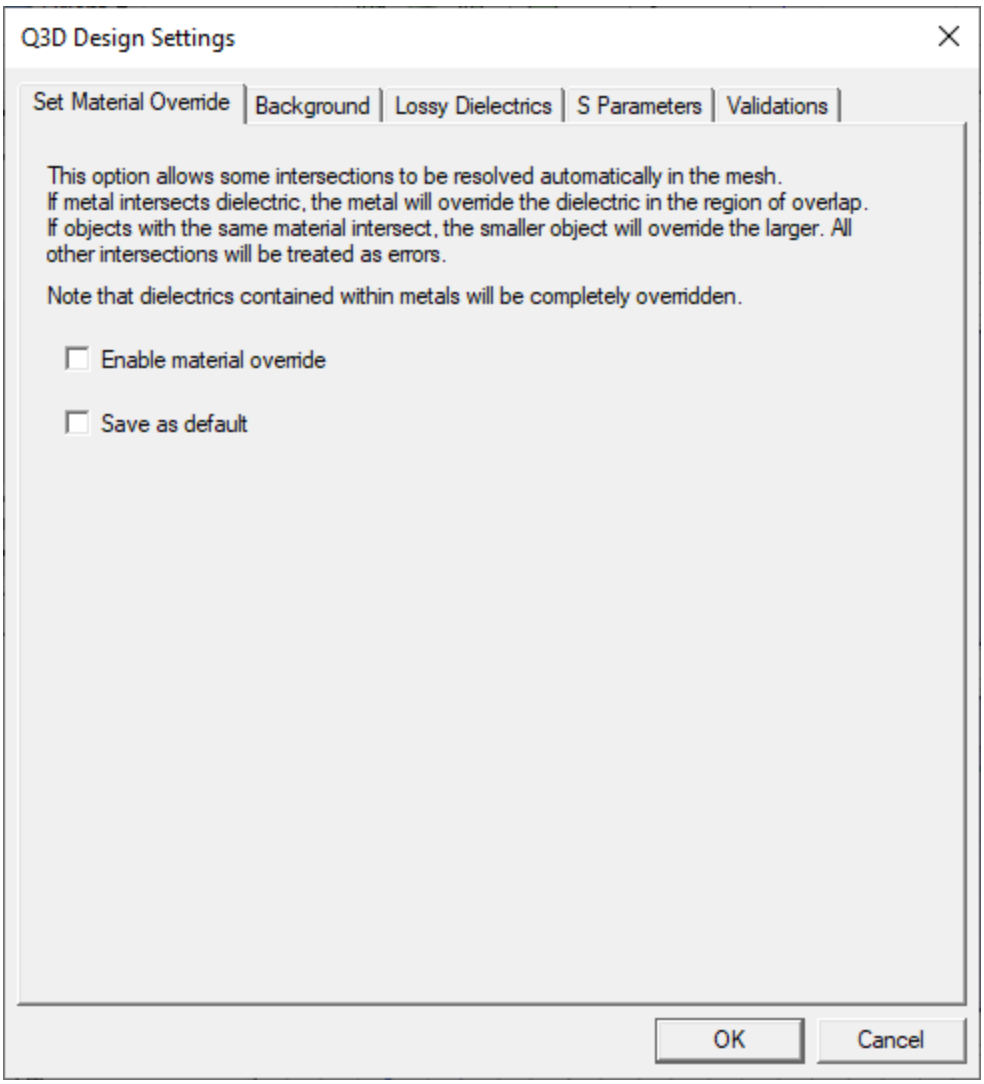
- HFSS
- HFSS Transient
- HFSS 3D Layout
- Icepak
- Mechanical
- Q3D
- 2D Extractor

Design Settings for Q3D Extractor

To set the design settings for Q3D Extractor:

1. Click **Q3D Extractor** > **Design Settings**.

   The **Q3D Design Settings** dialog box appears, on the **Set Material Override** tab.

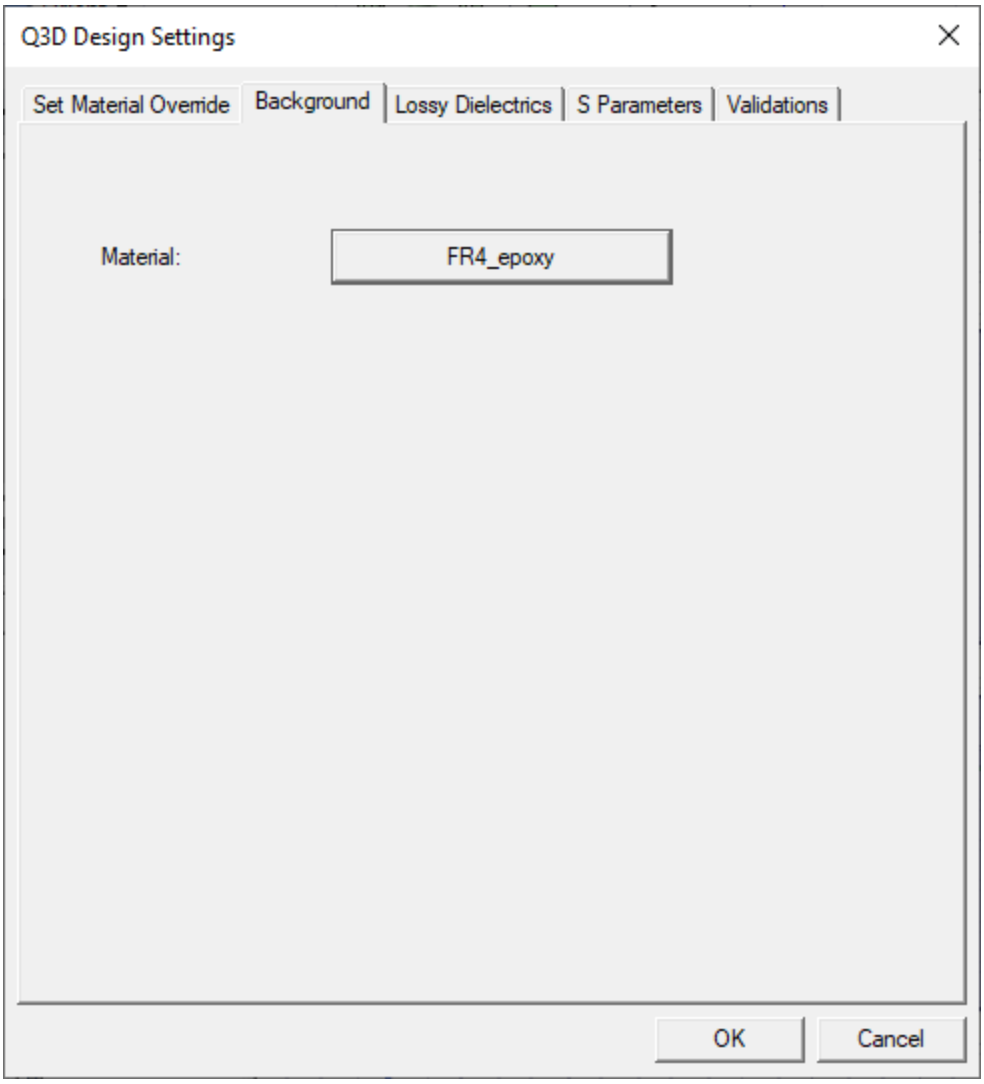


Enable material override allows some intersections to be resolved automatically in the mesh. If metal intersects dielectric, the metal overrides the dielectric in the overlap region, and the metal object is subtracted from the dielectric. If objects with the same material overlap, the small object overrides the larger - the small object is subtracted from the larger. All other intersections are treated as errors. Normally, the modeler considers any

intersection between 3D objects to be an error. In the meshing process, the dielectrics are locally overwritten by 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. An 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.

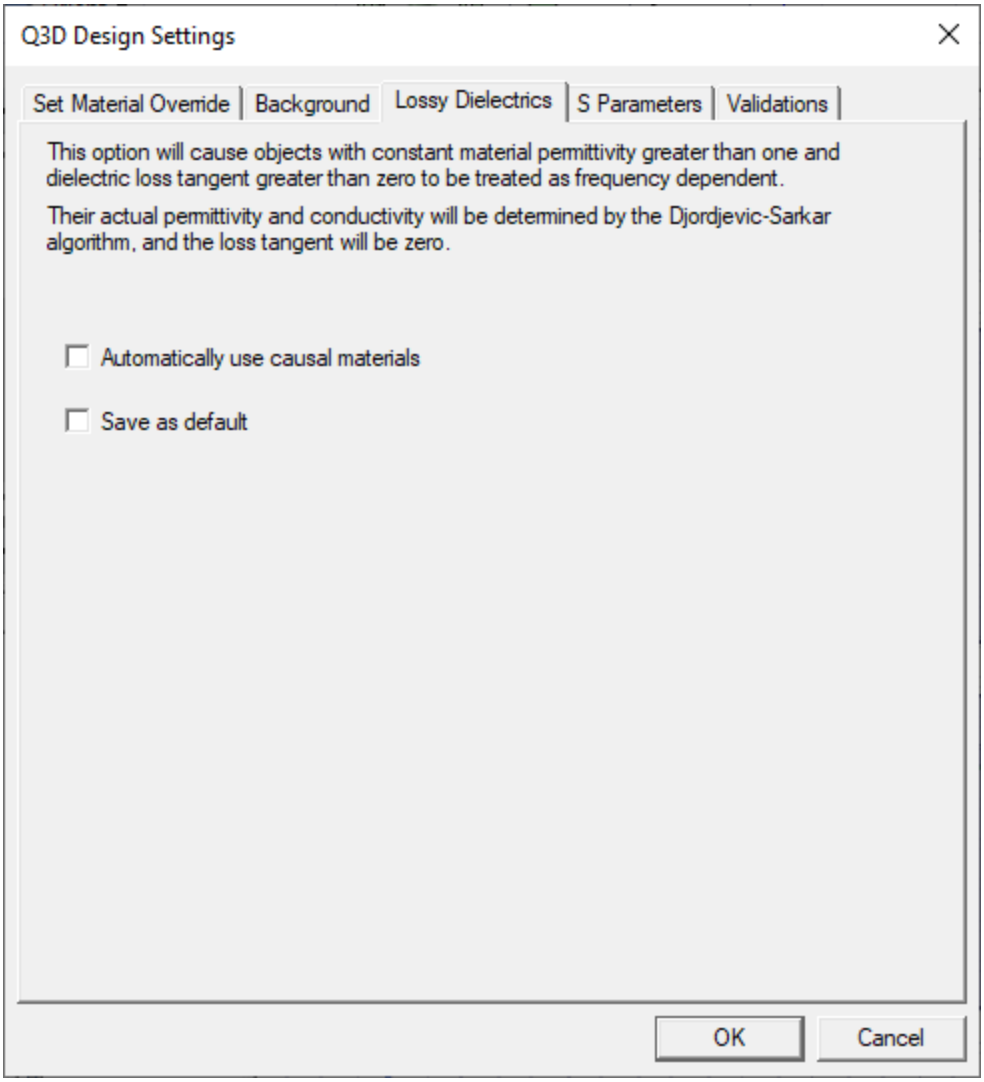
> **Note:**
>
> For DC-RL simulations, dielectric objects inside metal objects will always be treated as conductors irrespective of this option. To explicitly take into account the dielectric objects, use the **Subtract** operation to single out the conductive region.

2. In the **Design Settings** window, click through the tabs to set various options.
    - The **Background** tab shows the background material assignment.



Clicking the button opens the Materials window.

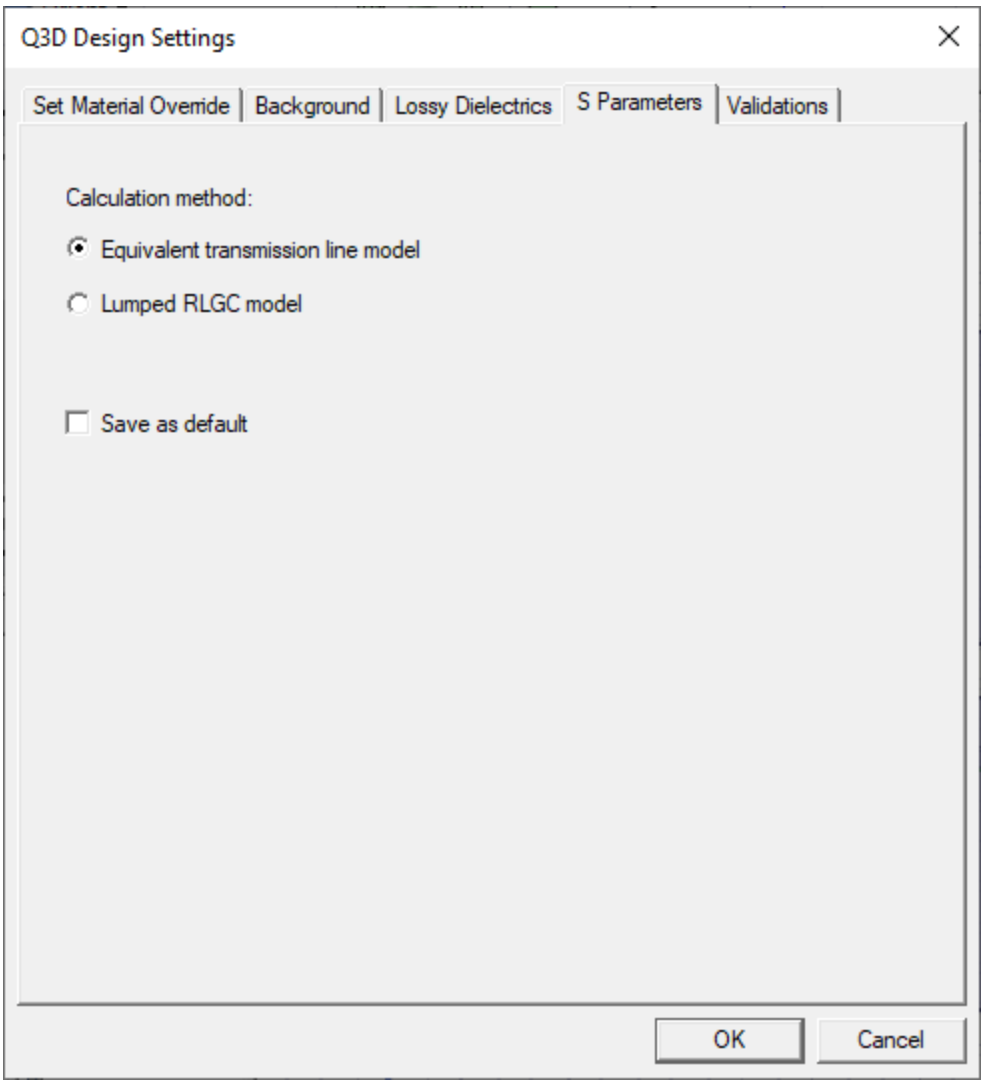
- The **Lossy Dielectrics** tab contains options for causal materials.



Automatically use causal materials causes objects with constant material permittivity greater than one and dielectric loss tangent greater than zero to be treated as frequency dependent. Their actual permittivity and conductivity will be determined by the Djordjevic-Sarkar algorithm, and the loss tangent will be zero.

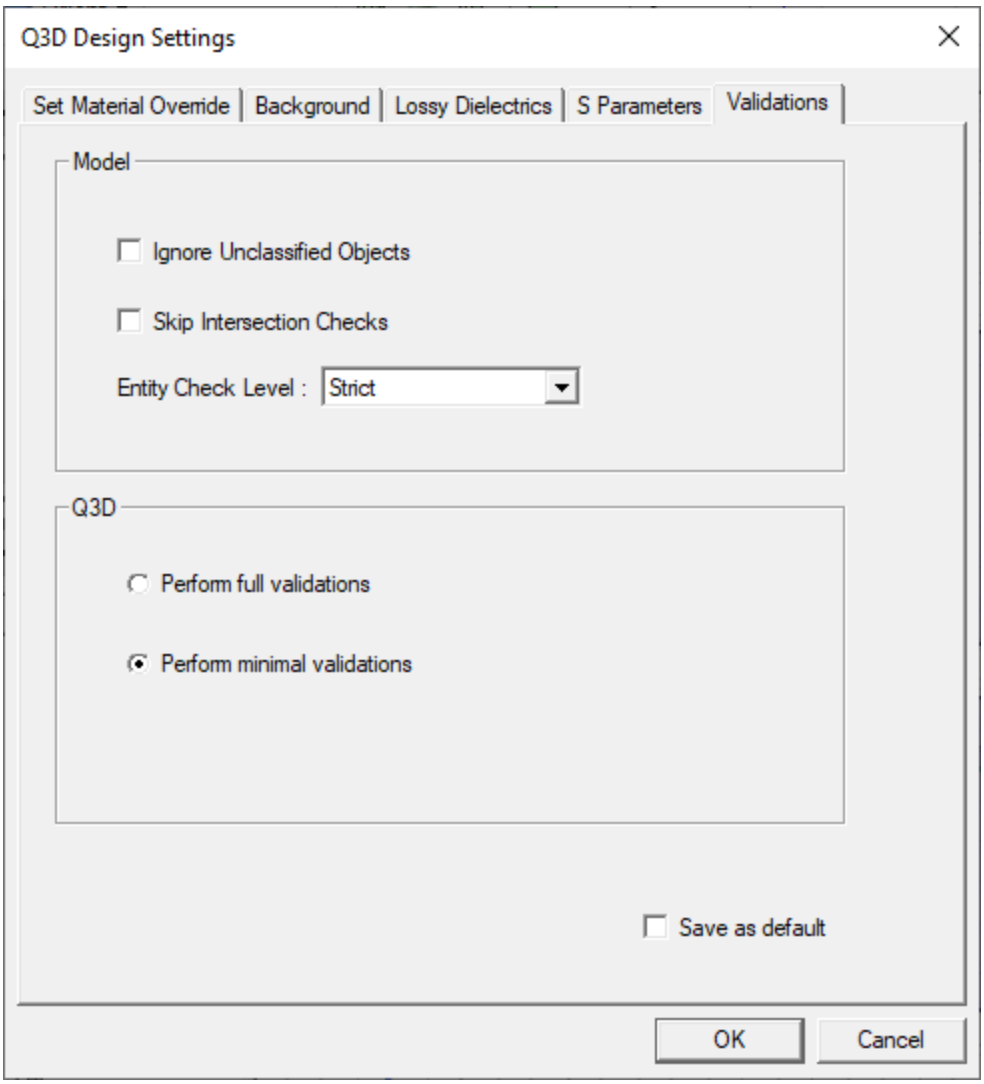
> **Important:**
>
> Automatic casual material calculations are not performed under the following circumstances:
>
> - If the material permittivity or loss tangent is anisotropic.
> - If the material permittivity or loss tangent is frequency dependent.
> - If the material itself is not a lossy dielectric.

- The **S Parameters** tab allows you to choose a calculation method.



You can select either the **Equivalent transmission line model** or the **Lumped RLGC model**.

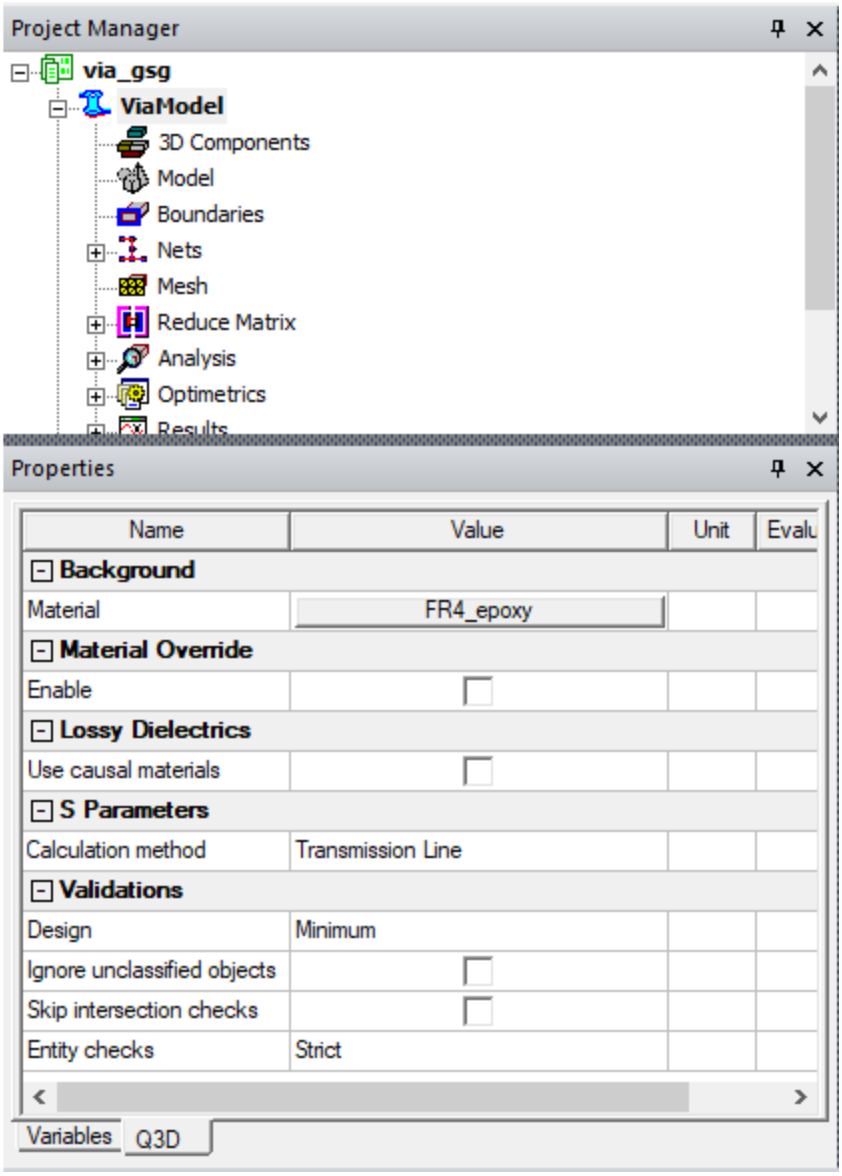
- The **Validations** tab controls the extent of validations performed and, therefore, the time involved.



  - Select model validation choices: either **Ignore unclassified objects** or **Skip intersection checks**.
  - Control the **Entity Check Level** as **Strict**, **Basic**, **Warning Only**, or **None**.
  - Choose **Perform full validations** to do all design validations, or **Perform minimal validations** to run all design validation except boundary overlap checks and net validations.

3. If desired, select **Save As Default** on any tab. You must be careful since these settings change the modeler's "ground rules" and may produce unexpected results.
4. Click **OK** to accept your selections and close the dialog box.

You can also view and set Design Settings by selecting the current project and then the **Q3D** tab in the **Properties** window.

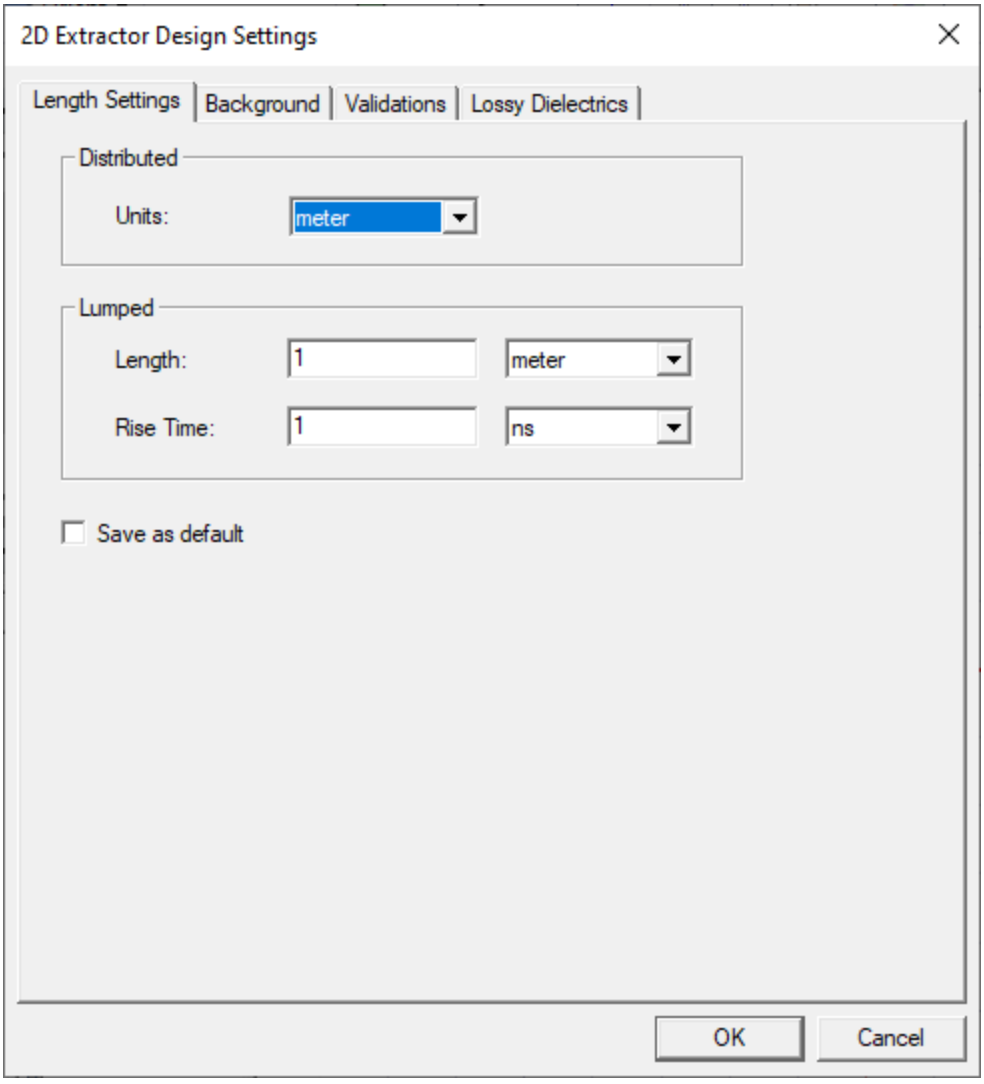


Design Settings for 2D Extractor

To set the design settings for 2D Extractor:

1. Click **2D Extractor** > **Design Settings**.

   The **2D Extractor Design Settings** dialog box appears, on the **Length Settings** tab.

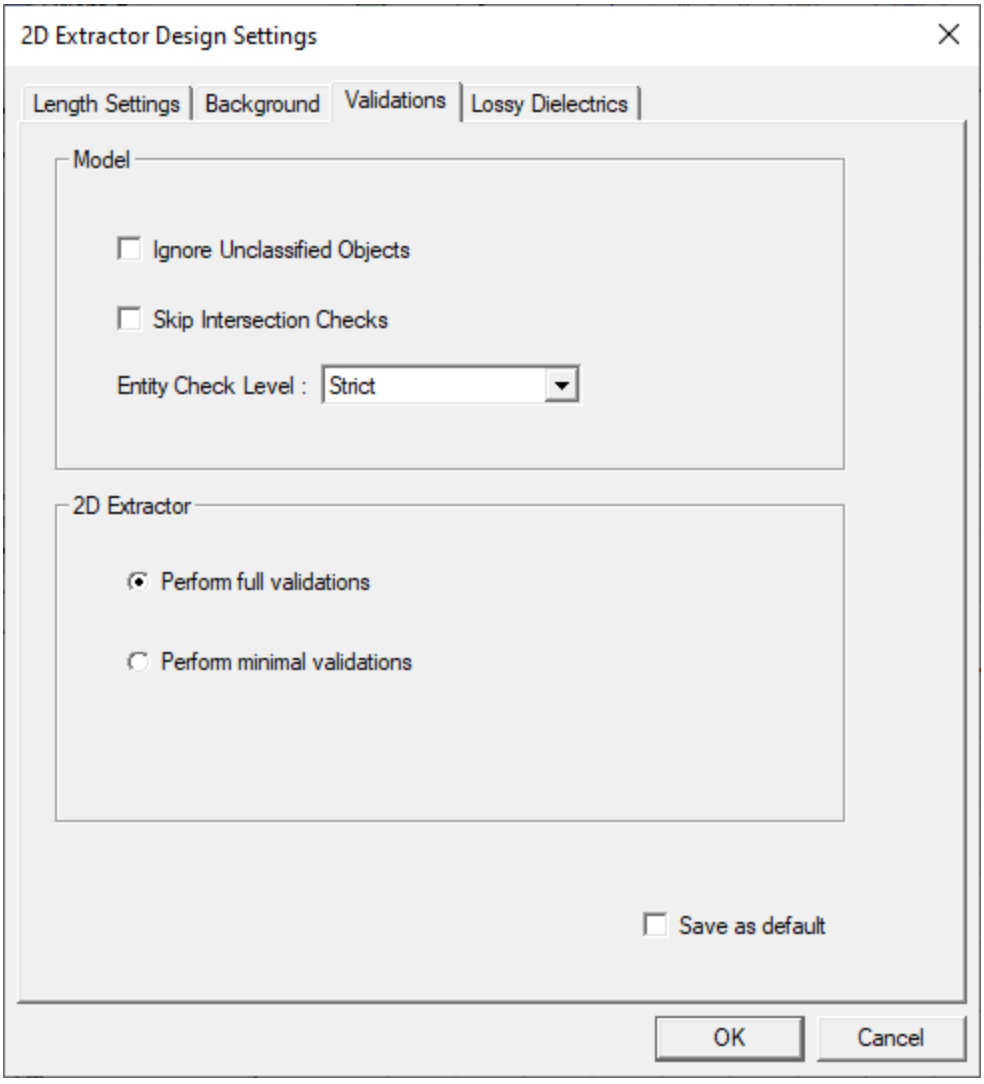


The **Length Settings** tab controls the Distributed Units, Lumped Length and Rise time.

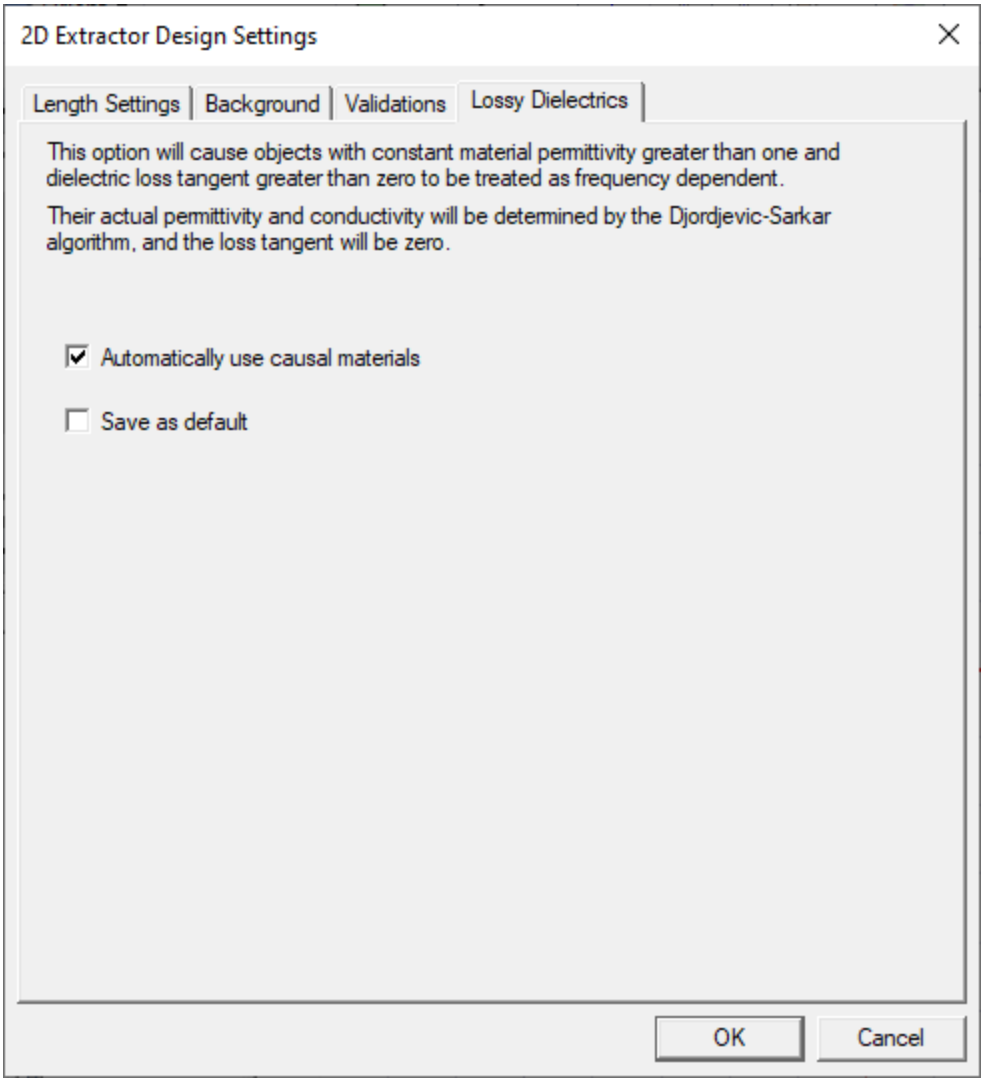
2. Click through the tabs to set various options.
    - The **Background** tab shows the current background material name:

    Clicking the name button opens the Materials window.

    - The **Validations** tab controls the extent of validations performed, and therefore the time involved:



  - Select model validation choices: either **Ignore unclassified objects** or **Skip intersection checks**.
  - Control the **Entity Check Level** as **Strict**, **Basic**, **Warning Only**, or **None**.
  - Choose **Perform full validations** to do all design validations, or **Perform minimal validations** to run all design validation except boundary overlap checks and net validations.
- The **Lossy Dielectrics** tab contains options for causal materials:

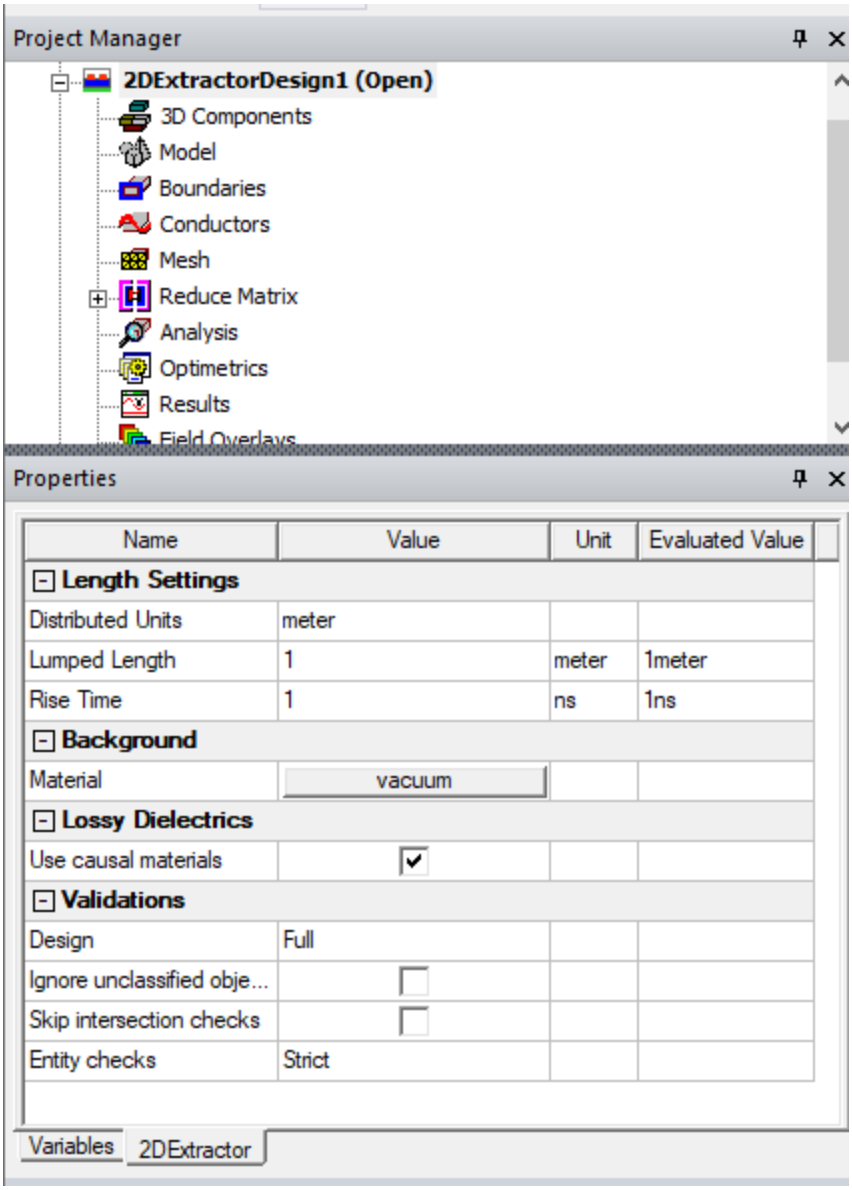


Automatically use causal materials causes objects with constant material permittivity greater than one and dielectric loss tangent greater than zero to be treated as frequency dependent. Their actual permittivity and conductivity will be determined by the Djordjevic-Sarkar algorithm, and the loss tangent will be zero.

> **Important:**
>
> Automatic casual material calculations are not performed under the following circumstances:
>
> - If the material permittivity or loss tangent is anisotropic.
> - If the material permittivity or loss tangent is frequency dependent.
> - If the material itself is not a lossy dielectric.

3. If desired, select **Save As Default** on either tab. You must be careful since these settings change the modeler's "ground rules" and may produce unexpected results.
4. Click **OK** to accept your selections and close the dialog box.

You can also view and set Design Settings by selecting the current project and then the **2D Extractor** tab in the **Properties** window.



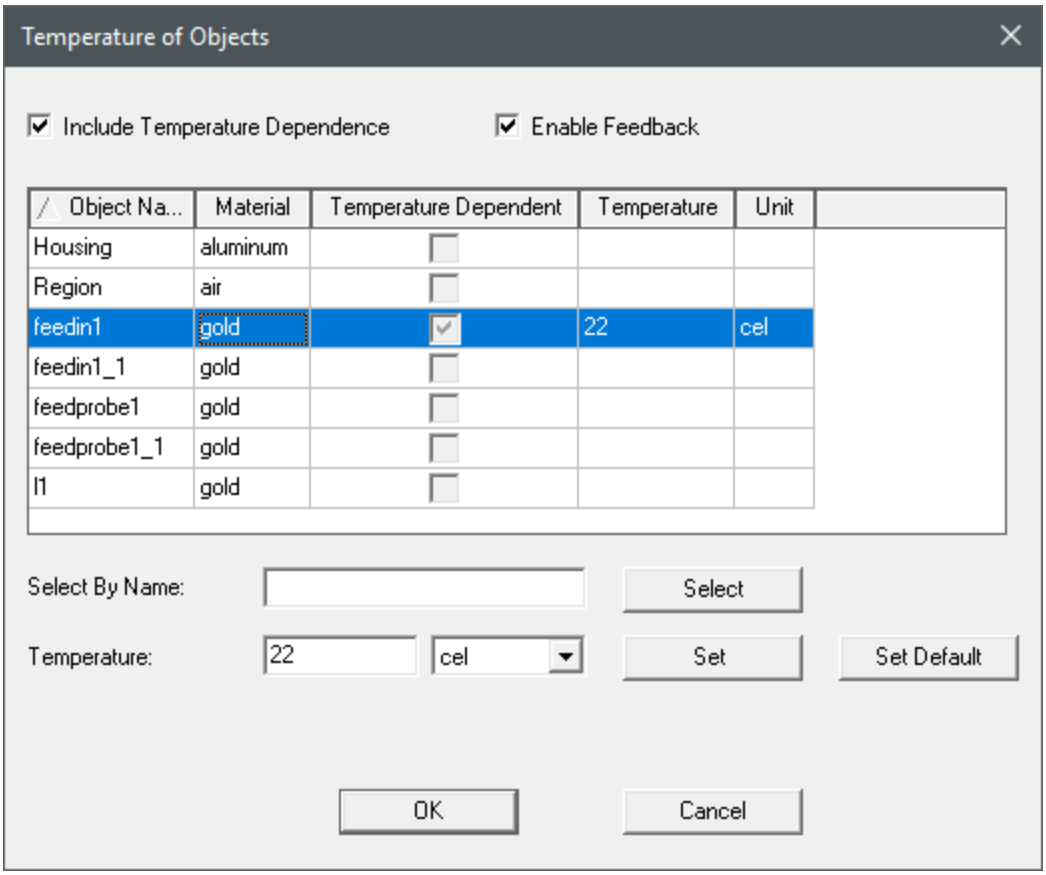
Setting the Temperature of Objects

To set the temperature of objects and to enable feedback for use in two-way coupling:

1. Use the **Q3D Extractor** > **Set Object Temperature** command to display the **Temperature of Objects** dialog box. You can also right-click the design in the project tree and select from the shortcut menu.

   This window includes a table of the objects in the design. In order, the table's columns list: object name, material, a check box to show whether each object has temperature-

dependent features, Temperature (if applicable), and unit. To enable temperature dependence, you must edit the material properties and include a thermal modifier. See: Specifying Thermal Modifiers.



2. To enable editing, select **Include Temperature Dependence**.

   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, a scroll bar will appear.

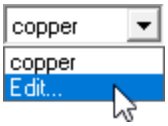
   You can also use the **Select by name** field to find objects by name. Enter the name of the object you want and click **Select**.

   Selected objects are highlighted, and you can make multiple selections.

3. To set the temperature for the selected object(s), type a value or an existing variable name in the text field.
4. Select the units from the drop-down menu.
5. Click **Set** to apply the value to the selected objects, or click **Set Default** to make the specified values the default.

If you click **Set Default**, the row(s) for the selected object(s) display the temperature value and units.

6. To edit the material for an object, click the object's material to display a drop-down menu.



Click **Edit...** to display the Materials dialog box.

7. To enable coupling with thermal designs, select **Enable Feedback**.
   - You can perform two-way coupling between an HFSS, Q3D, or Maxwell design and an Icepak design within the Electronics Desktop. In an Icepak design, you can specify the number of coupling iterations to be run. EM Loss data calculated by the source HFSS, Q3D, or Maxwell design is consumed by Icepak when calculating thermal data in the current coupling iteration. The thermal data is then consumed by the source design in the next coupling iteration's calculation.
   - Alternatively, you can perform thermal static and transient analysis based on a high frequency solution by coupling with other Ansys thermal systems in Ansys Workbench. The solver can then use per mesh element temperatures for the object's material properties. If **Enable Feedback** is enabled but feedback is not available, the temperature specified in the dialog is used for the temperature dependent properties in the design.

> **Note:**
>
> When **Enable Feedback** is selected, the right-click menu for **Analysis** and the analysis setup includes the **Revert to Initial Temperature** command. Selecting this command invalidates the current solution.

8. To close the dialog box and accept changes, click **OK**.

Model Preparation Commands

Use the **Modeler** > **Model Preparation** commands to prepare a model.

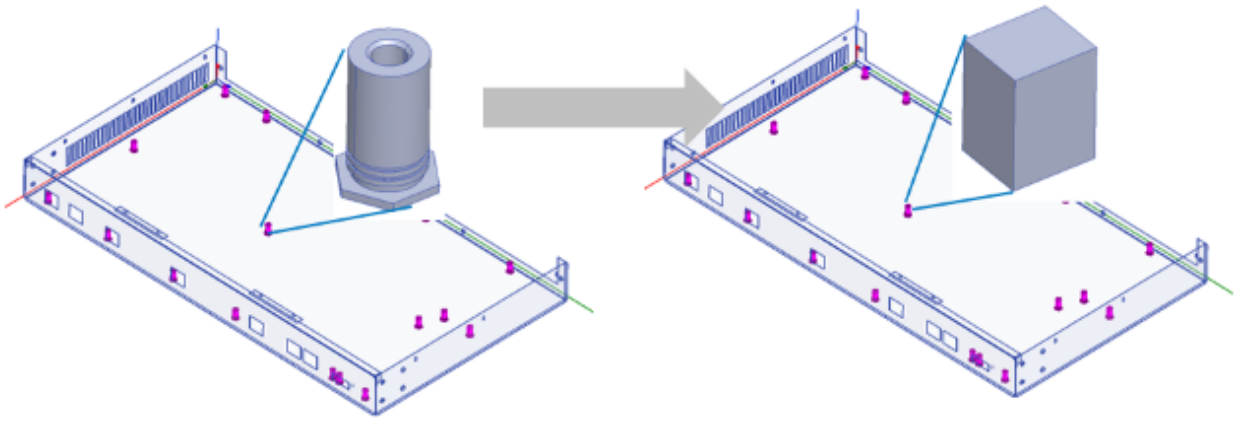
These commands include:

- Simplify
- Heal
- Stitch Sheets
- Align Faces

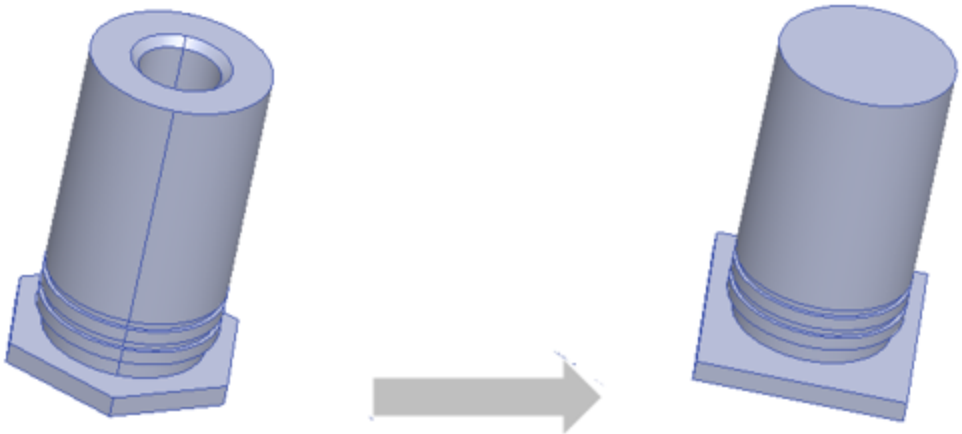
- Remove Faces
- Remove Edges
- Perform Explicit Subtractions

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.



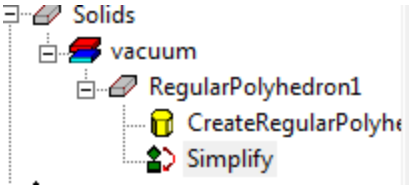
Bounding Box Simplification



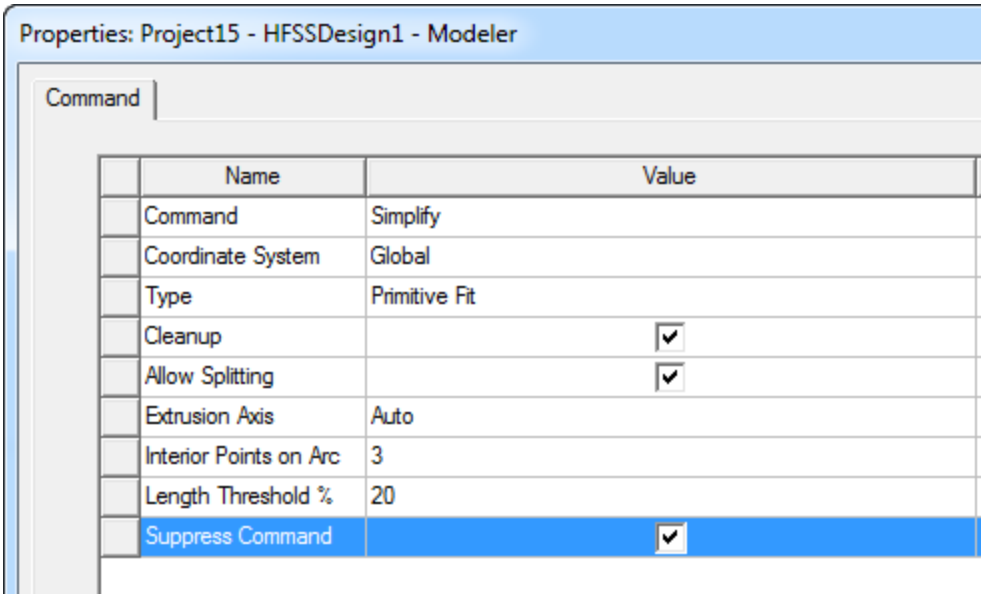
Primitive Fit Simplification

Polygon Fit Simplification

The History tree shows the **Simplify** command as being applied to the object.



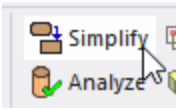
Any parameters you specify in the dialog you can also edit in the Properties dialog for the History tree **Simplify** command to get a different simplification. The properties display in the docked properties window and include only the parameters that are used by your currently selected **Simplify** command type. None of the parameters of simplify operation accept variables.



1. Select an object.

   This enables the Simplify command on the **Modeler** > **Model Preparation** submenu.

2. Click **Modeler** > **Model Preparation** > **Simplify** or on the **Model** ribbon, select **Simplify**.

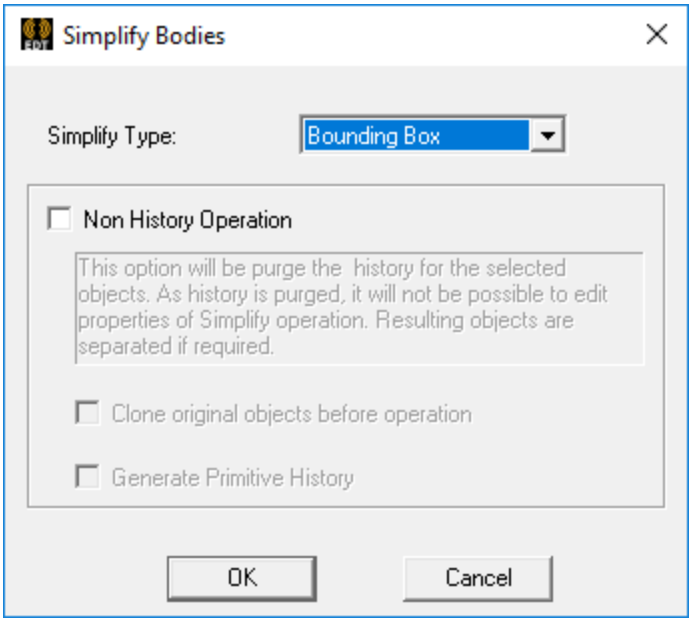


This opens the **Simplify Bodies** dialog box. A drop-down menu lets you select the degree of simplification, from Bounding Box, Primitive Fit, or Polygon Fit.

Bounding Box Simplify Type

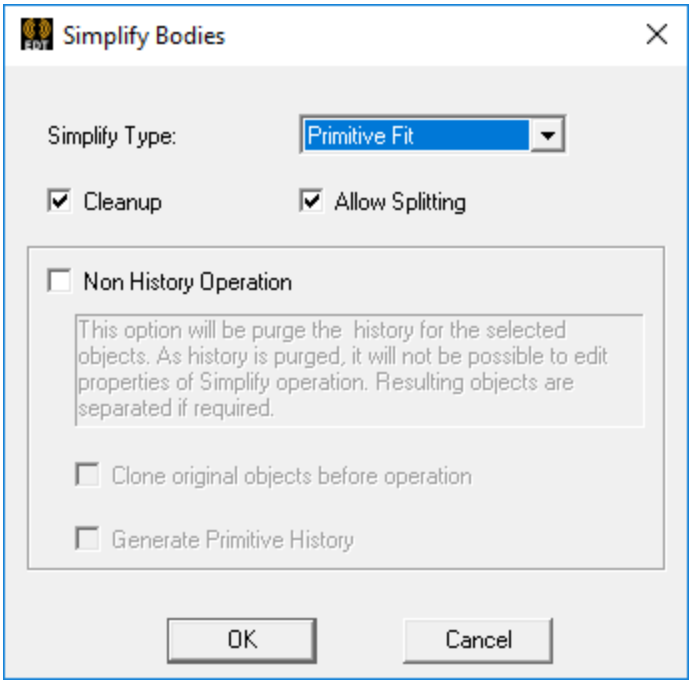
If you select Bounding Box as the Simplify Type, there are no additional parameters. The object is replaced by its exact bounding box as computed in operation's coordinate system. You can change the operation CS to get a bounding box in an appropriate orientation. For the **Non**

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, whether to Separate Bodies and Purge History, and if so, whether to clone original objects before the operation.

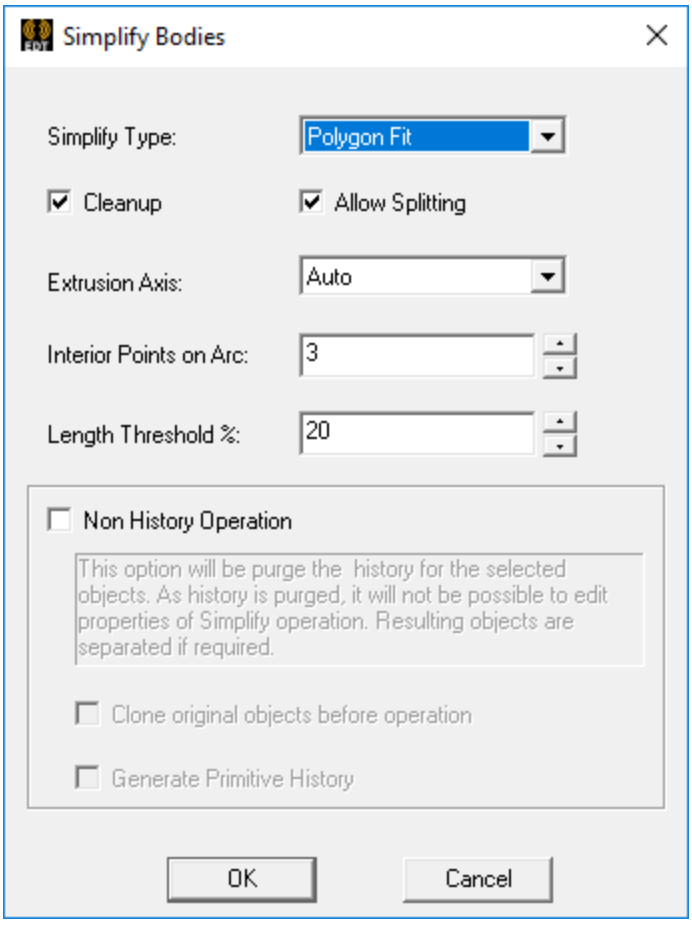


For Primitive Fit, the parameters are:

- **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.
- **Allow Splitting** - This option controls if object should be split 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.

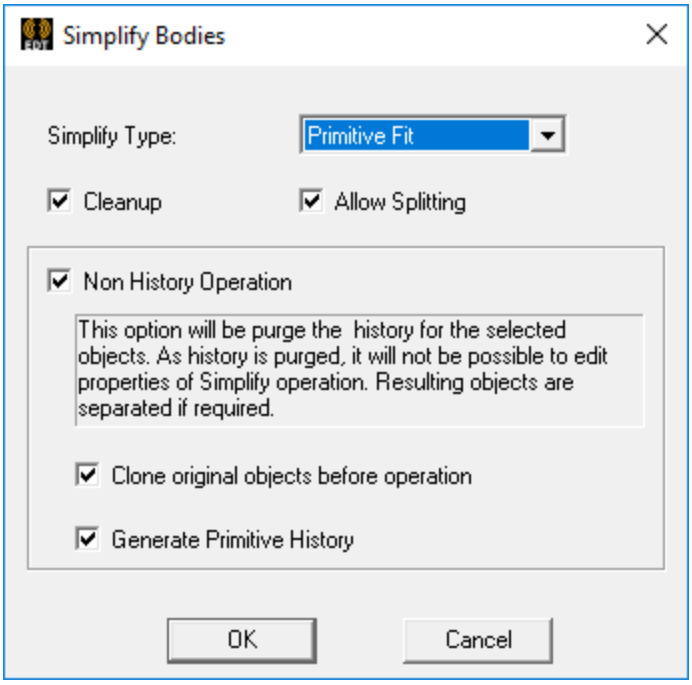


For Polygon Fit, you specify parameters for

- **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.
- **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 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.

Heal

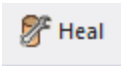
The **Heal** command provides a way to correct geometric violations and to remove specific kinds of small features. When models are imported, two types of errors can occur - geometry errors and topology errors. Geometry errors are errors in definition of the underlying geometry while topology errors are errors in how the underlying components like faces, edges and vertices are connected. Ansys Electromagnetics 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.

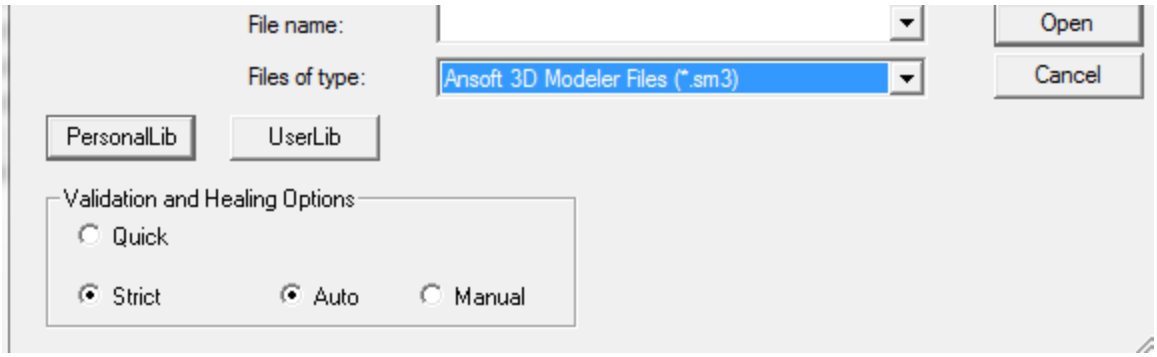
Healing can be invoked in different ways.

- The menu command **Modeler** > **Model Preparation** > **Heal** command, or the **Model** ribbon **Heal** icon 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),
  - CreoParametric files (*.prt, *.asm),
  - CATIA (*.model, *.CATpart), and
  - Parasolid file (*.x_t, and *.x_b).

  Selecting 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 box 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 Specifying 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**.

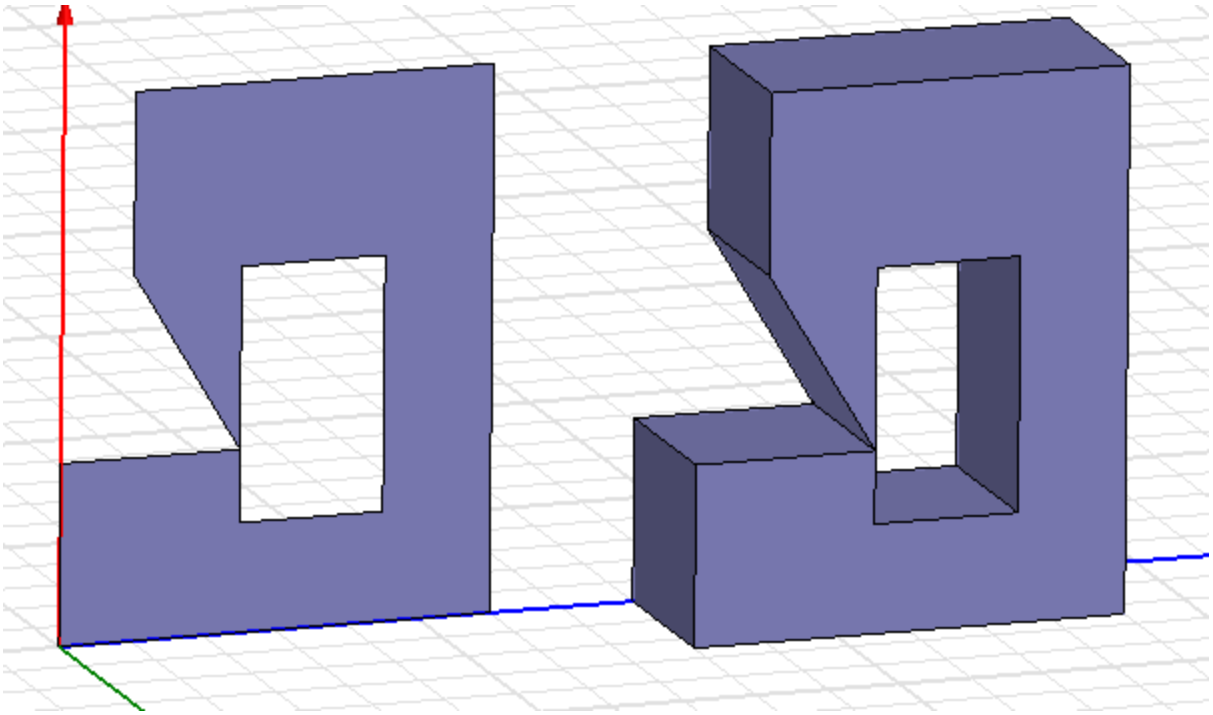
Validation and Healing Options for Import File

The **Import File** dialog box contains shows different Validation and Healing options for the seven formats listed here. The Quick option allows you to import these formats without healing.

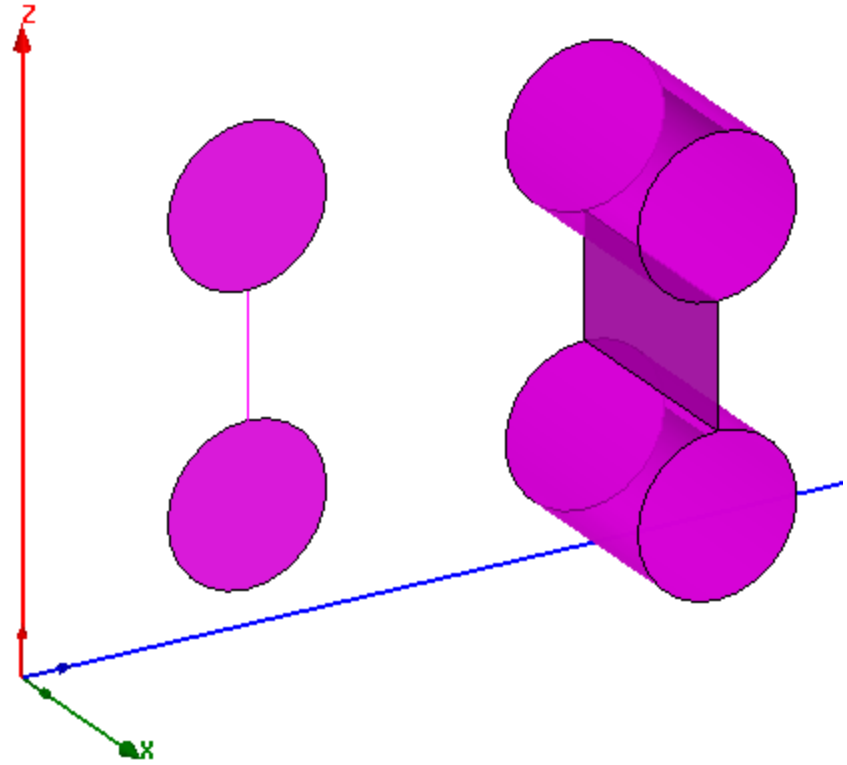
Format	Options Available			
3D Modeler file (*.sm3),	Quick	Strict		
SAT file (*.sat),	Quick	Strict	Auto or manual	
STEP file (*.step,*. stp),	Quick	Strict	Auto or Manual	Stitch Tolerance
IGES file (*.iges, *.igs)	Quick	Strict	Auto or Manual	Stitch Tolerance
Creo Parametric files (*.prt, *.asm)	Quick	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	

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 the face selection toolbar icons.

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 for all non-manifold edges.

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 box 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 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:

   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.

   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.

   Control Object Properties Change options:

   - **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 box.

Stitch Sheets

Use the **Modeler** > **Model Preparation** > **Stitch Sheets** command to stitch selected sheets.

1. Select two or more sheet objects.

   This enables the **Stitch Sheets** command on the **Modeler** > **Model Preparation** submenu.

2. Click **Modeler** > **Model Preparation** > **Stitch Sheets**; or, on the **Model** ribbon, click the **Stitch Sheets** icon.

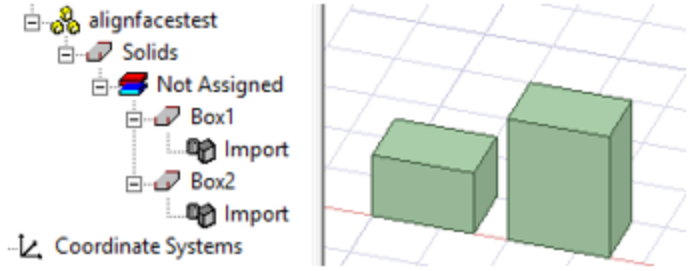
This displays a **Stitch** dialog box with a **Maximum Stitch Tolerance** field. The default value (auto) comes from the **Healing** dialog box's **Options** tab with Manual Healing selected. You may edit the value in the **Stitch** dialog box or in the **Healing Options**.

3. Click **OK**.

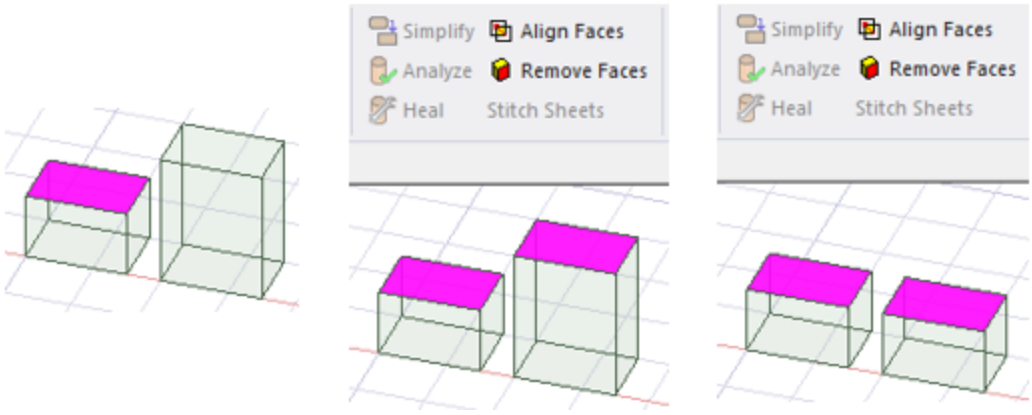
   This closes the dialog box and attempts to perform stitching on the selected sheets. If the sheets are separated beyond the stitch tolerance, stitching is not performed and a warning is issued.

Align Selected Faces

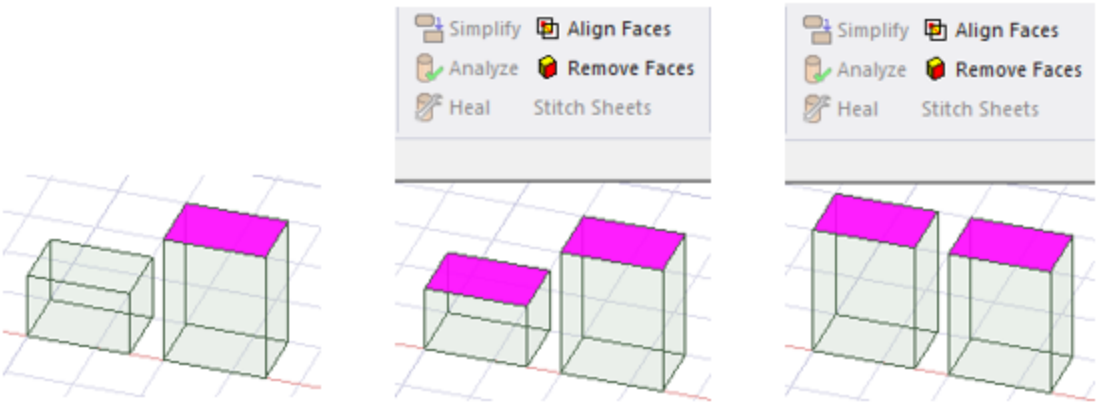
Use the **Modeler** > **Model Preparation** > **Align Faces** command to align the adjacent selected faces of imported objects that have only one operation in their History Tree. You can use **Undo/Redo** on this command.



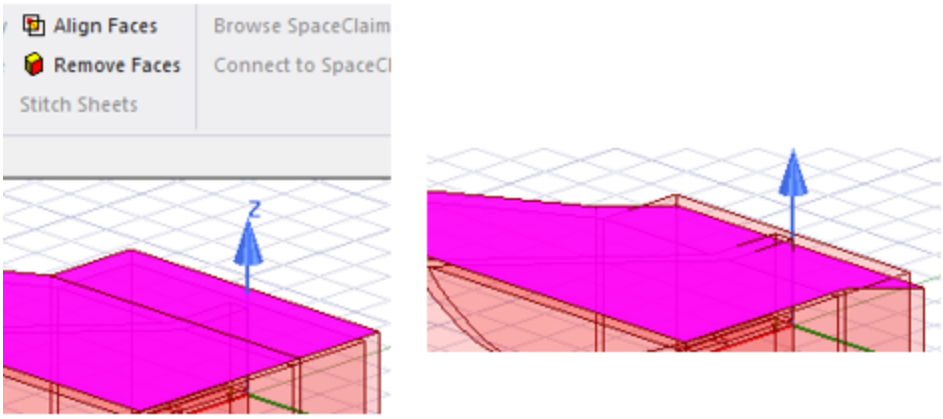
The **Model** ribbon also includes an **Align Faces** icon, which is enabled when you make appropriate face selection. The first selected face defines the plane of alignment for subsequent selection, that is, tool faces to blank faces.



Selecting faces in a different order changes the alignment.

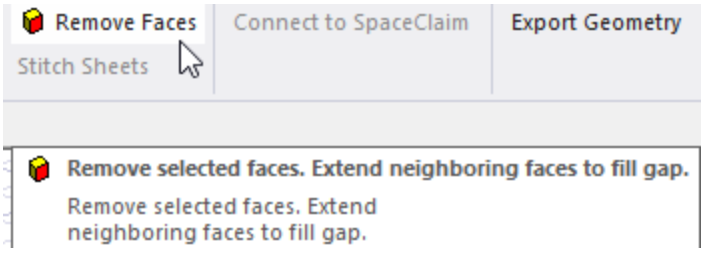


The selected faces can touch.



Remove Selected Faces

Use the **Modeler** > **Model Preparation** > **Remove Faces** command to remove the selected faces, extending the remaining faces to fill gaps. 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 Preparation** > **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.

Remove Selected Edges

Use **Modeler** > **Model Preparation** > **Remove Edges** to remove selected edges. You can also use the toolbar icon when you have made an appropriate edge selection.

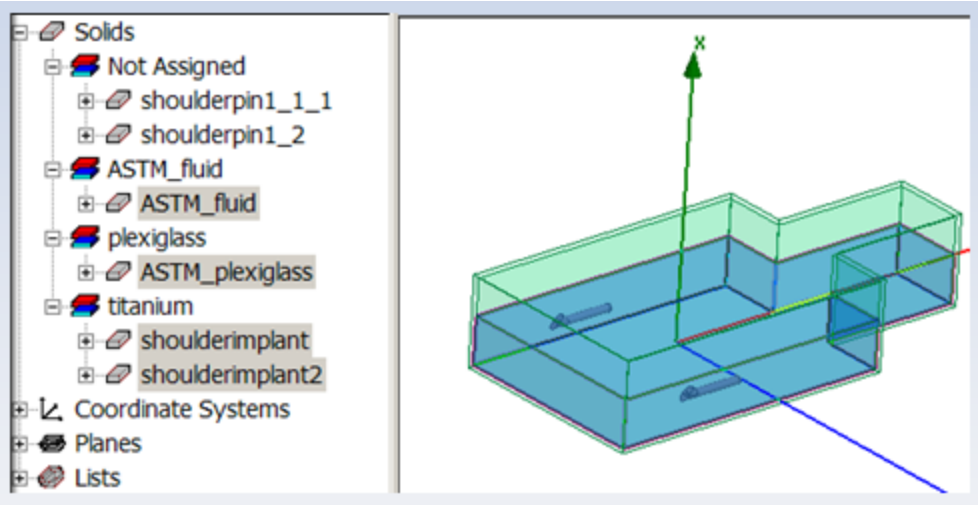
Perform Explicit Subtractions

Use **Modeler** > **Model Preparation** > **Perform Explicit Subtractions** to subtract selected History Tree objects that are fully contained by other parts.

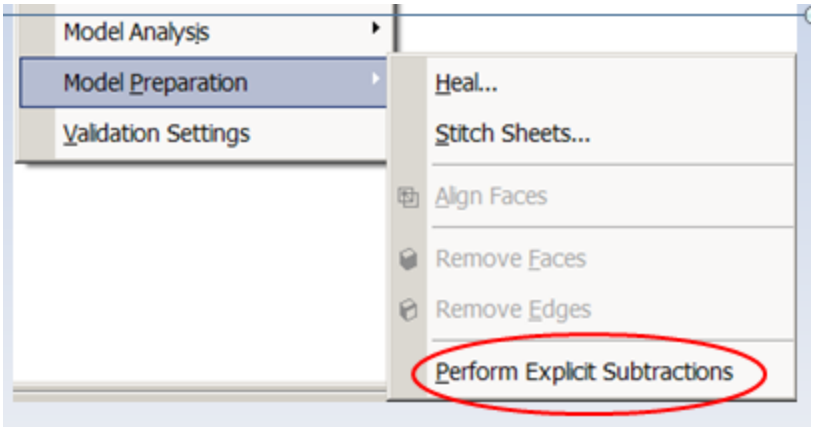
1. Select two or more history objects.

   This enables the **Modeler** > **Model Preparation** > **Perform Explicit Subtractions** submenu.

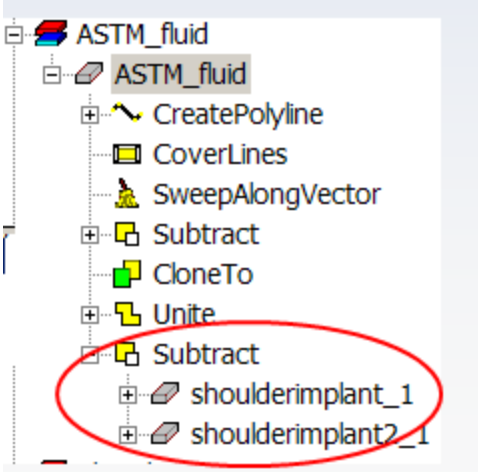
   For example, consider the following model.



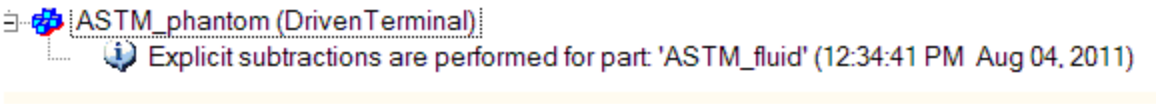
2. **Perform Explicit Subtraction** can be performed.



The results appear in the History tree as shown:



The Message window reports this successful action.



If no parts are fully contained by other parts, a message explains that no explicit subtraction is needed.

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 or IronPython scripts that you

build and compile. When user-defined primitives exist in your UserLib or PersonalLib directory (given the paths specified in **Tools** > **General Options**, Project Options tab), they appear in the menu bar under **Draw** > **User Defined Primitives**. Newly created UDPs will appear after a restart or after executing the command **Draw** > **User Defined Primitive** > **Update Menu.**

For Python-based primitives, see the **Q3D Extractor Scripting Guide**.

The modeler includes example C++ source and header files that can be used to generate DLLs. The files are located in the **UserDefinedPrimitives/Examples** subdirectory under the 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 **myDLL**.
    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 **Source** 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 resemble 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 or the **[*PersonalLibPath*]personallib/UserDefinedPrimitives** directory.
10. To view your primitives, click **Draw** > **User Defined Primitive** > **Update Menu** and then click **Draw** > **User Defined Primitive** > **[*PrimitiveToDraw*]**.

> **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 will have a **.so** extension and 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.

User Customization through User Defined Primitives (UDPs)

User Defined Primitives (UDPs) allow users to add customized geometric modeling commands to the Ansys Electronics Desktop. UDPs are C++ compiled or Python libraries that can be added to the desktop interface and shared between users with common modeling needs.

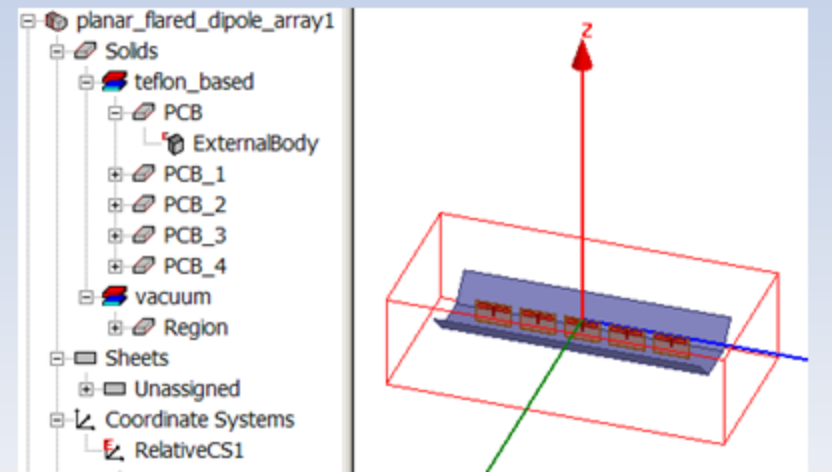
To create a UDP, see Creating a User Defined PrimitiveCreating a User Defined Primitive for requirements and the procedure for building a proper DLL or Python library.

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 given the paths specified in **Tools** > **General Options** under *General* > *Directories*. Placing an appropriately constructed DLL or Python library in this subdirectory and executing **Draw** > **User Defined Primitives** > **Update Menu** adds a new menu item in the **Draw** > **User Defined Primitives** menu to allow access to the UDP.

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 EM 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)

See the following sections:

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

UDM Properties

Library of Models for CAD Integration

Ansys EM to Ansys Geometry Transfer

Material Assignment Transfer

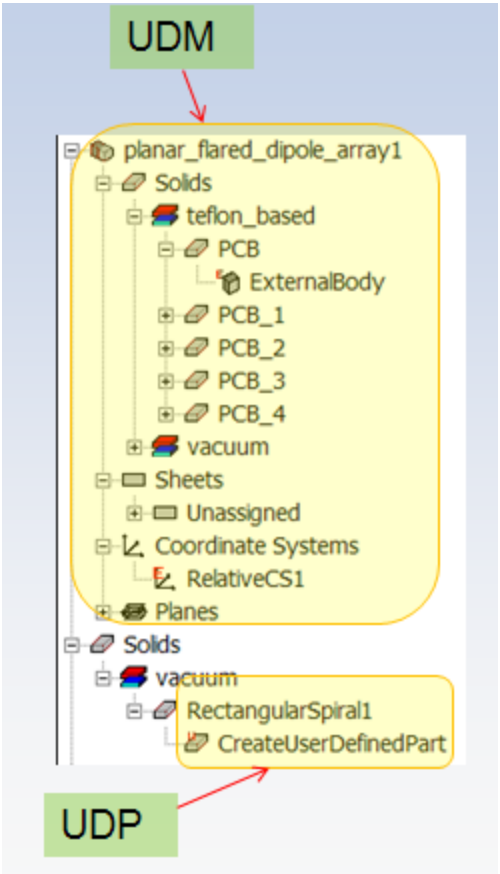
Geometry Transfer through Ansys DesignModeler (DM)

CAD Integration Functionality

See the **Q3D Extractor Scripting Guide** for information about Creating User Defined Primitives and User Defined Models in Python Scripts.

UDM Compared to User Defined Primitives

User Defined Models (UDM) resemble User Defined Primitives (UDP):



- Ansys EM 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 EM applications
- 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

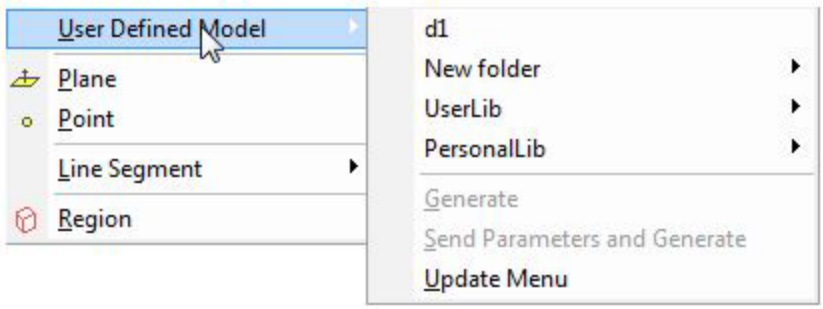
  UDP does not define part attributes or material

  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 or .py name, dll or .py location, version etc
- Option tab:
  - may have options if any

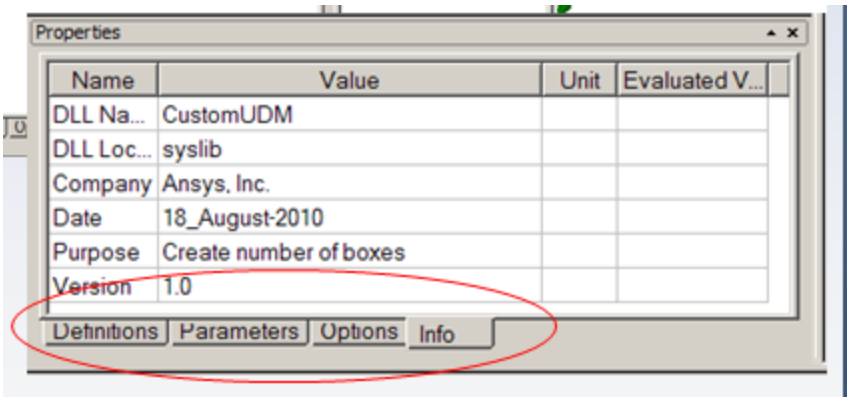
Insert UDM Command on Draw Menu

To insert a UDM into a design, use the User Defined Model command on the Draw menu for the Modeler window.



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 or .py name, dll or .py location, version etc

Option tab:

- may have options if any

UDM Parameters

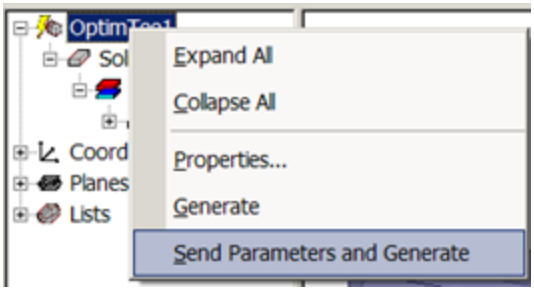
UDM geometry can be manipulated through its parameters.

- Can be mapped to design or project variable for animation, parametric analysis.
- IDs are persisted (allowing to retain boundaries) during parameter edits.

  UDM geometry in HFSS invoked through Workbench is not 'dynamic updated' upon parameter edits.

- In such cases, 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



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)

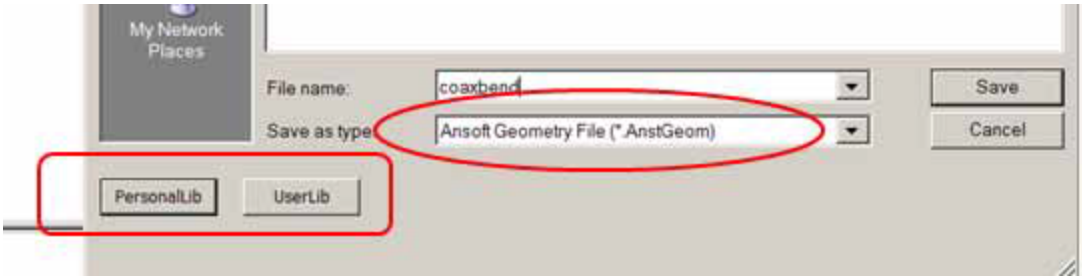
Following part attributes can be modified for UDM parts

- Model/Non Model flag
- Part orientation

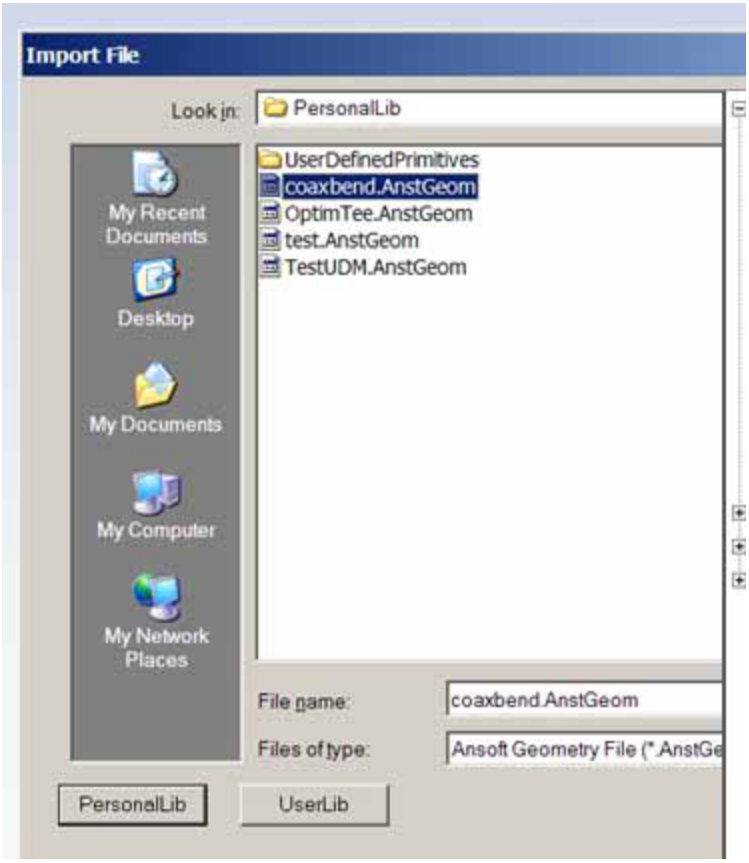
Library of Models for CAD Integration

UDM technology allows a library of models

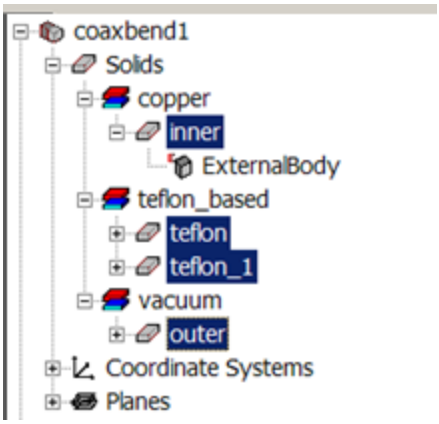
- Any Ansys EM model can be exported as 'Ansoft Geometry File'



- 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.

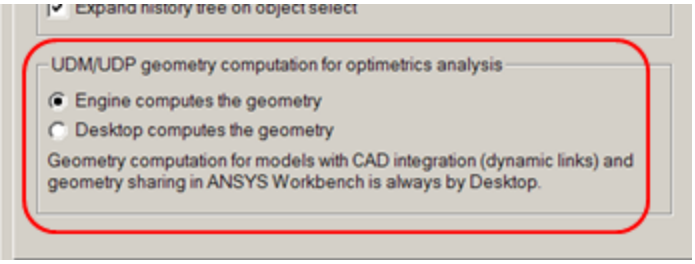
- Engine side (default):

  Requires deployment of UDM on each node

- Desktop side:

  UDM need not be deployed on each engine

  Desktop will be busy during parametric analysis



3D Modeler Discovery Link

The 3D Modeler Discovery Link allows you to create a User-Defined Model (UDM) from the geometry model in an Ansys Discovery project. Discovery Link can be used only if Discovery with the same version as Ansys Electronics Desktop is installed on the same machine. Discovery Link is only available on Windows.

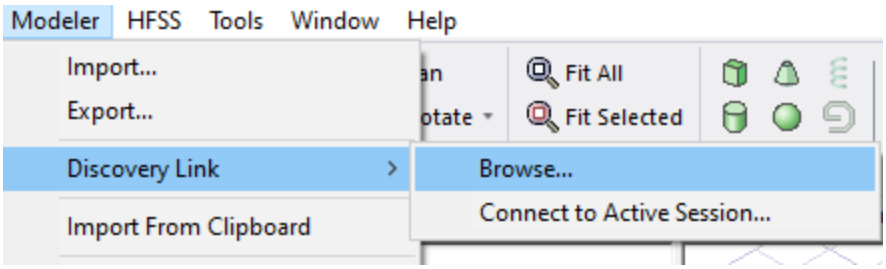
Discovery Link provides following functionality:

- Browse and select Discovery project files (*.dsco) to import geometry directly.
- Import geometry by connecting to an active Discovery session.
- The Discovery geometry is imported as a UDM along with geometry parameters with units, rendering attributes and material assignments. Object, material, and parameter names with non-ASCII characters are not supported.
- The Discovery geometries can be imported as faceted bodies if the design supports lightweight geometries.
- A Discovery document referenced by a Discovery link is included in the project archive automatically.
- You can break a Discovery link. In this case, all the parts in the UDM are converted to imported objects.

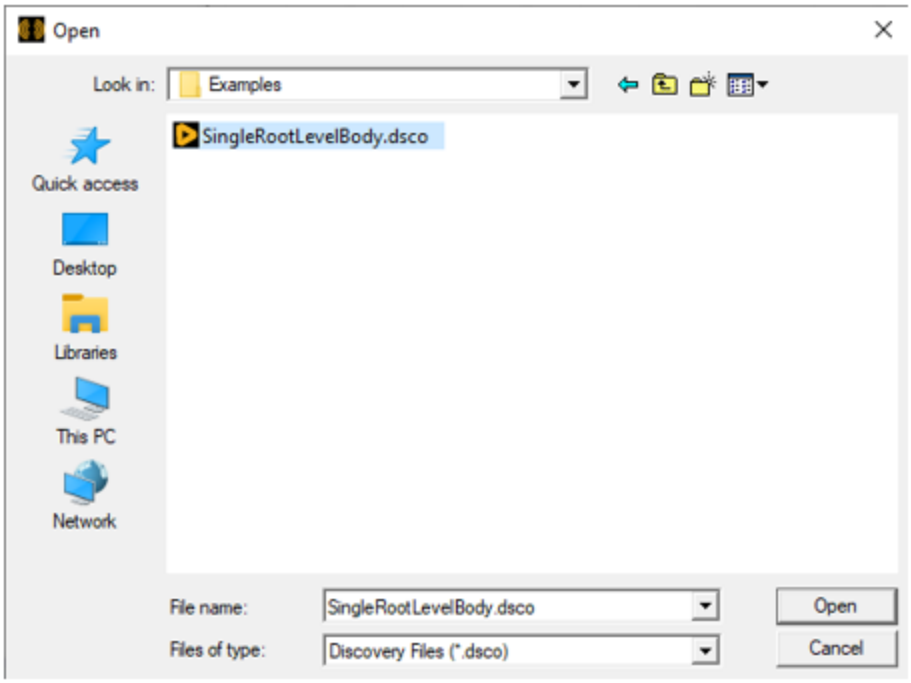
Importing Discovery Geometry into Ansys Electronics Desktop

To select a Discovery file:

1. Select **Modeler** > **Discovery Link** > **Browse** or **Browse Discovery** in the **Model** ribbon.



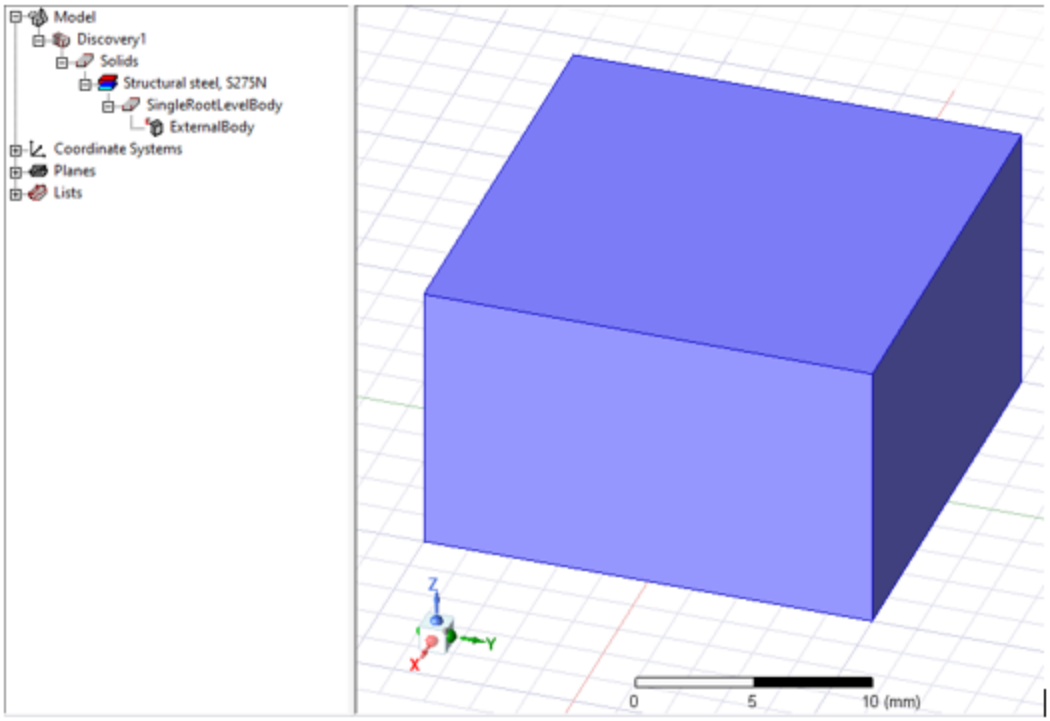
2. Navigate to and select a *.dsco project to load into the Modeler window.



3. Click **Open**.

   The progress bar displays import progress. When the model has loaded, it appears in the Modeler window, and in the history tree as a User-Defined Model (UDM) listed under

Discovery.

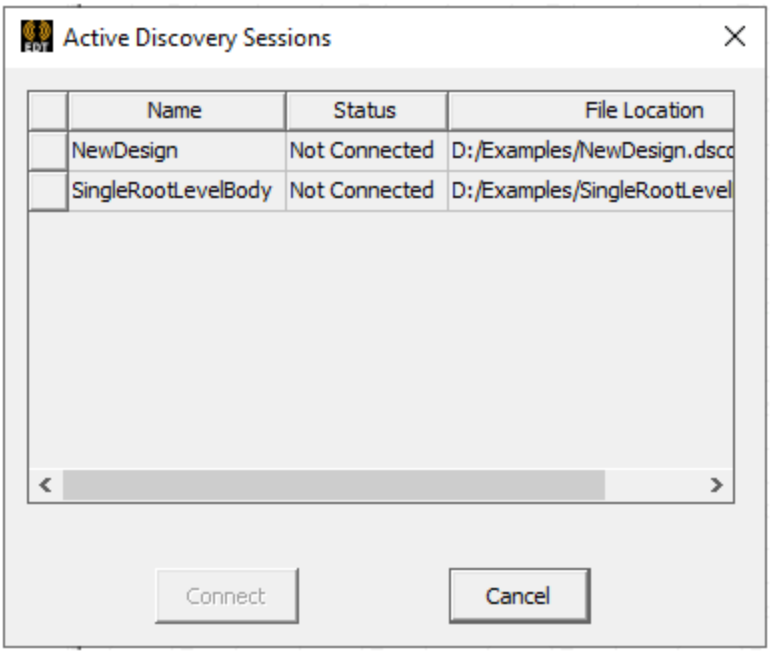


If there is an active Discovery session which is using the selected document, the created UDM will be linked to the active Discovery session. If there is not one, a new non-graphical Discovery session will be used, which makes the creating and refreshing slow. Afterwards, if an active Discovery session is launched and opens the document, the UDM will be linked to the active session.

Connecting to an Active Discovery Session

If one or more Discovery sessions are running on the machine, Discovery link can connect to them to import geometry.

When you select **Modeler** > **Discovery Link** > **Connect to Active Session** or **Connect to Discovery** in the Model ribbon, the Active Discovery Sessions dialog box appears.

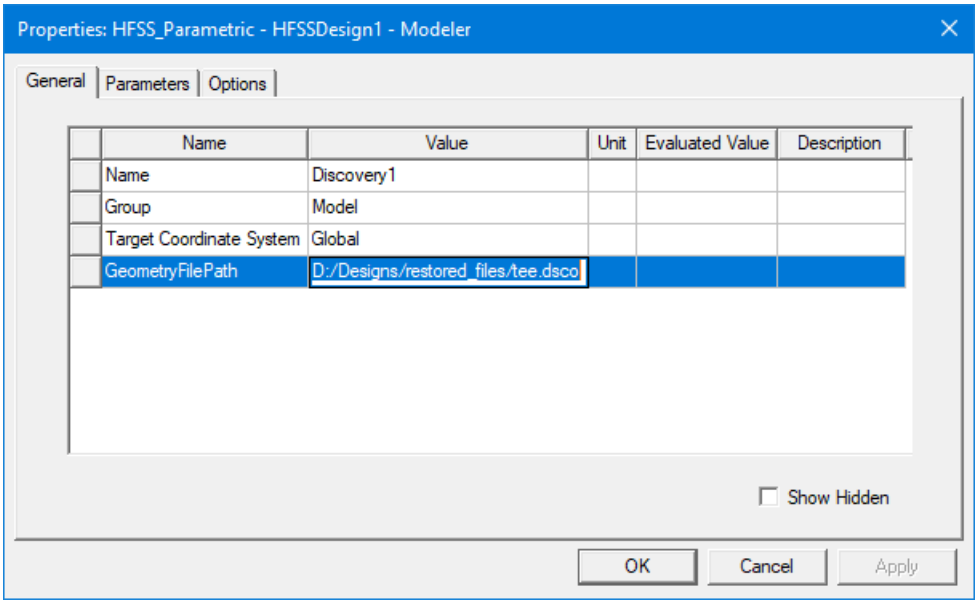


It shows all active Discovery sessions with their document names and connection status. Sessions which are not using a saved document are not included. You can select a session and click **Connect** to import geometry from that session. Once connected, that session will be used to regenerate the corresponding Discovery link UDM.

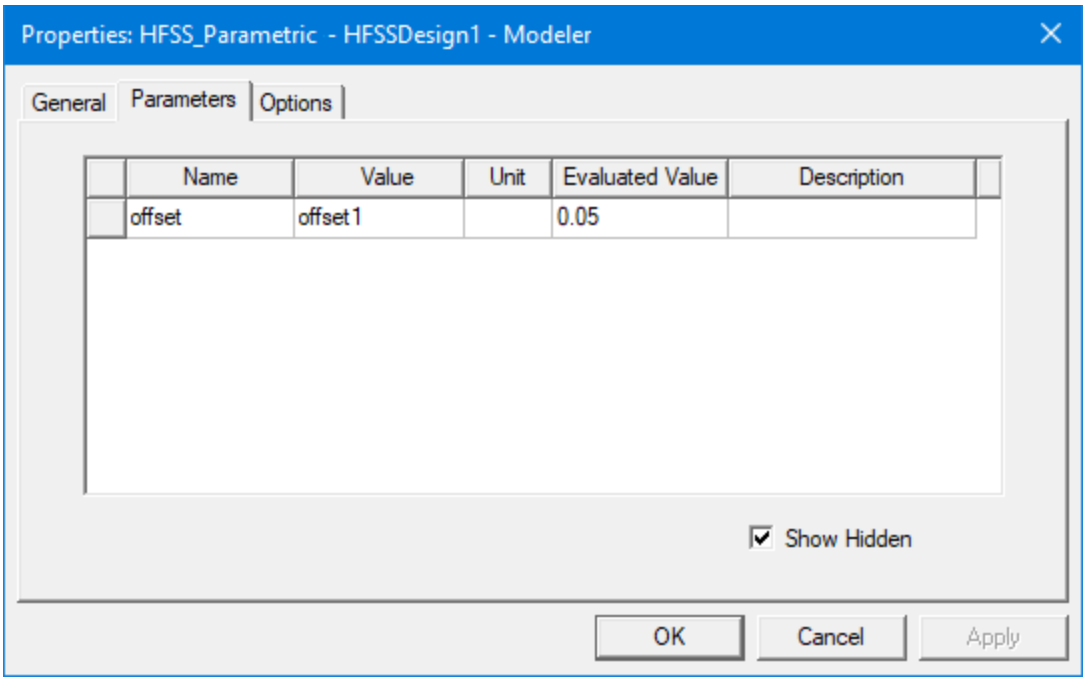
Property Window of Discovery Link UDM

The Property window of a Discovery link UDM has three tabs:

- The **General** tab shows general information such as Discovery document path. Note that the Geometry File Path is editable; users can update the path if the original file is moved.



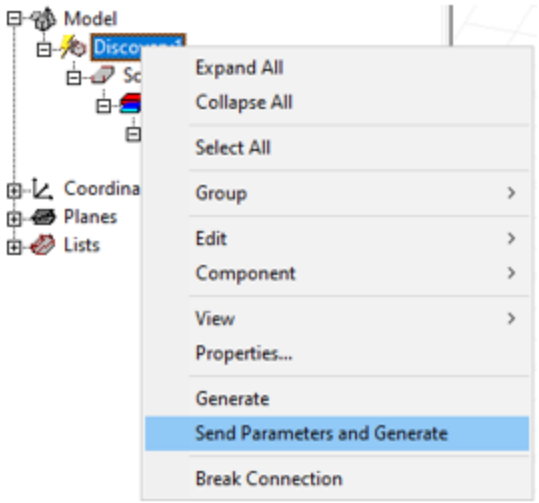
- The **Parameters** tab shows all parameters.



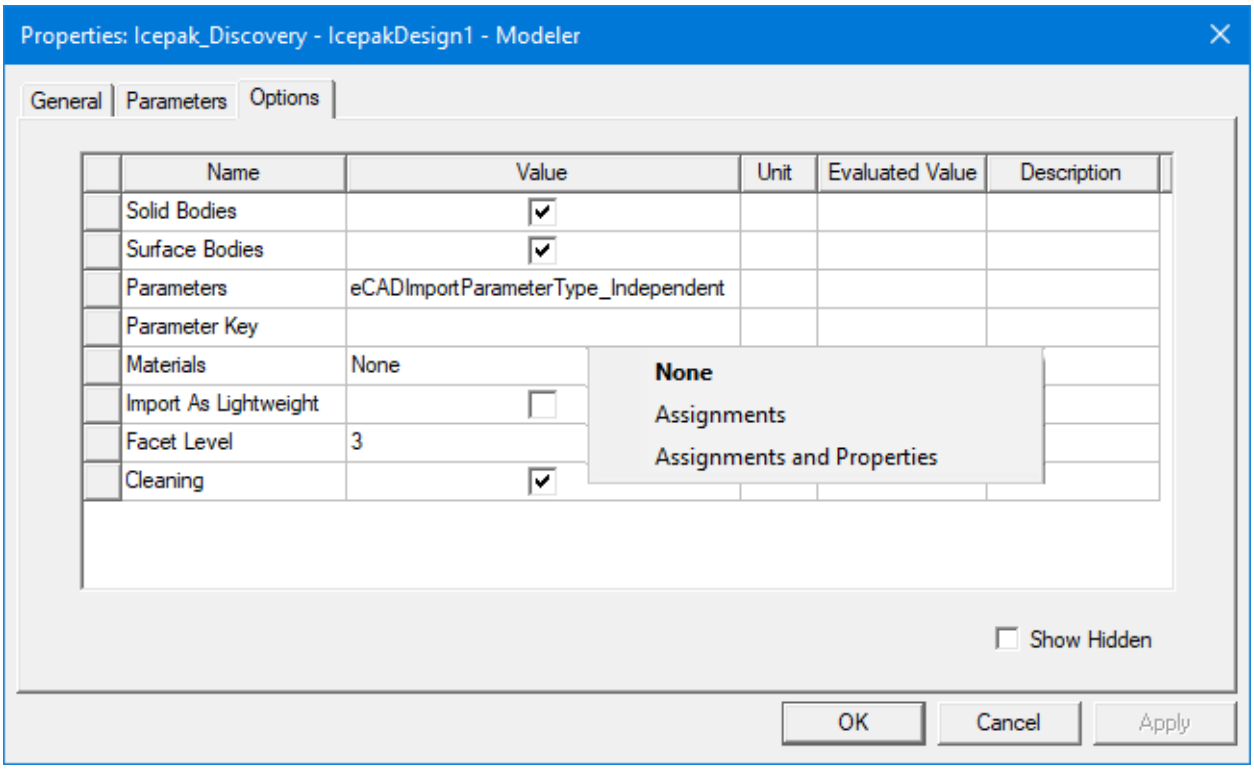
You can edit the parameters and send them back to Discovery to get the geometry updated accordingly.

After editing the parameter, you will need to use **Send Parameters and Generate** in the context menu of the Discovery link UDM. If there is an active Discovery session linked, its

model will be updated but not saved. If there is not an active Discovery session linked, the Discovery document will not be updated.



- The **Options** tab shows supported options. Note that Materials**Assignments and Properties** is only available for Icepak and Mechanical designs. See Discovery Link Options for details on the options in this window.



Archive a Project Containing Discovery Link

If you want to share a project with Discovery link with other users, you need to archive the project, because the linked Discovery file is not included in the project file. When archiving, the Discovery documents corresponding to the links will be included in the project archive automatically. Once the archive file is unarchived, referenced Discovery documents will be put in restored_files subfolder in the project directory. All definition paths are updated to refer to the reference files in restored_files folder.

Break Connection with Discovery Link

A **Break Connection** command is available in the context menu of the Discovery link UDM.

- **Break Connection** converts the UDM bodies to imported objects. The UDM will be removed.
- Any operations that you have performed on UDM bodies are preserved.
- Parameters of the UDM are lost. You lose the ability to create parametric variations with Discovery parameters.
- Attributes such as rendering attributes, materials, etc., are preserved.

SpaceClaim Integration

As of 2025R1, the direct link between the Ansys Electronics Desktop software and the SpaceClaim application has been discontinued. Users are encouraged to convert SpaceClaim documents to the Discovery application file format and to use the Discovery link (**Modeler** > **Discovery Link**) to import geometry into AEDT.

Legacy AEDT projects containing SpaceClaim links can be opened. Any regeneration of SpaceClaim geometry will open the SpaceClaim file in background session of the Discovery application. We recommend that users convert source *.scdoc* files to *.disco* files by opening and saving them in the Discovery application. Users can then modify the source file path under the UDM link to point to the Discovery file.

3D Component Libraries

Both Q3D Extractor and 2D Extractor designs support 3D components. These can be inserted into the design and then manipulated for simulation. See: Modifying Objects.

The typical workflow for using 3D components is:

1. Create 3D component(s).
2. Encrypt component contents.
3. Insert into design from Component Libraries window.
4. Simulate.

3D components used in a design are listed in the **Project Manager** under **3D Components**, and in the **History Tree** under **Model**.

3D Components in Q3D Extractor

Q3D Extractor provides access to predefined 3D component libraries. It can also import 3D component (*.a3dcomp) files or create custom 3D components from geometries and design data.

Libraries for Q3D Extractor include:

- Human Body Exteriors
- Any custom libraries inserted into the syslib, userlib, or personalib directory under \3DComponents or \3DComponents\Q3D Extractor

Components created in Q3D Extractor are compatible with other Electronics Desktop design types.

3D Components in 2D Extractor

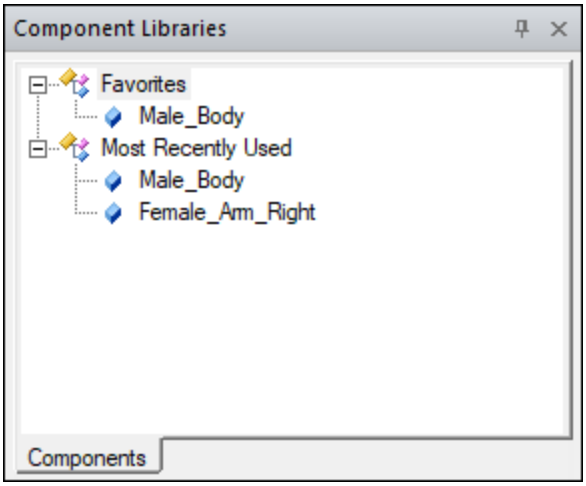
2D Extractor does not have any predefined 3D component libraries, but can create and use custom 3D components.

> **Important:**
>
> Components created in 2D Extractor are only compatible with 2D Extractor.

Component Libraries Window

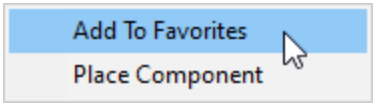
3D components are easily manipulated from the **Component Libraries** window.

To toggle display of the **Component Libraries** window, click **View** > **Component Libraries**.



The window displays a tree with:

- Any existing libraries, as described above.
- **Favorites** – displays components designated as favorites. To mark a component as a favorite, right-click it and select **Add to Favorites**.

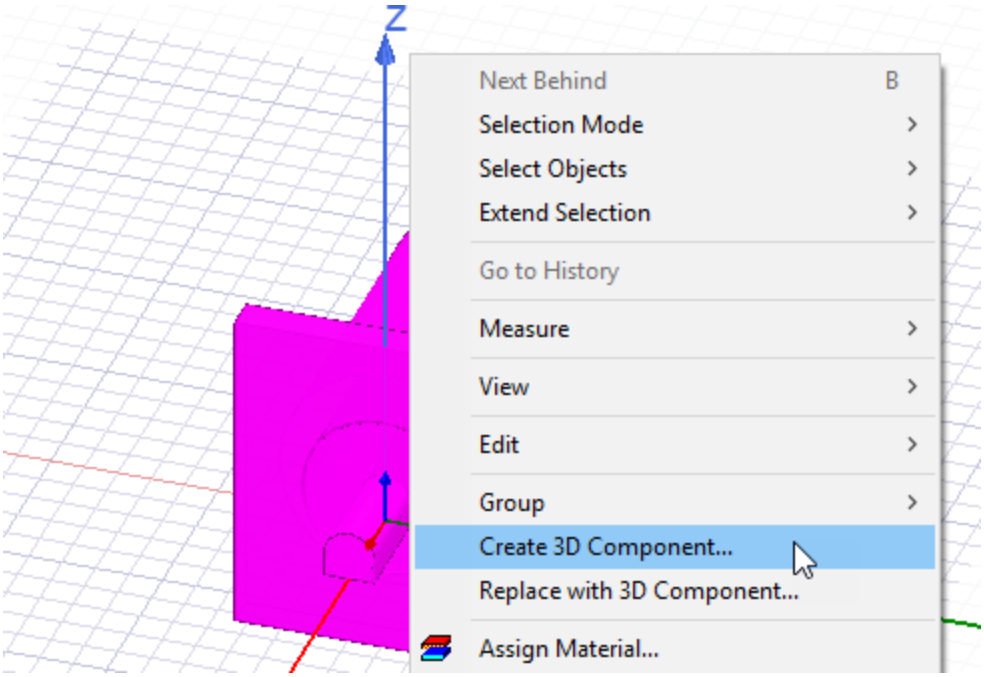


- **Most Recently Used** – automatically populates with recently used components.

Place any component into the design by dragging it into the modeling workspace, or by right-clicking it and selecting **Place Component**.

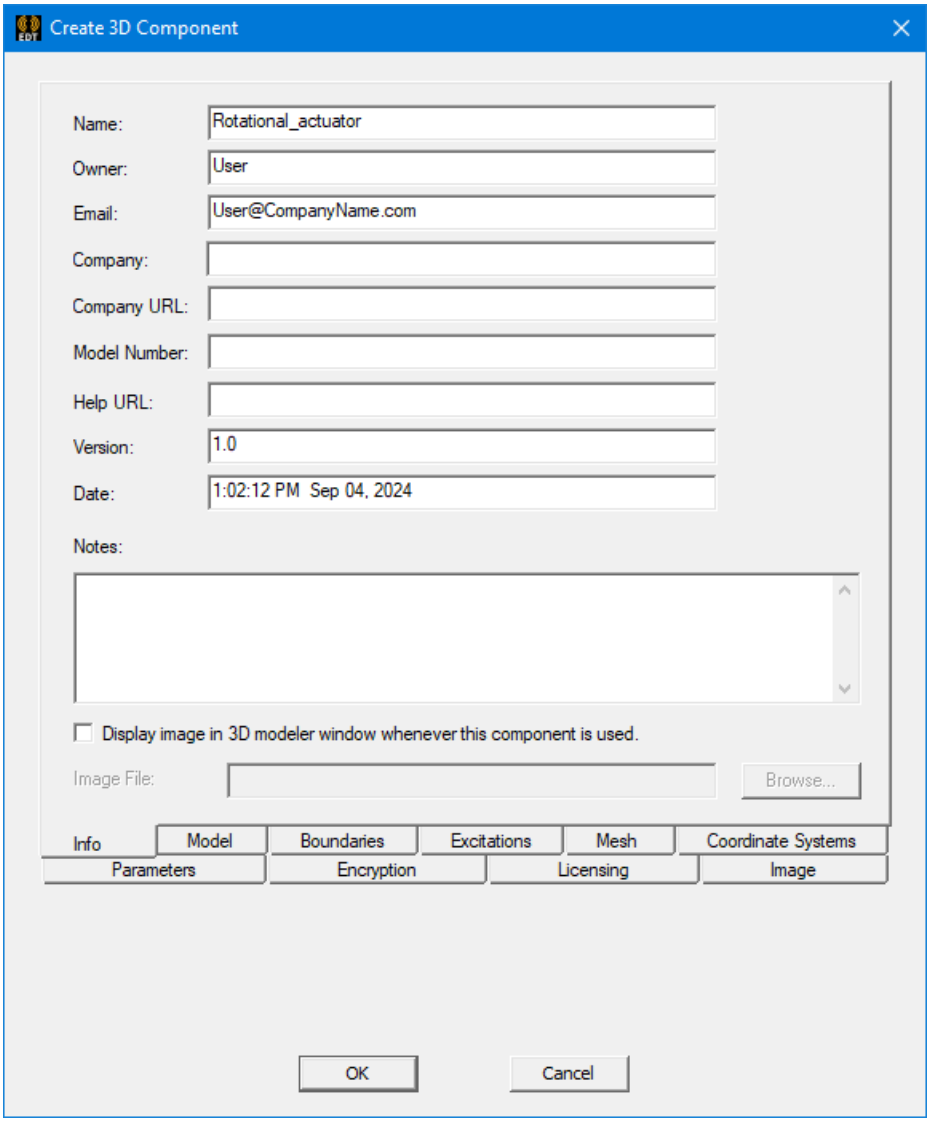
Creating a 3D Component from an Existing Model

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**. You can also go to the **Model** tab of the ribbon and select **Create 3D Component**.



This opens the *Create 3D Component* dialog box.

You cannot create components from a design with an array setup. However, you can create 3D component from a unit cell and use that to create a Multi-Unit Cell array.



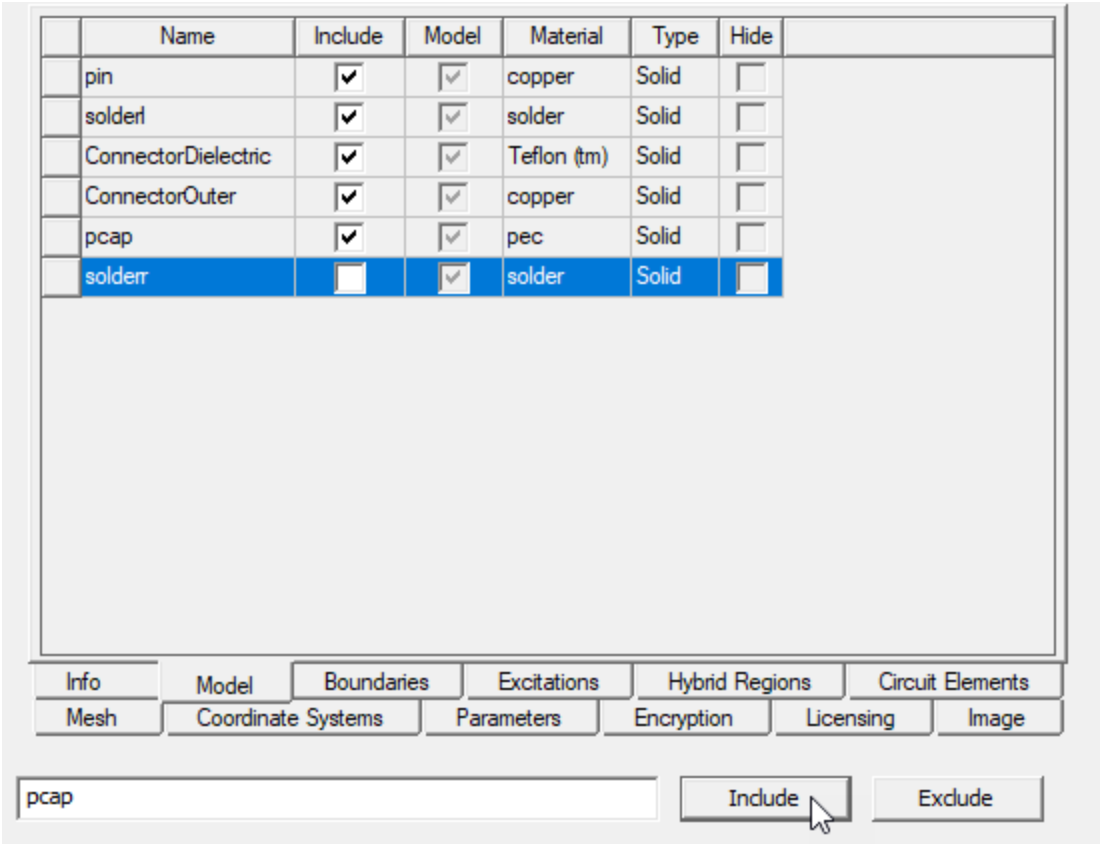
The tabs let you view the component features organized according to geometry, design data, and settings:

- Info tab
- Model tab
- Boundaries tab
- Excitations tab
- Circuit Elements tab
- Hybrid Regions tab
- Mesh tab
- Coordinate Systems tab

- Parameters tab
- Encryption tab
- Licensing Tab
- Image tab

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 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 **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 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, you see a dialog for saving the component to a location.

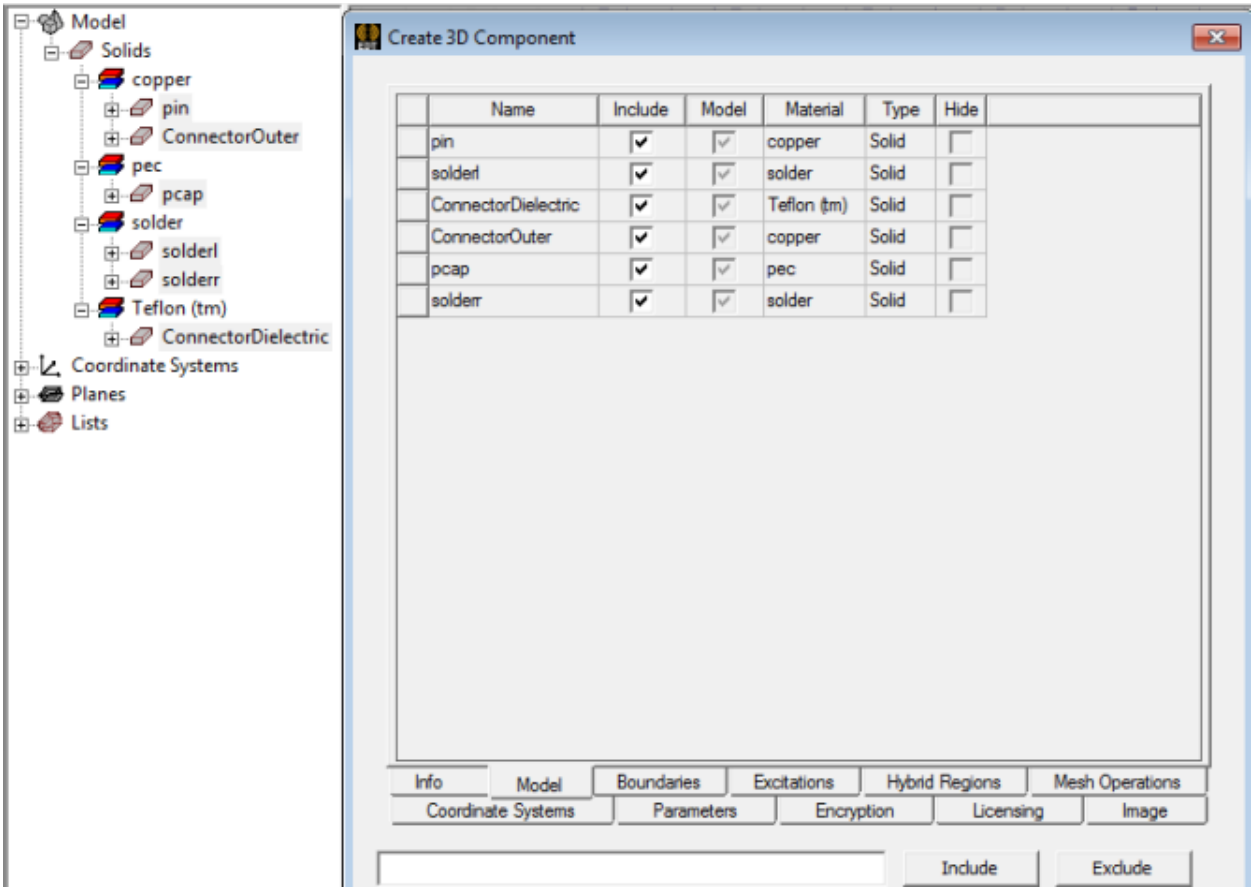
Create 3D Component: Model Tab

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 **Include** check box to include objects in the 3D component (selected), or to Exclude them from the component (cleared).

  Alternatively, you can type an object **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 objects with similar names in a single operation.

- The **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 Selected Contents** option selected on **Encryption** tab.



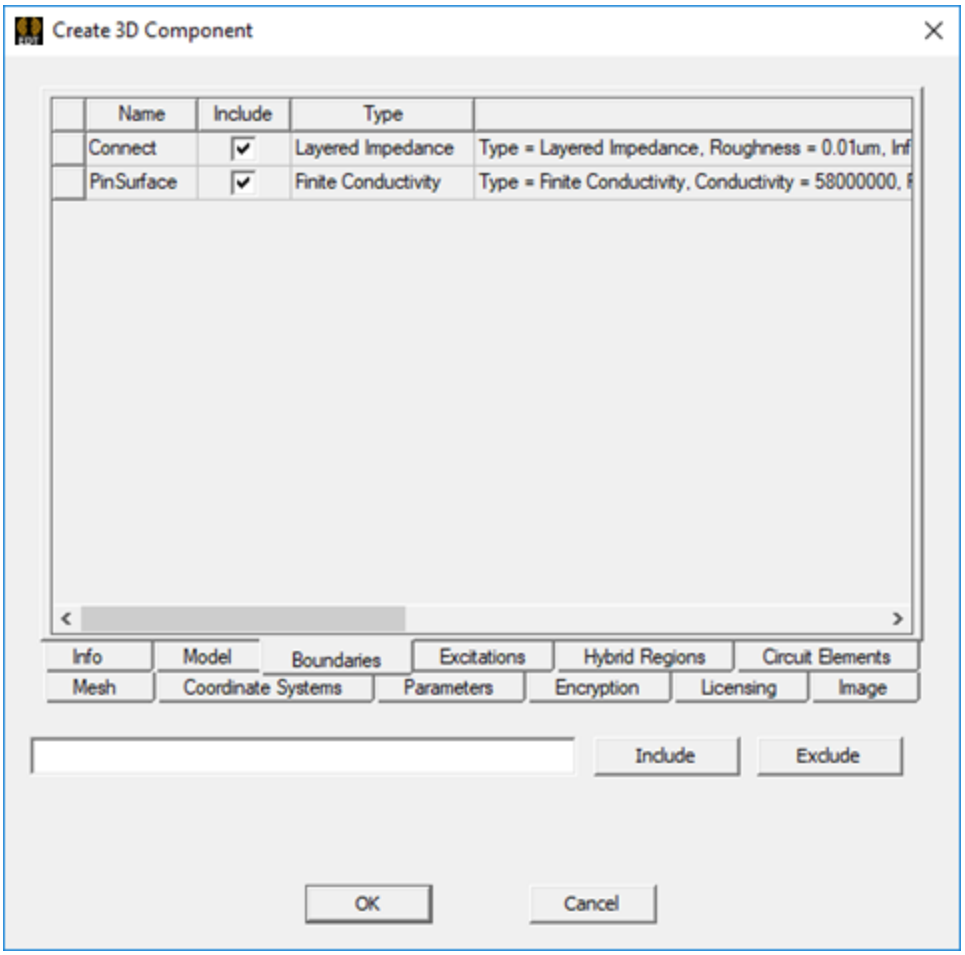
Create 3D Component Boundaries Tab

The list is populated with all the boundaries on 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.

- The *Type* and *Description* columns are read only.



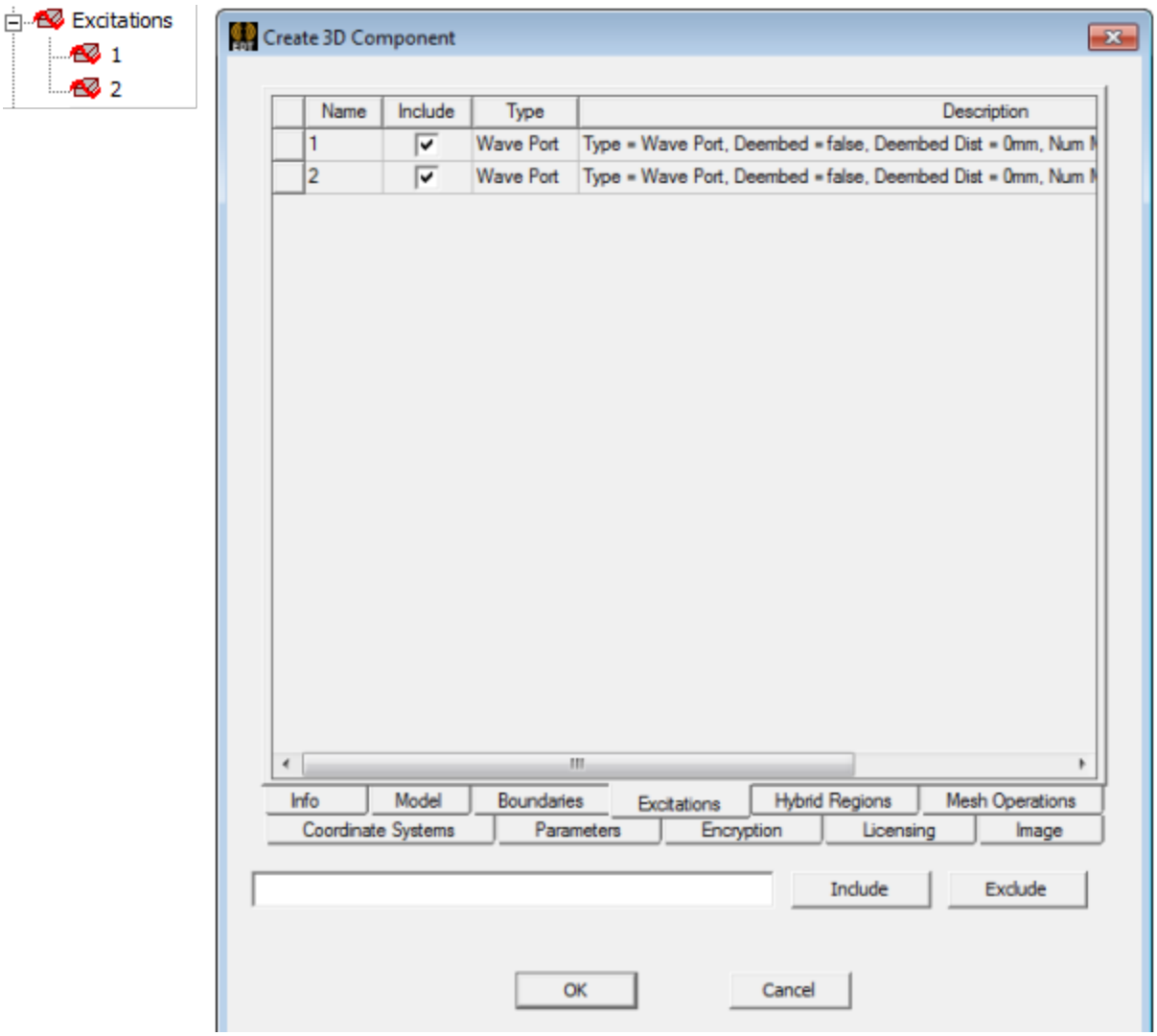
Create 3D Component Excitations Tab

The list is populated with all the excitations on selected objects.

- Use the **Include** check box to include excitations in the 3D component (selected), or to Exclude them from the component (cleared).

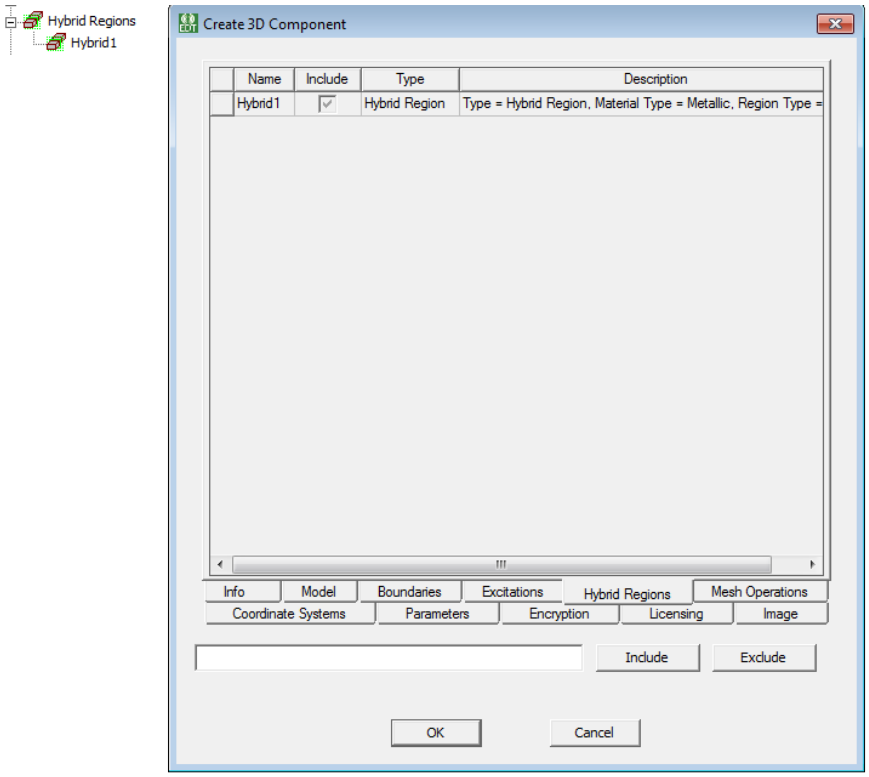
  Alternatively, you can type an excitation *Name* into the text box and click the **Include** or **Exclude** button. Wildcards (? and *) are supported when you use this method, so you can change the inclusion state of multiple excitations with similar names in a single operation.

- *Type* and *Description* columns are read only.



Create 3D Component Hybrid Regions Tab

If you have included Hybrid Regions for the model they are listed in the **Hybrid Regions** tab. You can use the Include check box to include them in the created model. However, if you create a 3D Component from an SBR+ Solution type design, it will not contain a **Hybrid Regions** tab and can only be used in an SBR+ Solution type design.



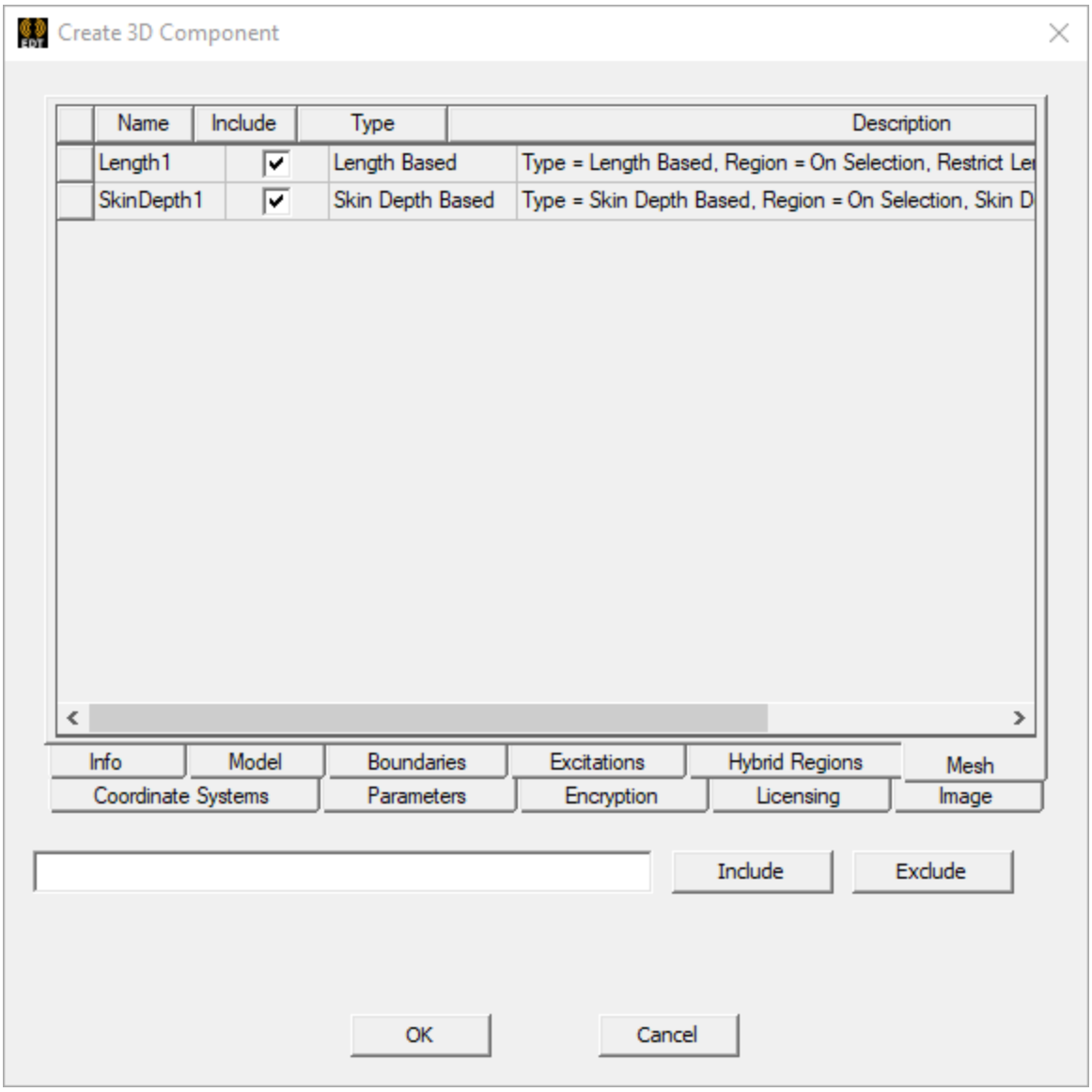
Create 3D Component Mesh Tab

The list is populated with all the mesh operations on selected objects.

- Use the **Include** check box to include mesh operations in the 3D component (selected), or to Exclude them from the component (cleared).

  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

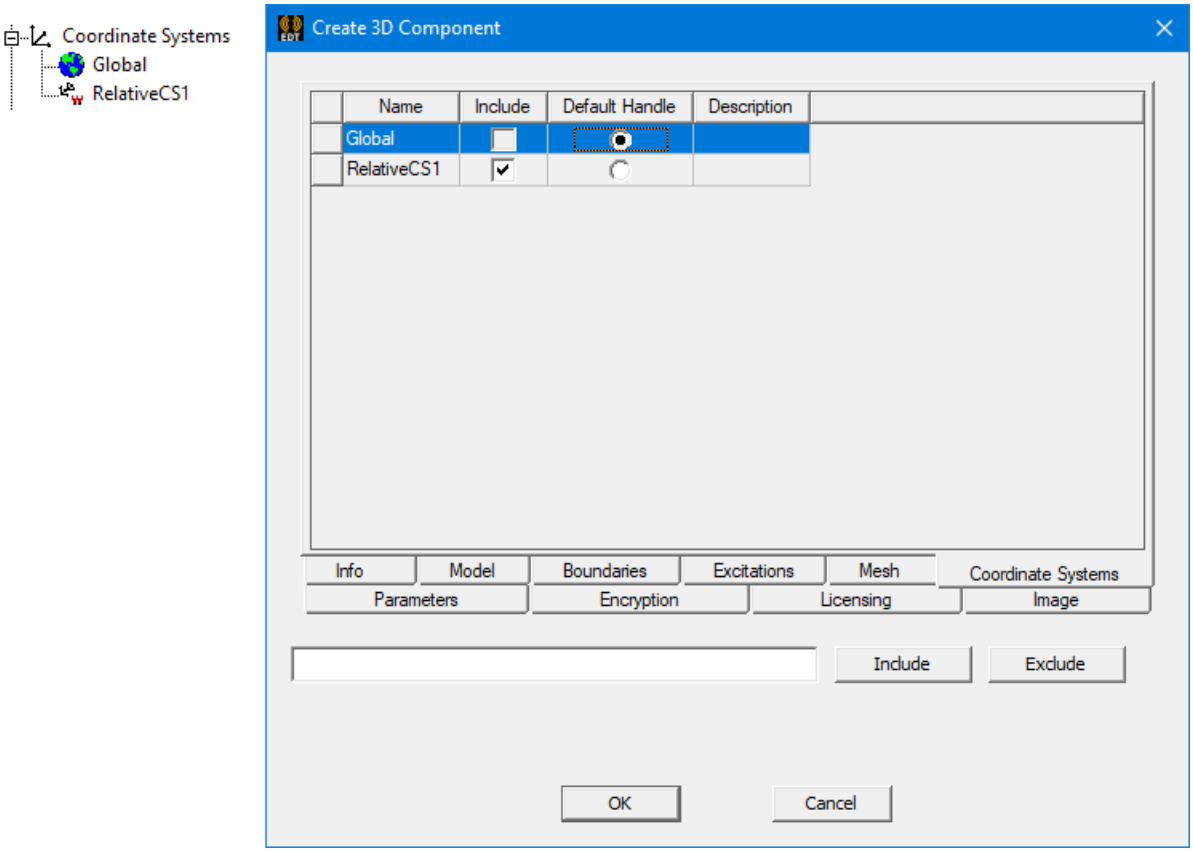
The list is populated with all the Coordinate Systems on selected objects. By default, only the Coordinate Systems used to define the object orientations are included. You can include Coordinate Systems created on specific model parts.

- Use the **Include** check box to include coordinate systems in the 3D component (selected), or to Exclude them from the component (cleared).

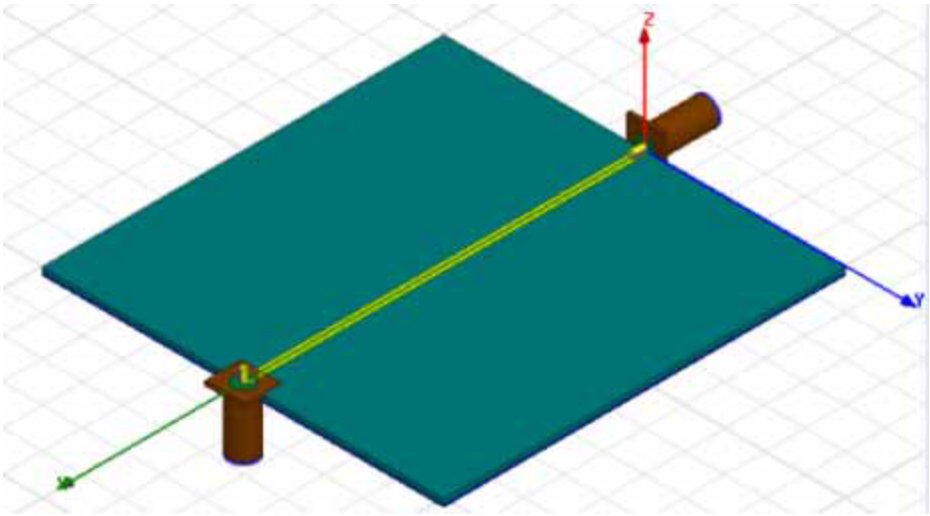
  Alternatively, you can type a coordinate system *Name* into the text box and click the **Include** or **Exclude** button. Wildcards (? and *) are supported when you use this method,

so you can change the inclusion state of multiple coordinate systems with similar names in a single operation.

- You can select any of the CS as the default handle. When this component is placed in assembly design, the handle CS is aligned to target CS in assembly design. Note that the handle CS can be changed after import; see Editing 3D Component Properties for more details.
- By default, the current working CS is designated as the default handle CS.
- The default handle CS must be included in the model (except for global).



The specified default handle affects the orientation of the component upon insertion:



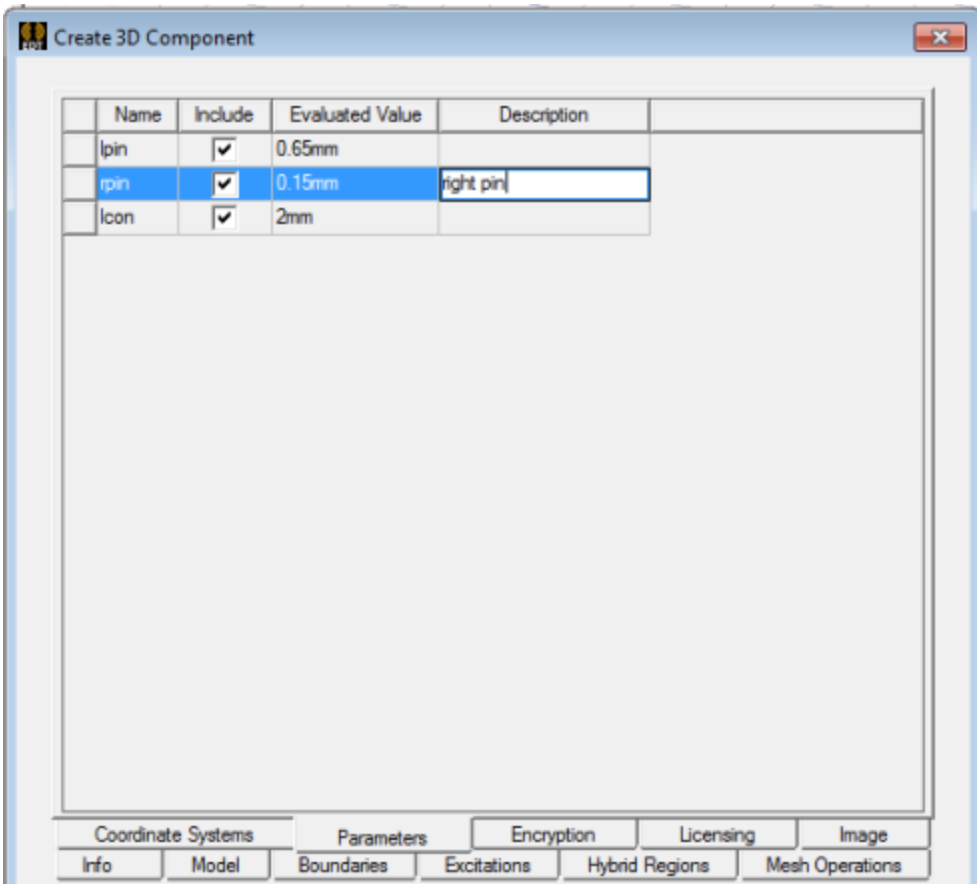
Create 3D Component Parameters Tab

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).

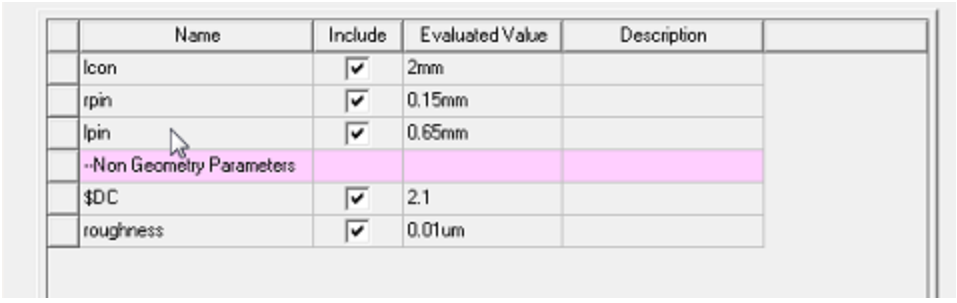
  Alternatively, you can type a parameter **Name** into the text box and click the **Include** or **Exclude** button. Wildcards (? and *) are supported when you use this method, so you can change the inclusion state of multiple parameters with similar names in a single operation.

- You can fill in the **Description** field, if desired.



Non-Geometry Variables in 3D Components

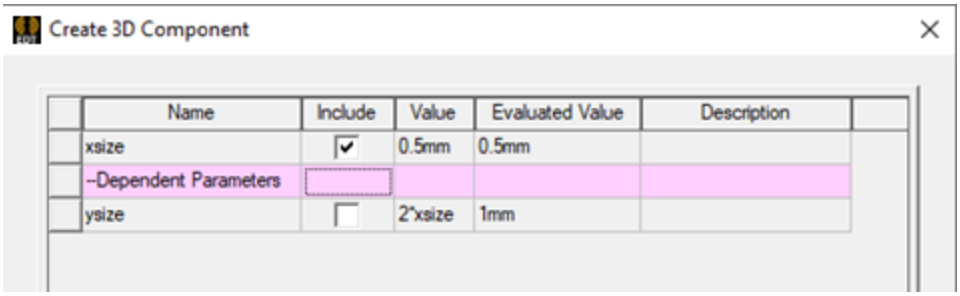
If you create non-geometry variables materials or boundaries, these are listed separately:



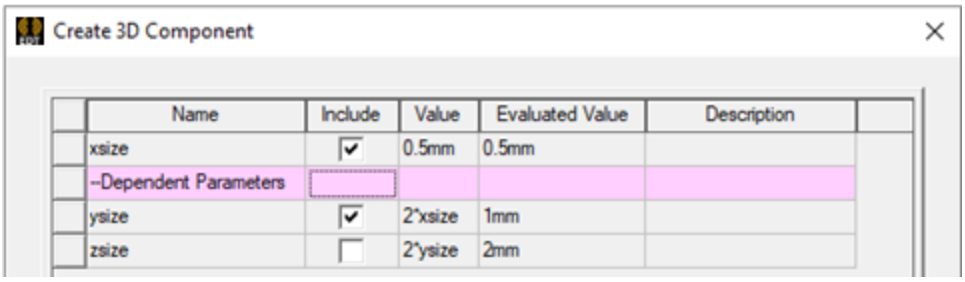
If you exclude any geometry, boundary, or material associated with a variable, the associated variable is automatically excluded on the **Parameters** tab.

Dependent Variables in 3D Components

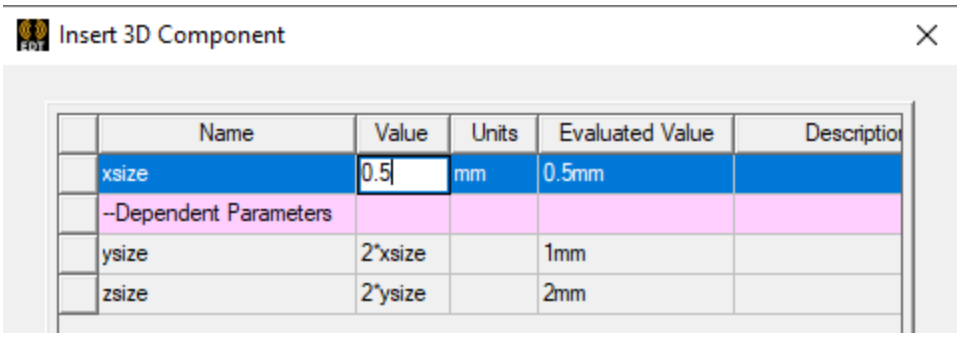
If you create dependent variables, you can choose to display the values.



If more variable dependency occurs for an Included variable, such variables are also displayed with the option to include.

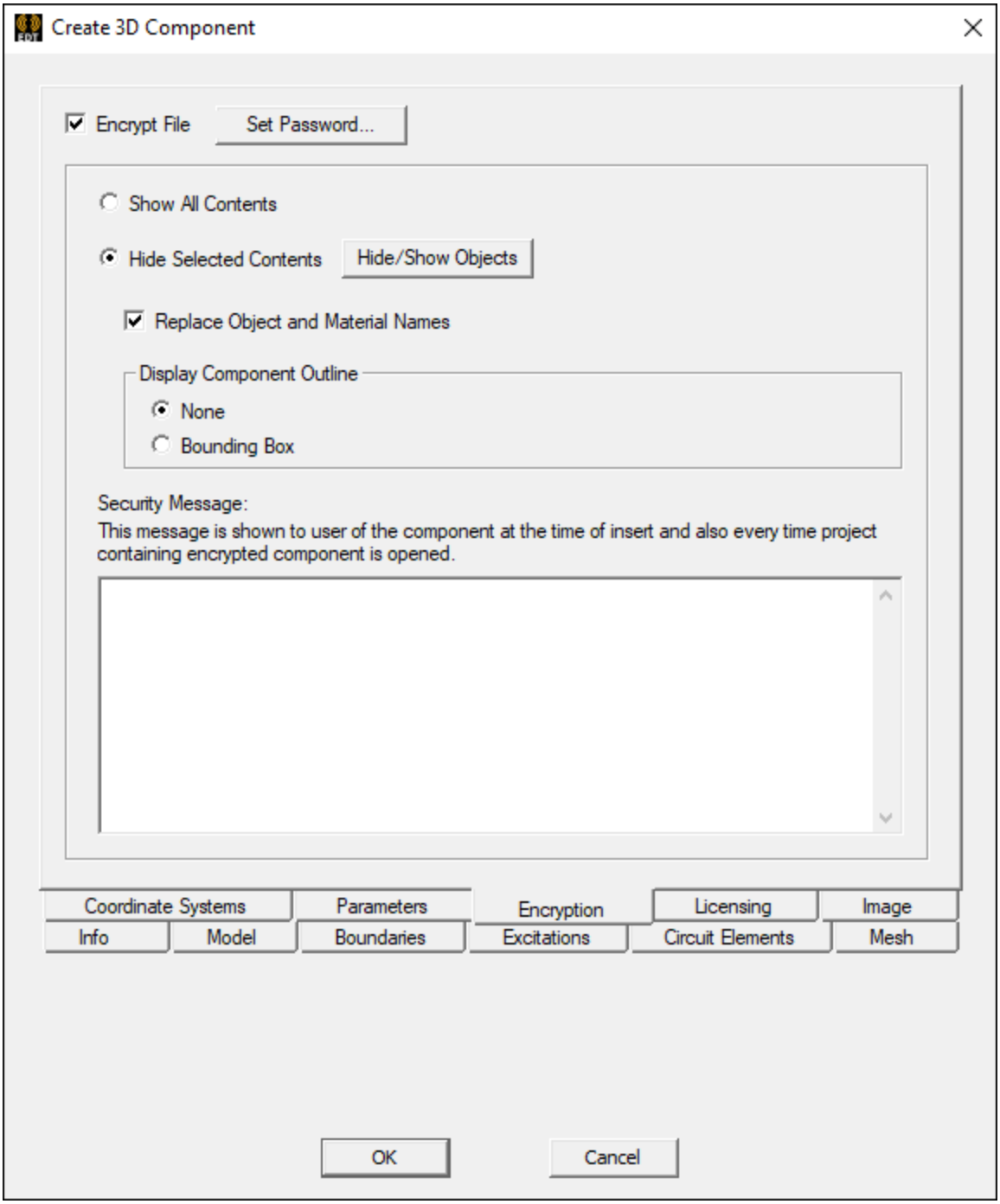


When you Insert a 3D Component with dependent Parameters included, you can edit the Independent variables, but can only view the evaluated values of the Dependent Parameters.



Create 3D Component Encryption Tab

The **Encryption** tab contains options for 3D component encryption.



Selecting the **Encrypt File** check box enables all options and automatically launches the **Set Passwords** window.

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 Electronics Desktop unless **Allow Component Edit with User Password** was selected when defining the component.
- 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 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.

    > **Important:**
    >
    > 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).

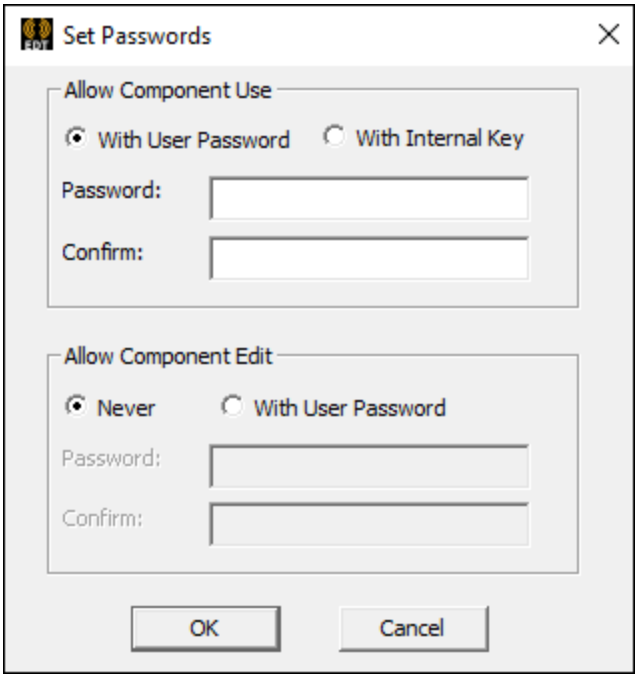
    See: Hiding Component Content.

- When using **Hide Selected Contents**, 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.
- The **Security Message** field specifies text that is displayed at component insertion and when the project containing the component is opened.
- 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 the file can be opened without entering a password.

> **Important:**
>
> 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 design contains an encrypted component with hidden contents, it will also protect solver inputs and other project files containing data that could be used to extract component information. However, note that solver inputs associated with SBR+ regions are not protected, even when encrypted components are in use.
> - For a hybrid design that contains a combination of SBR+, FEM, and IE regions: the FEM and IE regions will be fully protected, while the SBR+ regions will not.
> - Other design types, such as Maxwell, Q3D, or Icepak, do not include extended protection of solver inputs and other project files.

Set Passwords

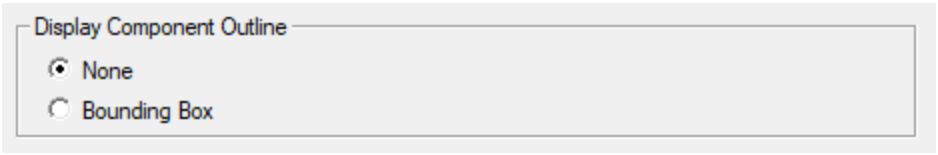
Clicking **Set Password** launches a window containing fields for setting component passwords. This is the same window that automatically launches upon enabling **Encrypt File**.



- **Allow Component Use** – allows you to select between encrypting with a user password or an internal key (which requires no additional authorization). If setting a password, enter it in both the **Password** and **Confirm** fields and click **OK**.
- **Allow Component Edit** – allows you to choose between never allowing edits to components or allowing edit with a user password. If selecting the latter, enter a password in both the **Password** and **Confirm** fields and **click** OK.
- When enabling a password for both component use and component edit, the password does not have to be the same.

Hiding Component Content

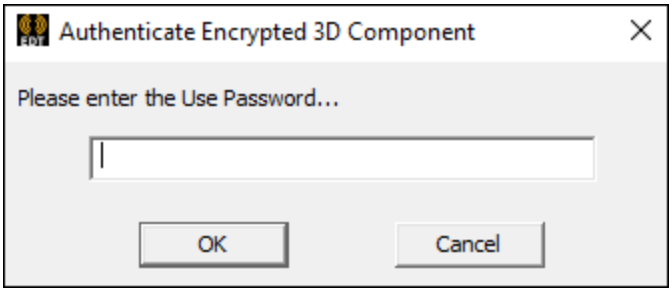
The **Hide Selected Contents** option allows selected parts of the component to be displayed either as a **Bounding Box** or without an outline (**None**).



Once encryption is set, visibility can be set on a per-object basis using the **Model** tab of the **Create 3D Component** window. Preview the component using the**Image** tab.

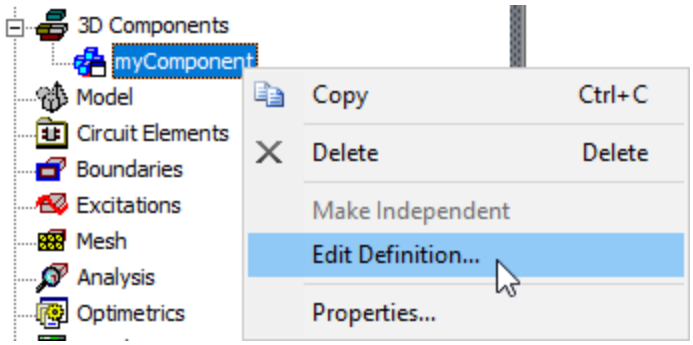
Inserting Encrypted Components

When an encrypted 3D component is added to a design, a prompt will appear asking for the password.

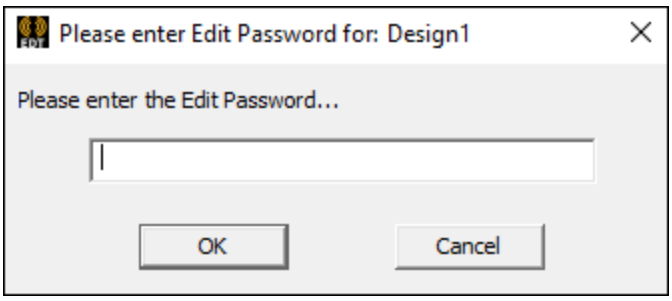


Editing Encrypted Components

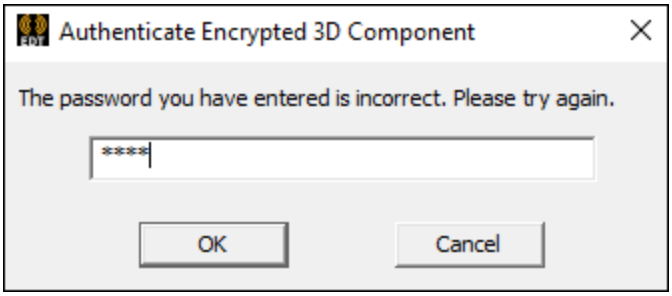
3D component definitions can be edited from the **Project Manager**:



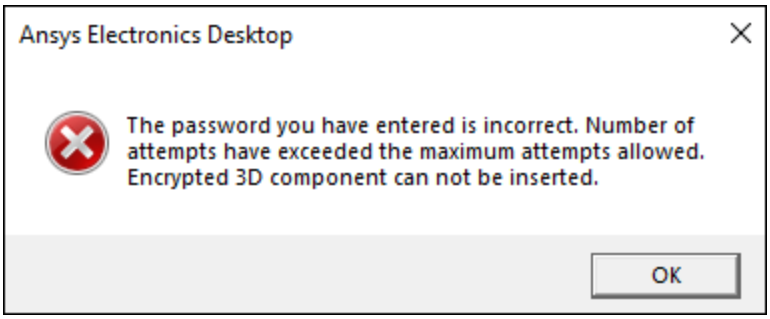
A prompt appears asking for the password:



If the password is incorrect, a warning appears:

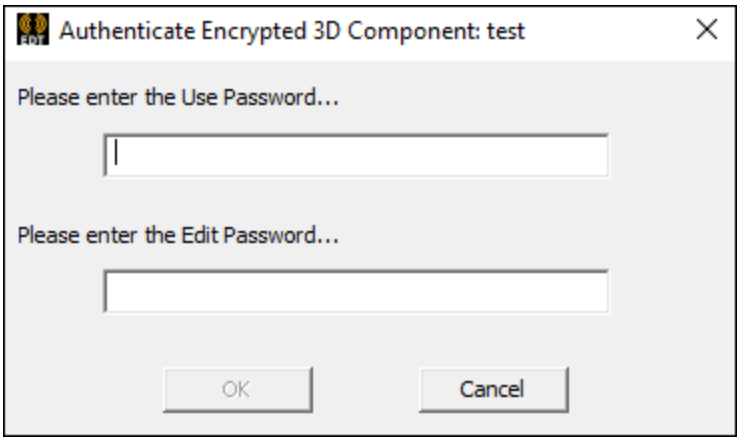


There is a three attempt maximum, after which a warning will appear and no further attempts may be made:



Opening a 3D Component File

When opening an encrypted 3D component file using **File** > **Open**, both the use password and edit password must be entered:

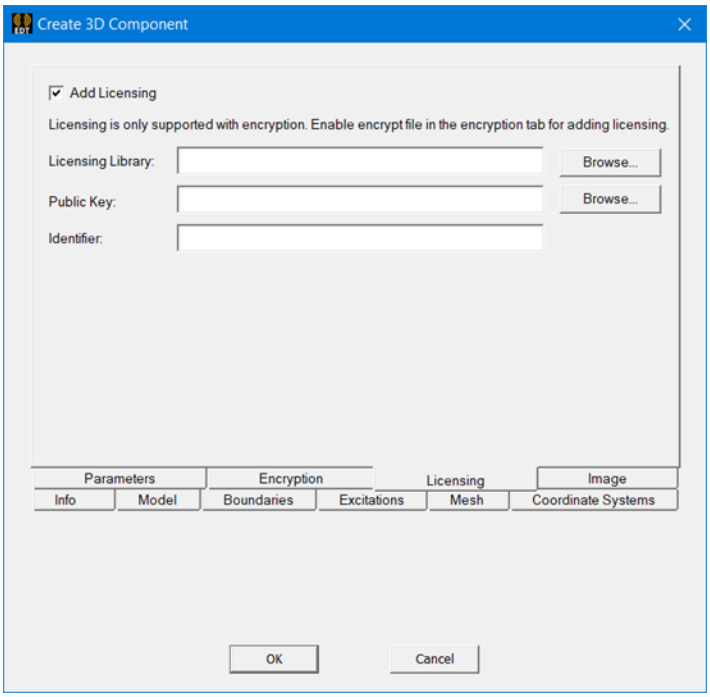


Copying and Pasting Encrypted Components

Encrypted components may be Copied and Pasted. They will be remain encrypted.

Create 3D Component Licensing Tab

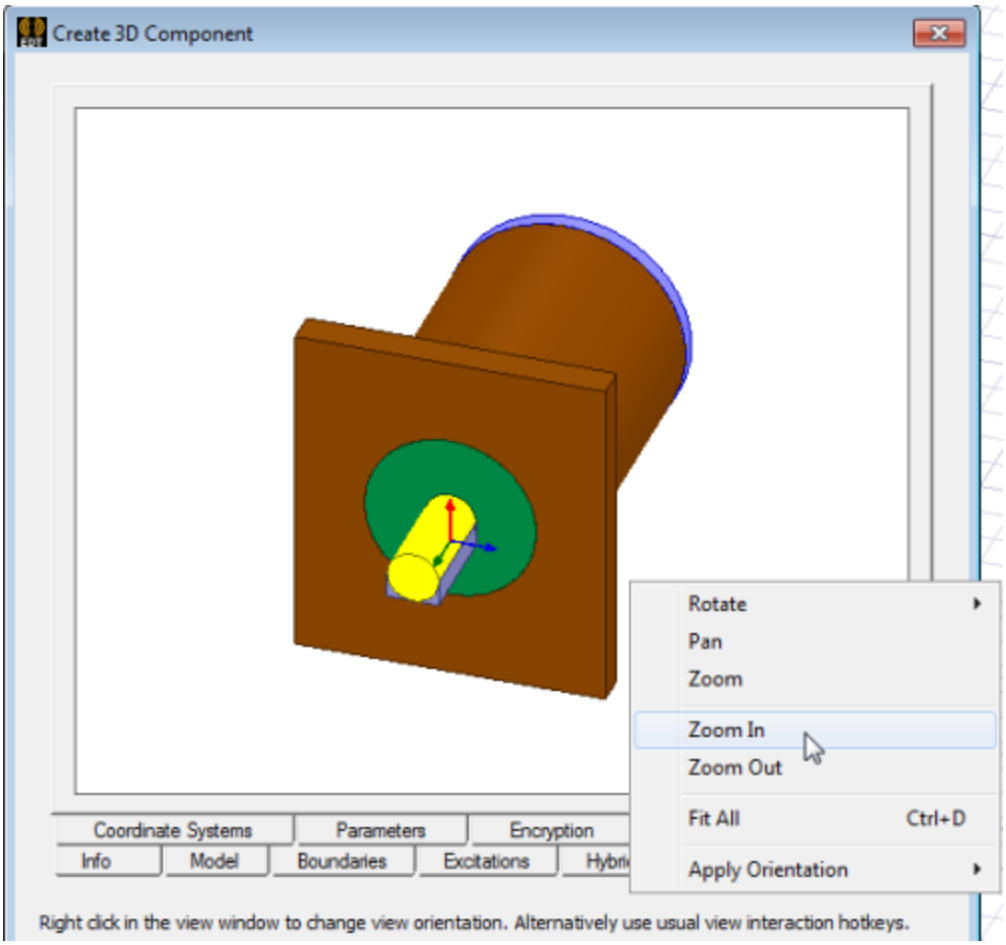
A 3D Component can be licensed using an external licensing library. This feature is for those who want to license and distribute their 3D Component models. On the **Licensing** tab, you specify a 3D Component Licensing dll, a public key and an optional component identifier. You can license a 3D Component only if it is also encrypted.



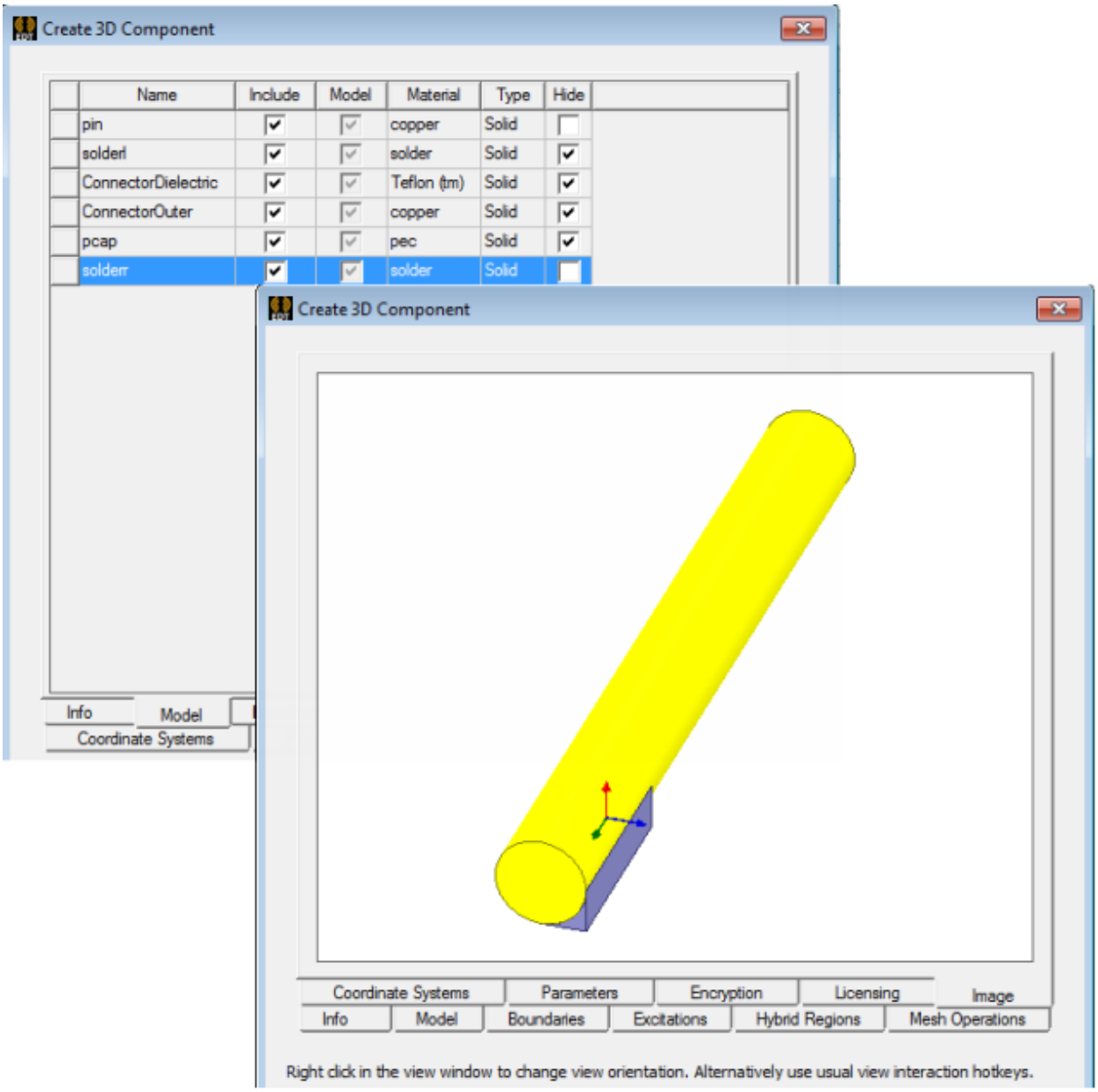
Create 3D Component Image Tab

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.

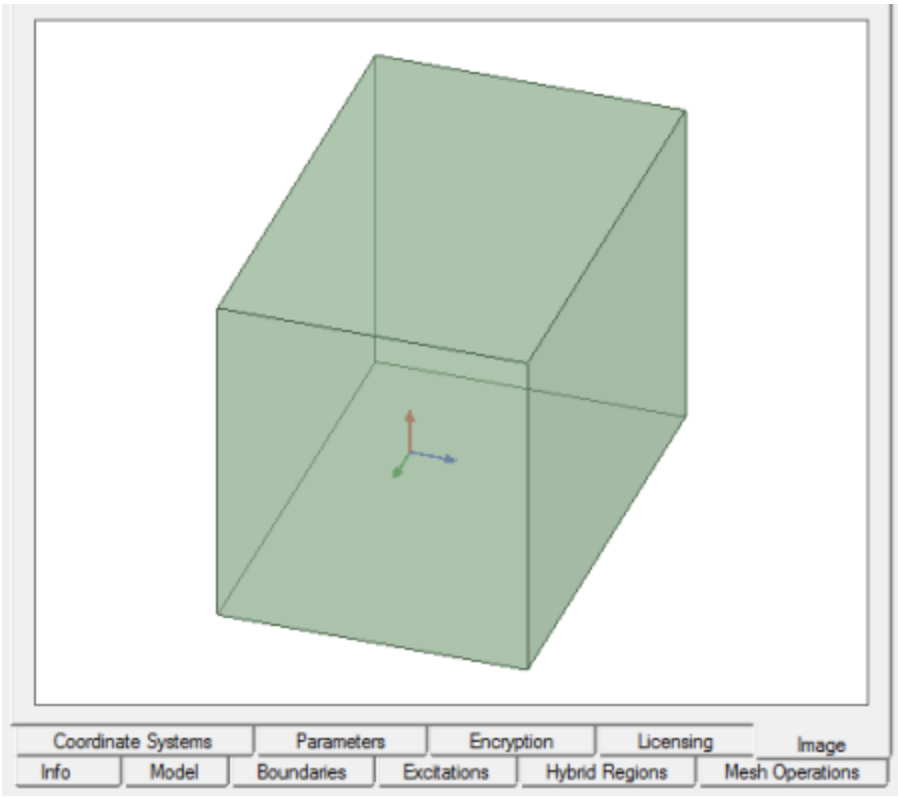
- You can right-click for a shortcut menu to change the view of the preview image.
- Changing the preview orientation does not affect the modeler window view.



If you use the **Encryption** tab and **Model** tab settings to hide/show the model or specified model objects, the **Image** tab responds. You must use encryption to enable the Hide check boxes on the **Model** tab.

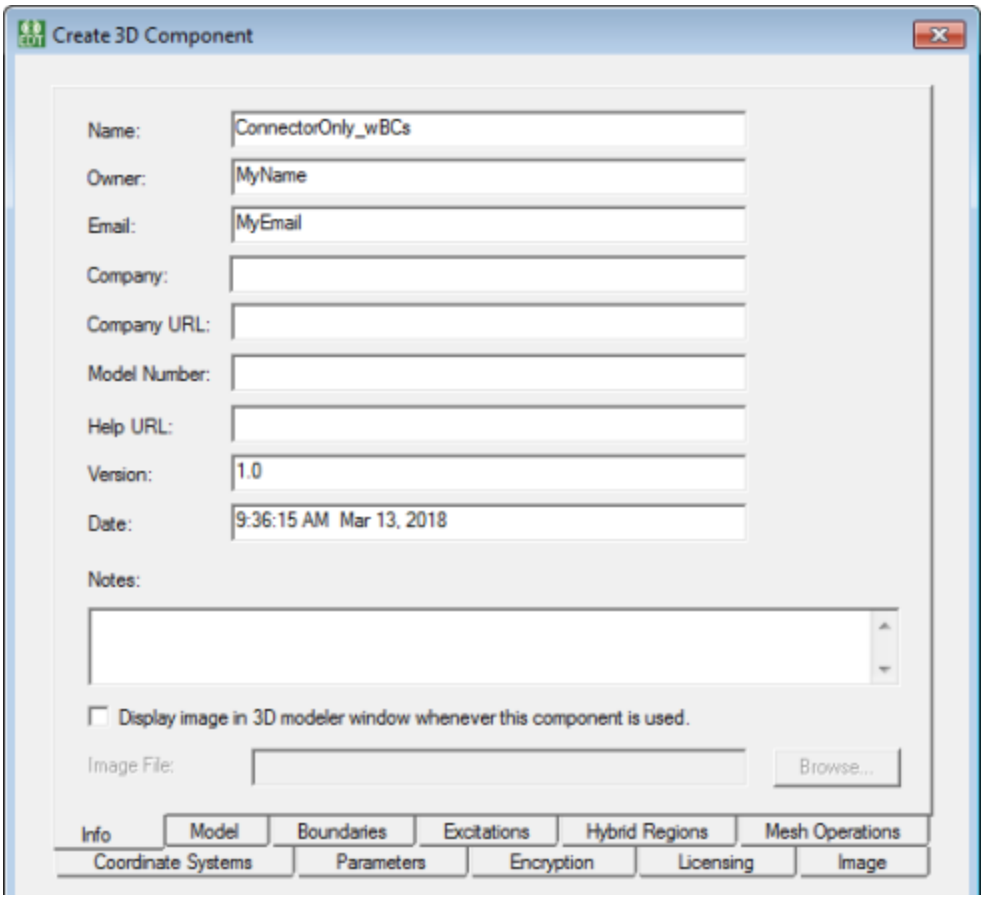


If you use the **Encryption** tab to display only an object outline, the **Image** tab responds.

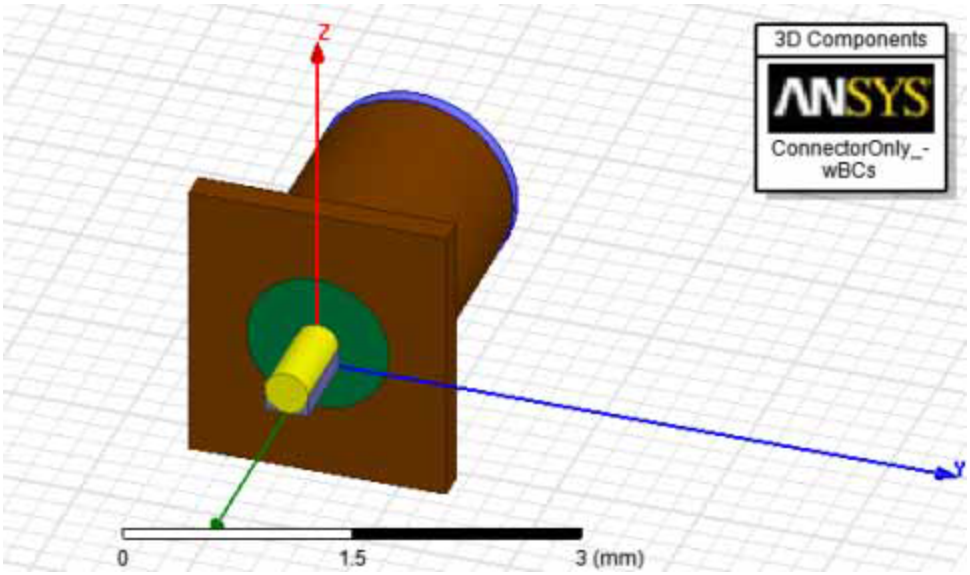


Create 3D Component Info Tab

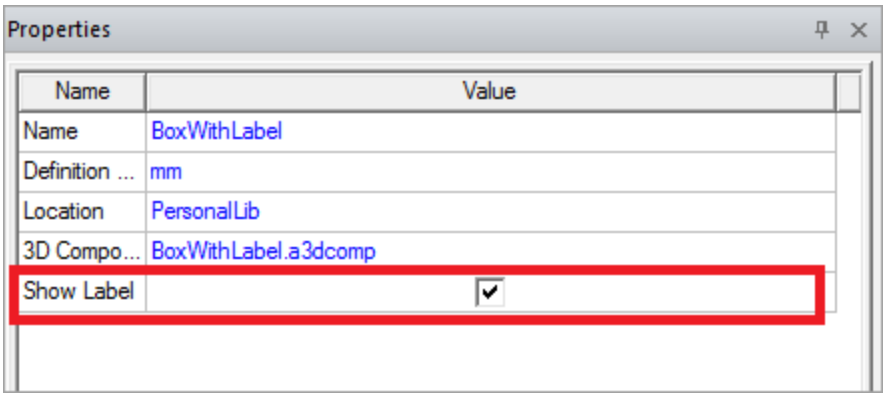
On the **Info** tab you can view and edit component information, as well as add notes. The *Name* in the first field is used when the component is inserted into a design. Other fields include *Owner, email, Company, Company URL, Model Number, Help URL version, Date,* and *Notes*.



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.

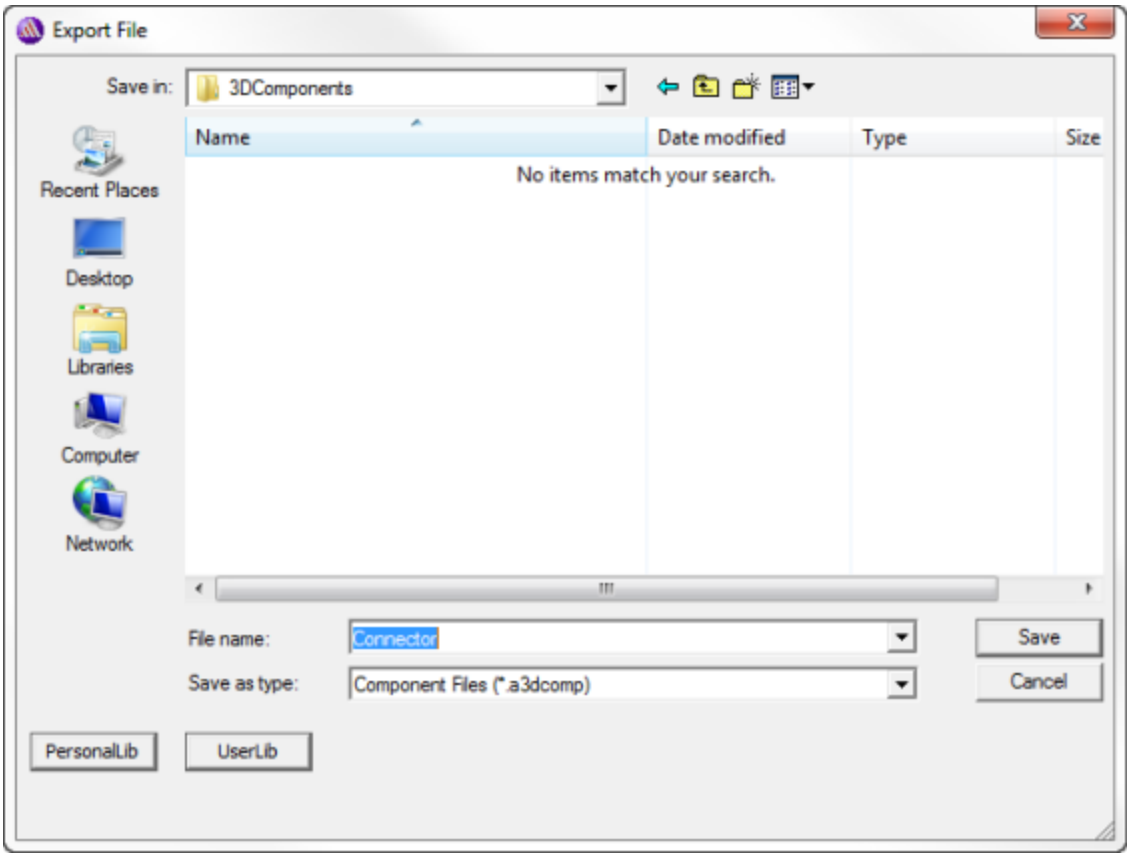


You can move the image in the modeling workspace by clicking the "3D Components" area of the box and dragging it. You can hide the image by editing the 3D component and clearing the **Display image...** check box, or by deselecting the Show Label check box in the Properties window:



Save 3D Component File

When you **OK** the *Create 3D Component* dialog box, an *Export File* dialog box displays. By default the **Save in** field shows the model source folder. Click **PersonalLib** or **UserLib** to display a 3D Components folder.



The default **File name** is the component *Name* specified in the *Info* tab.

Importing 3D Component Files

If you have an existing 3D component (*.a3dcomp) file, you can import it.

The component will be inserted into the design, and will appear in the **Component Libraries** window, under **Most Recently Used**.

> **Note:**
>
> Components created in older versions of the software may not be compatible.

To import a 3D component:

1. Click **Draw** > **3D Component Library** > **Browse**.

   The **Browse 3D Component** window appears, with a preview area.

2. Navigate to the file you want to import, and click **Open**.

   The **Insert 3D Component** window appears.

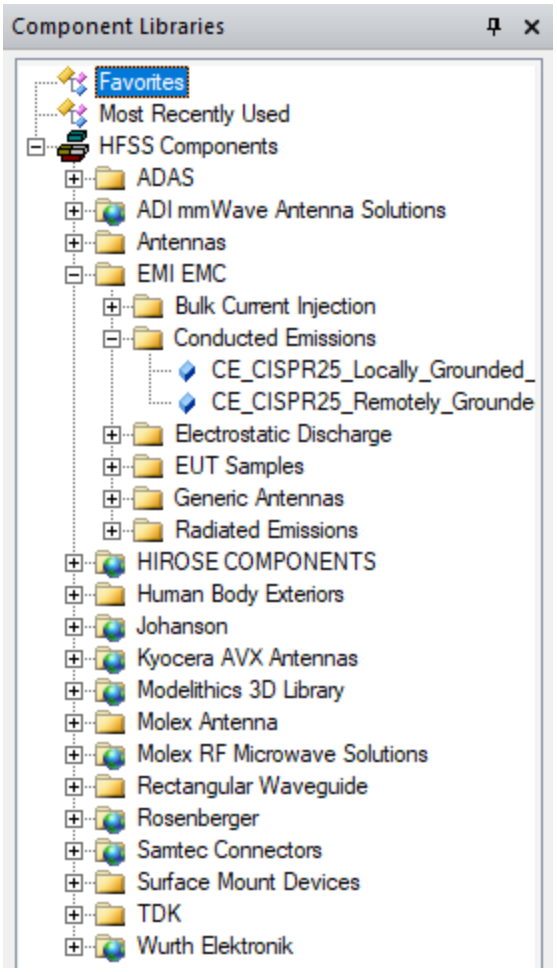
3. Review the component information, and use the drop-down menu to select a **Target Coordinate System**.
4. Click **OK**.

   The component is inserted into the design, and is listed in the **Component Libraries** window, **Project Manager**, and **History Tree**.

Inserting a 3D Component into a Design

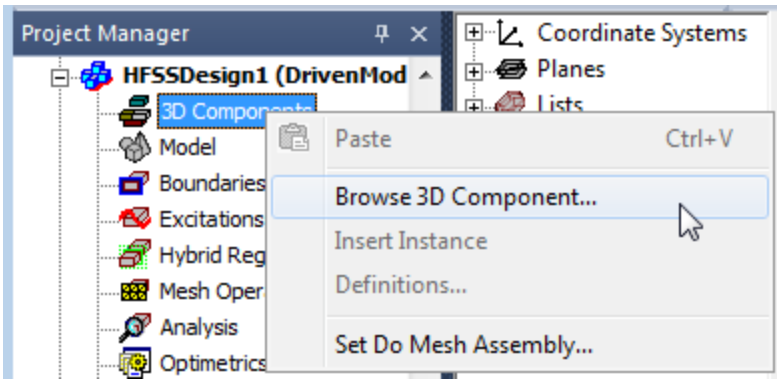
Once you save one or more components to a library, and create target coordinate systems in the design where you intend to place the component, you can right-click the 3D Component icon in the Project Manager or use the **Draw** > **3D Component Library** commands to browse your folders or libraries, or use the **View** > **Component Libraries** to display a Component Libraries window to navigate installed libraries. Alternatively, you can insert an instance of a component by right-clicking the 3D Component icon in the Project Manager and selecting **Insert Instance**.

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.

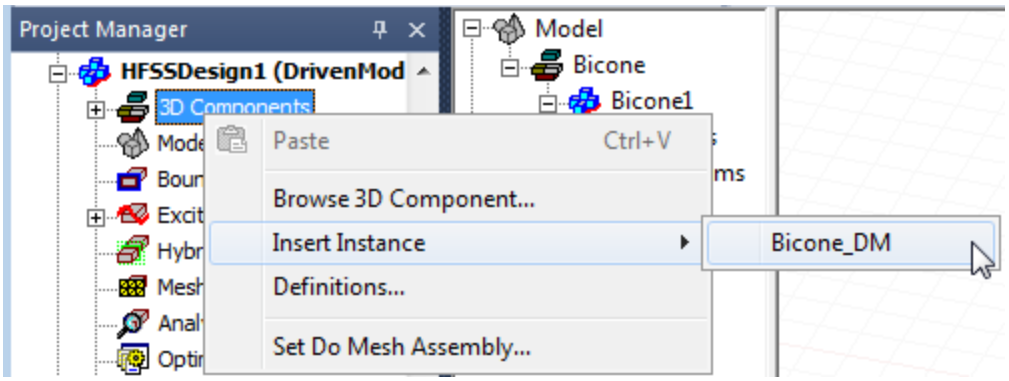


You can select any component from the library, and drag and drop to insert it in the design. Double-click any model in the component libraries to view the **Insert 3D Component Instance** dialog opened to the **Parameters** tab. You can also view the **Image** and **Info** tabs, and select the target coordinate system.

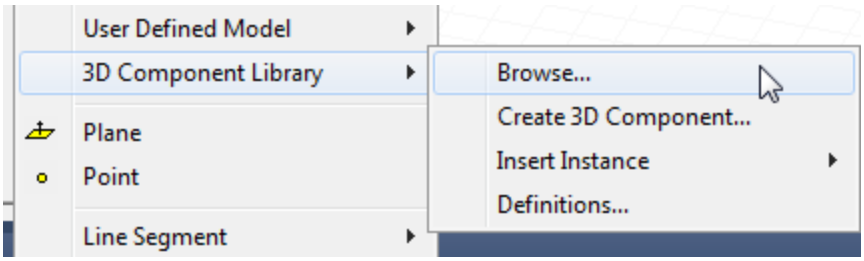
Selecting **Browse 3D Components** lets you navigate directories via a browser window.



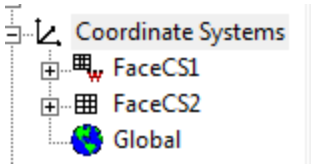
If you have previously inserted a 3D Component, the **Insert Instance** command is enabled on the menu. You can use this command to select from 3D Models in the design to easily insert another instance.



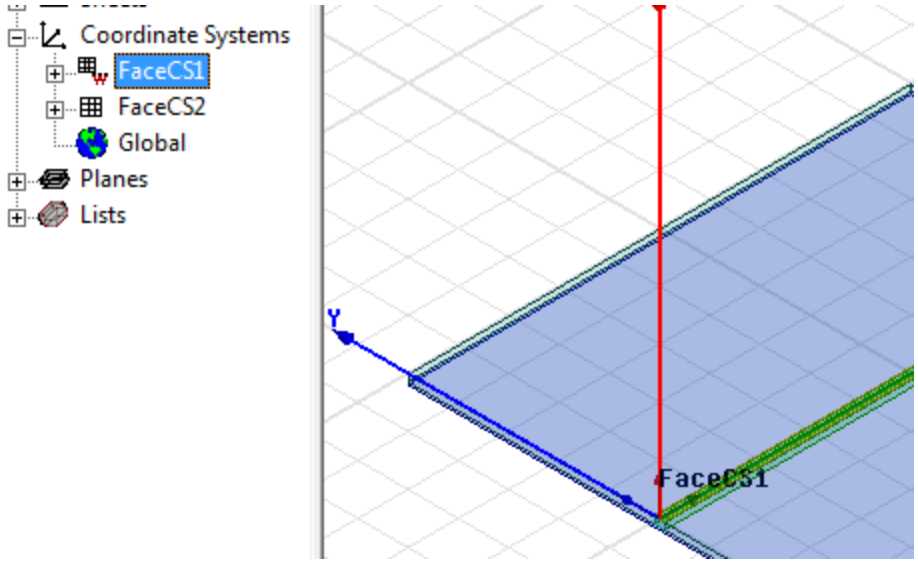
You can also click **Draw> 3D Component Library** to access the **Browse** and **Insert Instance** menus **Antennas, Rectangular Waveguides**, and **PersonalLib** to select a component to insert into a design.



A target coordinate system provides a location for a component. For example, a design includes two additional coordinate systems:

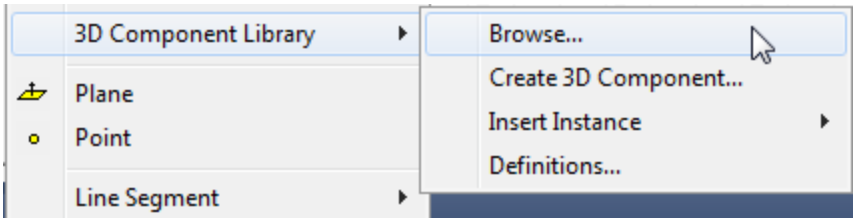


Selection of a coordinate system in the History tree displays a potential location for inserting a component.



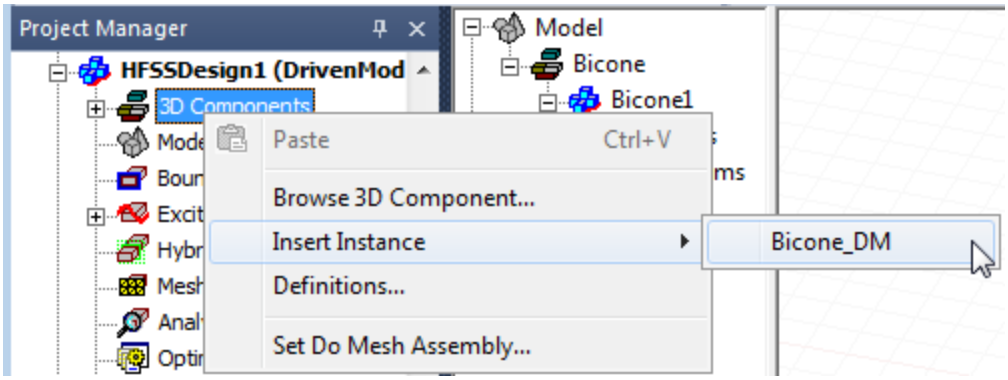
To Insert a 3D Component from the Menus

1. Right-click 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.

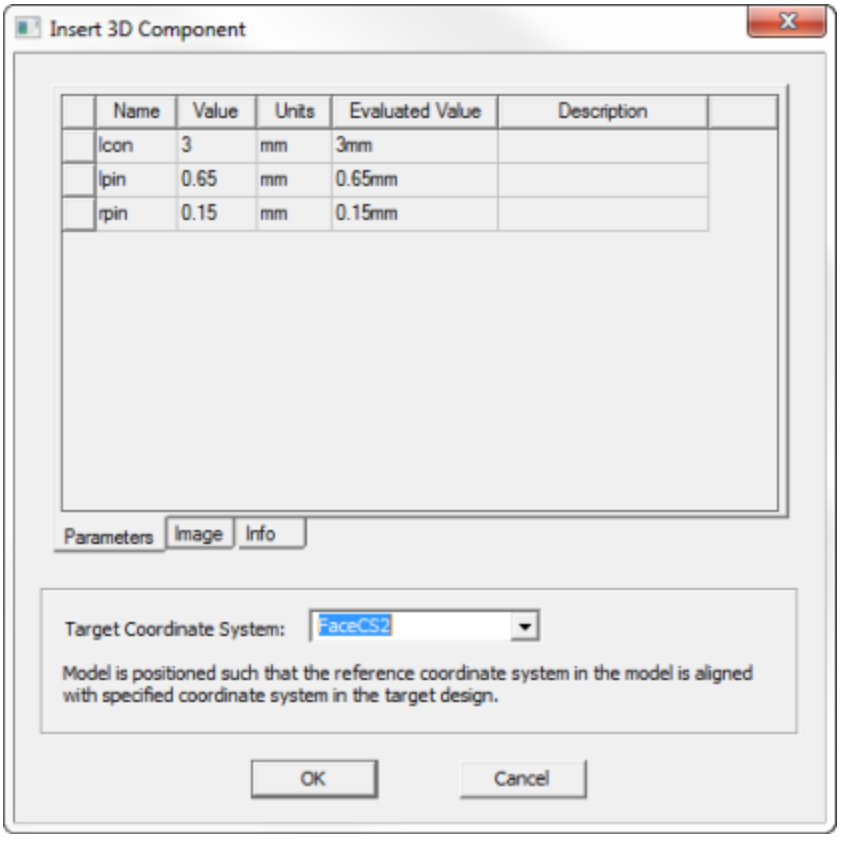


   Selecting a 3D component file causes a display of the component image and the File name.

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.

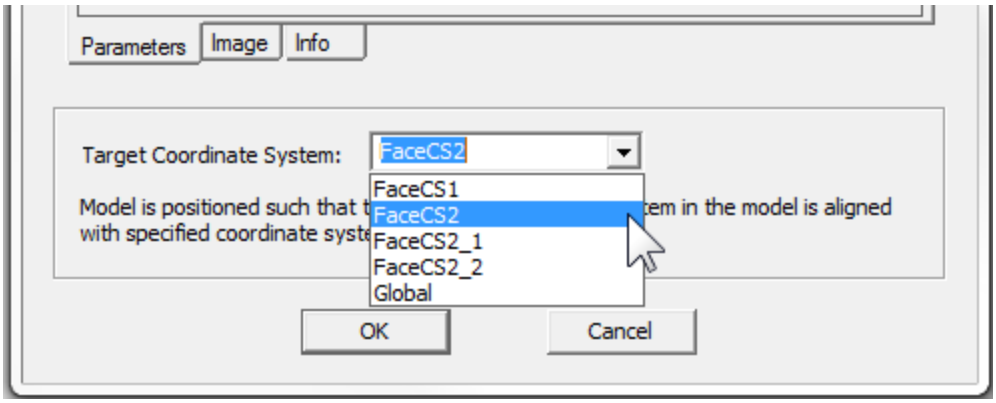


You see the **Insert 3D Component** dialog opened on the Parameters tab.

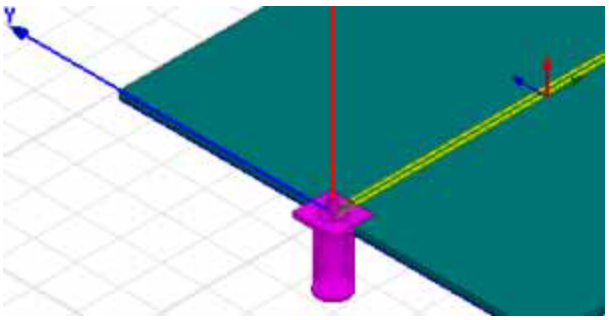


Tabs let you view the Parameters, Image, and Info. You can edit parameter values, and assign variables or expressions for parameters

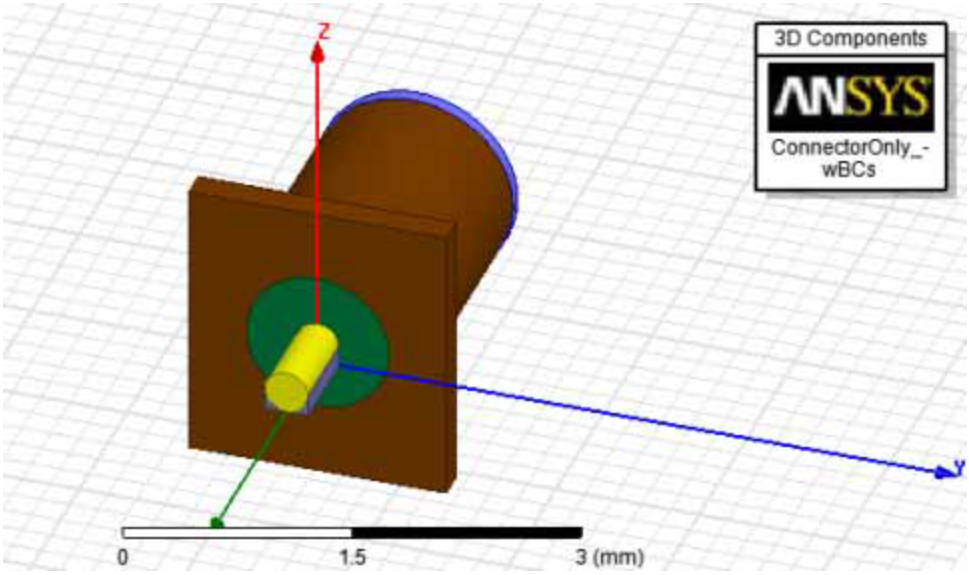
2. Use the menu to select the Target Coordinate System if any have been defined in addition to the Global coordinate system. The target coordinate system that you select is highlighted in the modeler window.



3. When you click **OK** the component is placed at the coordinate system you selected.

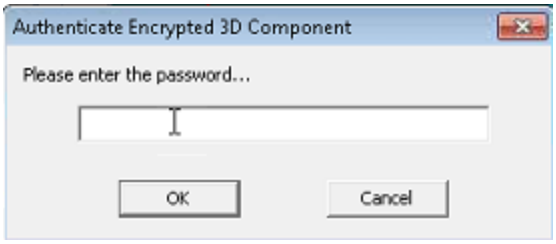


4. If the component has a logo defined, the image is always displayed in the upper right of the modeler window.



5. If you have used Browse, select a component and click **Open**. 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.



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.

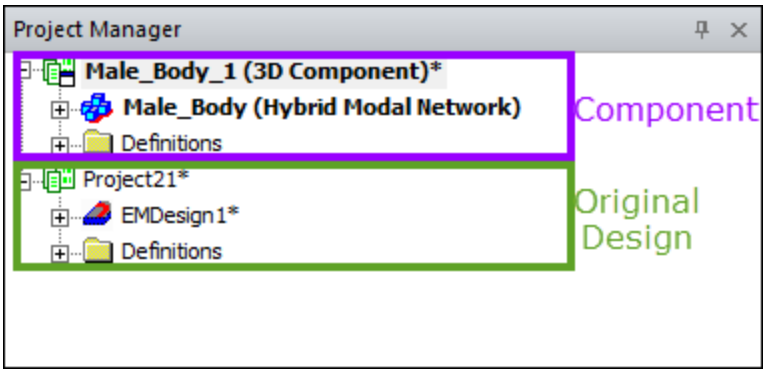
Editing a 3D Component Definition

> **Note:**
>
> If you reached this page looking for help on editing the definition of a **Layout Component**, as opposed to a *3D Component*, go here instead.

The **Edit Definition** feature allows the user to create new component definition using an existing, unencrypted 3D component file. This speeds up the task of creating new versions of components without having to start from the beginning. When **Edit Definition** is initiated, **Electronics Desktop** automatically creates a new project containing all 3D component data from the file in 'exploded' form. Component materials are added at the project level; variables are added at the project and design level; a detailed history is created for all objects in the component; and design data is added to the appropriate folders in the Project Manager.

> **Important:**
>
> Only 3D components that are **not encrypted** can be edited as described in this section.

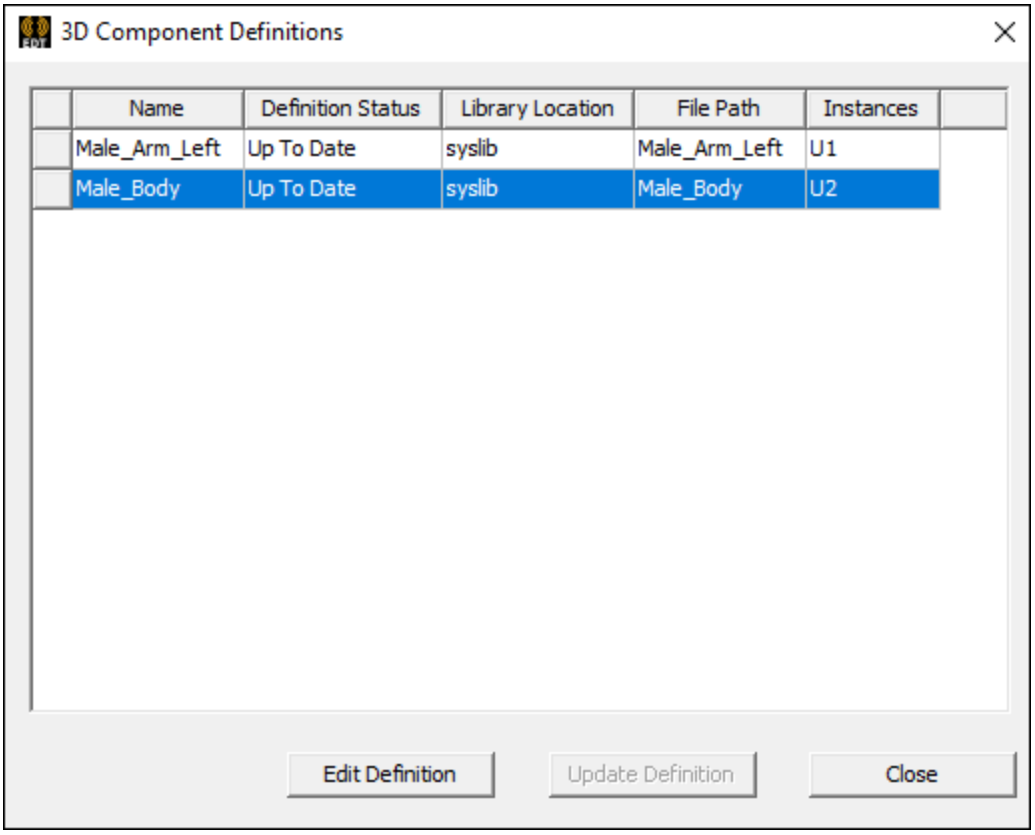
The **Edit Definition** command creates a new project, named for the component, in the **Project Manager** window.



When the edited component is saved, an **Update 3D Component** window appears. From the **Update 3D Component** window, choose whether to save the edited component as a local instance or to the original library. To edit a 3D component definition:

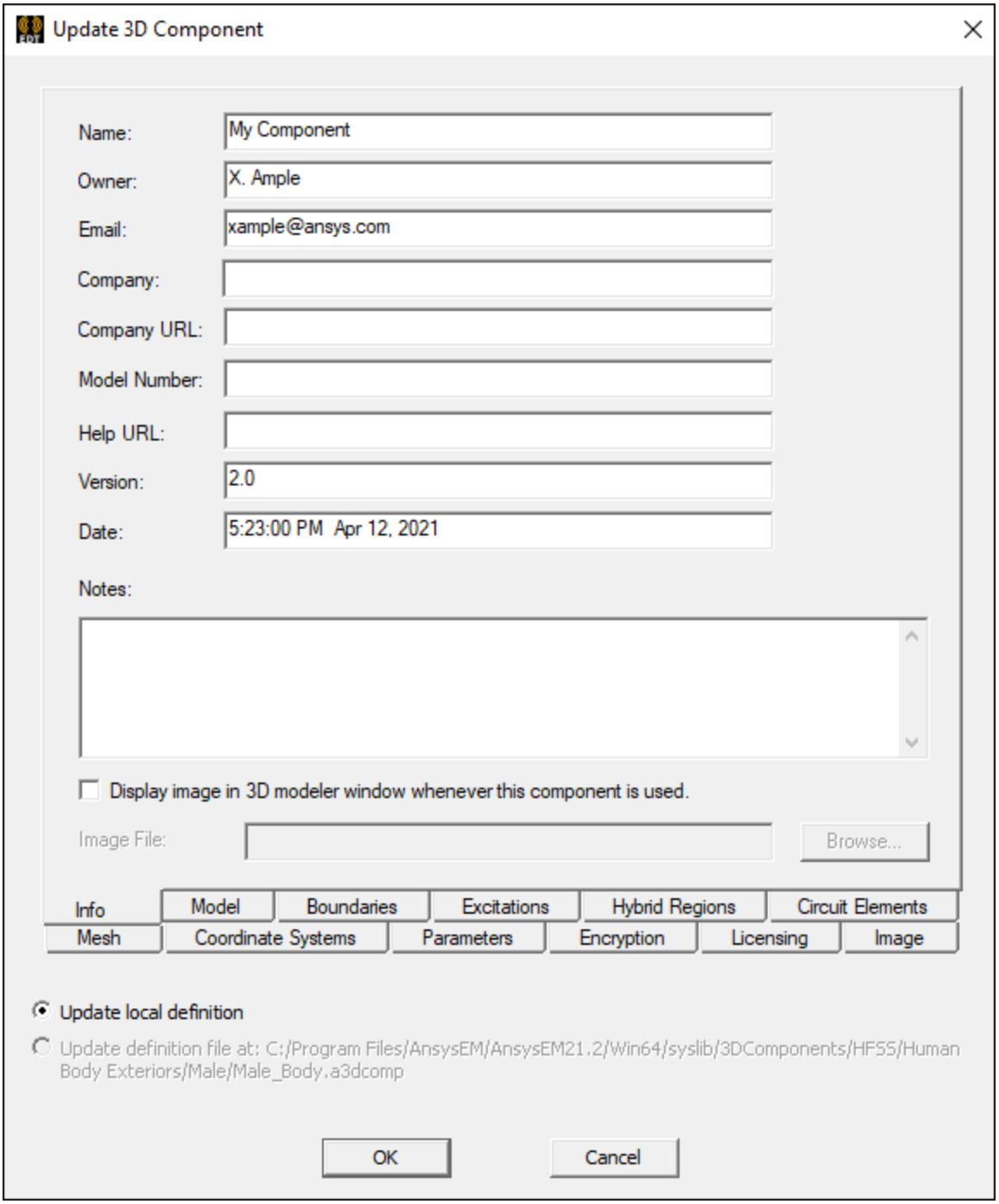
1. Navigate to the **Edit Definition** window by following one of the following paths:
    - From the **History Tree**, right-click the component and select **Edit Definition**.
    - From the **History Tree**, right-click an instance under the component and select **Components** > **Edit Definition**.
    - From the **Project Manager**, right-click **3D Components** and select **Definitions** to open the **3D Component Definitions** window. From the **3D Component**

Definitions window, select a component and click **Edit Definition**.



Regardless of which path is taken, a copy of the 3D component opens in the **Layout Editor**, and a new project folder appears in the **Project Manager** window, containing a copy of the original design.

2. Edit the component as desired.
3. To save the component as a new component (*.*a3dcomp*) or as its own **Electronics Desktop** (*.*aedt*) file, click **File** > **Save As**. Specify a file name and destination and click **Save**.
4. To save the component in the current project or for all projects, click **File** > **Save** to open the **Update 3D Component** window.



This window is populated with all entities (objects, boundaries, excitations, etc.) in the component design.

5. Use the radio buttons to select one of the following options:

- **Update Local Definition** – updates the component definition for use in the original design only.
- **Update Definition File At <Path>** – updates the component in the component library, affecting all other instances.

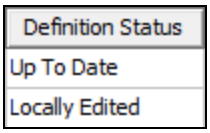
6. Click **OK**.

Verifying Updates to Components

To verify that a component has been edited:

1. Access the **3D Component Definitions** window:
   - From the **Project Manager**, right-click **3D Components** and select **Definitions**.

   The **Definition Status** column shows "Locally Edited" for any altered components.



2. Click **Close** to exit the window.

Updating Definitions

Once the updated component is saved, return to the original design and update all instances of that component. See: Updating 3D Component Definitions.

> **Note:**
>
> Definition update is not automatic even if **Edit Component** was initiated from that design. Instead, when an assembly design becomes active, a dialog box appears, prompting the user to update the 3D component definition. Then choose to update the definition from the prompt.

To update a component's definition:

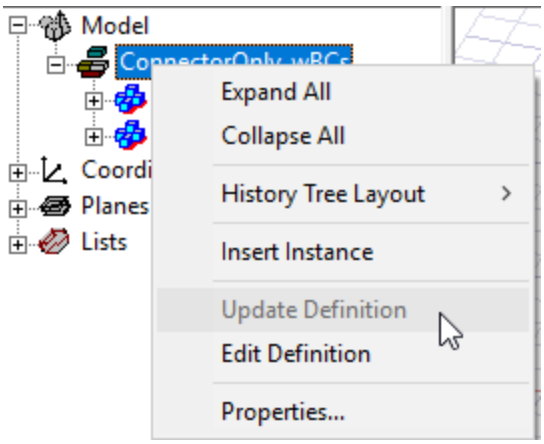
1. Access the **3D Component Definitions** window:
   - From the **Project Manager**, right-click **3D Components** and select **Definitions**.

   Components with updates available display "Update Available" in the **Definition Status** column.

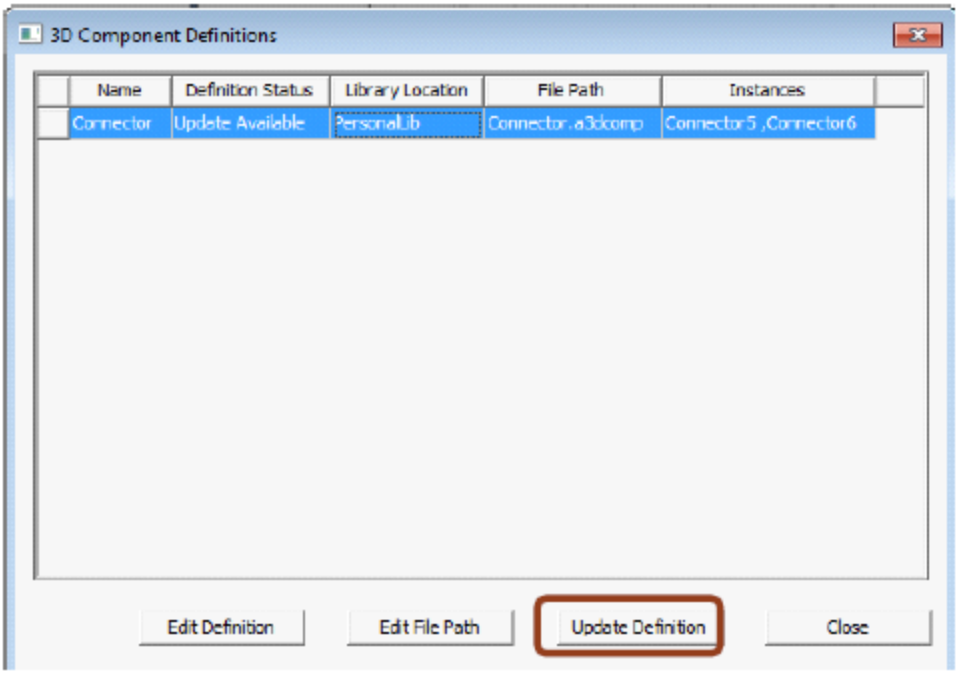
2. Select the desired component and click **Update Definition**.
3. Click **Close** to exit the window.

Updating a 3D Component Definition

If you create or obtain a new version of a component, and then open a project that contained an earlier version of that component, you can right-click 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 unavailable 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.
- The Definitions dialog box provides complete list of definitions in the design.
- Definitions can be updated from the dialog box.

Modifying Objects

You can quickly modify the position, dimensions, and other characteristics of objects created in the 3D Modeler window.

What do you want to do?

Object and History Editing	**Modify Object Appearance**
Modify Object Dimensions	**Modify Object Location or Orientation**
Surface or Edge Operations	**Boolean Operations on Objects**

Object and History Editing

- Copy and paste objects.
- Delete objects
- Delete Last Operation
- Cutting Objects
- Duplicate objects
- View and Edit Commands on History Tree Objects
- Purge History
- Generate History

Modify Object Appearance

- Assign color to an object.
- Assign transparency to an object.
- Also see Modifying the Model View

Modify Object Dimensions

- Assigning a Cross Section and Dimension to a Polyline
- Scale the size of objects
- Connect objects
- Move faces or edges
- Convert polyline segments
- Rounding the edge of an object (Fillet)
- Flattening the edge of an object (Chamfer)

Modify Object Location or Orientation

- Move objects
- Rotate objects
- Change the Orientation of an object
- Mirror objects about a plane.
- Offset an object (move every face of an object).
- Also see Modifying the Model View

Surface or Edge Operations for Objects

- Sweep objects.
- Cover lines.
- Cover faces.
- Uncover faces.
- Detach faces.
- Detach edges.
- Create a new object by taking a cross-section of a 3D object.
- Wrap Command
- Project Sheet
- Thicken Sheet

Boolean Operations on Objects

- Unite objects.
- Subtract objects.
- Create objects from intersections.
- Create an object from a face.
- Create an object from an edge.

- Split objects.
- Separate objects.
- Imprint Projection Commands
- Imprinting an Object

Assigning Color to Objects

1. Select the object to which you want to assign a color.

   > **Note:**
   >
   > If the *Properties* window is not visible on the desktop, click **View> Properties Window** or use **Edit> Properties**.

2. In the *Properties* dialog box or docked *Properties* window, click the **Attribute** tab.
3. Click the color sample in the **Value** column of the **Color** row.

   The *Color* palette appears.

4. Select a *Basic* or *Custom* color from the **Color** palette.

   Alternatively, you can specify a color by typing numbers between 0 and 255 to specify the following values:

   - **Hue, Sat** (saturation), and **Lum** (luminance), or
   - **Red, Green,** and **Blue**

   You can also specify a color by clicking within the color spectrum box and using the luminance slider to the right of this box.

5. Optionally, click **Add to Custom Colors** to save the settings as a custom color, which makes in easier to specify the same color for other objects.

   > **Note:**
   >
   > If you wish to overwrite a previously defined or default custom color, choose the custom color swatch to overwrite *before* specifying the new color parameters. To clear your customizations and restore default choices, click **Reset Custom Colors to Defaults**.

6. Click **OK**.

   The color is assigned to the selected object.

Setting the Default Color of Objects

The default color you specify for objects is applicable to when the objects are rendered as solids (shaded) or rendered as wireframes (that is, outlines only). The outline color is the same as the object color you specify according to the following procedure:

1. Using the menu bar, click **Tools**> **Options**> **General Options**, or, from the **Desktop** ribbon tab, click  **General Options**.
2. In the tree at the left side of the *Options* dialog box, select **3D Modeler**> **Display**> **Rendering**.
3. In the *Object Appearance* section, click the color sample button to the right of **Default color**.

   The **Color** palette appears.

4. Select a color from the **Color** palette, as described in the parent topic.

   Any objects you draw after this point will be assigned the default color you selected. However, if the *Material Appearance* option is selected within the *Attributes* tab of the object properties, material color and transparency assignments will override the default assignments.

   Alternatively, select **Use material appearance if available** within the *Options* dialog box to use the material color and transparency values by default (when they exist).

5. Click **OK**.

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 or docked *Properties* window, click the **Attribute** tab.
3. Click the *Value* in the **Transparent** row.

   The *Set Transparency* dialog box 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. Or, type a decimal number between 0 and 1 (inclusive) in the box to the right of the slider.
5. Click **OK**.

Setting the Default Transparency of Objects

1. Using the menu bar, click **Tools**> **Options**> **General Options**, or, from the **Desktop** ribbon tab, click **General Options**.
2. In the tree at the left side of the *Options* dialog box, select **3D Modeler**> **Display**> **Rendering**.
3. In the *Object Appearance* section, click and drag the **Default Transparency** slider. Move the slider to the right to increase the default object transparency. Move the slider to the left to decrease the default object transparency. Or, type a decimal number between 0 and 1 (inclusive) in the box to the right of the slider.

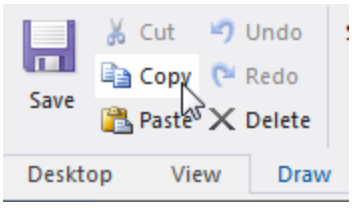
   Any objects you draw after this point will be assigned the default transparency you specified. However, if the *Material Appearance* option is selected within the *Attributes* tab of the object properties, material color and transparency assignments will override the default assignments.

   Alternatively, select **Use material appearance if available** within the *Options* dialog box to use the material color and transparency values by default (when they exist).

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** or in the **View/Draw/Model** tabs of the ribbon, click the **Copy** icon:

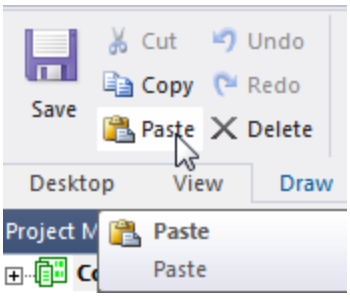


   The objects are copied to the Clipboard, a temporary storage area. The selected items are not deleted.

   To cut an item to the clipboard and deleting the original, use the scissors icon on the toolbar.

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 pasted relative to the current working coordinate system.

6. Click **Edit**>**Paste** or in the **View/Draw/Model** tabs of the ribbon, click the **Paste** icon:



   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. Designs and projects containing encrypted components can be copy pasted for products using 3D Modeler

Import a Model from the Clipboard

You can import a model from the Clipboard in order to use the geometry from a different design. To use 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 is 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 from the source design. 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 in the same project as the design from which you copied the items or in a different project.
4. Click in the **3D Modeler** window.
5. Click **Modeler**> **Import From Clipboard**.

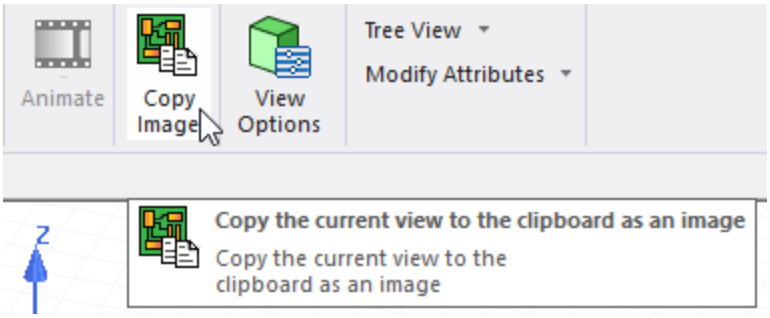
   The geometry is pasted from the Clipboard with the IDs preserved. Additionally, the source design's coordinate systems, materials, appearances, and other attributes are preserved and duplicated in the target design.

Copy Image

You can import images of the **3D Modeler** window or of Reports into any other application. The image has to be copied to the clipboard, so that it can be imported into the other application.

To copy an image of the Modeler window and paste it into another application:

1. Make the Modeler window from which you wish to capture an image active.
2. Copy an image to the clipboard using one of the following three methods:
   - From the menu bar, click **Edit> Copy Image**.
   - Right-click the Modeler window and select **Copy Image** from the shortcut menu.
   - On the **View** ribbon tab, click **Copy Image**:



   A raster image of the contents of the *Modeler* window is copied to the Clipboard.

3. Open the application into which you want to paste the image of your model and follow that application's image pasting procedure.

To copy an image of a Report to paste into another application:

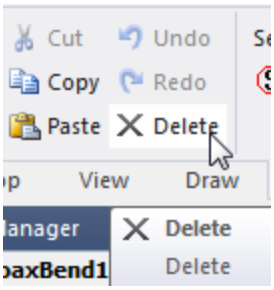
1. Make the report the active window and use the **Copy Image** command from the **Edit** menu, short-cut menu, or **View** ribbon tab.

   An image of the report is copied to the Clipboard.

2. Open the application into which you want to paste an image of the report and follow that application's image pasting procedure.

Deleting Objects

1. Select the objects to delete.
2. Delete the selected objects using *one* of the following three methods:
   - From the menu bar, click **Edit**> **Delete**.
   - From the **View, Draw,** or **Model** ribbon tab, click the **Delete** icon:



   - Press the **Delete** key.

   The objects are deleted.

> **Note:**
>
> Boundaries, excitations, hybrid regions, and mesh operations that were associated only with deleted objects are also deleted along with the objects. However, associated design variables remain, and they can be assigned to other geometry.

Deleting Start Points and Endpoints

For individual segments of polylines, you can delete the start point and/or the end point, with the following conditions:

- For the *first* segment, you can only delete the *Start Point*, which effectively removes the entire segment.
- For the *last* segment, you can only delete the *End Point*, which effectively removes the entire segment.
- For all remaining (internal) segments, you can delete either the *Start Point* or the End Point. The new point becomes the opposite point of the adjacent segment, and the total number of segments is reduced by one.

To delete *Start Points* or *End Points* of polyline segments:

1. In the History Tree, locate the polyline that contains the segment you want to delete and expand the branch to expose the *CreateLine, CreateArc, CreateAngularArc,* or

CreateSpline entries. These entries represent the individual polyline segments.

2. In the History Tree, right-click the polyline segment you want to edit and choose either:
   - **Delete Start Point**, which removes the preceding segment, or
   - **Delete End Point**, which removes the subsequent segment.

   Alternatively, you can select the segment and choose **Edit> Delete Start Point** or **Edit> Delete End Point** from the menu bar.

Delete Last Operation

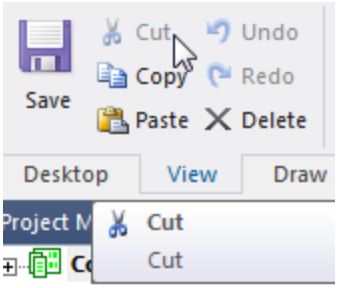
You can delete the last geometry operation performed on an object. For certain situations, this functionality can offer an advantage when compared with the *Undo* command. For example, if you performed several valid operations (such as assigning boundaries, excitations, or mesh operations) after operating on an object's geometry, you could revert the last geometry change without undoing the subsequent assignments.

1. Select the object.
2. From the menu bar, click **Modeler> Delete Last Operation**.

   This command undoes the last geometry operation performed on the selected object, including removing that operation from the History Tree and updating the context for the *Undo* and *Redo* commands.

Cutting Objects

1. Select the objects to cut.
2. From the menu bar, click **Edit> Cut** or, on any ribbon tab, click the **Cut** icon:



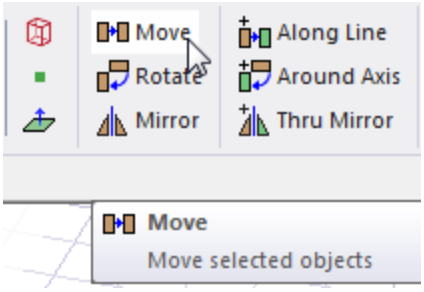
   The objects are copied to the Clipboard and removed from the design. They can subsequently be pasted elsewhere, such as into a different design.

Moving Objects

The process for moving objects differs depending on the Drawing Data Entry Mode for the Drawing Options, whether Point mode (F3) or Dialog mode (F4).

For Point mode (click F3 to change to Point Mode):

1. Select the objects to move.
2. Click **Edit**> **Arrange**> **Move** or, on the **Draw** ribbon tab, click the **Move** icon:



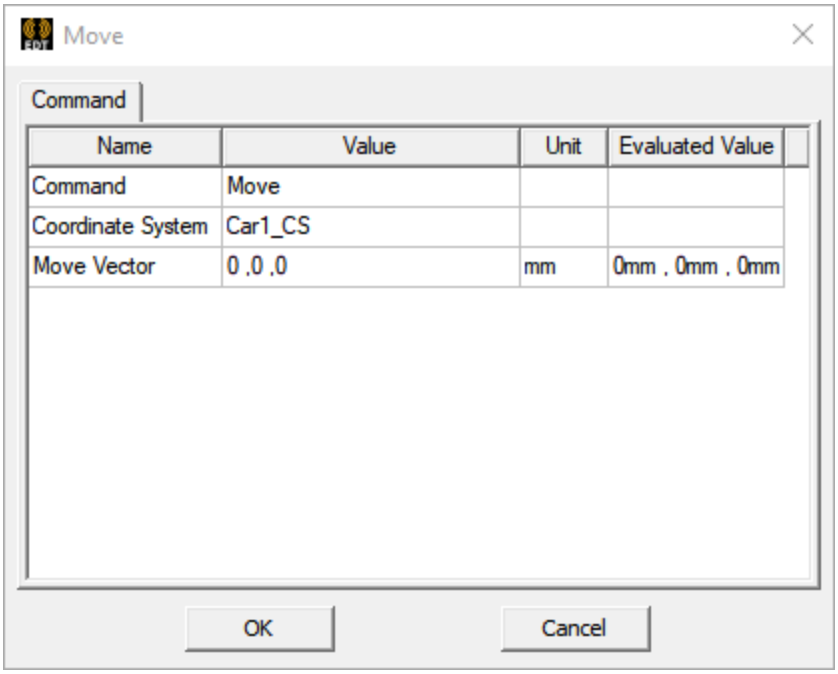
3. Select an arbitrary anchor point in one of the following ways:
   - Click the point. (The *Measure Data* window shows the coordinates of the points you click.
   - Enter the point's coordinates in the **X, Y,** and **Z** boxes.
4. Select a target point in one of the following ways:
   - Click the 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.

For Dialog mode (click F4 to change to Dialog mode):

1. Select the objects to move.
2. From the menu bar, click **Edit**> **Arrange**> **Move** or, on the **Drawing** ribbon tab, click the **Move** icon.

   A *Move* command dialog box appears.



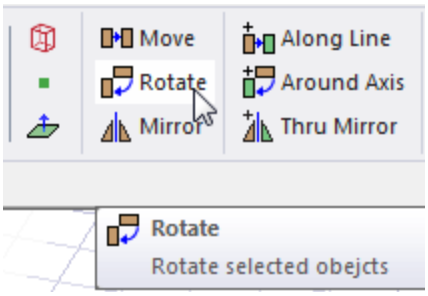
3. Enter the coordinates of the location to which the object should move and click **OK**.

Rotating Objects

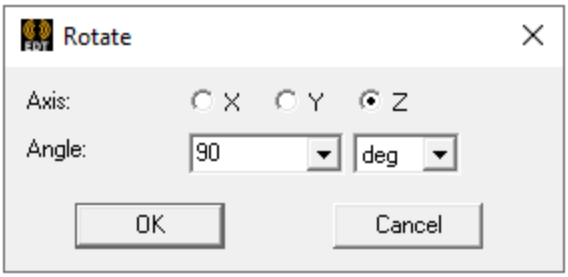
Rotate objects about the x-, y-, or z-axis using the **Edit> Arrange> Rotate** command.

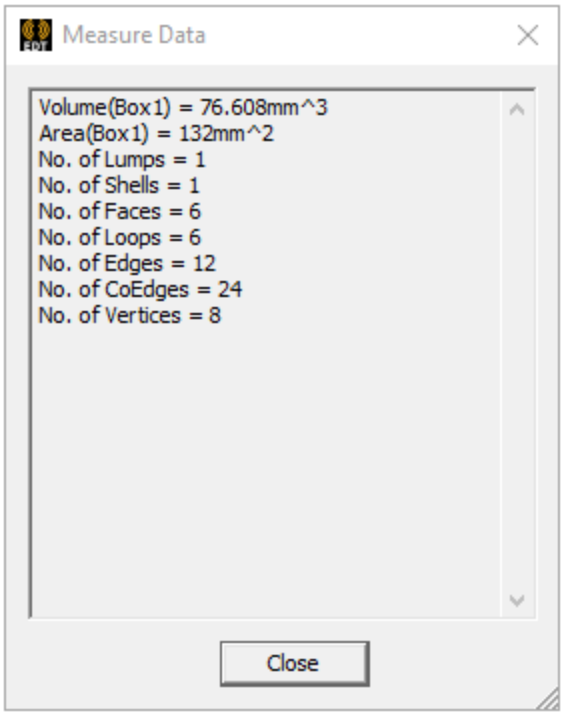
To rotate objects about an axis:

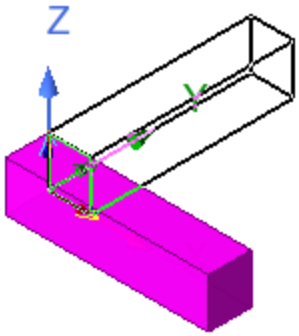
1. Select the objects to rotate.
2. From the menu bar, click **Edit**> **Arrange**> **Rotate** or, on the **Draw** ribbon tab, click the Rotate icon:



The **Rotate** dialog box appears along with the **Measure Data** window. A visualization of the proposed change appears in the Modeler window.

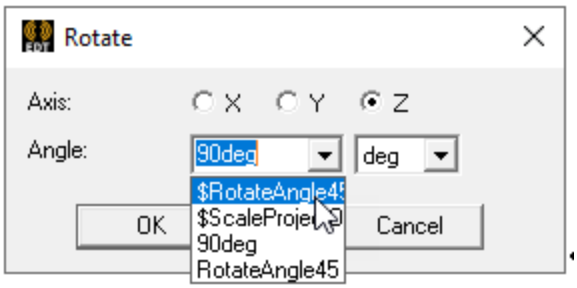






3. Select the axis about which to rotate the objects: **X, Y,** or **Z**.
4. Type the angle to rotate the objects in the **Angle** box. If you have previously specified values, or have created Project or Design Variables, this are listed in the drop-down menus.

   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* and *copy* objects, use the **Edit> Duplicate> Around Axis** command.

Changing the Orientation of an Object

Each object has an **Orientation** property that specifies whether the coordinate system it uses is Global or a user-defined orientation relative to the Global coordinate system.

To change an object's orientation:

1. Define the coordinate systems you want to have available.
2. Open the **Properties** dialog box for the object, or select the object and use the docked **Properties** window.
3. Click on the **Orientation** property and select the coordinate system from the drop-down menu. If no orientations other than **Global** have been defined, none appear on the list.
4. Click **OK** to close the dialog box and apply the changes.

> **Note:**
>
> The object will translate and rotate to make the geometry conform to the new coordinate system. If your intention was to change the material orientation, but the object had already been drawn in the correct physical orientation, you will have to redefine it's properties. The coordinates and dimensions used to define the object geometry must be based on the same coordinate system that is needed for proper material orientation. To avoid this problem, it is best to choose the correct coordinate system for material orientation before drawing the object.

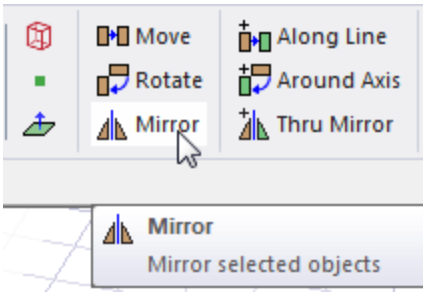
Mirroring Objects

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:

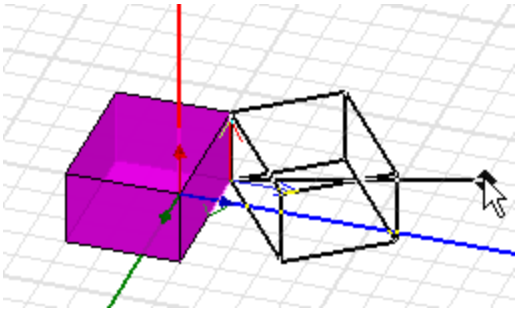
1. Select the object or objects that you want to mirror. You can select multiple objects.
2. From the menu bar, click **Edit** > **Arrange** > **Mirror** or, on the **Draw** ribbon tab, click the **Mirror** icon:



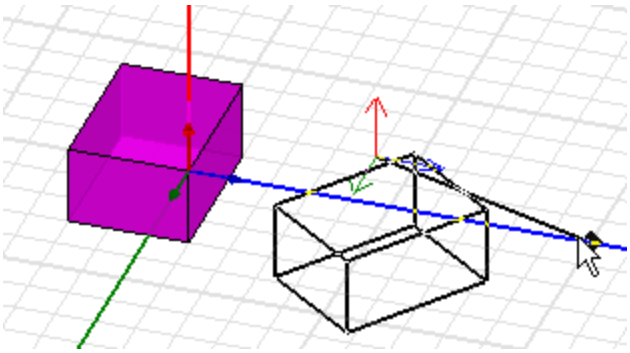
3. Select a point on the plane around which you want to mirror the object.

   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. 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.



4. Select a second 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 **Enter** to complete the command.

   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.

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. From the menu bar, 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 menu.
4. Click **OK**.

   The selected object's faces are moved the distance you specified.

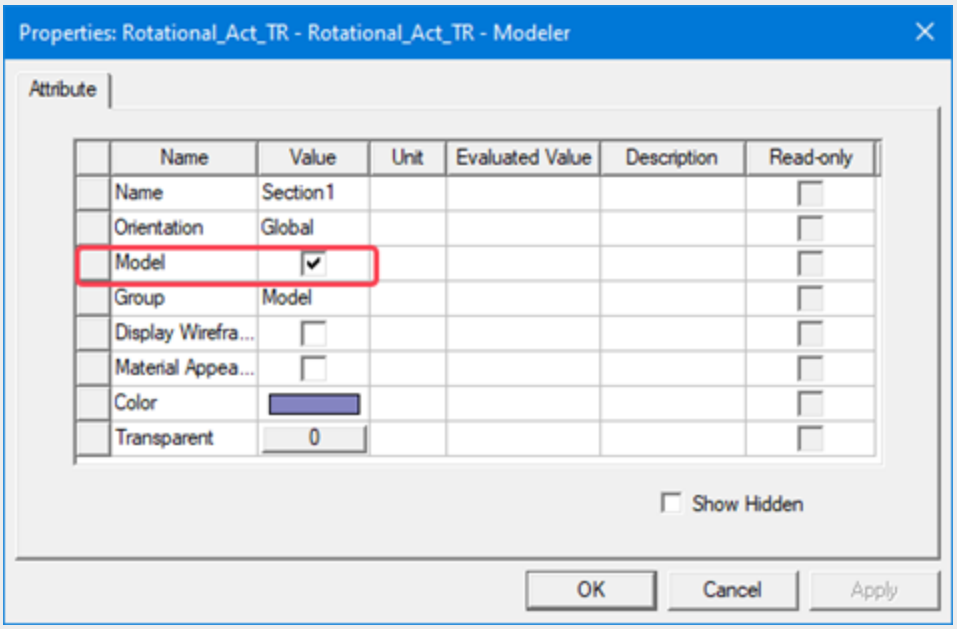
Duplicating Objects

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 will show the duplication command, illustrating which commands affect all duplicates (those performed before the duplication) and which commands would 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.

Note: When you duplicate an object, the Model\Non-Model status of the new object is not copied from the parent, but is determined by the **Modeler** > **New Object Type** menu selection. You can change the Model\Non-Model status of an existing object from its Properties window:



Choose from the following commands:

Edit > Duplicate > Along Line	Duplicates the parent object along a straight line. The child object can be designated as attached to the parent object, but if so, no ports or boundary conditions are duplicated.
Edit > Duplicate > Around Axis	Duplicates the parent object around an axis. The child object can be designated as attached to the parent object, but if so, no ports or boundary conditions are duplicated.
Edit > Duplicate > Mirror	Duplicates a mirror image of the parent object about a plane.

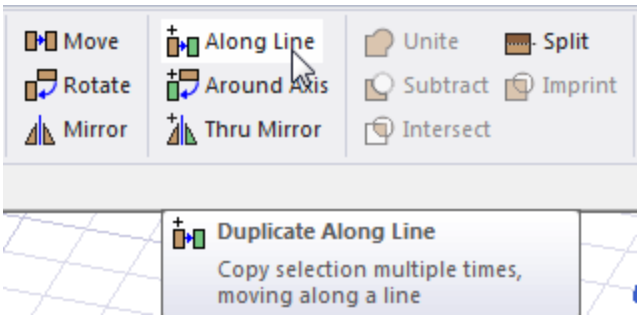
To copy objects to another design, use the **Edit** > **Copy** and **Edit** > **Paste** commands.

> **Note:**
>
> There is currently no method for dissolving the parent/duplicate relationship once a duplicate has been created.

Duplicating Objects Along a Line

To duplicate an object along a straight line, use 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. From the menu bar, click **Edit** > **Duplicate** > **Along Line** or, on the **Draw** ribbon tab, click the **Duplicate Along Line** icon:



3. Specify the vector along which the object will be duplicated. The *Measure Data* window appears to help you choose points:

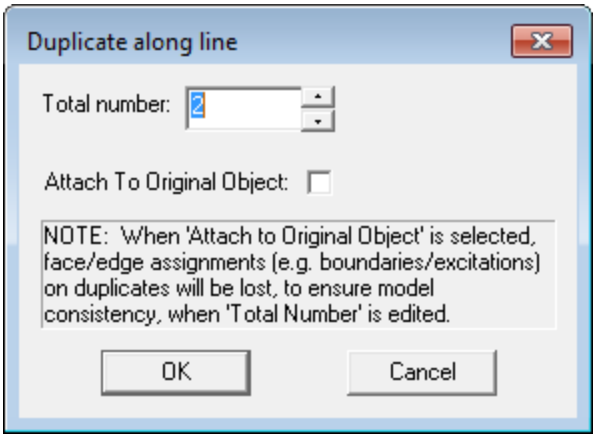
a. Select an arbitrary anchor point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the in the **X, Y,** and **Z** boxes.

   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, where **d** is the distance from the previously selected point.

   This point defines the direction and distance from the anchor point to duplicate the object.

   The *Duplicate Along Line* dialog box appears:



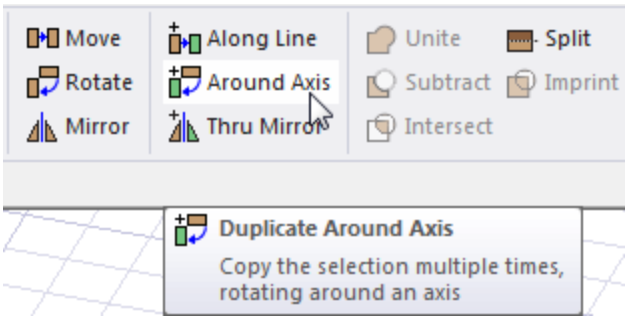
4. Type the total number of objects, including the original, in the **Total Number** box.
5. Optionally, check the **Attach to Original Object** check box. If this is checked, original object and its duplicates will be combined into one part. No ports or boundary conditions are duplicated for duplicates. If this option is used, it is recommended to run validation to ensure that the original and duplicates are not intersecting or touching each other.
6. If this option is used, it is recommended to run validation to ensure that the original and duplicates are not intersecting or touching each other.
7. Click **OK**.

   The duplicates are placed along the vector you specified.

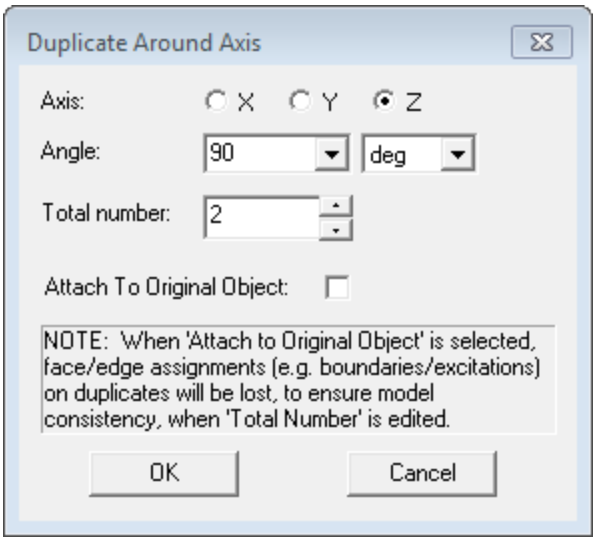
Duplicating Objects Around an Axis

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. From the menu bar, click **Edit** > **Duplicate** > **Around Axis** or, on the **Draw** ribbon tab, click the **Duplicate Around Axis** icon:



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. Optionally, check the **Attach to Original Object** check box. If this is checked, original object and its duplicates will be combined into one part. No ports or boundary conditions are duplicated for duplicates. If this option is used, it is recommended to run validation to ensure that the original and duplicates are not intersecting or touching each other.
7. Click **OK**.

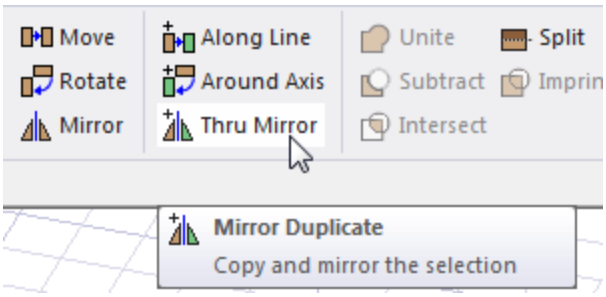
   The object is duplicated around the axis at the angle you specified.

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. From the menu bar, click **Edit** > **Duplicate** > **Mirror** or, on the **Draw** ribbon tab, click the **Thru Mirror** (Mirror Duplicate) icon:



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.

   A duplicate of the object appears on the plane you specified, oriented according to the normal point you specified.

Scaling Objects

Scale an object's dimensions in one or more directions using the **Edit** > **Scale** command.

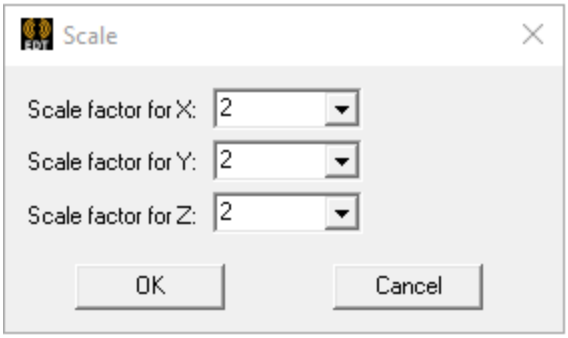
The scale of an object is determined by the distance of each of its vertices from the origin of the model coordinate system. When an object is scaled, the distance of each vertex from the origin is multiplied by the scaling factor, causing the object to be resized and/or moved.

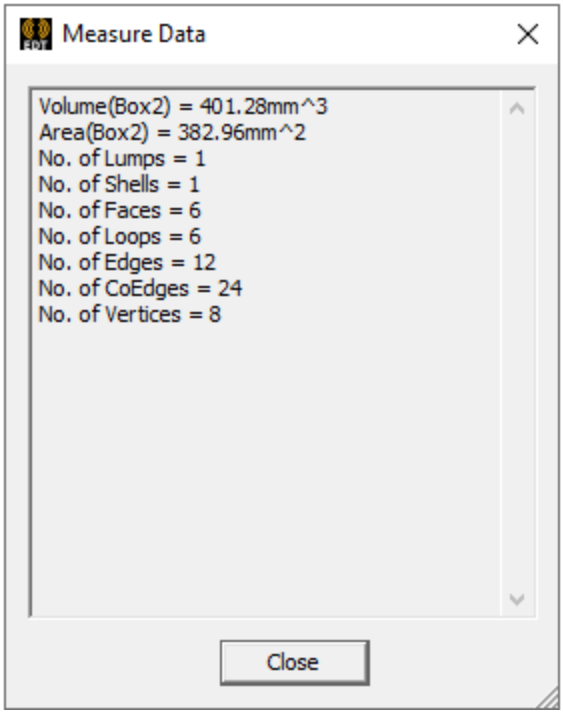
For example, if you specify a scaling factor of 2 in the X direction, each vertex in the model will be moved so that the distance to its origin is doubled. Note that a vertex located at the origin will not move. You can alter an object's proportions by scaling it in one direction.

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. From the menu bar, click **Edit** > **Scale**.

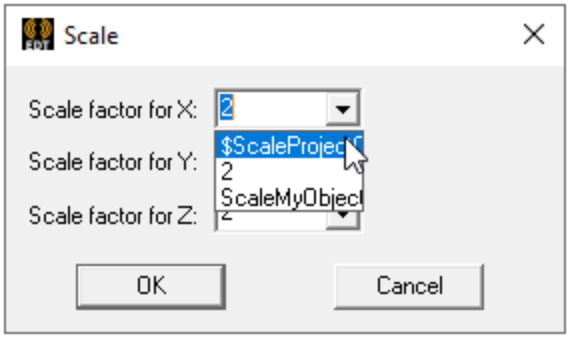
   The **Scale** dialog box appears with the **Measure Data** window.





4. Type the **Scale factor** for each axis. If you have created Project or Design Variables, or have previously used a relevant value, these may be listed for selection in the drop-down

menu.



5. Click **OK**.

   The object is scaled about the working coordinate system's origin.

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.
- Sweep open 1D objects, such as polylines, which results in open 2D sheet objects.
- Sweep one or more faces of a 3D object to create a new 3D object. (See Sweep Faces Along Normal).

Sweeping Around an Axis

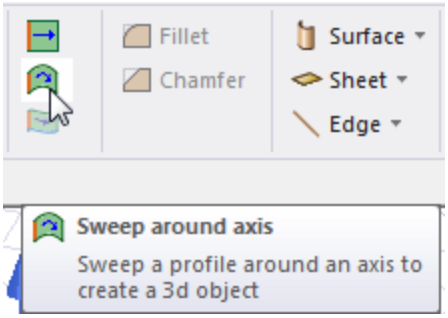
Sweep a 1D or 2D object around the X, Y, or Z axis using the **Draw** > **Sweep** > **Around Axis** command. Sweeping circles around an axis is a convenient way to create an open coil loop.

Before using this command, keep the following guidelines in mind:

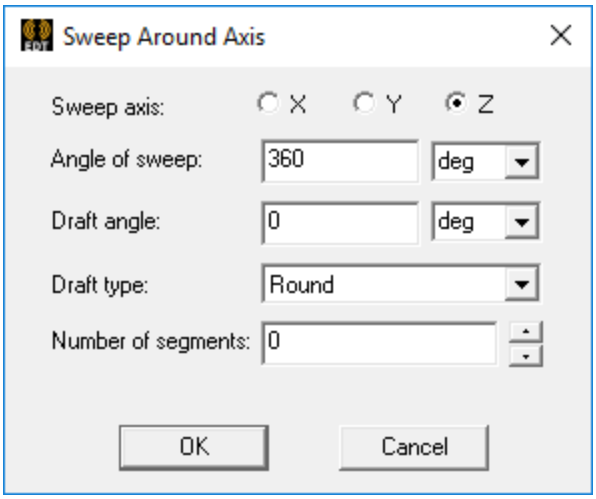
- The object and the axis you are sweeping around must lie in the same plane. For example, if you are sweeping an object around the Z-axis, the object must lie in a plane that includes the Z-axis, such as XZ or YZ.
- The normal of the object's plane faces must be perpendicular to the axis around which 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 the **Sweep axis** (that is, the axis you want to sweep the object around: **X, Y,** or **Z**).
4. Type an angle into the **Angle of sweep** text box and select the desired unit of measure.

   This is the angle about the specified sweep axis through which you want the object to be twisted or rotated as it is swept. 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 and click **OK**.

   > **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.

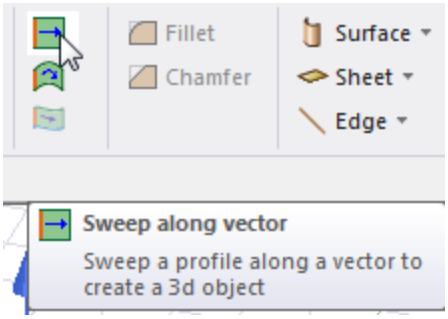
   The object is swept around the axis. The new object has the properties of the original object. The *Properties* dialog box appears, enabling you to modify the object's properties.

8. Click **OK**.

Sweeping Along a Vector

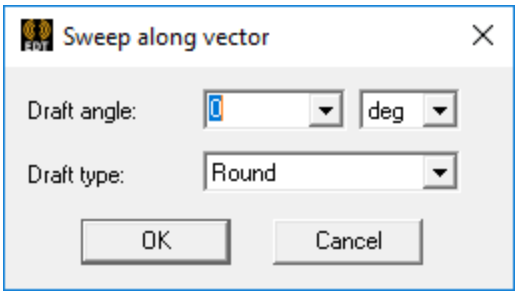
Sweep a 1D or 2D object along a vector using the **Draw** > **Sweep** > **Along Vector** command.

1. Select the object you want to sweep.
2. 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.

Best Practices

Adhere to the following best practices for good sweep outcomes using the Parasolid geometry kernel:

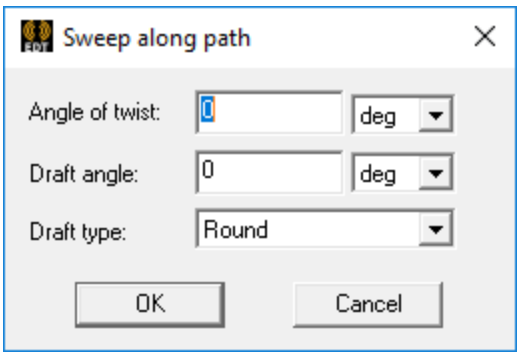
- When sweeping along a path that is not closed (i.e., the start and end point are not the same), ensure that the start point of the path is on the profile being swept.
- When sweeping a profile around a closed path, ensure that the path intersects the profile.
- Parasolid sweep operator preserves the angle between the profile and path. Ensure that the profile is perpendicular to the path if it is supposed to be perpendicular.
- If the sweep profile is rotationally symmetric (e.g., circle, rectangle, square), position the profile so that the path intersects the profile at the center of the profile. If the profile does not have a center, allow the path to go through or start at the vertex of a profile.

> **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 to be used as a path.
2. Select the object you want to sweep and then select the new polyline.
3. Click **Draw** > **Sweep** > **Along Path** or, from the **Draw** ribbon tab, click the **Sweep along path** icon:

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 you want the profile to 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: Draft Types.

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.

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. Right-click in the Modeler window and choose **Select Faces** from the shortcut menu.

   Alternatively, with the Modeler window active, press **F** to switch to the *Face* selection mode.

2. Select the faces of the object you want to sweep.
3. From the menu bar, click **Modeler** > **Surface** > **Sweep Faces Along Normal**.

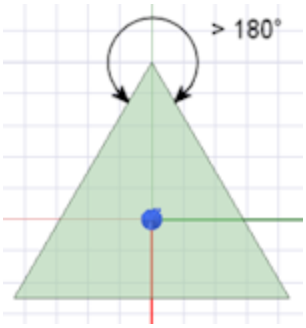
   The *Sweep Faces Along Normal* dialog box appears.

4. Type the distance you want to sweep the object face from its origin.
5. Click **OK**.

   The face is swept the distance you specified to create a new 3D object.

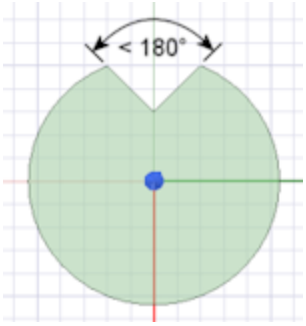
Draft Type Options

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

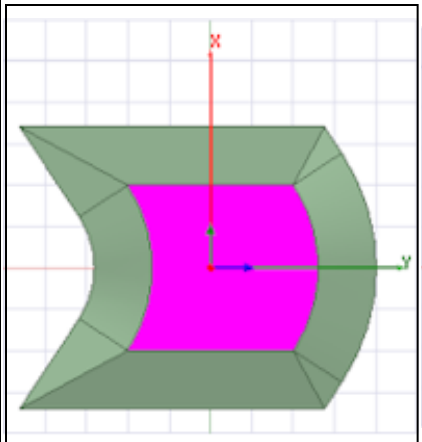
entire swept direction. However, the side faces may be split into three separate tangential faces, as needed and as described in the preceding bullet.

- **Round:** This option rounds all sharp corners of the original object, when possible, as it is swept along the specified direction.

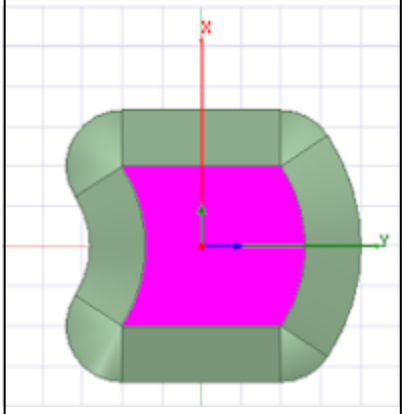
Examples

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.

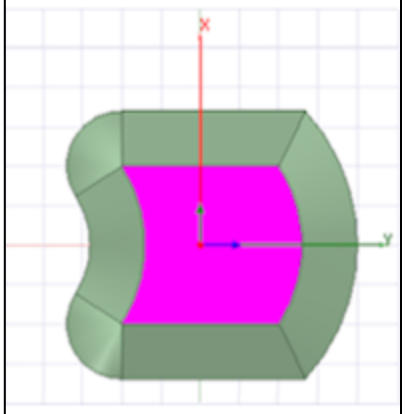
Sweep Along Vector (Positive Draft Angle):



1a: Extended

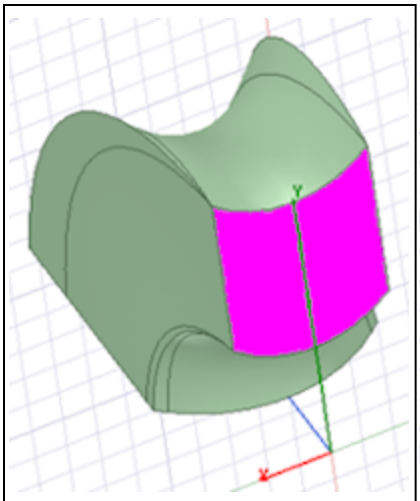


1b: Round

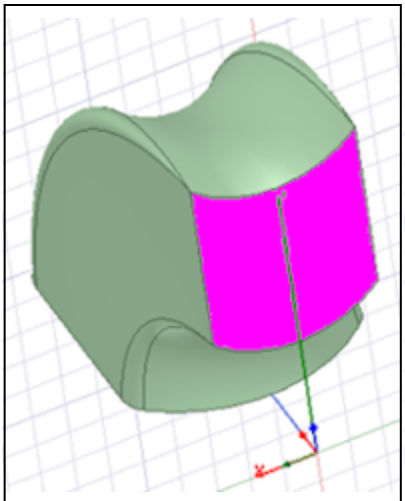


1c: Natural

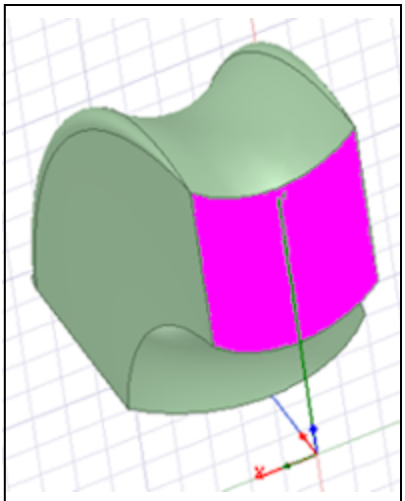
2. **Sweep About Axis** (Positive Draft Angle):



2a: Extended

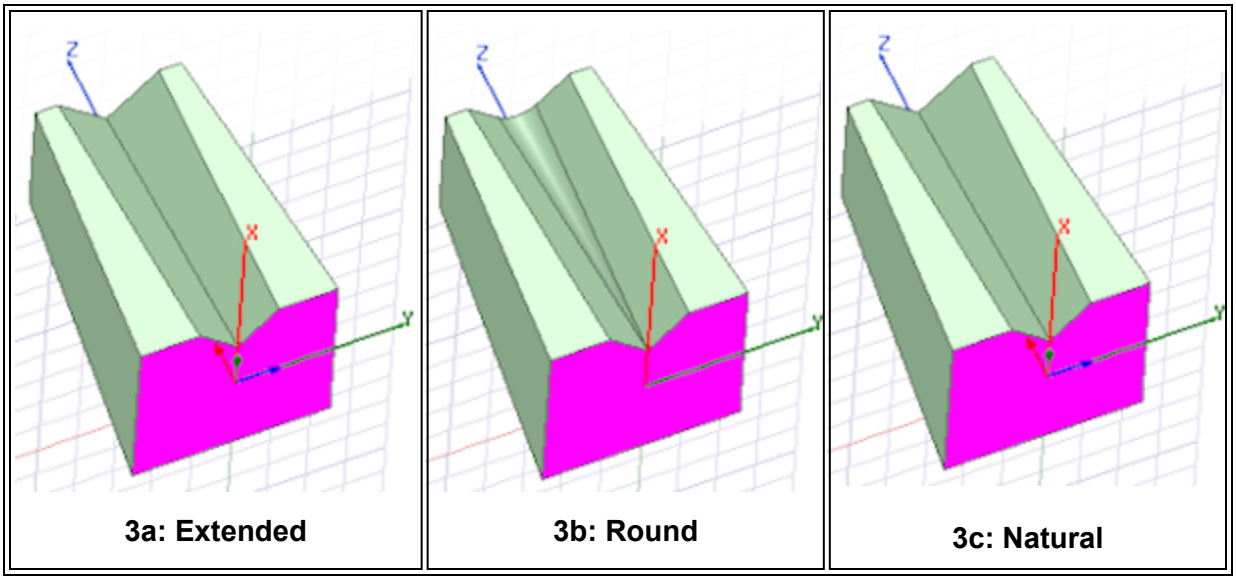


2b: Round



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.

  In 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.

  In example **3a** (negative draft angle), no face is split and no edge is rounded.

- **Round Option:** In both examples **1b** and **2b**, 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).

  In 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.

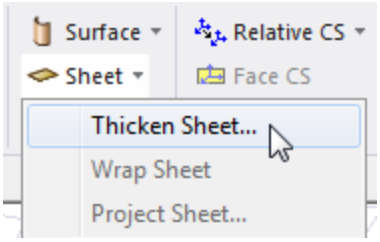
- **Natural Option:** In both examples **1c** and **2c**, the corners adjacent to the convex curved face remain sharp. However, the corners adjacent to the concave curved face are progressively rounded in both cases (with the radius increasing from zero at the plane of the original object to a maximum radius at the end of the sweep).

  In example **3c** (negative draft angle), no face is split, and no edge is rounded.

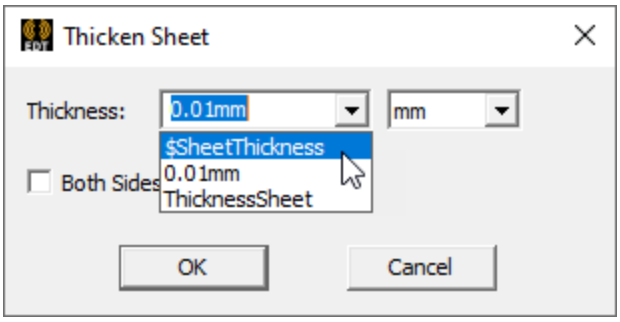
Thicken Sheet

To thicken one or more sheet objects to make 3D objects:

1. Select the sheet or sheets you want to thicken into 3D objects.
2. From the menu bar, click **Modeler** > **Surface** > **Thicken Sheet** or, on the **Draw** ribbon tab, choose **Sheet** > **Thicken sheet**:



The *Thicken Sheet* dialog box appears:



3. Specify the **Thickness**. If you have set Project or Design variables with relevant values, or have previously assigned values, this are listed in the dropdown.
4. If necessary, specify the units by selecting from the drop-down menu.
5. If you want to thicken symmetrically (half of the thickness on each side of the sheet or sheets), select the **Both Sides** check box.

   The resulting solid is the requested thickness, but the original sheet location corresponds to the middle of the resulting solid's thickness.

6. Click **OK**.

   The dialog closes and the sheets are changed into 3D objects 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 of bringing them into contact.

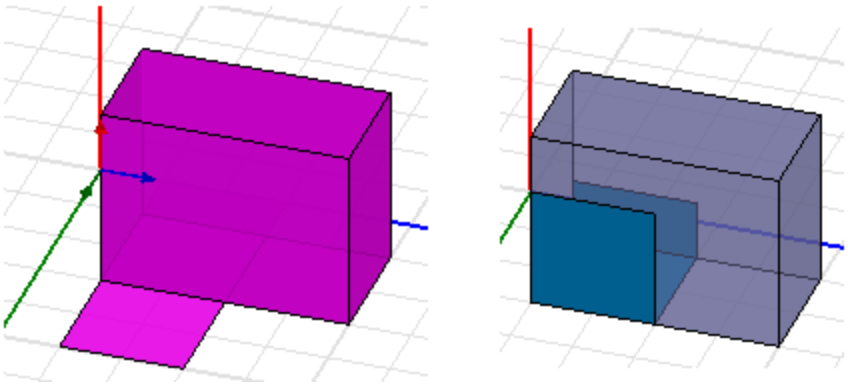
- If an edge from the sheet body and the target body are close enough and at a small angle (10 degrees), the edges are brought into alignment before the wrap.

- 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 criterion are not matched, a vertex from the sheet body is moved to the closest face of the target body.

If 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 moving to match or align 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 following figure:

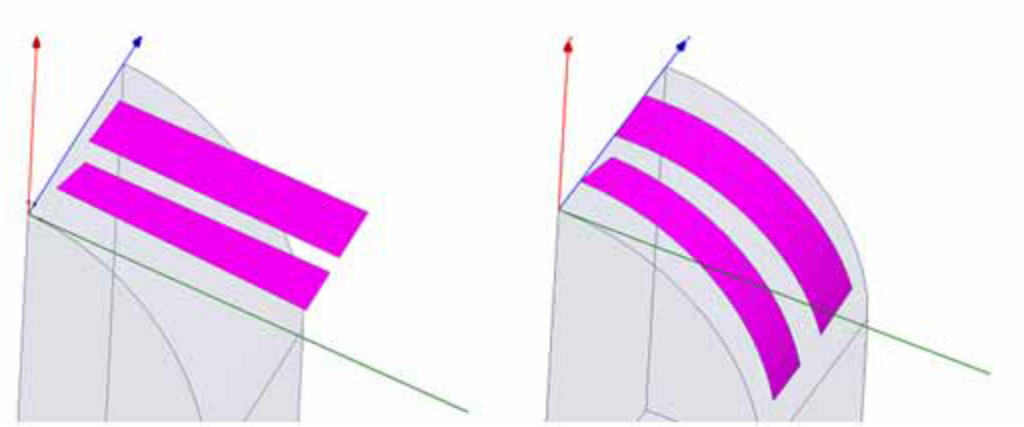


If you delete the 3D object, the wrapped sheet retains the form it took when wrapped.

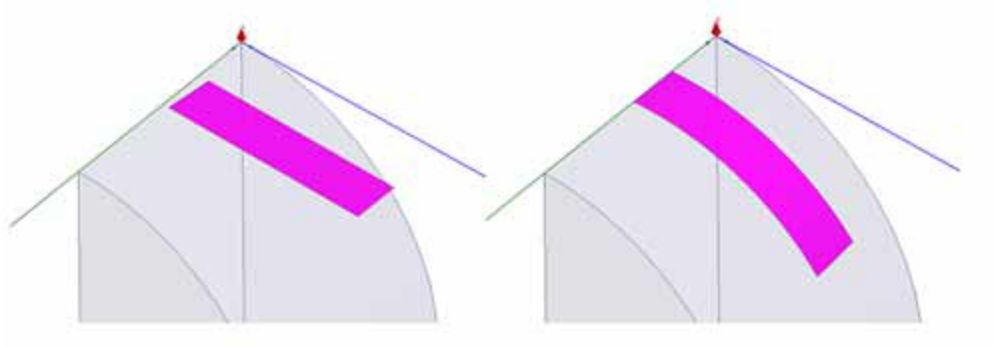
A sheet object that overlaps corners may not wrap in straightforward fashion, depending on both the angle(s) involved, and the sheet object. While it is possible, it is not recommended.

Examples:

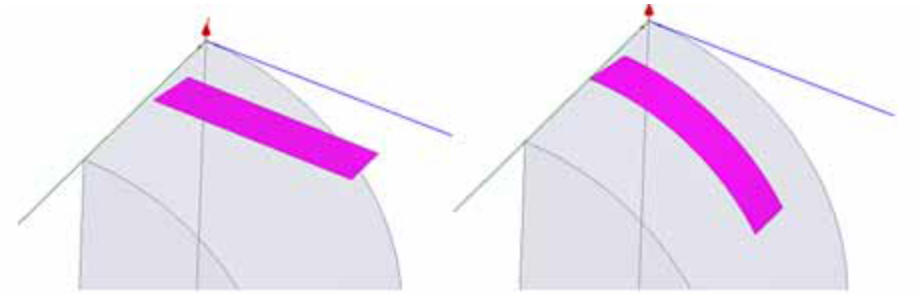
The following figure illustrates wrapping on a target surface with the snap to an edge. The surface will snap to an edge if the distance is close relative to the length of the edge. If the angle between the edge is less than 10 degrees, the edge to be wrapped will be aligned to the target body. In the 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 vertex will not be moved to the target edge unless it is within modeling tolerances.



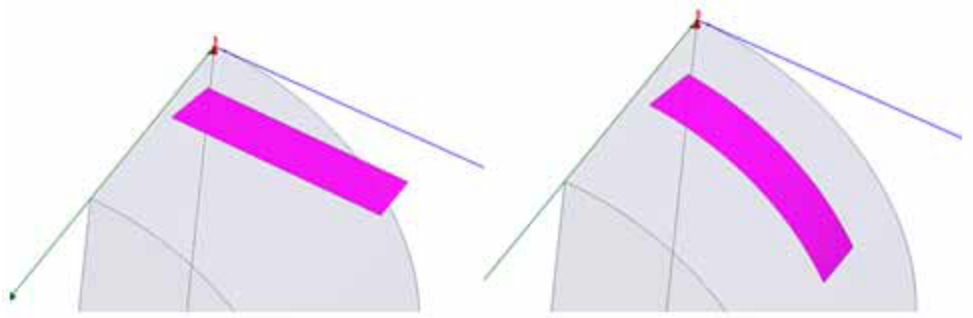
The following figure shows the situation when you move the edge to the edge before wrapping.



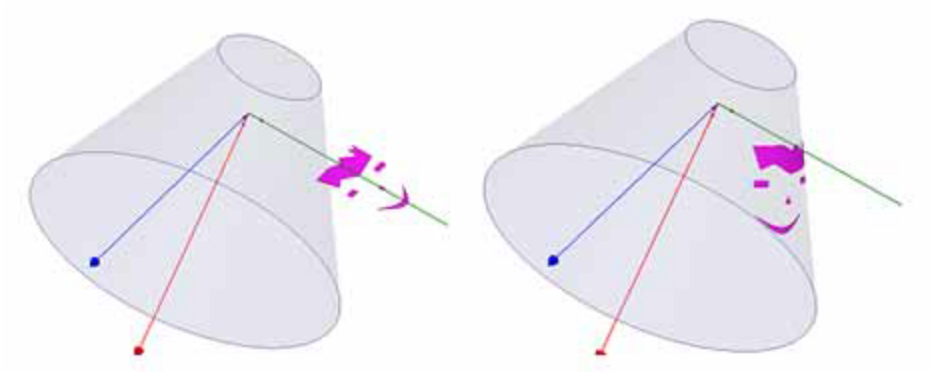
The following figure shows the situation when you move a vertex to an edge before wrapping.



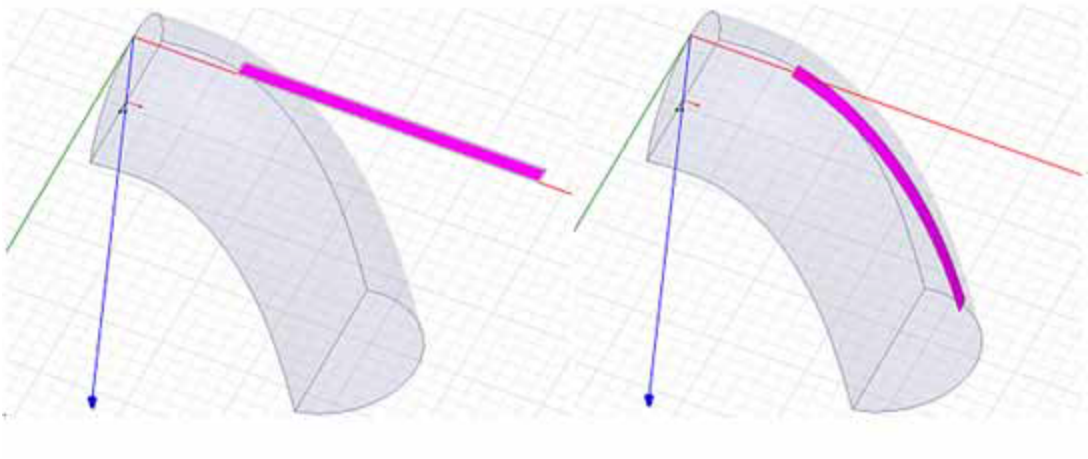
The following figure shows the situation where you move a vertex to a face before wrapping.



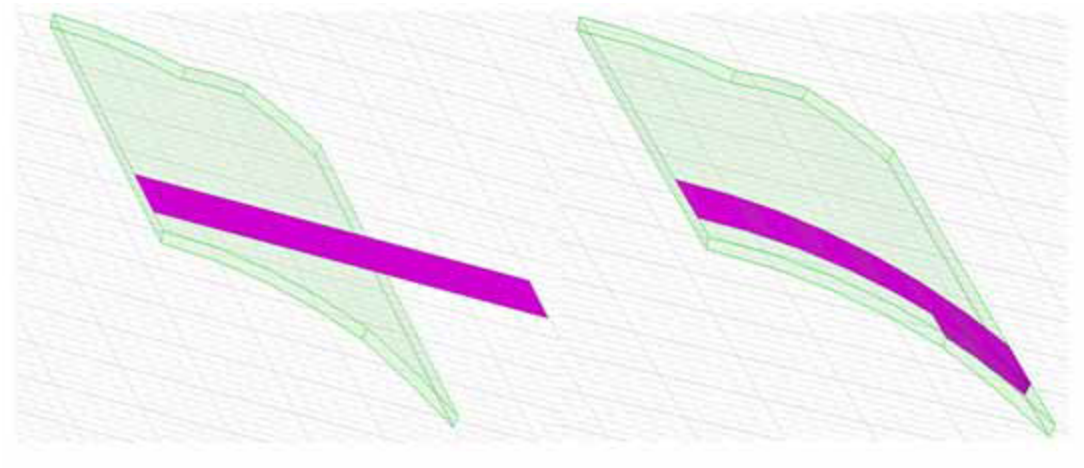
The following figure shows the situation when you move a multilump sheet to touch the target body. A rectangle that encloses all lumps is used to position the sheet on the target.



The next figure illustrates wrapping to a non-developable surface. The area of the sheet changed from 0.7000 mm^2 before wrapping to 0.6986 mm^2 after wrapping.



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:

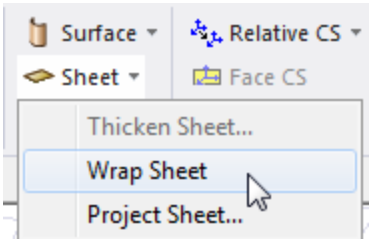
- Wrapping on non-developable surfaces creates distortion in lengths.
- Wrapping is not allowed across curved boundaries between faces.
- Working with tolerant edges is successful only if the gaps can be tightened sufficiently for the operation.
- Wrapping on the pole of a surface (such as sphere or 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. From the menu bar, click **Modeler** > **Surface** > **Wrap Sheet** or, on the **Draw** ribbon tab, select **Sheet** > **Rap 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** > **General Options**, 3D Modeler options to automatically perform a **Modeler** > **Boolean** > **Imprint** command after performing a **Wrap.**

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 closed polyline object to a sheet object:

1. Select the closed polyline object you want to cover.
2. From the menu bar, 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 *3D Modeler > Drawing* section of the *General Options*. 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. From the menu bar, 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 the *Face* selection mode: Click **Edit** > **Selection Mode** > **Faces**.
2. Select a face of the object you want to uncover.

   From the menu bar, 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.

   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 section plane is set.
2. Select the object from which you want to create a cross section.
3. From the menu bar, click **Modeler** > **Surface** > **Section**.

   The *Section* dialog box appears.

4. Select the **Section Plane** you will use to divide the object: **XY, YZ,** or **XZ**.
5. Click **OK**.

   A closed polyline object is created at the intersection of the 3D object and the specified section plane. The original 3D object is unmodified.

Connecting Objects

Use the **Modeler** > **Surface** > **Connect** command to perform the following operations:

- Connect two or more 1D polyline objects. Ansys Electronics Desktop 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. Ansys Electronics Desktop 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.

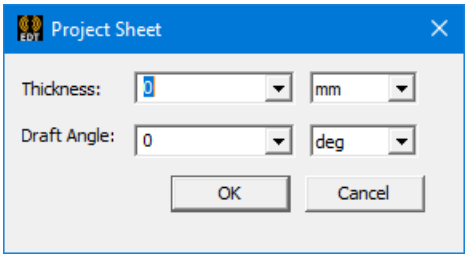
To connect objects:

1. Select the objects you want to connect. Select 3D objects as objects, not as faces.
2. Click **Modeler** > **Surface** > **Connect**.

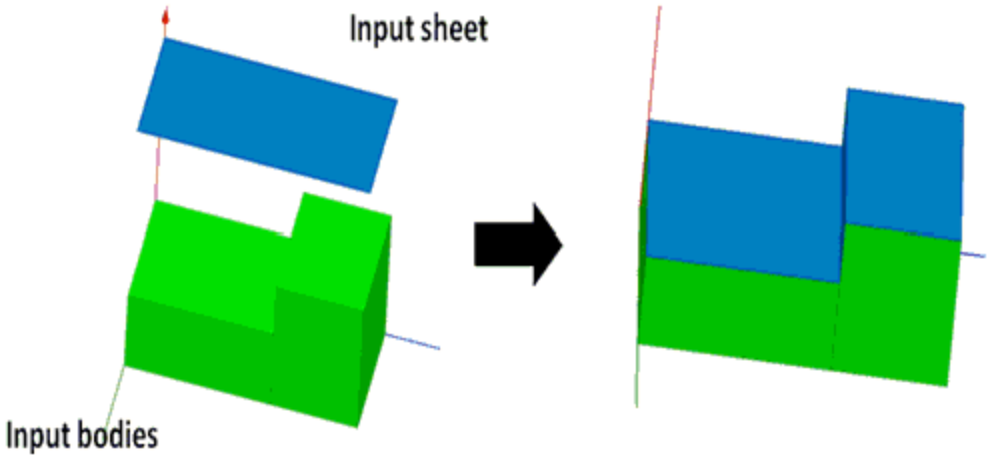
   A new object is created that connects the objects you selected. The first object you selected was modified to create the new object and all subsequently selected objects were deleted.

Project Sheet Object

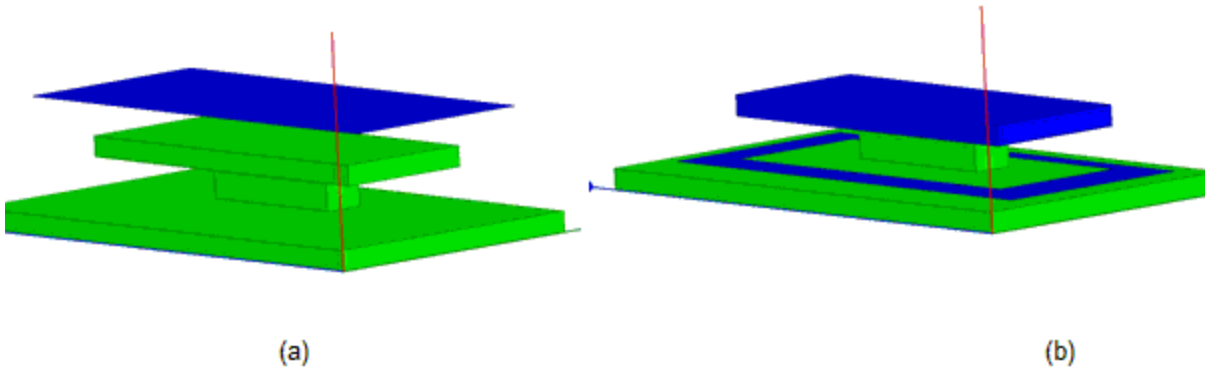
Use the **Modeler** > **Surface** > **Project Sheet** command to apply a selected planar sheet object to a suitable 3D object. This permits easy modeling of thin conformal deposits. (The **Wrap Sheet** and **Imprint Projection** commands are not suitable for this use.) For modeling conformal deposits when using **Project Sheet**, you can specify a thickness and a draft angle.



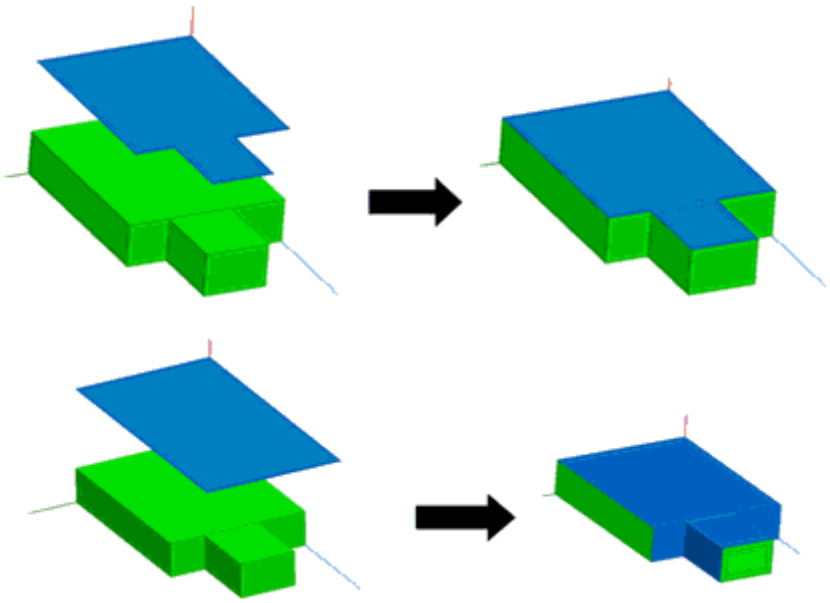
To use the **Project Sheet** command you must select one planar sheet and at least one solid body. If you select multiple solids, the sheet is project on all bodies as if they have been united. After the commend executes, a new sheet that lies on the surface of the selected bodies is created. The solid bodies are not changed.



Faces that are hidden from the sheet being projected are not covered, due to the projected sheet.

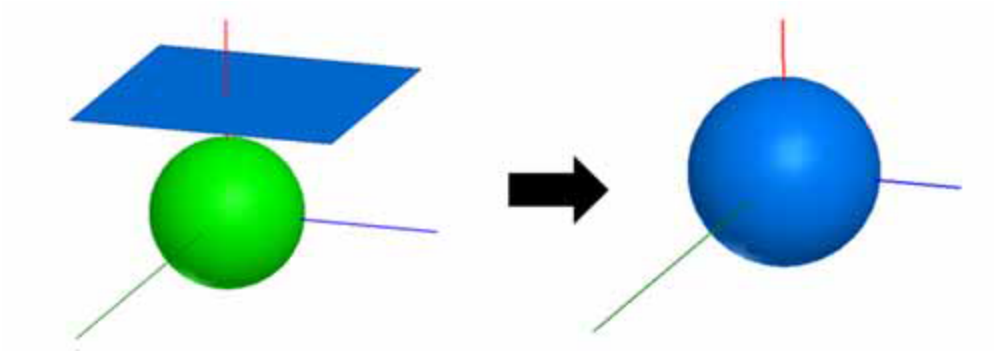


Only vertical faces (or faces parallel to the projection direction) that lie completely within the projection of the sheet will be covered by the projected sheet. Vertical faces that lie on the boundary of the projection are not covered.



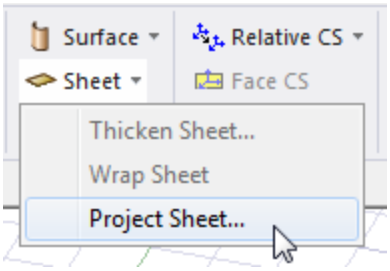
If the sheet cross-section exactly matches the body cross-section, no vertical faces are covered. If the projected sheet exceeds the cross-section of solids, faces that lie with in the sheet projection are covered.

If a surface like a cylinder or sphere lies completely within the sheet projection, the entire surface will be covered, though there may be a warning that “Output may be incorrect because a self-obstructing face was found.”



To Project a Sheet:

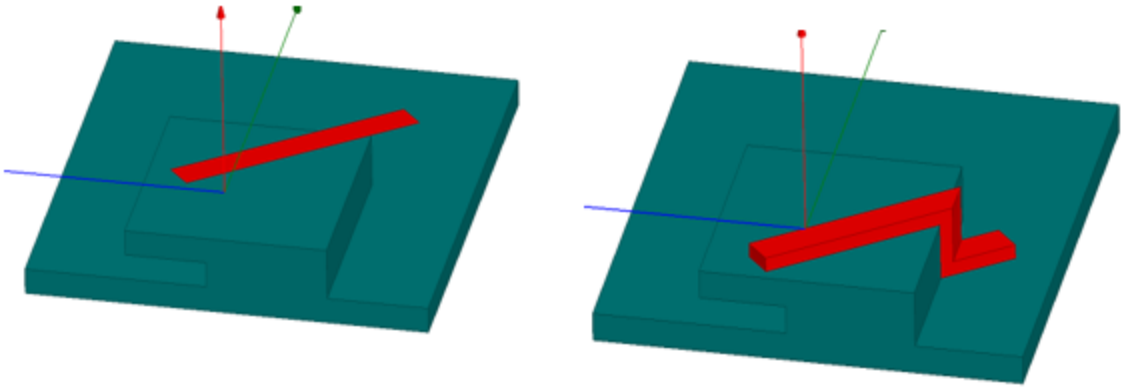
1. Select a planar sheet object and one or more appropriate solid objects.
2. From the menu bar, click **Modeler** > **Surface** > **Project Sheet**, or on the **Draw** ribbon tab, click **Sheet** > **Project Sheet**:



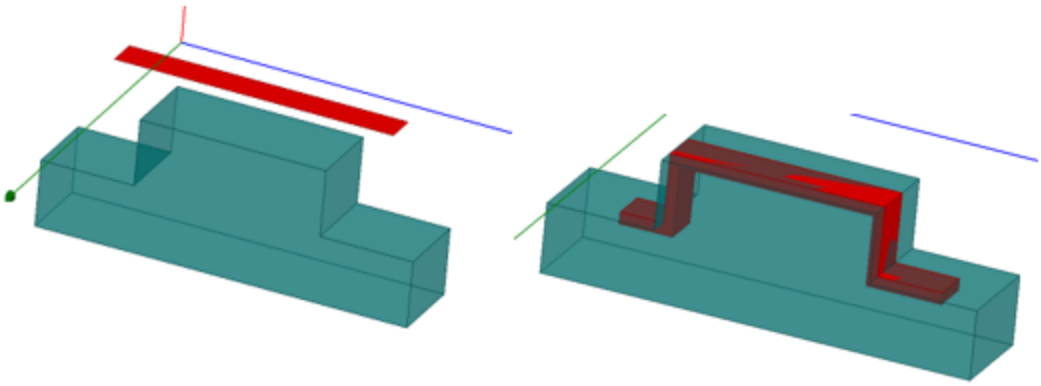
   For modeling conformal deposits, you can also specify a thickness and optional draft angle.

After you perform **Project Sheet**, the History Tree shows the **Project Sheet** command and the create command for the imprinted object. If you select the **Project Sheet** command in the History Tree, you can suppress the command via the docked *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.

Specifying a Thickness value has different effects depending on whether the value is positive or negative. The following figure shows the effect of a positive thickness, with the Projected sheet taking its thickness outward from the target object.

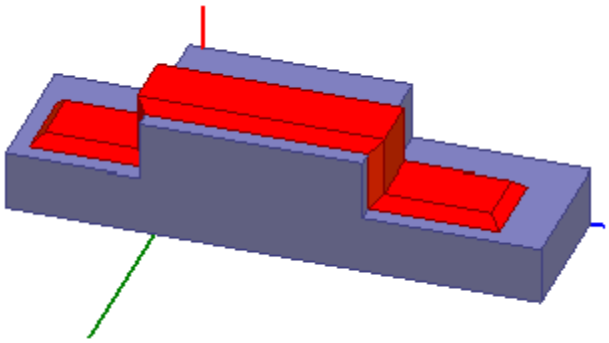


The following figure shows the effect of a negative value for thickness.

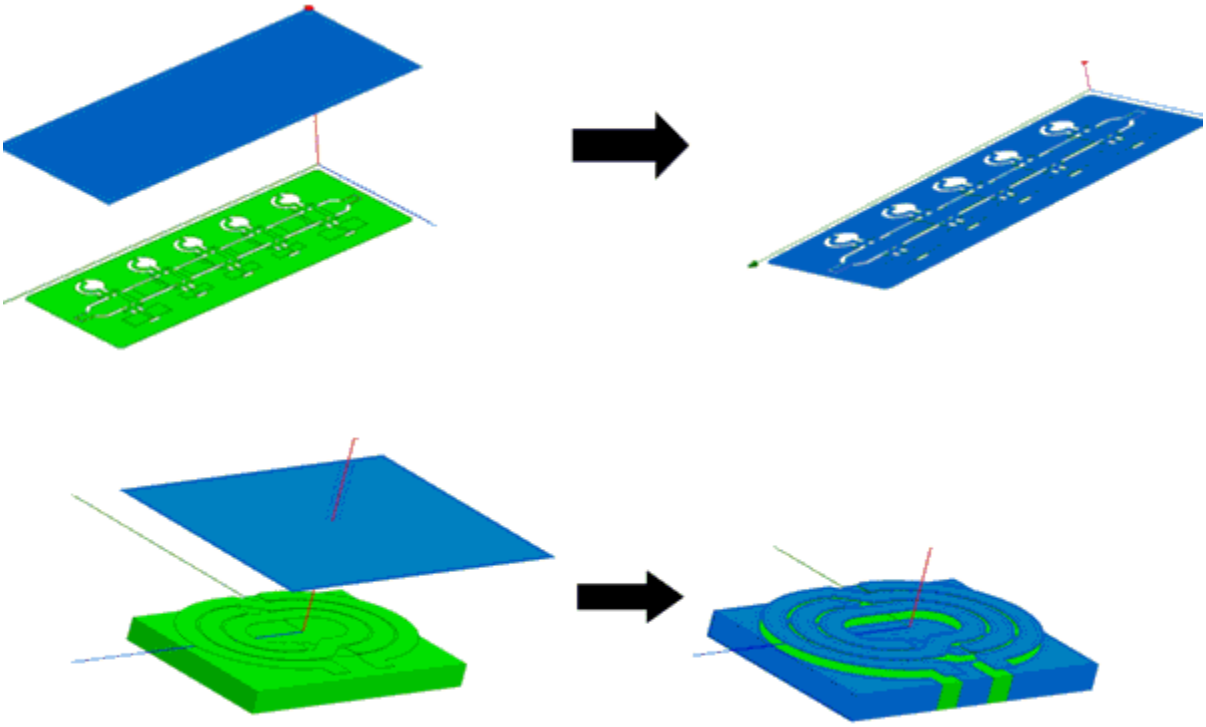


When a negative value is given the thickness will be limited by the target solid body. That is, the thickness can not be beyond the original solid body which was chosen. If you use a negative thickness it is also possible that not all overhangs will be eliminated.

The following figure shows the effect of a 45 degree draft angle.



Other examples of Project Sheet on complex models are show in the following figure.



Moving Faces or Edges

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.

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. Right-click in the Modeler window and click **Select Faces** on the shortcut menu.
2. Select the face of the object you want to move.

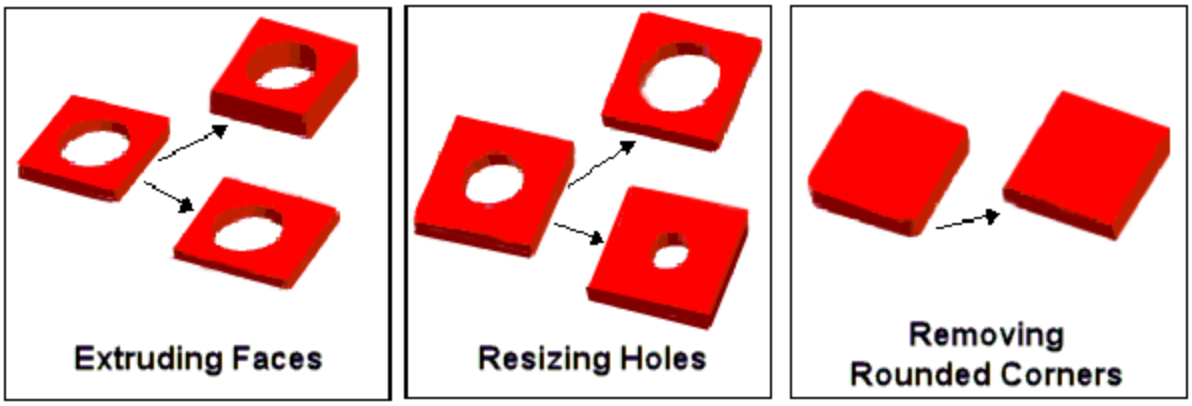
   If you have created a suitable face list, right-click the list and click **Select Assignment** from the shortcut menu. You can operate on faces in the list.

3. Click **Modeler> Surface> Move Faces**> **Along Normal**.

   The *Move Faces Along Normal* dialog box appears.

4. Type the distance you want to move the object face.
5. Click **OK**.

   The face will be moved the distance you specified.



To move every face of an object normal to its surface, use the **Edit> Arrange> Offset** command.

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.

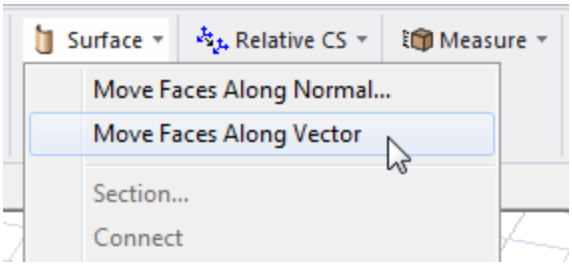
This command is useful for relocating holes in an object, as shown below.

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.

   If you have created a suitable face list, right-click the list and click **Select Assignment** from the shortcut menu. You can operate on faces in the list.

3. From the menu bar, click **Modeler> Surface> Move Faces> Along Vector**, or, on the **Draw** ribbon tab, click **Surface> Move Faces Along Vector**:



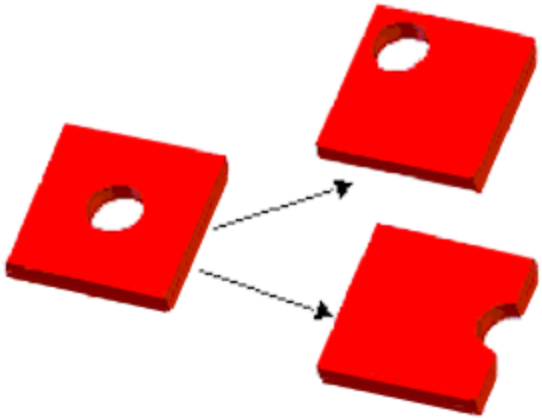
4. Specify the vector along which the face will be moved:
   a. Select an arbitrary anchor point in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the **X, Y,** and **Z** boxes.

      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.



Relocating Holes

To move every face of an object normal to its surface, use the **Edit>Arrange>Offset** command.

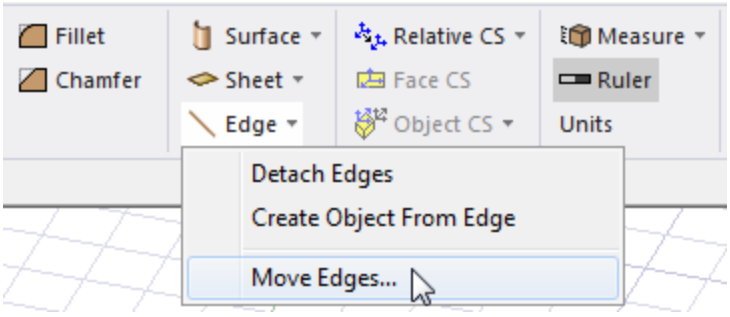
Moving Edges Along Normal

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. Right-click in the Modeler window and choose **Select Edge** on the shortcut menu.
2. Select the edge of the object you want to move.
3. From the menu bar, click **Modeler> Edge> Move Edge** or, on the **Draw** ribbon tab, click **Edge> Move Edges**:



   The *Move Faces Along Normal* dialog box appears.

4. Type the distance you want to move the selected edge.

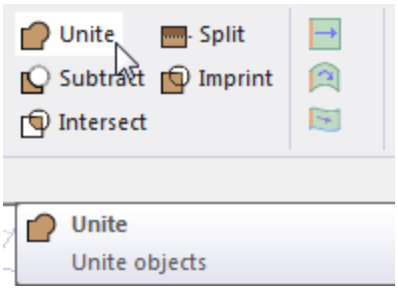
5. Click **OK**.

   The edge of the object is moved based on the value you specified.

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.

1. Select the objects you want to join.
2. From the menu bar, click **Modeler> Boolean> Unite** or, on the **Draw** ribbon tab, click the **Unite** icon:

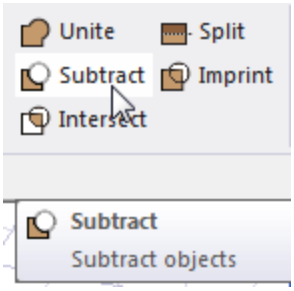


   The objects are united.

> **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 tool objects before uniting** in the *3D Modeler> Operation* section of the *General Options*. 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. From the menu bar, click **Modeler**> **Boolean**> **Subtract** or, on the **Draw** ribbon tab, click the **Subtract** icon:



The *Subtract* dialog box appears.

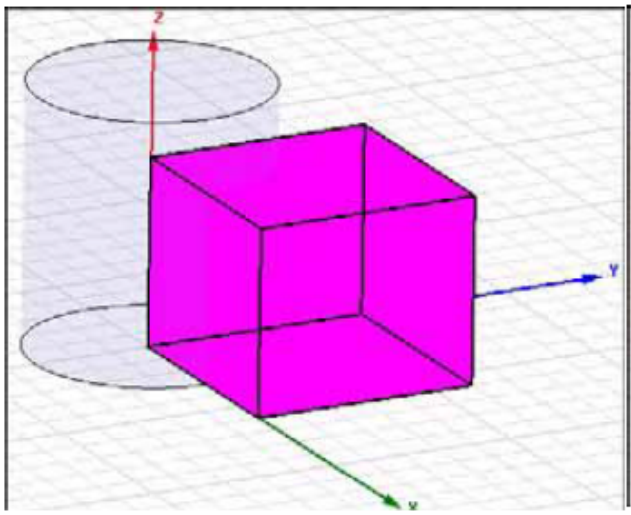
Objects listed in the *Tool Parts* list will be subtracted from the object or objects listed in the *Blank Parts* list.

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.

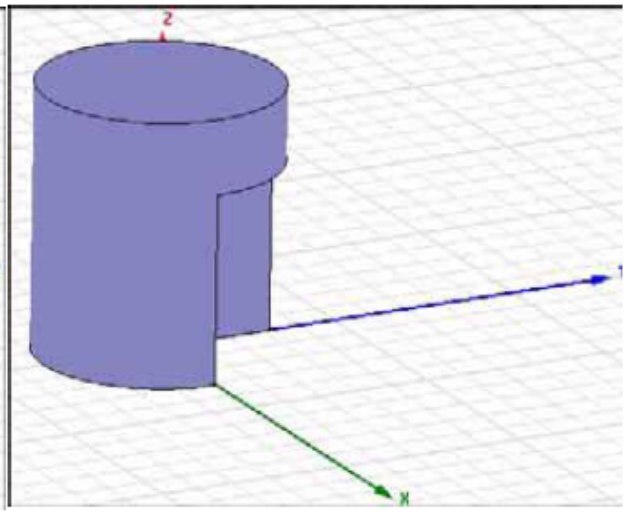
   Though you can type the name of objects into either list, you cannot access the *Subtract* command unless you first select two or more model objects. Therefore, the lists will initially be populated with the selected objects.

5. Optionally, select **Clone tool objects before subtracting**. This option instructs Ansys Electronics Desktop to keep a copy of the original objects being subtracted.
6. Click **OK**.

   The revised *Blank Parts* retain their original name, color, and material.



An intersecting box and cylinder



A box subtracted from a cylinder (the cylinder was selected first)

Note:

By default, the *Tool Parts* (that is, the objects being subtracted from the *Blank Parts*) are *not* preserved for later use. If you want to keep a copy of the objects being subtracted from the blank parts, do one of the following:

- In the *Subtract* dialog box, select **Clone tool objects before subtracting**, as stated in step 5 above. This action is a one-time override of the defdefault behavior.
- In the *3D Modeler*> *Operation* section of the *General Options*, select **Clone tool objects before subtracting**. This option instructs Ansys Electronics Desktop to always (that is, by default) keep a copy of the original objects being subtracted.

Creating Objects from Intersections

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.

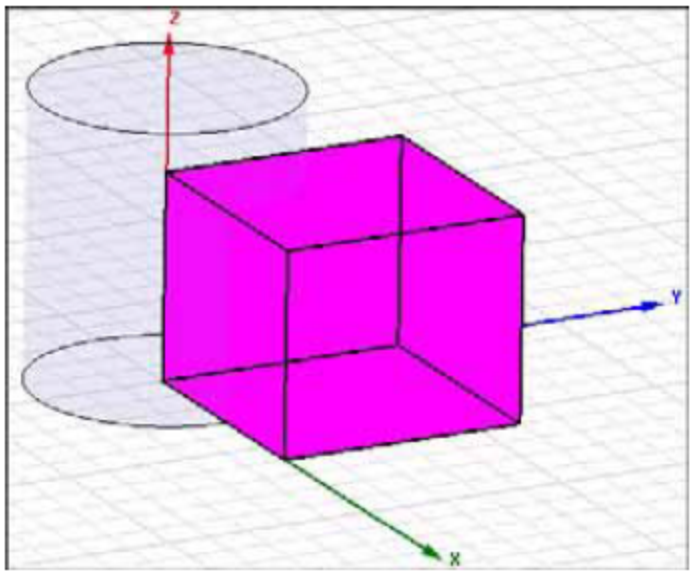
   Warning:

   If the objects you selected do not overlap, the result is a null object, and both objects vanish. When three or more objects have been selected, an intersection is created only if a common volume exists in which all the selected objects overlap. Otherwise, a null object is created and all original objects vanish.

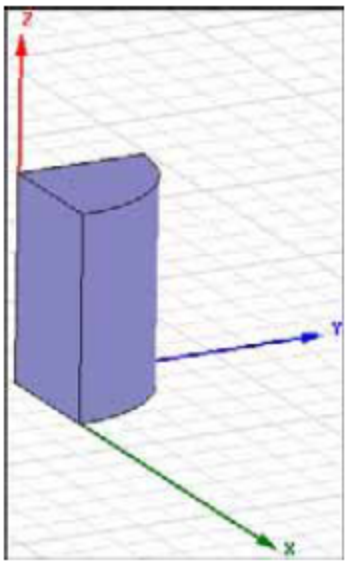
   For example, assume that Part 1 overlaps Part 2, and Part 2 overlaps Part 3, but all three parts do not overlap in a common volume. This situation results in no intersection being created, and all three parts vanishing.

2. From the menu bar, click **Modeler** > **Boolean** > **Intersect**. (You can also access this command from the **Draw** ribbon tab.)

   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:

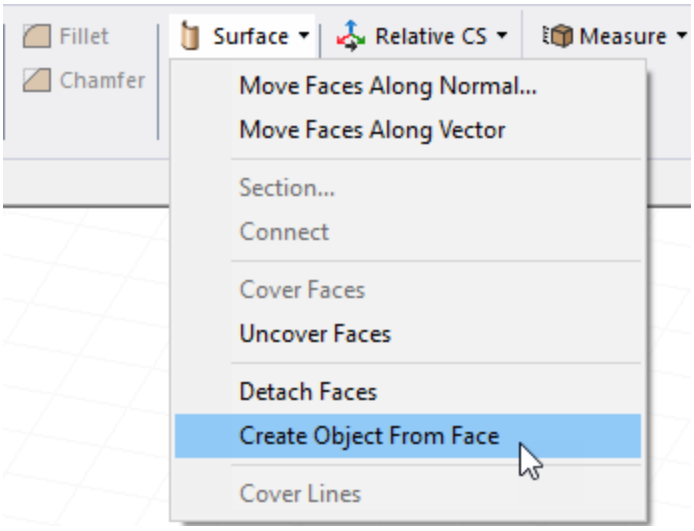
- Copy the objects, and then paste them back into the design after creating the new object from the intersection.
- Select **Clone tool objects before intersecting** in the *3D Modeler> Operation* section of the *General Options*. This option instructs the modeler to always keep a copy of the original tool parts (the second and subsequent objects selected) that intersect the blank part (the first object selected). The blank part is not preserved. To preserve the blank part, do as instructed in the preceding bullet.

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 choose **Select Faces** from the shortcut menu.
2. Select the object face you want to copy. If you select multiple faces, each becomes a new object.
3. From the menu bar, click **Modeler**>**Surface>Create Object From Face** or, on the **Draw** ribbon tab, click **Surface**> **Create Object From Face**:



The face is copied, resulting in a new 2D sheet object in the same location.

Tip:

This command is useful for assigning a boundary to the partial intersection of two faces. To do this, first select the faces and create objects from them using the procedure shown above. Next, make sure the **Clone tool objects before intersecting** option is cleared in the *3D Modeler> Operation* section of the *General Options*. Then, select the two sheet objects and use the **Modeler> Boolean> Intersect** command to create a sheet object that includes only the intersection of the two faces. Finally, assign the boundary to the new sheet object.

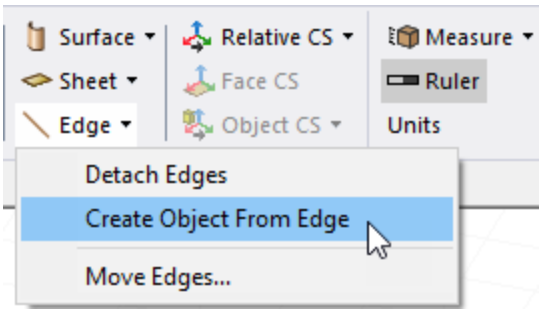
Creating an Object from an Edge

The **Modeler> Edge> Create Object From Edge** command copies a selected edge, resulting in a new line object.

To create a new object from an edge:

1. Right-click in the modeler window and choose **Select Edges** from the shortcut menu.
2. Select the object edge you want to copy. If you select multiple edges, each edge becomes a new line object.

3. From the menu bar, click **Modeler**> **Edge**> **Create Object From Edge** or, on the **Draw** ribbon tab, click **Edge**> **Create Object From Face**:

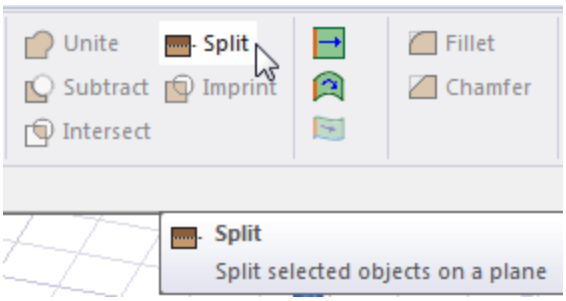


The edge is copied. The resulting object appears in the History Tree as a line object.

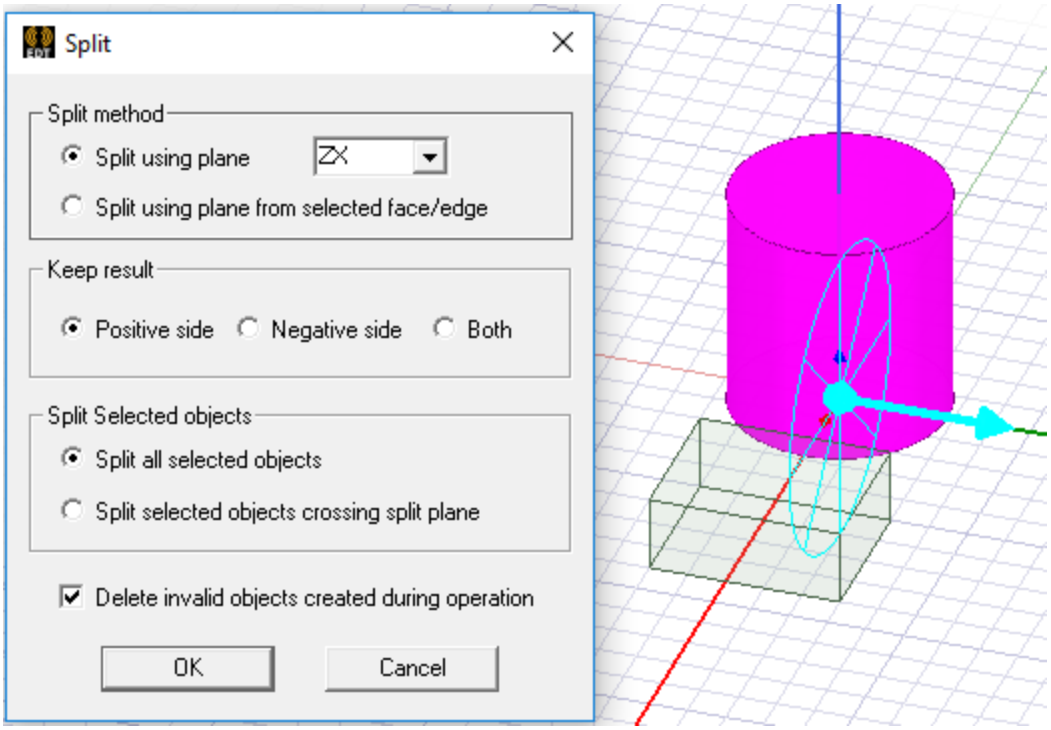
Splitting Objects

To split an object or several objects using one of the global planes (XY, YZ, or XZ) or a plane based on a selected face or arc, use the **Modeler> Boolean> Split** command.

1. Select the object you want to split. You can select more than one object.
2. From the menu bar, click **Modeler> Boolean> Split** or, on the **Draw** ribbon tab, click the **Split** icon:

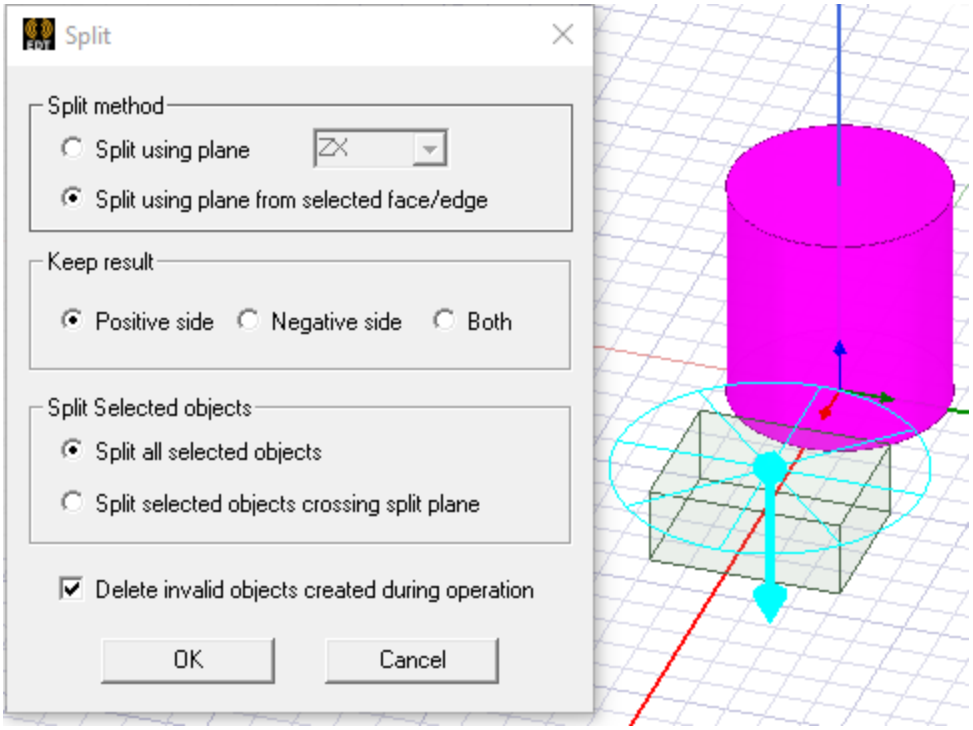


The *Split* dialog box appears. Also, a cyan colored wheel shows the current split plane and points toward the positive side of the plane.



3. Choose the desired *Split method* and specify the splitting plane:
   - **Split using plane:** Select the desired splitting plane (**XY, YZ,** or **XZ**) from the drop-down menu, or
   - **Split using plane from selected face/edge:** Select a single face or edge (specifically, a planar face or arc) to define the splitting plane.

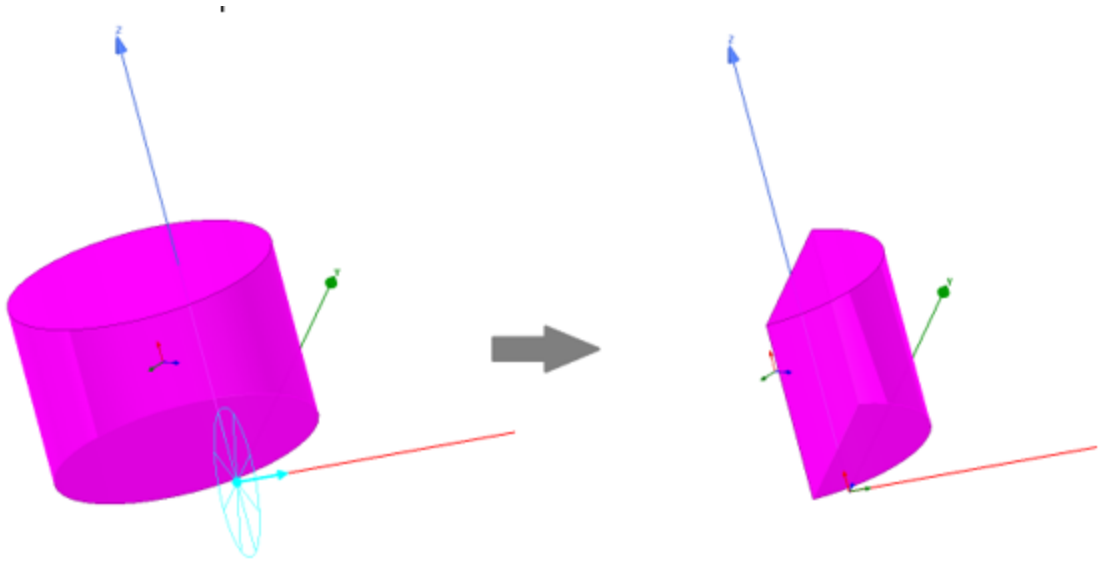
     The following example shows the top face of the box being used to define the splitting plane for the cylinder:



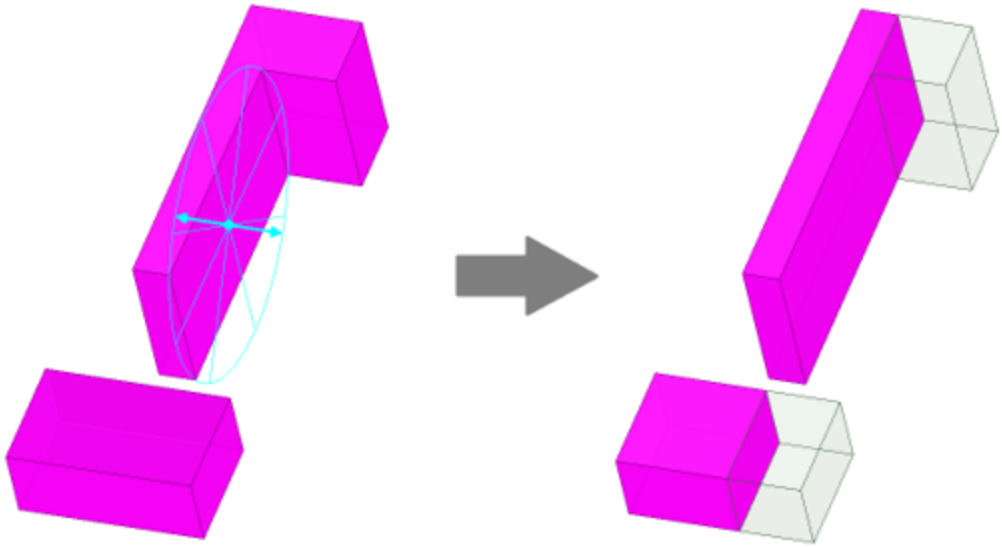
4. Select one of the following *Keep result* options to specify which object fragments you want to keep:
   - **Positive side:** Keep objects on the positive side of the splitting plane.
   - **Negative side:** Keep objects on the negative side of the splitting plane.
   - **Both:** Keep objects on both sides of the splitting plane.

   The split plane is shown using a cyan colored wheel with spokes, as seen in the images above and below left. A cyan arrow indicates which side of the split plane is the positive side.

   In the following example, the *YZ* plane (green and blue axes) is the splitting plane, and the result on the *positive side* was kept. The picture on the left is before the split. The picture on the right is the result of the completed *Split* command:



The next example shows the second *Split method* (*Split using plane from selected face/edge*). You can only select one entity, a planar face or an arc (selected as an edge). The face or edge used to define the splitting plane need not belong to one of objects selected to be split. You can select any face or arc edge in the model.. You can pan, zoom, or rotate the model viewpoint as needed to make the selection easier. Also, the *Keep result* for *Both* sides option was selected. (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 of the completed *Split* command:



5. Select one of the following *Split Selected objects* options:
   - **Split all selected objects:** Select this option if you do not want to preserve objects that are not crossing the split plane but are part of the selection.
   - **Split all selected 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 software versions are not changed.
6. Optionally, select the **Delete invalid objects created during operation**. If you choose to *Split all selected objects*, the operation could create invalid objects (if split plane does not cut any of selected objects and any are on the side not being kept). This option deletes any such invalid objects that might be created.
7. Once the proper selections have been made, click **OK**.

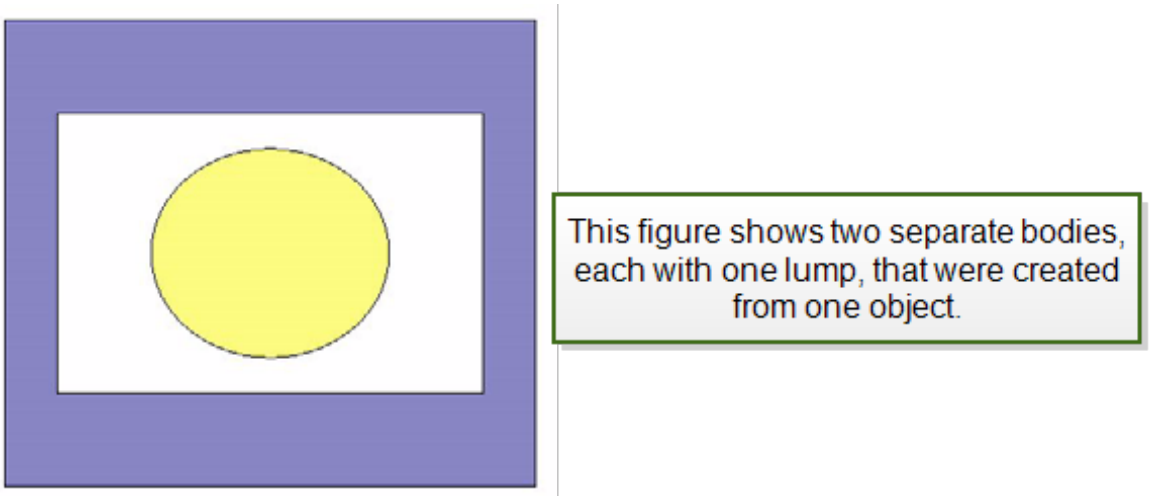
   The objects are divided and retained as specified.

Separating Bodies

To separate an object with multiple lumps (geometric bodies) into individual single-body objects:

1. Select the object you want to separate.
2. From the menu bar, click **Modeler**> **Boolean**> **Separate Bodies**.

   The bodies are separated into multiple objects.



Converting Polyline Segments

A polyline is a single object that includes any combination of straight line, arc, or spline segments. You can convert a polyline segment from one type to another. The following

conversions are supported:

- Straight line segments to arc or spline segments.
- Arc 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 tree.
2. Click the polyline segment operation you want to change to select it.

   The segment properties appear in the *Segment* tab of the docked *Properties* window.
3. Click in the **Value** text box in the **Segment Type** row and select the desired polyline segment type from the drop-down menu.

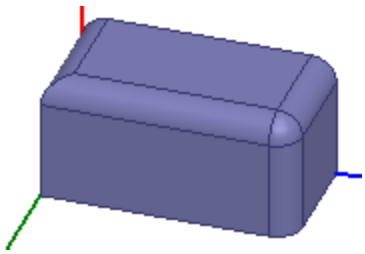
   The polyline segment you selected is changed to the new type.

   > **Note:**
   >
   > Converting an arc 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.
4. By default, curved segments are generated as smooth (true) curves. If you want to approximate an arc or spline using multiple discrete straight segments, enter an integer of 2 or greater in the **Number of Segments** value. A value of zero produces a smooth curve.

   Segmented curves can be useful when sweeping polylines into surfaces. Segmented surfaces provide some regularity to the mesh and can enable you to visualize surface twisting when you sweep along a path with a twist angle specified.
5. If the changes are not what you expected, undo the change using the **Edit> Undo** command or by pressing **Ctrl+Z**.

Rounding the Edge of Objects (Fillet Command)

The fillet command rounds the object at the original edges and vertices. The edges and vertices are replaced by new rounded surfaces that are tangential to the original adjacent surfaces, providing a smooth transition.



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:

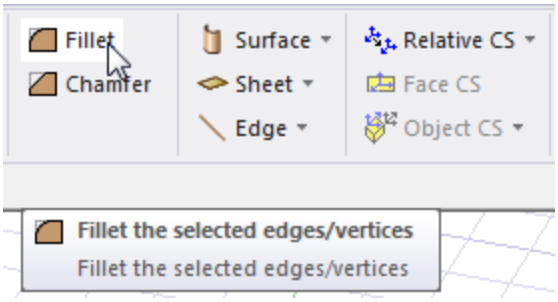
- Right-click the desktop, and choose **Select Edges** from the shortcut menu.

To round the edge of an object:

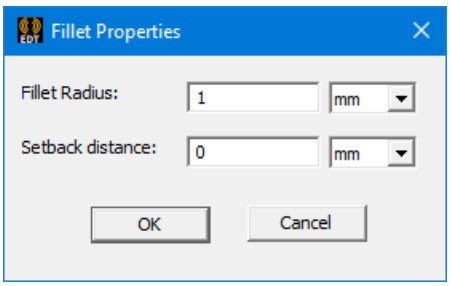
1. Select the edge you want to change.

   The edge is highlighted and the *Fillet* command is enabled.

2. From the menu bar, click **Modeler** > **Fillet** or, on the **Draw** ribbon tab, click the **Fillet** icon:



   The *Fillet Properties* dialog box is displayed:



3. Enter a value for the **Fillet Radius** and select units from the drop-down menu.
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 the *Setback distance* has no effect. Also, you will get an error if it is greater then the length of the edge. Note that the setback feature only works on corners where three or more edges meet and only if all edges meeting at the vertex are selected.

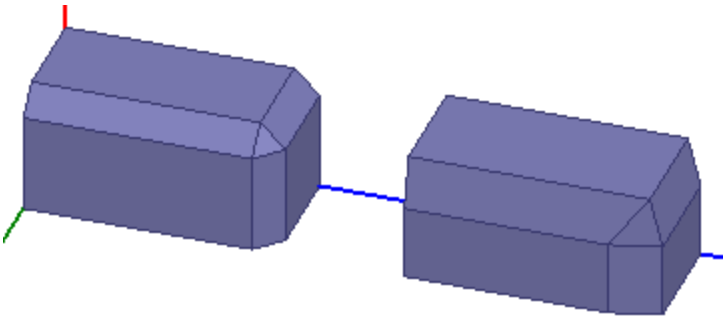
5. Click **OK** to apply the change to the edge.

   The dialog box closes and the object edges and vertices are rounded as specified.

Flattening the Edge of Objects (Chamfer Command)

The chamfer command flattens the selected edges and vertices of the object. Selected edges and vertices are replaced by new flat surfaces known as chamfers.

Chamfers can be symmetric, or asymmetric:



The angle between the original faces and the chamfer's face varies depending on the angle between the original faces, and it also depends on the left and right chamfer distances. For example, for two faces at 90° with an equal distance (symmetrical) chamfer, the edge chamfer's face will be 45° from both of the original faces.

Vertices are only going to be replaced by new flat faces if all the edges connecting to the original vertex are selected; otherwise, the vertex is preserved but moved (if necessary).

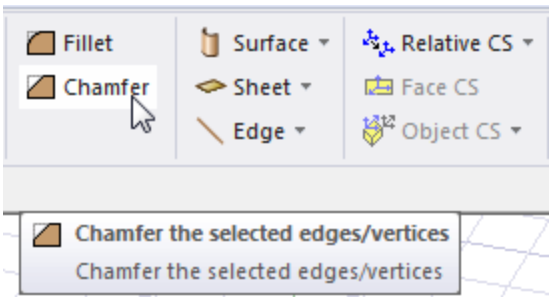
The chamfer command is disabled if an edge is not selected.

To flatten an object's edge for a symmetric chamfer:

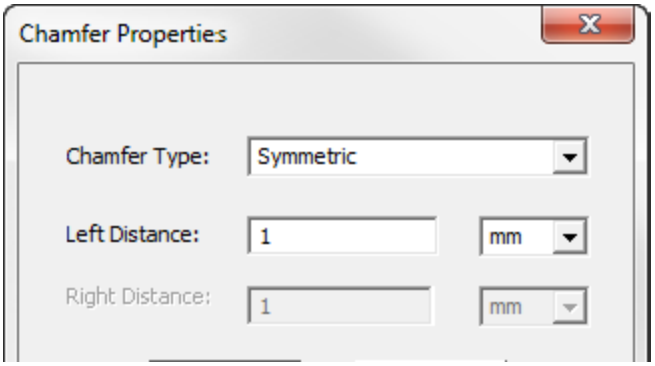
1. Right-click the desktop, and select **Select Edges** from the shortcut menu.
2. Select the edge (or edges) you want to change.

   The edge(s) is highlighted, and the **Chamfer** command is enabled.

3. From the menu bar, click **Modeler> Chamfer** or, on the **Draw** ribbon tab, click the **Chamfer** icon:



The *Chamfer Properties* dialog box appears.

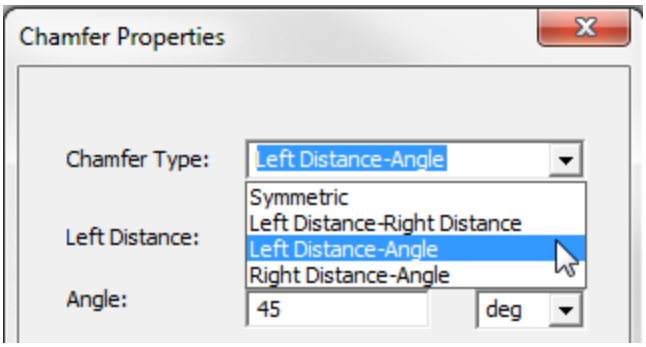


4. Type a **Left Distance** value and select the units from the drop-down menu.
5. Click **OK** to apply the change to the selected edges.

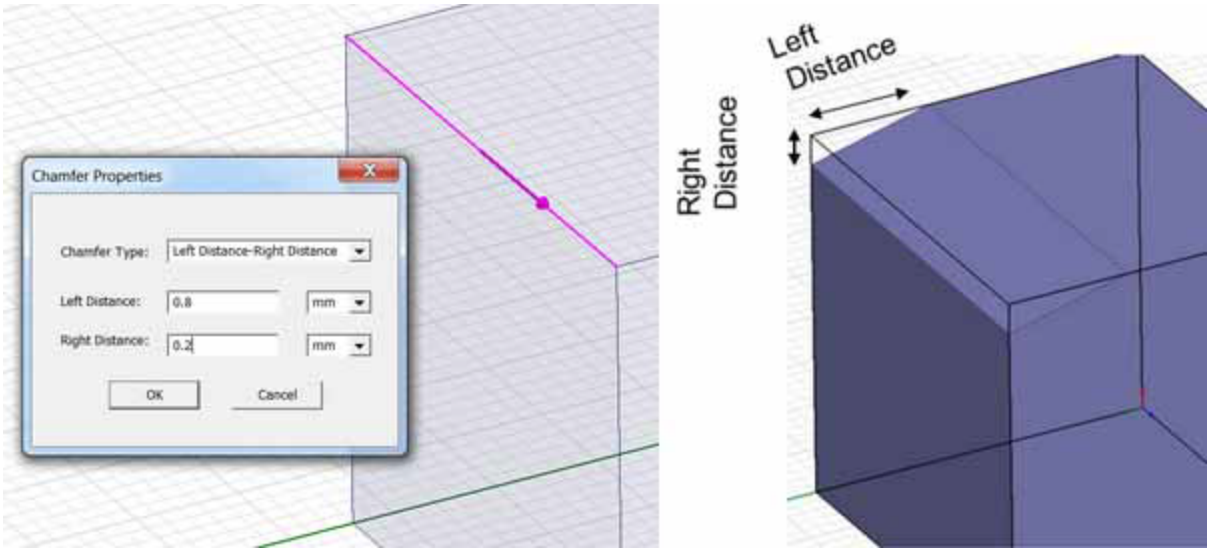
To flatten object's edge for an asymmetric chamfer:

1. Select one or more edges and click **Modeler> Chamfer** to open the *Chamfer Properties* dialog box.
2. Use the **Chamfer type** drop-down menu to select the type:

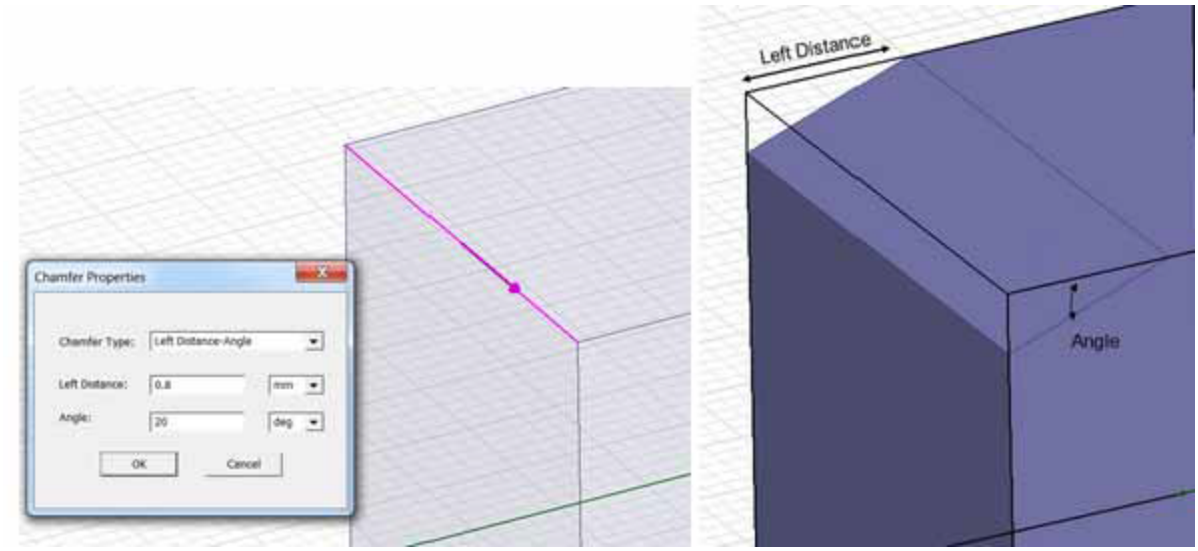
   You control an asymmetric chamfer by selecting a type that defines the chamfer asymmetry as **Left Distance-Right Distance**, as **Left Distance-Angle**, or as **Right Distance-Angle**. Notice that red directional arrows on the selected edges provide the direction against which left distance and right distance are determined.



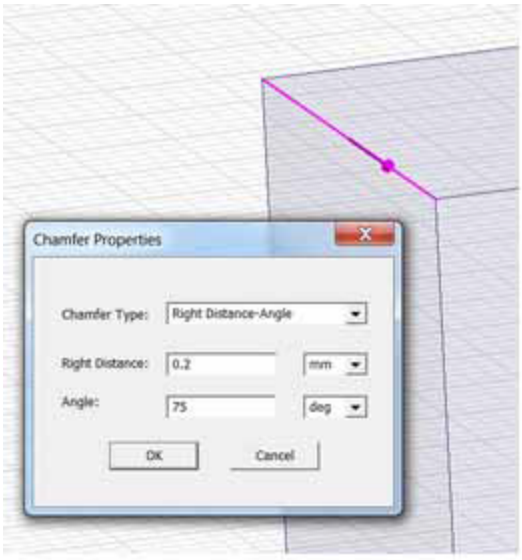
Example 1: Left Distance-Right Distance

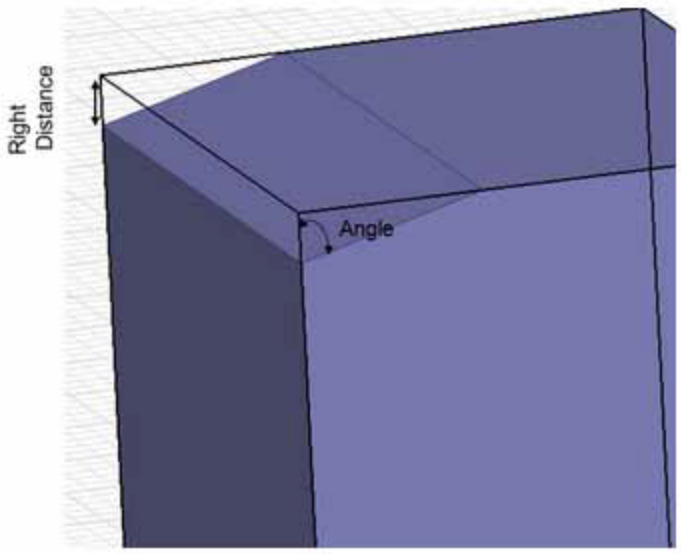


Example 2: Left Distance-Angle



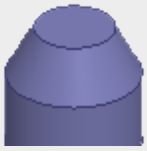
Example 3: Right Distance-Angle





Note:

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.

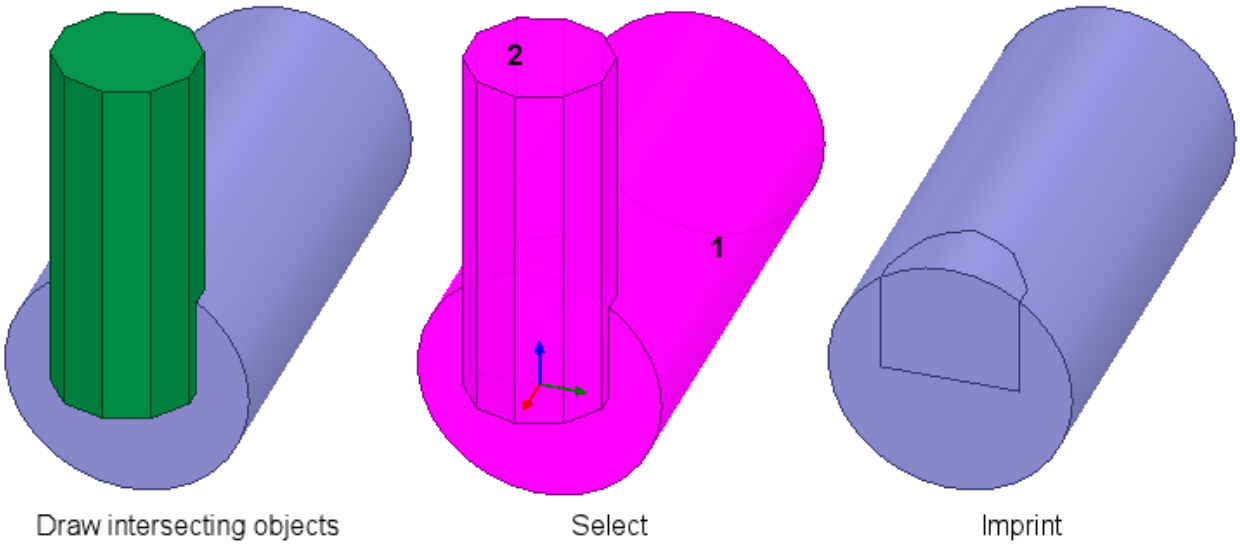
3. Click **OK** to apply the chamfer to the selected edges.

Imprinting an Object

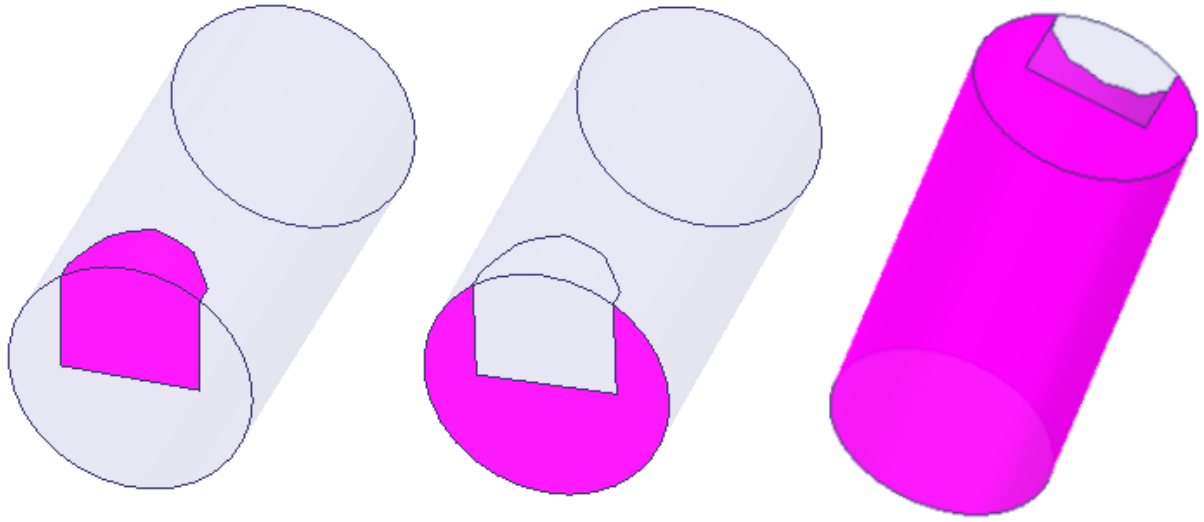
The **Boolean** > **Imprint** command lets you imprint the geometry of one object upon another. For example, you could draw a polyhedron intersecting a cylinder, and then imprint the intersecting lines on the cylinder. This process splits the larger face into two or more faces, with the intersecting lines representing the edges of the imprinted face. The original face is cut so that it no longer extends into the imprinted area.

Imprinting is often an essential capability for finite element analysis. It enables a boundary or excitation to be applied to the exposed area of a model face while excluding it from areas of intersection with other parts. One example is a thermal analysis where convection is applied to

the exposed surfaces of a heat sink. The area in contact with a semiconductor (the source or heat) is not exposed to ambient air and should not receive a convection boundary. In this case, you would imprint the semiconductor on the heat sink, producing a separate heat sink face in the contact area. This face is excluded from the selection set when applying the convection boundary.

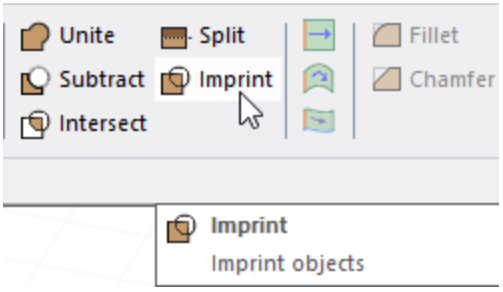


You can select the faces of the imprinted surface separately and assign properties as needed.

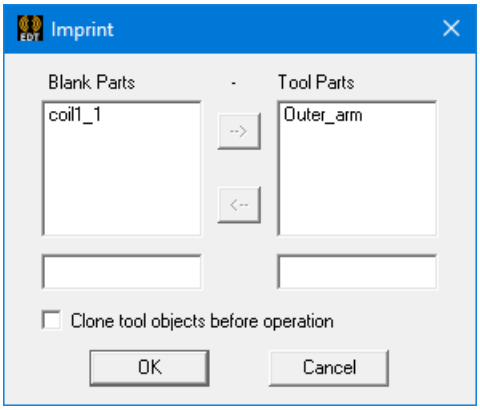


To imprint one object with another:

1. Select the intersecting objects.
2. From the menu bar, click **Modeler** > **Boolean** > **Imprint...** or, on the **Draw** ribbon tab, click the **Imprint** icon:

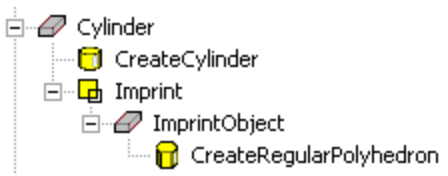


The **Imprint** dialog box displays in which you designate which objects are the *Blank Parts*, and which the *Tool Parts*. If necessary, you can select the objects in lists, and use the arrow keys to move them. If desired, you can clone the tool objects before the imprint operation.



3. Optionally, select **Clone the tool objects before operation** to retain the tool part. Otherwise, only its imprint on the blank part will remain.
4. Click **OK** to close the dialog box and perform the Boolean imprinting.

   The History Tree retains the ImprintObject command and the Create<object> command for the imprinted tool object:

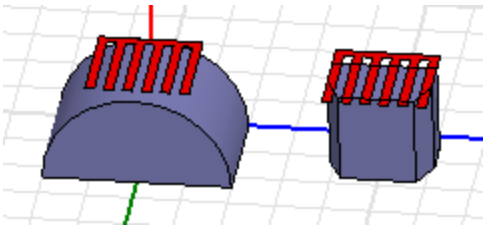


If you select the Imprint command in the History tree, you can suppress the command via the docked *Properties* window. If you select the *Create<object>* icon, you can edit the properties of that object. The changes you apply carry over to the imprinting operation, modifying the imprinted face accordingly.

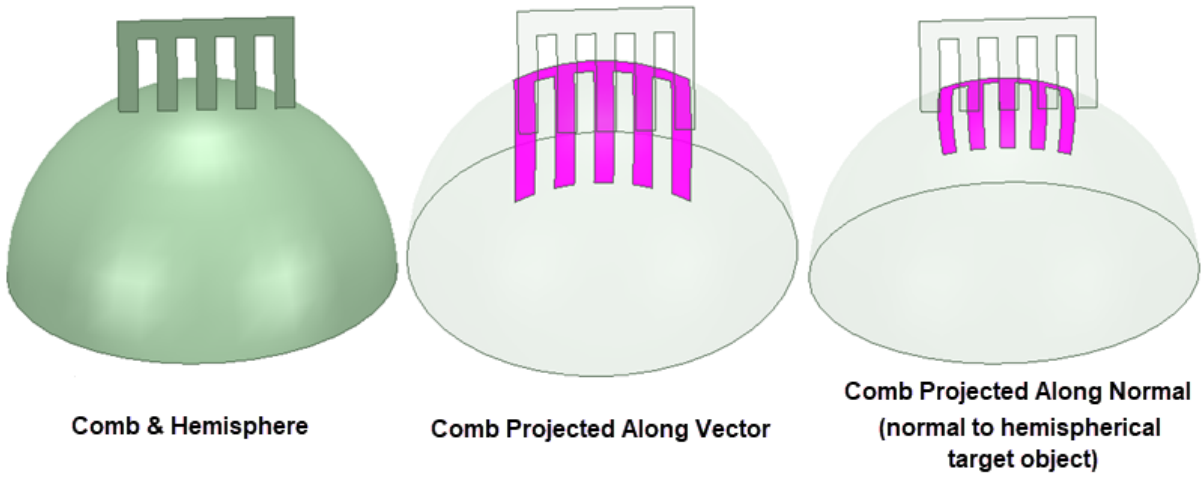
In the *3D Modeler* > *Operation* section of the *General Options*, you can also enable the **Automatically imprint wrapped sheets** option to automatically perform an *Imprint* operation command after executing the **Modeler** > **Surface** > **Wrap** command.

Imprint Projection Commands

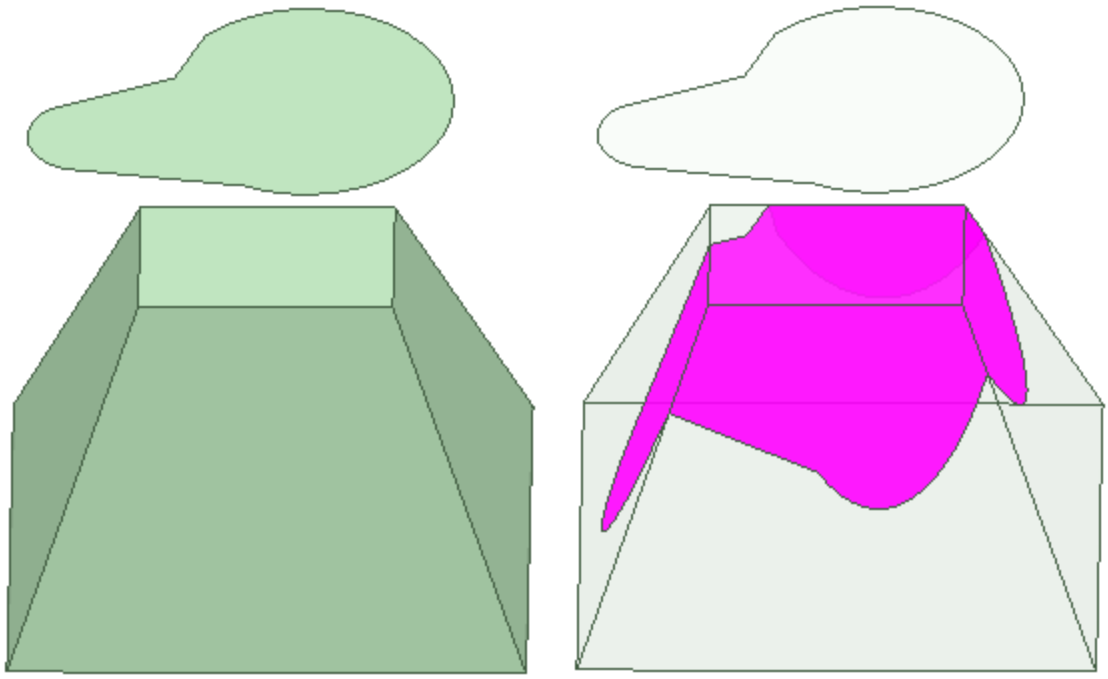
The **Boolean** > **Imprint Projection** commands lets you project the form of a sheet object onto the face or faces of another object (either solid or sheet). The receiving object faces can be flat, curved, or faceted.



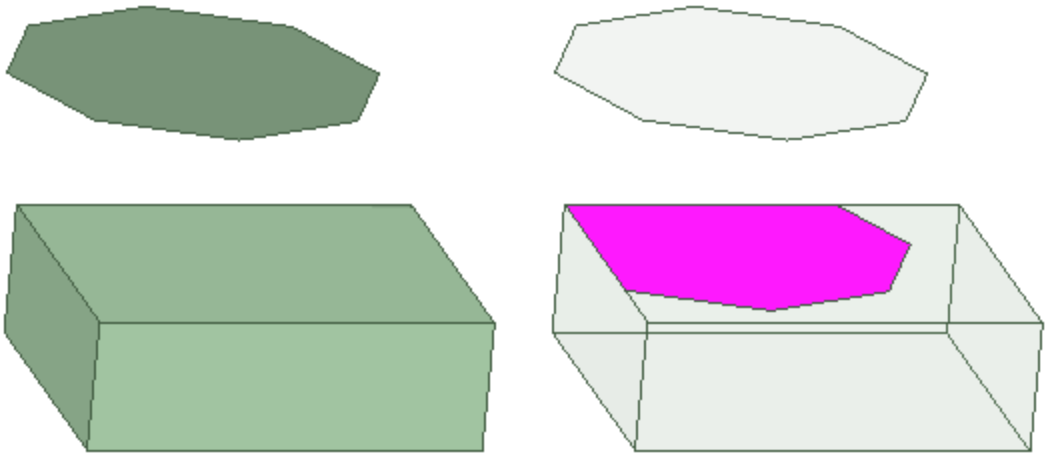
If the receiving surface is curved or inclined, and you choose to project *Along Normal* direction, the dimensions of the projection are affected. The reason is that the projection direction is normal to the curved or inclined face of the receiving object and not normal to the object being projected.



The projected shape will wrap around the target object, from the face nearest to the center of the imprinting area and extending to the adjacent target object faces:



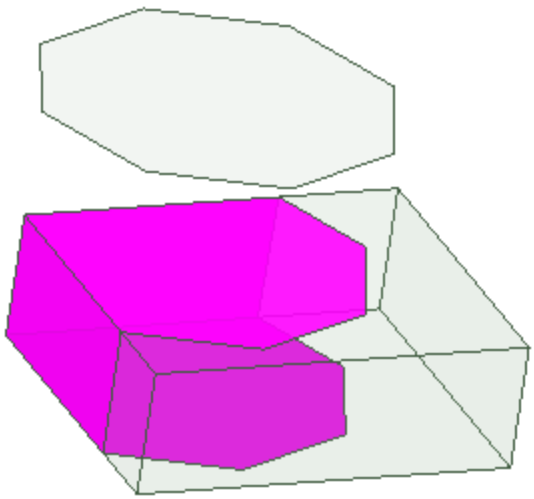
If the projection of the imprinting object extends beyond the extents of the receiving object, the imprinted shape is truncated:



You can select the imprinted faces on the target object separately, and perform subsequent operations (such as creating sheet objects from the faces, assigning boundaries or excitations, or selecting for field overlays).

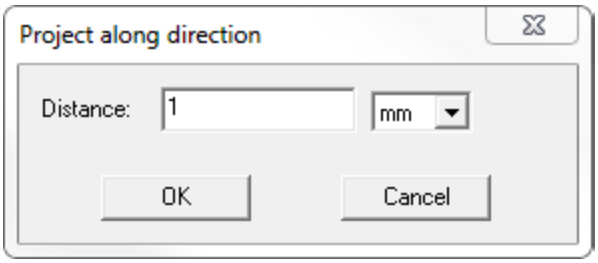
To perform an Imprint Projection:

1. If you want to preserve the tool object that will be imprinted on the target object, use one of the following methods:
   - Copy the tool object to the clipboard so that you can paste it back into the model after the imprinting operation.
   - In the *3D Modeler > Operation* section of the *General Options*, select the **Clone tool objects before projecting** option.
2. Select the projecting and target *objects*.
3. From the menu bar, click **Modeler** > **Boolean** > **Imprint Projection** > **Along Normal** or **Modeler** > **Boolean** > **Imprint Projection** > **Along Direction...**
   - If you select **Along Normal**, the projection occurs along the direction normal to the target faces and the operation is completed immediately. Imprinting occurs only on target faces visible from the imprinting object's point of view.
   - If you select **Along Direction**, you must specify two points that define the imprinting vector direction (step 3 below). Once you have done this, you see a dialog box for specifying the projection **Distance**. This option allows you to limit the projection distance so that only near faces are affected. Unlike the *Along Normal* option, the *Along Direction* option can imprint onto the target object faces that are not visible from the imprinting object's point of view (that is, those on the opposite side of the target object):



4. If performing an imprint projection operation using the **Along Direction** option:
   a. Specify the imprinting vector direction. You can click in the Modeler window or use the coordinate entry text boxes in the Status Bar.

      The *Project along direction* dialog box appears:



b. Specify the maximum projection **Distance** and select the length unit from the drop-down menu.
c. Click **OK**.

   The dialog box closes and the Boolean imprinting operation is performed.

After you perform the imprinting, the History Tree shows the ImprintProjection 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 docked *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 operation.

Purging Object History

Each object is a sequence of modeler-based operations. The history for each object is shown under its name in the History 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 wish 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. From the menu bar, 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

Generate History

Generate History to Reproduce Portions of Model

If a primitive object (cylinder, cone, box, circle, 3D polygon, etc.), polyline object (line, spline, or arc), circle, or ellipse is imported, or if the history was previously purged, you can select the

object and use the **Generate History** command to reproduce, in the History Tree, the primitive object definition or the individual line segments used to create a polyline. *Generated History* uses the original coordinate system in which the objects were created. *Generate History* also retains boundary conditions.

To reproduce the line segments in the model history tree:

1. Select the primitive or polyline object.
2. From the menu bar, click **Modeler> Generate History**.

Selecting Items in the 3D 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 Submodels
- Select Objects
- Select Faces
- Select Edges
- Select Vertices
- Select Multi
- Coordinates in the drawing space
- Select By Area including optional filters
- Select by History Tree Group
- Select by Submodel
- Select by intersection error message
- Selecting the Face or Object Behind

Selection menu for:

The **Edit>Select Objects** menu and **Modeler** window shortcut menu includes selection for:

- By Name
- By Material
- By Variable
- All Model Objects
- All Non-model Objects
- All Primitive Shapes
- All Non Primitive Shapes
- All Solids

- All Sheets
- All Lines

Selections enabled for features of selected Object:

Some **Edit>Extend Selection** menu commands are enabled when you select an appropriate object:

- All Object Faces
- All Object Edges
- All Face Edges
- Faces on Plane
- Select Connected Vertices
- Select Connected Edges
- Select Connected Faces
- Select Edge Chain
- Select Face Chain
- Select Sheet Edges

History Tree Icon Menu Selection

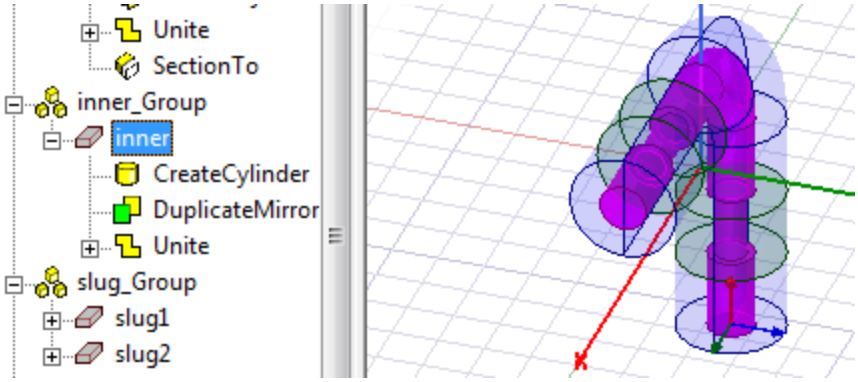
You can also right-click **History** tree icons to **Select All** for ungrouped:

- Model
- Group
- Solids
- Selected Material, for example copper or air.
- Sheets
- Lines

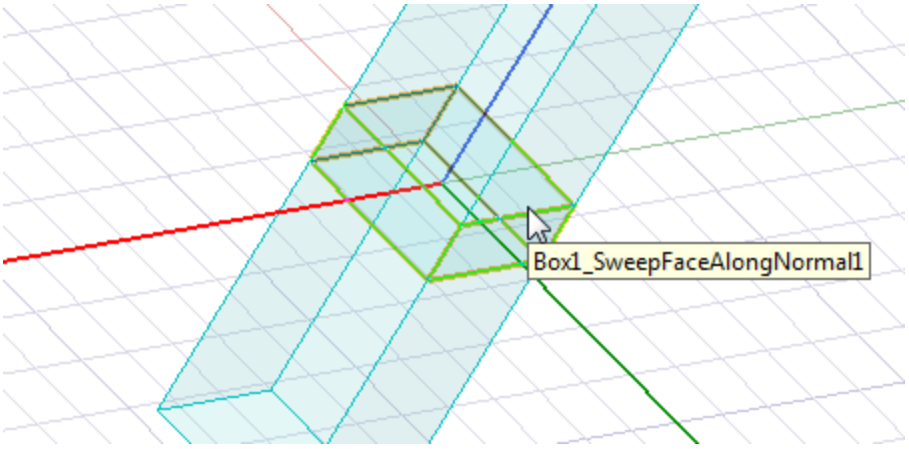
> **Note:**
>
> If selected objects do not display correctly, for some graphics card, you can improve performance by changing settings in the *NVIDIA Control Panel* application. In the *Select a Task* tree, navigate to **3D Settings> Manage 3D Settings**. Then, under the **Global Settings** tab, choose **Workstation App - Dynamic Streaming** from the **Global presets** drop-down menu.
>
> For more information about graphics cards, navigate to the appropriate Windows or Linux Ansys Installation Guide and search for **OpenGL**.

Selecting Objects

By default, the modeler is in *object selection mode*. Simply click an object in the view window or an object name in the history tree and it will be selected. All other objects become relatively transparent. Selected objects become the color specified under the **Display** tab of the **Modeler Options** dialog box.



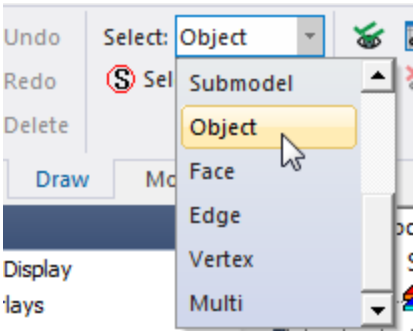
When the mouse hovers over an object in the view window, that object is highlighted, which indicates that it will be selected when you click. As you hover the cursor over an entity, tooltips indicate its type/ID (object name in the case of objects, Face_id in the case of faces, and so on). This feature helps you distinguish between face-of-sheet-object pick versus sheet-object pick.



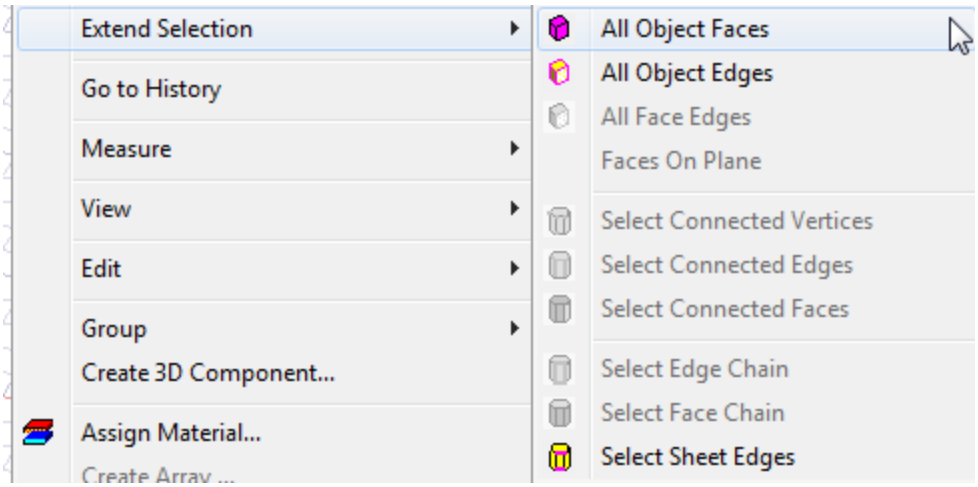
If the modeler is not currently in object selection mode, you can switch to it using one of the following methods:

- Press the **O** shortcut key while the Modeler window is active.
- Right-click in the Modeler window and click **Selection Mode**> **Select Objects**.
- From the menu bar, click **Edit**> **Selection Mode**> **Objects**.

- On the **Draw** or **Model** ribbon tabs, choose **Object** from the **Select** drop-down menu:



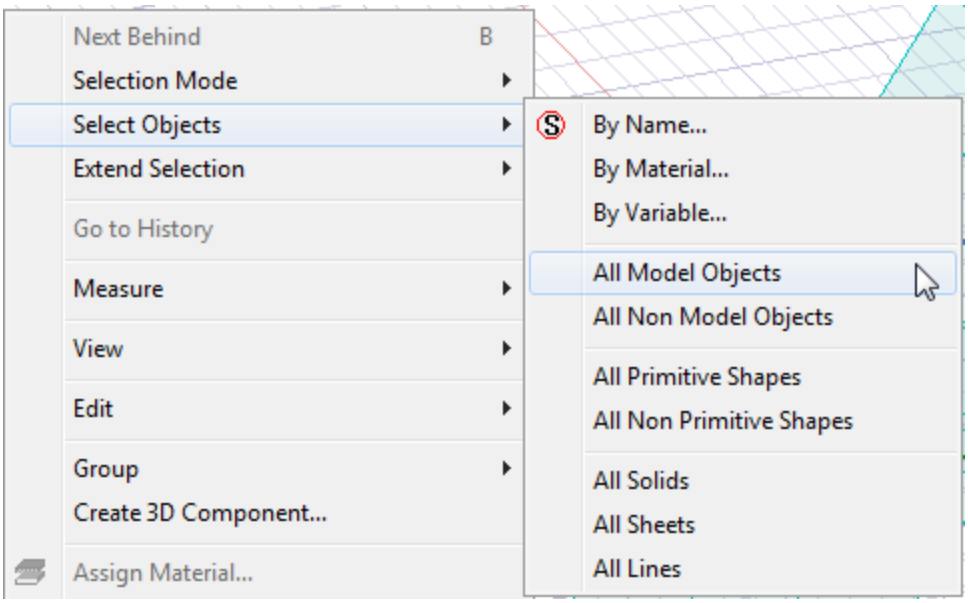
When you've selected an object, the menu bar's **Edit**> **Extend Selection** submenu and the **Extend Selection** submenu in the Modeler window's shortcut menu show the following options:



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 **O** shortcut key. You can always select objects in the History Tree.
2. Select several objects in one of the following ways:
   - Hold down **Ctrl** and click the objects in the Modeler 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**, which is an automatically created list of all model objects.

- From the menu bar, click **Edit> Select All** to select all objects that were drawn in the active Modeler window, including objects that are not currently visible.
- Press **Ctrl+A** or, from the menu bar, click **Edit> Select All Visible** to select all visible objects in the active Modeler window.
- From the menu bar, click **Edit> Select Objects>** or right-click in the Model window and go to the **Select Objects>** submenu. Then, choose a command from the submenu suitable for the objects you want to select:



- Right-click a History Tree icon for Model, Group, object, category (such as Solid or Sheet), or material to perform context-sensitive selections. See Select Objects in the History Tree.

To deselect all objects, do one of the following:

- From the menu bar, click **Edit> Deselect All**.
- Press **Ctrl+Shift+A**.

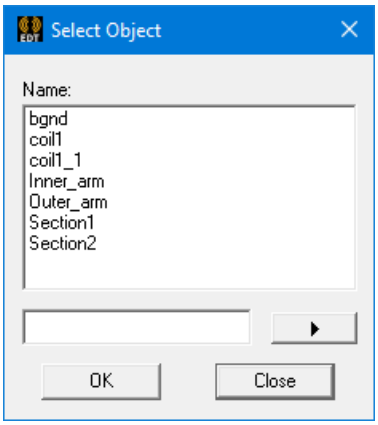
Selecting Objects by Name or Material

1. Make sure that the modeler is in the *Object* selection mode by pressing the **O** shortcut key or by choosing **Object** from the **Select** drop-down menu on the **Draw** or **Model** ribbon tab. Then, proceed to one of the following two procedures.

Selection by Object Name

2. Access the *Select Object* dialog box using one of the following three methods:
   - From the menu bar, click **Edit** > **Select Objects** > **By Name**.
   - On the **Draw** or **Model** ribbon tab, click **Select by Name**.
   - Right-click in the *Modeler* window and choose **Select Objects** > **By Name** from the shortcut menu.

   The *Select Object* dialog box appears, populated with a list of the objects in the model, by name:



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 in the empty text box, click the arrow to the right of the text box, and choose **Select** or **Deselect**.

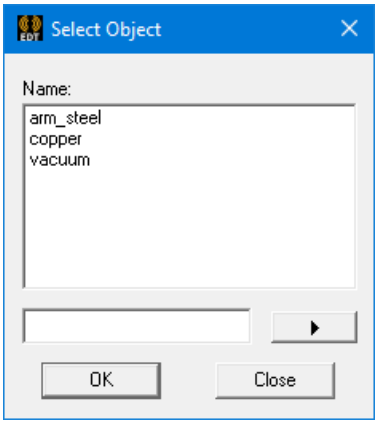
4. Click **OK**.

   The object or objects specified are selected.

Selection by Object Material

2. Access the *Select Object* dialog box using one of the following two methods:
   - From the menu bar, click **Edit** > **Select Objects** > **By Material**.
   - Right-click in the *Modeler* window and choose **Select Objects** > **By Material** from the shortcut menu.

   The *Select Object* dialog box appears, populated with a list of the materials comprising the model objects:



3. In the *Name* list, click the material of the object or objects you want to select. Use the **Ctrl** key to select more than one material.

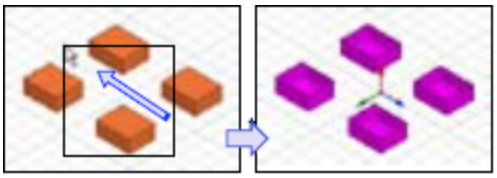
   Alternatively, type the name of a material in the empty text box, click the arrow to the right of the text box, and choose **Select** or **Deselect**.

4. Click **OK**.

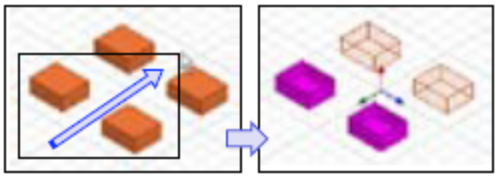
Select By Area

You can select items by area in the Modeler window by clicking and dragging a rubber band (selection box) around objects. Rubber band selection works differently depending on the direction you drag the mouse:

- **From right-to-left:** Selects all items that are wholly or partly enclosed within the rubber band.



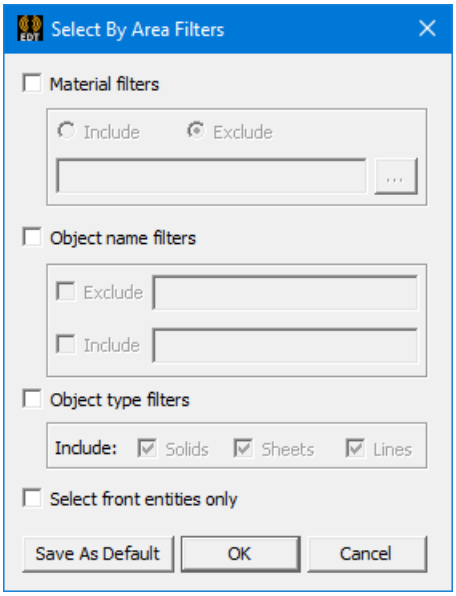
- **From left-to-right:** Selects only the items that are wholly enclosed within the rubber band.



Area selection works with Object, Face, Edge, and Vertex selection modes, but not for the Multi selection mode.

You can control which objects to include or exclude from area selections based on material, object name, or object type. To do this:

1. From the menu bar, click **Edit** > **Select By Area Filter** to display the following dialog box:



2. Check **Material filters** to enable the **Include** and **Exclude** radio buttons. Use the text box to specify filters by name, or use the ellipsis [...] button to display the *Select Definition* dialog box, from which you can browse and select materials.
3. Check **Object name filters** to enable the **Exclude** and **Include** check boxes and associated text boxes in which you can specify object names.
4. Check **Object type filters** to enable the *Include* check boxes for including **Solids, Sheets,** and/or **Lines**.
5. Check **Select front entities only** to prevent entities in the background (that is, those hidden behind other entities) from being selected.
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 box.

   Now when you 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.

Setting the Default Color and Transparency of Selected Objects

To set the color of objects when they are selected:

1. From the menu bar, click **Tools> Options> General Options** or, on the **Draw** ribbon tab, click  **General Options**.

   The *Options* dialog box appears.

2. In the tree at the left side of the dialog box, select the **3D Modeler> Display> Rendering** group of settings.
3. In the *Rendering Defaults* section, choose **Select** from the **Default color** pull-down menu.
4. Click the color button to the right of *Select*.

   The *Color* palette appears.

5. Select a color and then click **OK**.

   Any objects you select after this point will temporarily become the default color you selected.

By default, the modeler shows selected objects as nearly opaque and shows non-selected objects as nearly transparent. This features helps you to easily distinguish between selected and non-selected objects and to see selected objects on the back side of the model.

To set the transparency of selected and non-selected objects:

1. From the menu bar, click **View> Options**.

   The *3D UI Options* dialog box appears. The *When there is a selection* region contains check boxes for setting the transparency for selected and non-selected objects.

2. Ensure that the check box is selected for any value you want to change.

   Each check box enables the associated value box. 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**.

Setting the Default Color of Highlighted Objects

1. From the menu bar, click **Tools> Options> General Options** or, on the **Draw** ribbon tab, click  **General Options**.
2. In the tree at the left side of the dialog box, select the **3D Modeler> Display> Rendering** group of settings.

   The *Options* dialog box appears.

3. In the *Rendering Defaults* section, choose **Highlight** from the **Default color** pull-down menu.
4. Click the color button to the right of *Highlight*.

   The *Color* palette appears.

5. Select a color and then 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 group multiple objects for convenient selection and processing. 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 **O** shortcut key. 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 appears in the History Tree under **Lists**. Selecting an object list displays the properties of that list in the docked *Properties* window. One of the properties is a list of objects contained in the list.

   To rename the Object list, edit its Name in the docked *Properties* window. Object lists are sorted in alphanumeric order.

The object list is treated as one volume when you plot and perform fields calculations. It will be listed in the *Geometry* window of the Fields Calculator when you select **Volume**.

There is an automatically created list called AllObjects. Select this list selects all objects in the model. 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.

- Line objects in the list do not affect where the fields are saved. If all objects in the list are lines, fields will be saved on all objects.
- To post processing on line object when saving fields on object/face list, include both the line and its containing object/face in the list.

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 will be plotted only at the intersection of the plane and the objects in the list.

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 **O** shortcut key. 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 box with the existing object lists is displayed. (They appear in the History Tree under **Lists**.)

4. Select the list to which you want to assign the selected object(s) and click **OK**.

   The object is reassigned to the selected list, replacing previous list members. The *Objects* property of the List shows the objects contained in the list, with each name separated by a comma.

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**.

Using or Viewing List Objects

To view the objects included in an Object list:

1. In the History Tree, expand the **Lists** branch.
2. Right-click the ***list*** you want to select, and click **Select Assignment**.

   The objects that are included in that list are selected in the Modeler window, and the attributes appear in the docked *Properties* window.

To view the properties of the object list (including a list of the objects included):

1. In the History Tree, expand the **Lists** branch.
2. Under *Lists*, right-click the ***list*** you want to view and click **Properties**.

   The *Properties* dialog box appears. The objects included in the list appear as comma-separated names in the *Objects* row.

3. Click **OK** or **Cancel** to close the *Properties* dialog box.

To use an object from a list in another operation:

1. In the History Tree, expand the **Lists** branch.
2. Right-click the ***list*** you want to select and click **Select Assignment**.

   The objects that are included in that list are selected in the Modeler window, and the attributes appear in the docked *Properties* window.

3. Holding **Ctrl**, select any additional objects you might want to use in the operation.
4. Complete the operation.

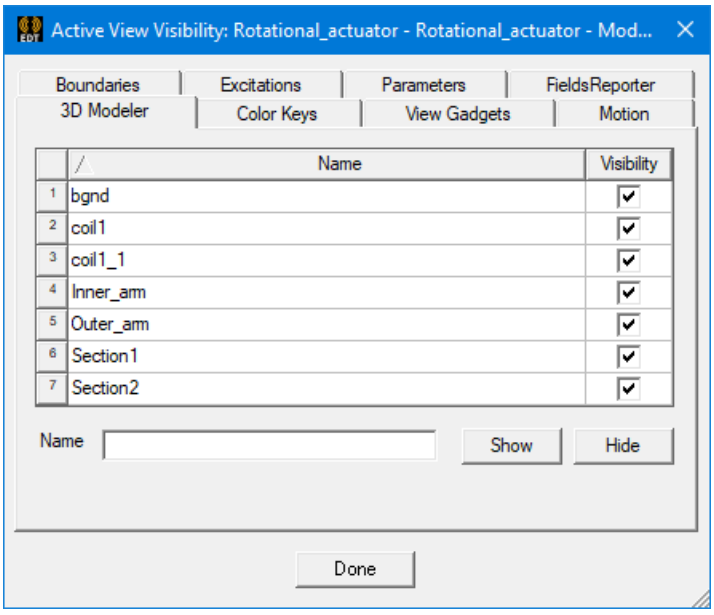
   For example, you could select an object list and an additional object, and then execute one of the Boolean commands (such as *Unite* or *Subtract*).

Selecting Objects and Surfaces That Lie Inside Other Objects

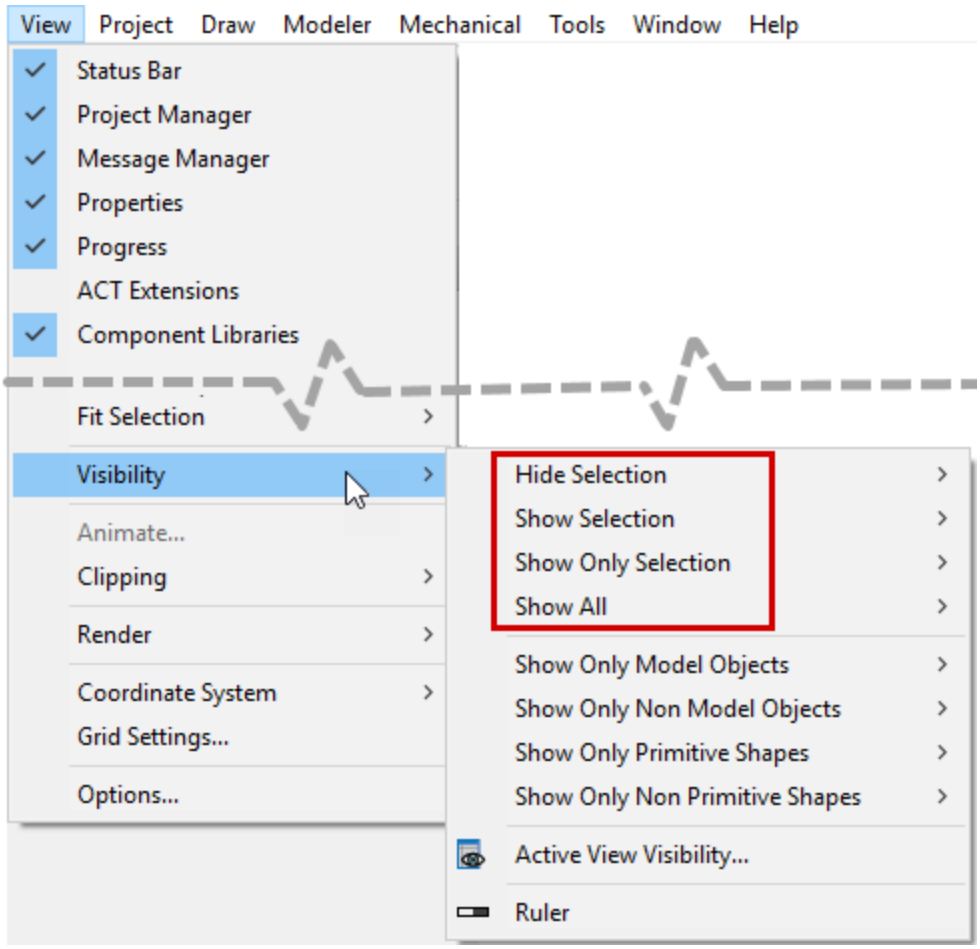
To select objects and faces that lie inside other objects (such as an object that lies within an air box, a conductive shield, or the background object), do one of the following:

- From the menu bar, **View** > **Visibility** > **Active View Visibility...** to access the *Active View Visibility* dialog box. Then, under the *3D Modeler* tab, use the check boxes in the **Visibility** column to control which objects are visible and which are hidden.

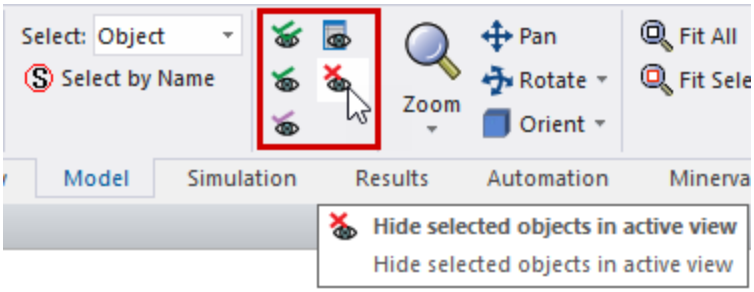
  This method, and the method that follows, is useful when you want to select objects using the mouse, since the mouse cannot select invisible objects.



- Make the objects on the outside of the model invisible using the menu bar's other **View** > **Visibility** > commands:



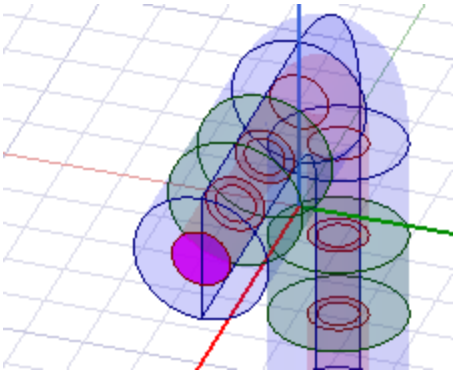
or the same commands on the **View, Draw,** or **Model** ribbon tabs:



- Use the **Select by Name** command to select Objects or Faces inside the model.
- Use the **Next Behind** command in the *Edit* menu or Modeler shortcut menu, or use the **B** shortcut key. This command selects the object that lies behind the one you initially selected. This command does nothing if no objects have previously been selected or if no object is behind the one you initially selected at the location you clicked.

Selecting Faces

If the modeler is in face selection mode, click an object face in the view window to select it. The selected face is rendered in the section color (magenta by default) and is relatively opaque by default. All other objects become relatively transparent.

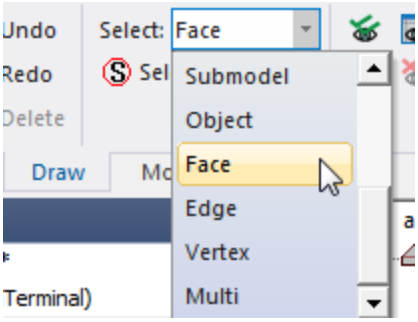


When the mouse hovers over a face in the view window, that face is highlighted, which indicates that it will be selected when you click. Selected faces become the color specified under the *3D Modeler> Display> Rendering* section the *General Options*.

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 Select Face dialog box.

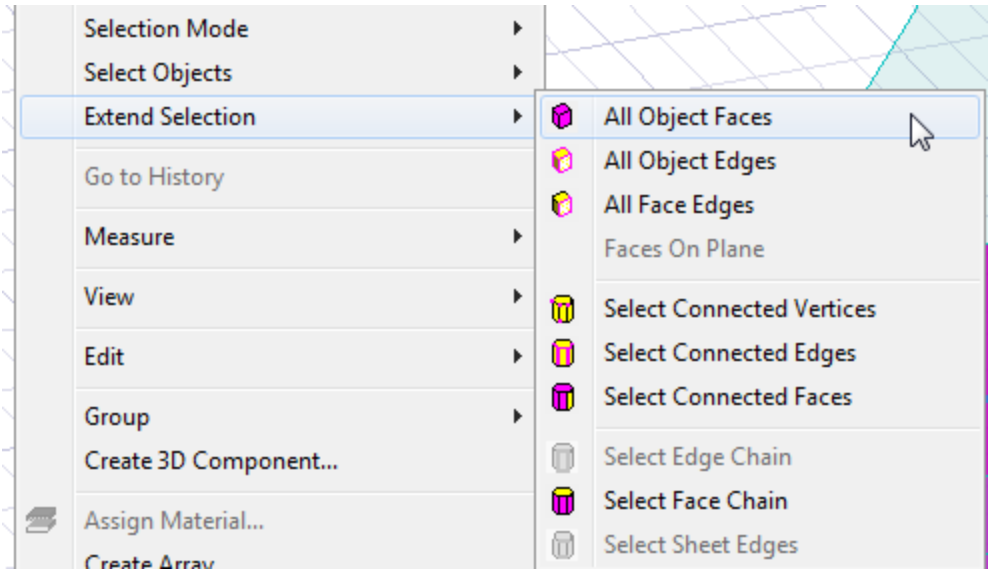
Switch to face selection mode using one of the following methods:

- From the menu bar, click **Edit> Selection Mode> Select Faces**
- Right-click in the Modeler window and choose **Selection Mode> Select Faces** from the shortcut menu.
- With the Modeler window active, press the **F** shortcut key.
- On the **Draw** ribbon tab, choose **Faces** from the **Select** drop-down menu.



You can also select faces in the Select Multi mode.

When you've selected a face, the menu bar's **Edit> Extend Selection** submenu and the **Extend Selection** submenu in the Modeler window's shortcut menu show the following options:



Various tools are available for face selection, as described in the *Related Topics* listed below.

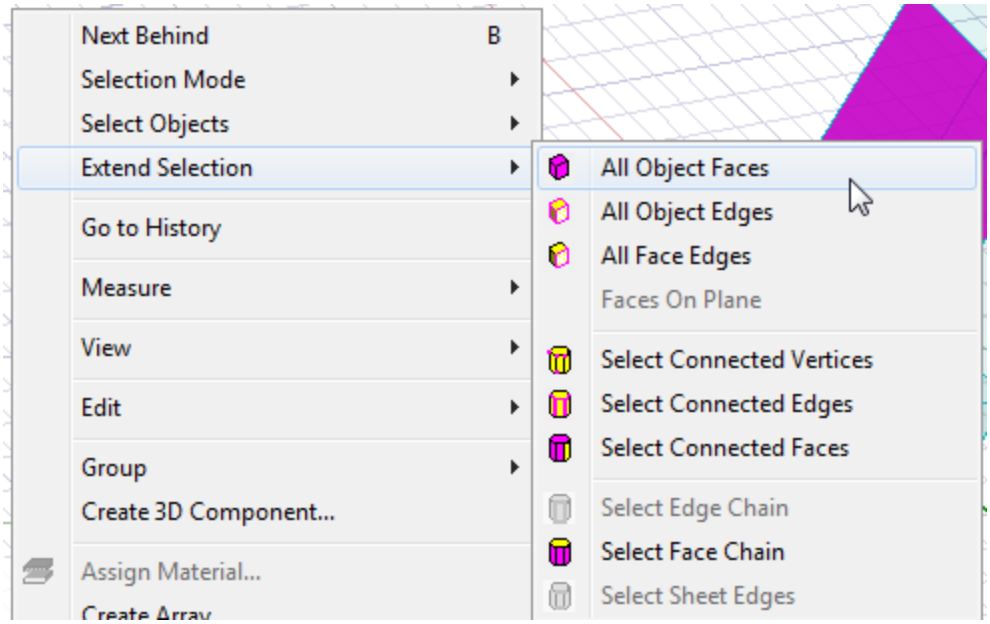
Selecting All Faces of an Object

1. With the Modeler window active, switch to the face selection mode by pressing the **F** shortcut key.
2. Click a face on the object of interest to select it.

   You can select a face on each of two or more objects if desired.

3. Use one of the following methods of selecting all faces of the objects of interest:
   - From the menu bar, click **Edit> Extend Selection> All Object Faces.**
   - Right-click in the Modeler window and choose **Extend Selection> All Object**

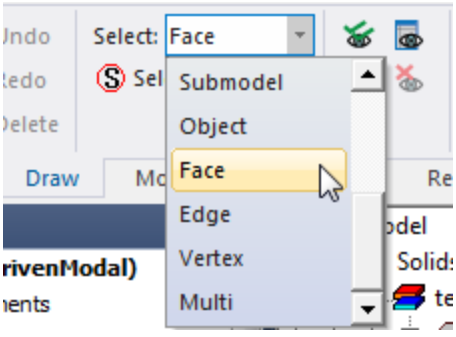
Faces from the shortcut menu:



The selection is extended to include all faces of each object that had at least one face selected originally.

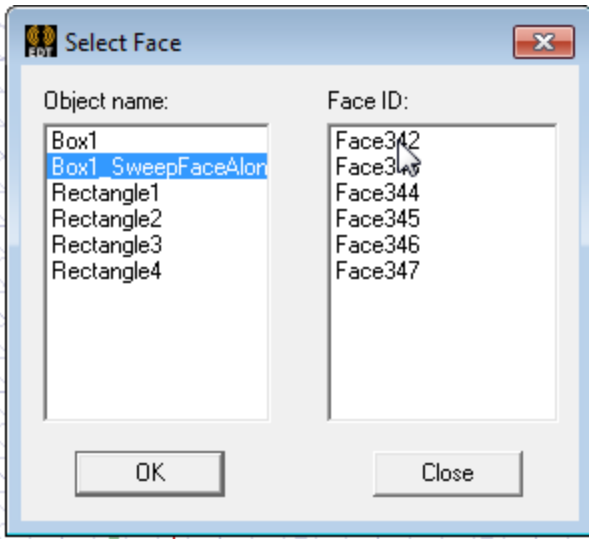
Selecting Faces by Name

1. Make sure that the modeler is in face selection mode using one of the following methods:
   - With the Modeler window active, press the **F** shortcut key.
   - From the menu bar, click **Edit> Selection Mode> Faces**
   - On the **Draw** or **Model** ribbon tabs, select **Face** from the **Select** drop-down menu.



2. From the menu bar, click **Edit> Select Objects> By Name** or, on the **Draw** or **Model** ribbon tabs, click **Select by Name**.

   The *Select Face* dialog box appears:



3. In the *Object name* box, click the name of the object with the face or faces you want to select.

   The object's faces are listed in the *Face ID* box.

4. Click the ***face*** you want to select in the *Face ID* column. Use **Ctrl+click** to select more than one face or **Shift+click** to select a range of faces.

   The selected faces are highlighted in the list and in the Modeler window.

5. Click **OK**.

Selecting Faces by Plane

To select a face that is aligned with a global plane, use one of the following two methods.

Method 1:

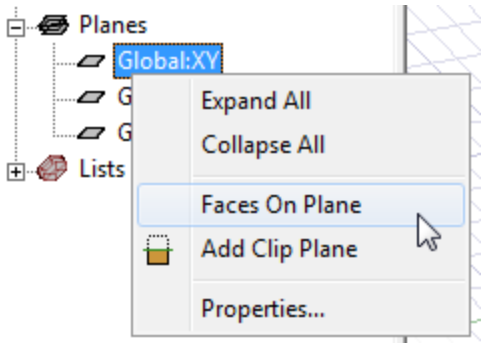
1. Make sure that the modeler is in face selection mode by pressing the **F** shortcut key.
2. In the History Tree, expand the **Planes** branch and select a plane to display (a global plane, a user-defined plane, or a plane associated with a relative coordinate system).
3. From the menu bar, click **Edit> Extend Selection> Faces on Plane** or right-click in the Modeler window and choose **Extend Selection> Faces on Plane**.

   All faces that lie on the chosen plane are selected.

Method 2:

1. In the History Tree, expand the **Planes** branch.
2. Right-click one of the listed planes (a global plane, a user-defined plane, or a plane associated with a relative coordinate system).

3. On the shortcut menu, click **Faces on Plane**.



All faces that lie on the chosen plane are selected.

Creating a Face List

Create a face list when you when you want to group multiple faces for convenient selection and processing. Creating a face list is a convenient way to identify and select a specific set of surfaces for field plots, calculations, boundaries or excitations. 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 **F** shortcut key.
2. Select the object faces you want to include in the face list.
3. From the menu bar, 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.

4. Optionally, 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 change the **Name** value in the docked *Properties* window.

   If necessary, the lists in the History Tree are resorted to restore the proper alphanumeric order.

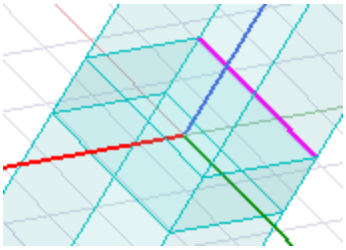
The face list will be treated as one selection of surfaces when you are plotting and performing fields calculations. The face list will be listed in the *Geometry* window of the *Fields Calculator*, when you select the *Surface* option.

- Line objects in the list do not affect where the fields are saved. If all objects in the list are lines, fields will be saved on all objects.
- To post processing on line object when saving fields on object/face list, include both the line and its containing object/face in the list.

If you right-click an existing face list and choose **Select Assignment** from the shortcut menu, you can make boundary assignments or execute the **Move Faces** geometry operations for all faces in the list.

Selecting Edges

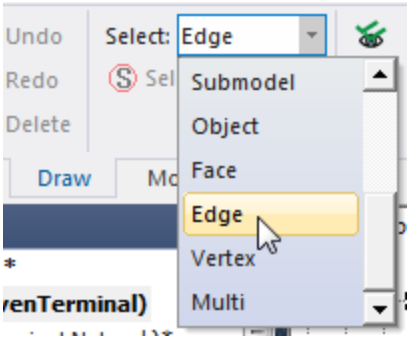
If the modeler is in edge selection mode, simply click an object's edge in the view window and it will be selected. The edge is rendered as nearly opaque using the default *Select* color. All other objects become relatively transparent.



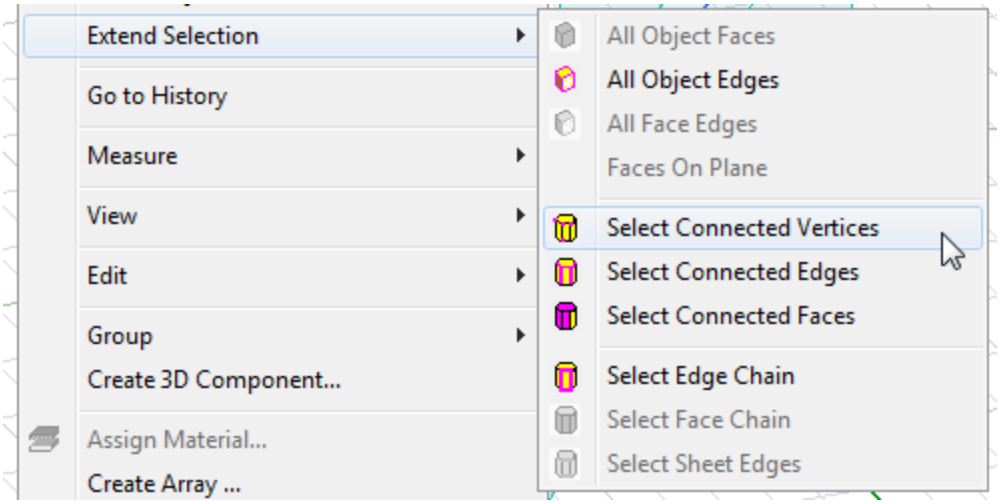
When the mouse hovers over an edge in the view window, that edge is highlighted, which indicates that it will be selected when you click.

Switch to edge selection mode using one of the following methods:

- From the menu bar, click **Edit> Selection Mode> Edges**.
- Right-click in the Modeler window and click **Selection Mode> Edges**.
- With the Modeler window active, press the **E** shortcut key.
- On the **Draw** or **Model** ribbon tab, choose **Edge** from the **Select** drop-down menu.



Selecting an edge enables the following commands in the menu bar's **Edit> Extend Selection>...**submenu and the **Extend Selection>...** submenu that appears when you right-click in the Modeler window.



You can use these commands to modify the current selection.

- All Object Edges
- Select Connected Vertices
- Select Connected Edges
- Select Connected Faces
- Select Edge Chain

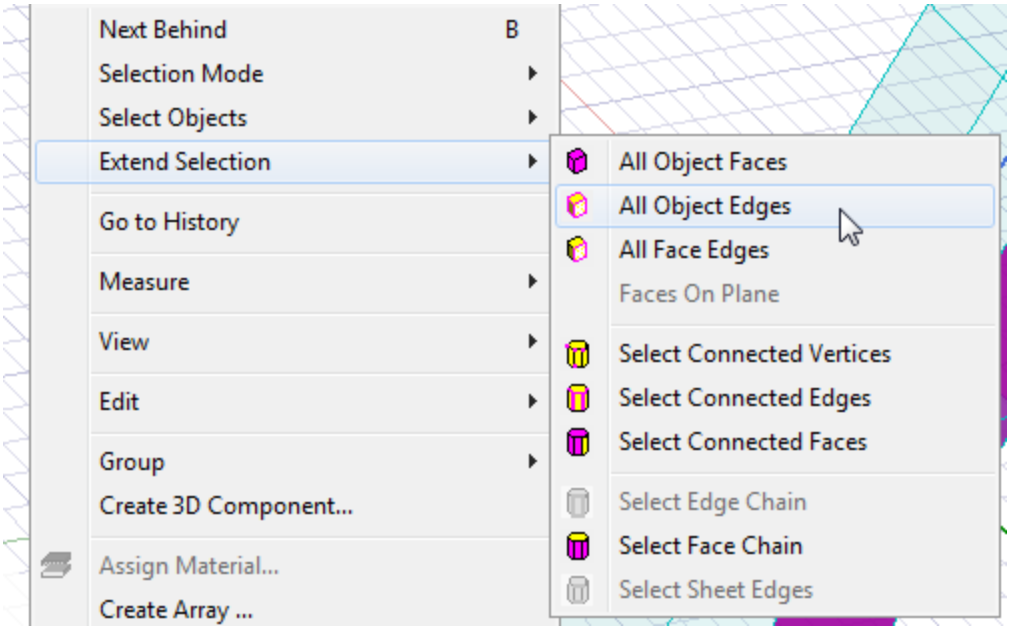
You can also select multiple edges by holding the **Ctrl** key as you click additional edges.

Selecting All Edges of an Object or Face

To select all edges of an object or face:

1. Select a single edge or face that belongs to the object for which you wish to select all edges.

   You may also select the object itself. The type of entity you initially choose affects which selection tools will be available in the next step:



2. If you selected an **object** or an **edge** in step 1, the following *All Edge* command is available:
   - From the menu bar click **Edit> Extend Selection> All Object Edges** or right-click in the Modeler window and choose **Extend Selection> All Object Edges** from the shortcut menu.

     All edges of the object are selected.
3. If you selected a **face** in step 1, the following two *All Edge* commands are available:
   - From the menu bar click **Edit> Extend Selection> All Face Edges** or right-click in the Modeler window and choose **Extend Selection> All Face Edges** from the shortcut menu.

     All edges belonging to the face are selected.
   - From the menu bar click **Edit> Extend Selection> All Object Edges** or right-click in the Modeler window and choose **Extend Selection>All Face Edges** from the shortcut menu.

     All edges of the object to which the face belongs are selected.

If you selected multiple objects, all edges of those objects are selected.

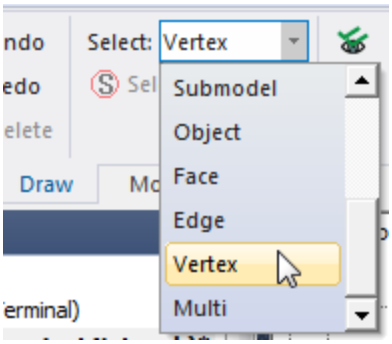
Selecting Vertices

If the modeler is in vertex selection mode, simply click an object's vertex in the view window and it will be selected. To select multiple vertices, hold the **Ctrl** key as you click the vertices.

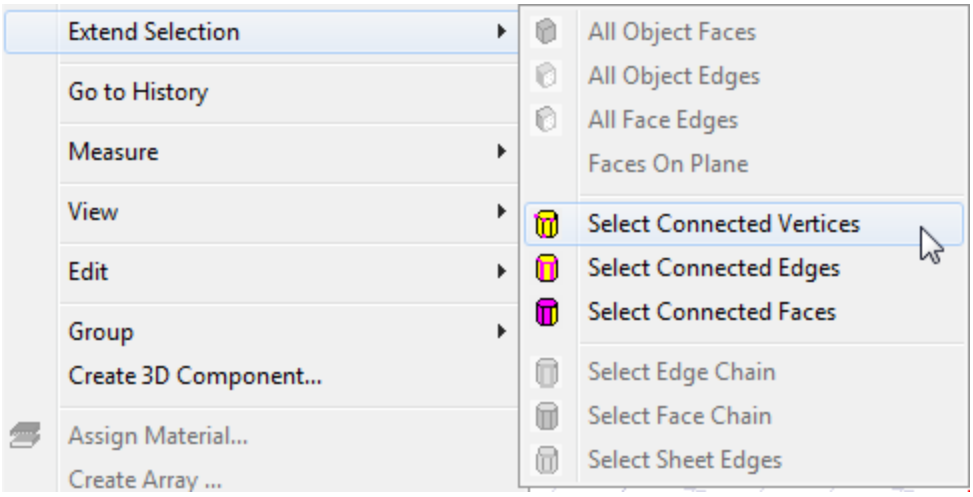
When the mouse hovers over a vertex in the view window, that vertex is highlighted, which indicates that it will be selected when you click. Selected vertices are rendered as using the default *Select* color. All other objects become relatively transparent.

Switch to vertex selection mode using one of the following methods:

- From the menu bar, click **Edit**> **Selection Mode**> **Vertices**.
- Right-click in the Modeler window and click **Selection Mode**> **Vertices**.
- With the Modeler window active, press the **V** shortcut key to begin the vertex selection mode.
- On the **Draw** or **Model** ribbon tabs, choose **Vertex** from the **Select**drop-down menu.



Selecting a vertex enables the following commands in the menu bar's **Edit> Extend Selection>...** submenu and the **Extend Selection>...** submenu that appears when you right-click in the Modeler window.



You can use these commands to modify the current selection:

- **Select Connected Vertices:** Selects the vertices of all edges that touch the initially selected vertices.

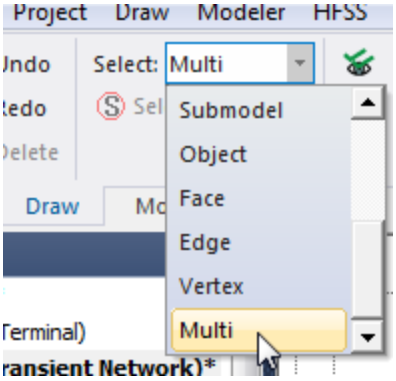
- **Select Connected Edges:** Selects all edges that touch the initially selected vertices.
- **Select Connected Faces:** selects all faces touching the initially selected vertices.

> **Note:**
>
> Multiple vertices can exist at the same coordinates, one vertex for each object that intersects at that point. There's no visual way to discern to which object a selected vertex belongs. If you click a vertex and use any of the commands listed above, only the entities that belong to the same object will be selected. To select the associated entities for all objects, start by box-selecting the vertices at the point of interest (clicking and dragging from left-to-right). This method selects all coincident vertices at the point. Then, use one of the above three commands.
>
> You can also pick a vertex and press **B** (*Next Behind*) to select a different vertex at the point of interest. Use the trial and error method to select the related vertices, edges, or faces of the desired object. For example, you may have to press *B* two or more times to select the vertex of the desired object.

Selecting Multi

The Select Multi mode permits you to select a mixture of objects, faces, vertices, or edges, depending on where you click. This mode can be very useful in conjunction with Measure Mode, for measuring the distances between different entities. Enter the *Multi* selection mode using one of the following methods:

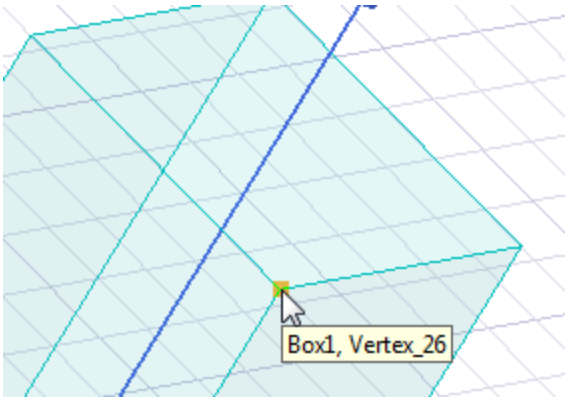
- With the Modeler window active, press the **M** shortcut key.
- Right-click in the Modeler window and choose **Selection Mode**> **Multi** from the shortcut menu.
- From the menu bar, click **Edit**> **Selection Mode**> **Multi**.
- On the **Draw** or **Model** ribbon tabs, choose **Multi** from the **Select** drop-down menu.



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 at least 10 pixels away from any vertex).
- To select an object, click little farther from any edge (between 10 and 20 pixels away).
- To select a face, click anywhere else on the interior of a face.

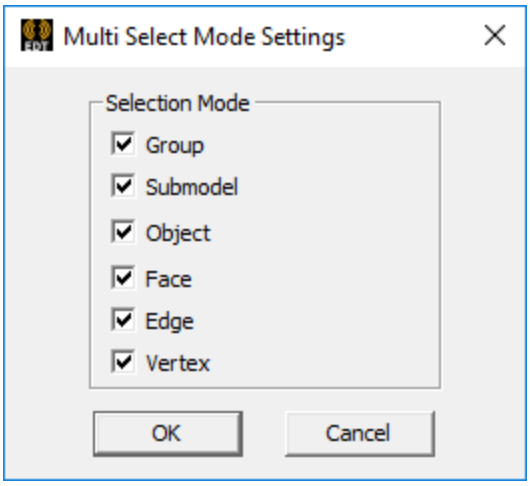
As you hover the cursor over an entity, a Tooltip indicates the type/ID of the entity (object name in the case of objects, Face_id in the case of faces, and so on). For example, this feature helps you to distinguish between the selection of the face of a sheet object versus a sheet object.



By holding down the **Ctrl** key, you can make multiple selections.

Controlling the Selection in Multi Mode

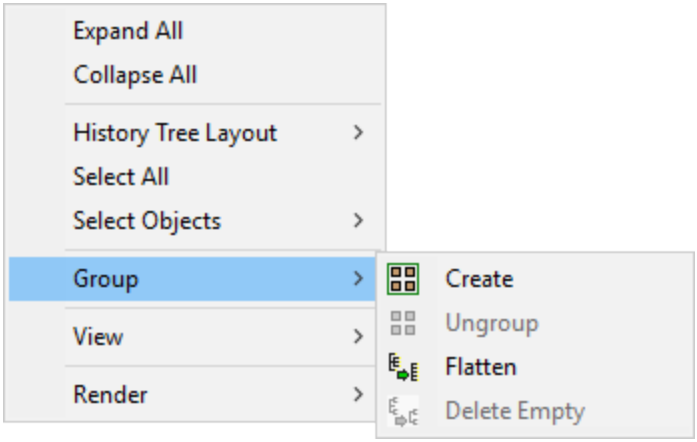
You can control the behavior of this selection mode. From the menu bar, click **Edit> Selection Mode> Multi Mode Settings**. This action displays a dialog box with check boxes for enabling the selection of specific types of entities (Group, Submodel, Object, Face, Edge, and Vertex):



Clear a box filters out the associated item, preventing that type of entity from being selected while the *Multi* selection mode is active.

Group Commands for Modeler Objects

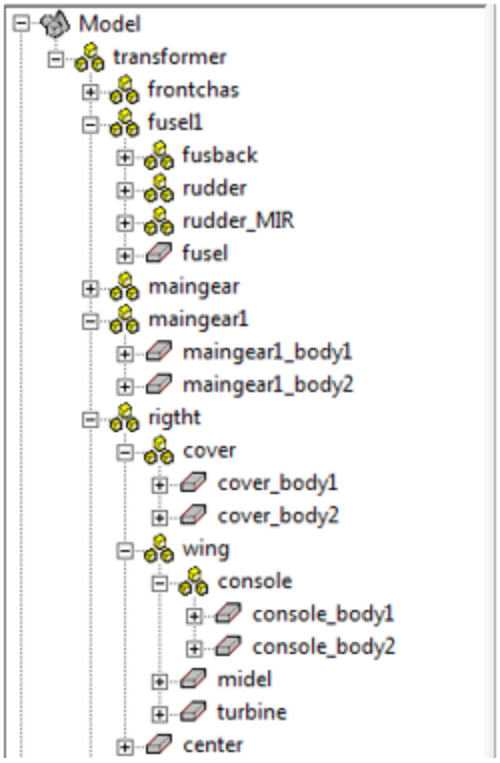
The 3D Modeler allows you to group objects in the History tree. Besides predefined folders under **History Tree Layout** for solids, sheets, and material assignments, a set of **Group** commands allows you to create groups, ungroup objects, flatten a selected group's hierarchy, and delete empty groups. Groups are not compatible with 3D components, although, you can organize 3D components by definition in the History tree.



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 history tree. They do not affect 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 following important distinctions 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. Delete of an object in one group could cause object in another group to be also deleted.

Accessing Group Commands

You can access Group commands in several ways: via **Modeler > Group**, via the right-click shortcut menu in the History tree, via the right-click shortcut menu in the Modeler window, and via the Group icons on the Model tab.

Group Command Descriptions

Group > **Create** works with or without a selection. If the History Tree Layout>Show Groups command is disabled, a dialog box reports that groups are currently hidden, and that if you OK to continue, groups will be shown. If there is no selection of objects for new group, an empty group is created under Model. If there are selections, 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 the 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** is enabled when you select a group. You can select multiple groups at different levels. All of the selected group's contents (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. If the History Tree Layout > Show Groups command is disabled, a dialog box reports that groups are currently hidden, and that if you OK to continue, groups will be shown.

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. If the History Tree Layout > Show Groups command is disabled, a dialog box reports that groups are currently hidden, and that if you click **OK** to continue, groups will be shown.

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 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

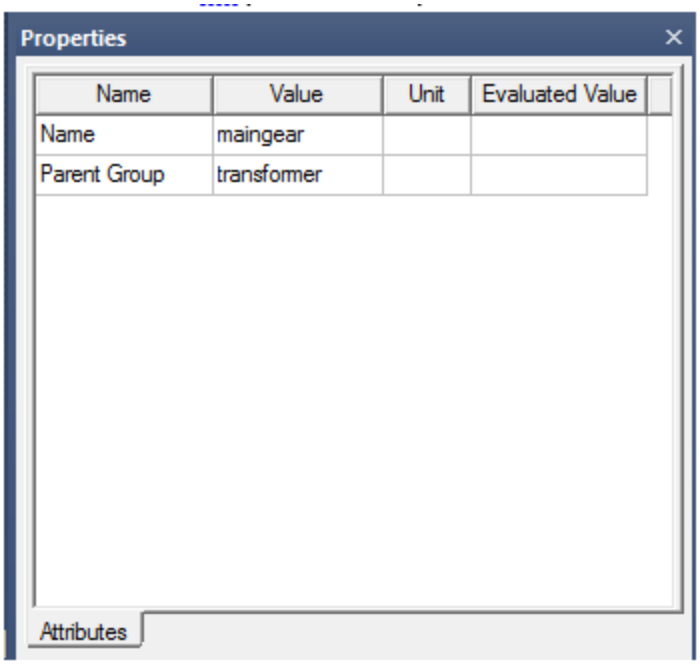
You can select groups to enable duplicate operations. Afterward, the History tree shows a duplicate operation under every object of that group. Newly created objects appear under a new group. There will be new group for every instance of duplicate.

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.

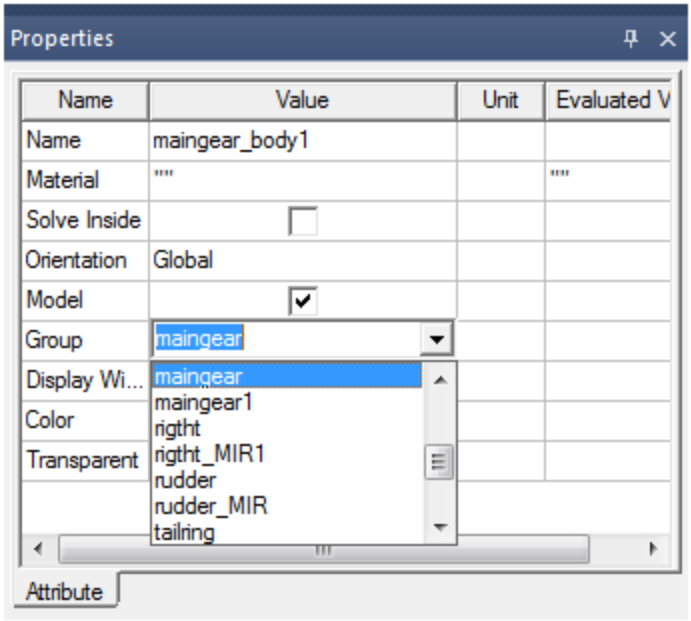
Group Properties

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

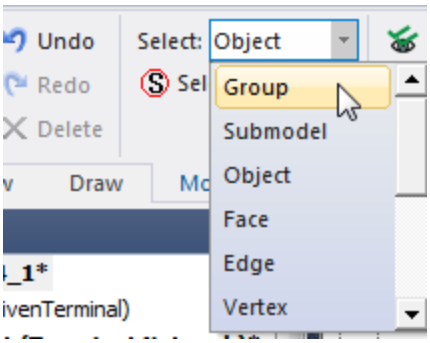
Selecting Groups and Submodels

With the *Group* selection mode, the parent group of selected object is selected. Selecting a 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.

The *Group* selection mode is only available when the model contains at least one defined group.

Use one of the following methods of setting the selection mode to *Group*:

- From the menu bar, click **Edit**> **Selection Mode**> **Groups**.
- With the Modeler window active, press the **G** shortcut key.
- Right-click in the Modeler window and choose **Selection Mode**> **Groups** from the shortcut menu.
- On the **Draw** or **Model** ribbon tabs, choose **Group** from the **Select** drop-down menu. You may have to scroll up to see this option.



If your design contains submodels, you can set the selection mode to *Submodel*:

- From the menu bar, click **Edit**> **Selection Mode**> **Submodels**.
- With the Modeler window active, press the **U** shortcut key.
- Right-click in the Modeler window and choose **Selection Mode**> **Submodels** from the shortcut menu.
- On the **Draw** or **Model** ribbon tabs, choose **Submodel** from the **Select** drop-down menu. You may have to scroll up to see this option.

Clearing a Selection

To clear an object, face, edge, or vertex selection, do one of the following:

- Click in an empty area of the Modeler window.
- To clear an object selection, click a point away from the object name in the History Tree.
- From the menu bar, click **Edit**> **Deselect All**.
- Press **Shift+Ctrl+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:

- From the menu bar, click **Edit> Next Behind**.
- Right-click in the Modeler window and choose **Next Behind** from the shortcut menu.
- Press the **B** shortcut key.
- Press **Ctrl+B**.

> **Note:**
>
> The desired entity must be behind the point where you clicked, not just behind the initially selected face, edge, vertex, or object.

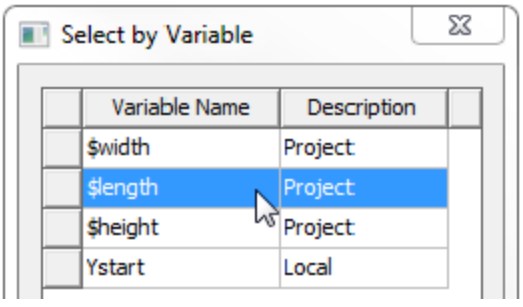
This option is useful when you are trying to select a face, edge, vertex, or object that is in the interior of a model, or when you do not want to change the model viewpoint to select an item.

Selecting Objects by Variable

You can select an object based on a variable that affects it. If your design includes variables, do so as follows:

1. From the menu bar, click **Edit> Select Objects> By Variable**.

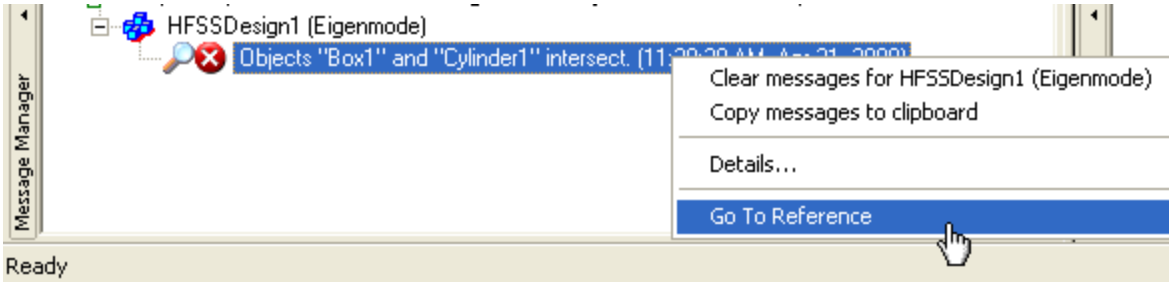
   The *Select by Variable* dialog box appears, which lists the variables in your design. Both columns are sortable by clicking the header. You can resize and move the dialog box. When you next open it, it remembers that size and location.



2. Select the variable of interest and click **OK**.

   The dialog box 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.

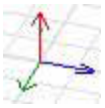
Assigning Coordinates to New Objects

When you insert and draw new 3D, 2D, or 1D objects in model, you may need to define coordinate systems, locations of points, distances between objects, and other geometry items.

- Setting the Reference Point
- Defining Cartesian 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 box opens. As you move the cursor over the Modeler window, the top line in the *Measure Data* dialog box 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 action moves the reference point marker to the new location. The *Measure Data* dialog box updates. The coordinates boxes in the Status Bar change to accept Relative distance information. If you prefer, rather than setting the reference point with the cursor, you can press **Tab** to jump to a coordinate box in the status bar, where you can specify the coordinates numerically. Navigate among the three coordinate boxes by pressing the **Tab** key again.

Defining Cartesian Coordinates

When you draw an object, define a point using Cartesian coordinates by typing the point's x, y, and z distances from the origin in the **X, Y,** and **Z** text boxes, respectively. When defining a second point, specify its x, y, and z distances from the previously selected point in the **dX, dY,** and **dZ** text boxes, respectively.

1. Select the desired drawing command.
2. Select **Cartesian** from the drop-down menu in the Status Bar.
3. Type the point's x, y, and z coordinates in the **X, Y,** and **Z** text boxes.

   > **Tip:**
   >
   > Press **Tab** to move from one coordinate text box to the next. Press **Ctrl+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the Modeler window.
4. When drawing objects other than polylines and helices, the second point you select is relative to the first point. Type the second point's x, y, and z distances from the previously selected point in the **dX, dY,** and **dZ** text boxes, respectively.

Defining Cylindrical Coordinates

To define a point using cylindrical coordinates, specify 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. When defining a second point, specify its distance from the previously selected point in the in **dR, dTheta,** and **dZ** text boxes.

1. After clicking the desired drawing command, select **Cylindrical** from the pull-down menu in the Status Bar.

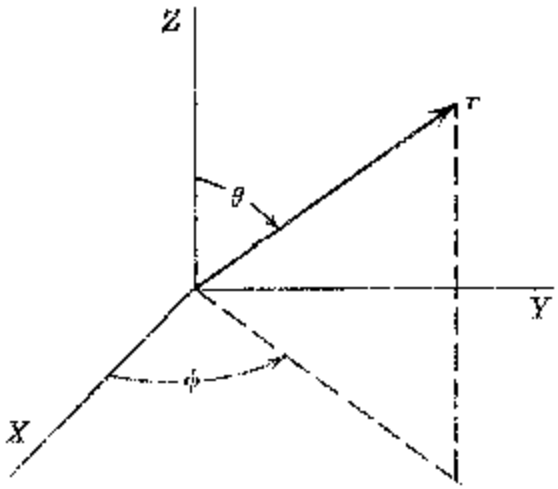
2. Type the point's r, theta, and z coordinates in the **R, Theta,** and **Z** boxes.

   > **Tip:**
   >
   > Press **Tab** to move from one coordinate text box to the next. Press **Ctrl+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the Modeler 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** text boxes.

Defining Spherical Coordinates

To define a point in spherical coordinates, specify the point's radius, measured from the origin, in the **Rho** text box, the angle from the positive z-axis in the **Theta** text box, and the angle from the positive x axis in the **Phi** text box. When selecting a second point, specify its distance from the previously selected point in the in **dRho, dTheta,** and **dPhi** text 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 drop-down menu in the status bar.

2. Type the point's r, theta, and phi coordinates in the **Rho, Theta,** and **Phi** text boxes in the Status Bar.

   > **Tip:**
   >
   > Press **Tab** to move from one coordinate text box to the next. Press **Ctrl+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the Modeler 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 **dRho, dTheta,** and **dPhi** text boxes.

Using Absolute Coordinates

When entering a point's location, you can specify it using *absolute* or *relative* coordinates. Absolute coordinates are relative to the working coordinate system's origin (0, 0, 0). This entry mode 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.

To enter a point's absolute coordinates:

1. Click the desired drawing command.
2. Select **Absolute** from the **Absolute/Relative** drop-down menu in 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 text boxes in 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** drop-down menu in the Status Bar. Be sure to select **Absolute** from this drop-down menu if you want the second point to be relative the origin of the working coordinate system.

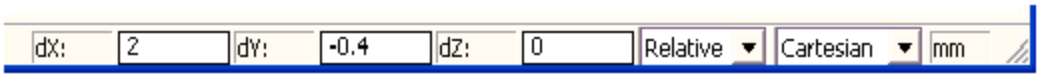
Using Relative Coordinates

When entering a point's location, you can specify it using *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** drop-down menu in the Status Bar.

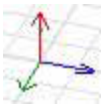
   When you are in *Relative* mode, the text boxes for a coordinate show an "d" before the coordinate description, to indicate "distance from" the working reference. For example:



3. Specify the point's coordinates in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the appropriate text boxes in the Status Bar.

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 when you click a drawing command:



To change the reference point, move the cursor to the desired point and **Ctrl+click** the point.

When creating 1D, 2D, or 3D geometry graphically (that is, by clicking points instead of by defining coordinates numerically), points that you can select are dependent on the current *Movement Mode* setting. You can move the cursor and place a point at one of the following locations:

- 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, you can snap to one of its points – any snapable point in 3D space (3D movement mode).
- Along the X-axis of the working coordinate system (CS)
- Along the Y-axis of the working CS
- Along the Z-axis of the working CS

Drawing commands that you choose may temporarily override the current movement mode. For example, when you draw a box or cylinder, the first two points you click (defining the base) may be in the drawing plane, but the third (defining the height) must be out of plane relative to the base. In this case, you do not have to manually change the movement mode; it is automatically overridden for the third click.

Note:

For all six movement modes covered in this section, the following rules apply regarding mode persistence:

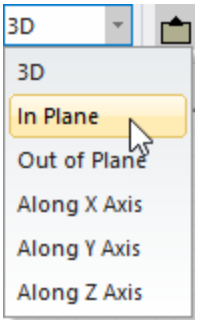
- If you change the movement mode *before* executing a drawing command, the new mode setting will be retained until you change it again.
- If you change the movement mode *after* executing a drawing command, it will be treated as a temporary override. Upon completion of the drawing operation, the mode will revert to the previous setting.

Moving the Cursor In Plane

Use the *In Plane* movement mode to specify points of objects being drawn that are *on the same plane* as the reference point:

Before or after clicking a desired drawing command, change the movement mode using one of the following methods:

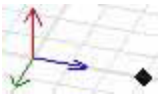
- From the menu bar, click **3D Model**> **Movement Mode**> **In Plane**.
- On the **Draw** ribbon tab, choose **In Plane** from the **Movement mode** drop-down menu:



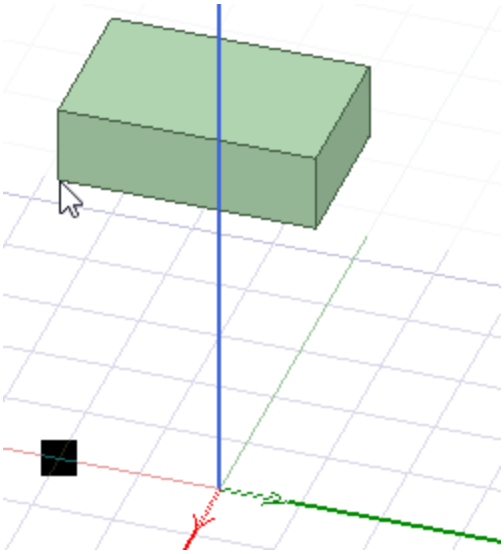
The following additional method is available only after clicking a drawing command:

- Right-click in the Modeler window and choose **Movement Mode**> **In Plane** from the shortcut menu.

The next point you select will be on the same plane as the reference point. The cursor's location is displayed with a black diamond that indicates where it will snap to the drawing plane grid.



If you click a snapping point on an object in 3D space (that is, one not on the drawing plane), the point location is projected to the drawing plane along a perpendicular vector. In the following example, a filled black square indicates the projected snapping point:



The symbol used for projected point matches the type of snapping point on the object (vertex, edge center, face center, and so on).

The *Measure Data* window and the coordinate text boxes in the Status Bar provide precise cursor location information.

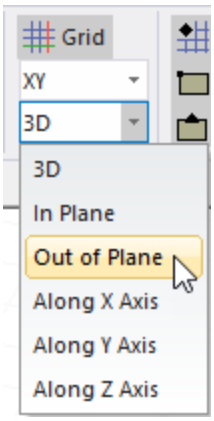
Moving the Cursor Out of Plane

Use the *Out of Plane* movement mode to specify points of objects being drawn that are along a vector *perpendicular* to the drawing plane and intersecting it at the reference point:

Before or after clicking a desired drawing command, change the movement mode using one of the following methods:

- From the menu bar, click **3D Model**> **Movement Mode**> **Out of Plane**.
- On the **Draw** ribbon tab, choose **Out of Plane** from the **Movement mode** drop-down

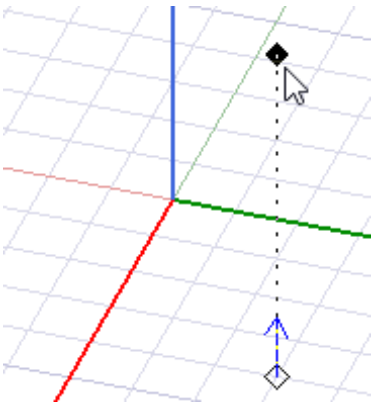
menu:



The following additional method is available only after clicking a drawing command:

- Right-click in the Modeler window and choose **Movement Mode**> **Out of Plane** from the shortcut menu.

A dashed line perpendicular to the drawing plane is displayed, and a filled black diamond indicates the cursor's location:



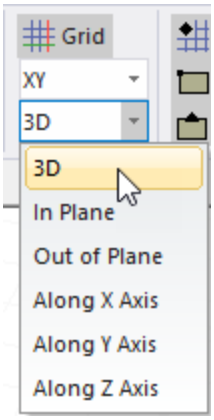
The *Measure Data* window and the coordinate text boxes in the Status Bar provide precise cursor location information.

Moving the Cursor in 3D Space

To snap to drawing plane grid points or to object snapping points anywhere in 3D space, use the *3D* movement mode.

Before or after clicking a desired drawing command, change the movement mode using one of the following methods:

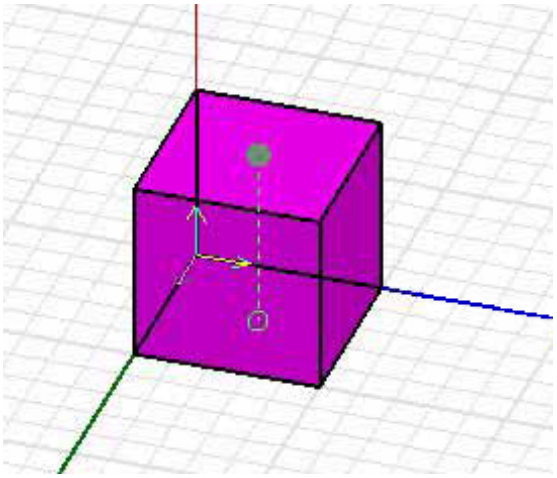
- From the menu bar, click **3D Model**> **Movement Mode**> **3D**.
- On the **Draw** ribbon tab, choose **3D** from the **Movement mode** drop-down menu:



The following additional method is available only after clicking a drawing command:

- Right-click in the Modeler window and click **Movement Mode**> **3D** from the shortcut menu.

If one of an object's snapping points (vertex, edge center, face center, and so on) is within snapping range of the cursor, the geometry you are drawing will snap to that point. The symbol will indicate the type of snapping point. In the following example, the cursor will snap to the face center point of the boxes top face:



If an object is not within snapping range of the cursor, the *3D* movement mode is identical to the *In Plane* mode. That is, the geometry being drawn will snap to a grid point on the drawing plane.

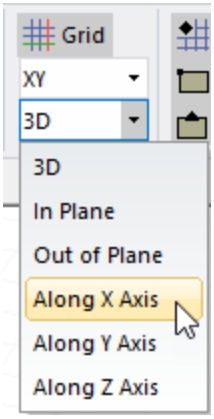
In either case, the *Measure Data* window and the coordinate text boxes in the Status Bar provide precise cursor location information.

Moving the Cursor Along the X-Axis

Use this movement mode to constrain the cursor to a moving only in the +/- X direction relative to the reference point. Movement is constrained to the X direction as defined by the working coordinate system, not necessarily the global X direction. The line of motion passes through the reference point, so it may be parallel to the X-Axis and not necessarily along the actual X-axis.

Before or after clicking a desired drawing command, change the movement mode using one of the following methods:

- From the menu bar, click **3D Model**> **Movement Mode**> **Along X Axis**.
- On the **Draw** ribbon tab, choose **Along X Axis** from the **Movement mode** drop-down menu:



The following additional methods are available only after clicking a drawing command:

- Press and hold the **X** key while clicking the next point.
- Right-click in the Modeler window and choose **Movement Mode**> **Along X Axis** from the shortcut menu.

The next point you select will be on the same plane as the reference point and in the positive or negative X direction relative to the reference point.

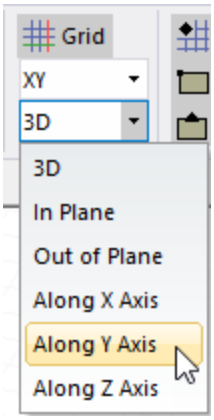
The *Measure Data* window and the coordinate text boxes in the Status Bar provide precise cursor location information.

Moving the Cursor Along the Y-Axis

Use this movement mode to constrain the cursor to a moving only in the +/- Y direction relative to the reference point. Movement is constrained to the Y direction as defined by the working coordinate system, not necessarily the global Y direction. The line of motion passes through the reference point, so it may be parallel to the Y-Axis and not necessarily along the actual Y-axis.

Before or after clicking a desired drawing command, change the movement mode using one of the following methods:

- From the menu bar, click **3D Model**> **Movement Mode**> **Along Y Axis**.
- On the **Draw** ribbon tab, choose **Along Y Axis** from the **Movement mode** drop-down menu:



The following additional method is available only after clicking a drawing command:

- Right-click in the Modeler window and choose **Movement Mode**> **Along Y Axis** from the shortcut menu.
- Press and hold the **Y** key while clicking the next point.

The next point you select will be on the same plane as the reference point and in the positive or negative Y direction relative to the reference point.

The *Measure Data* window and the coordinate text boxes in the Status Bar provide precise cursor location information.

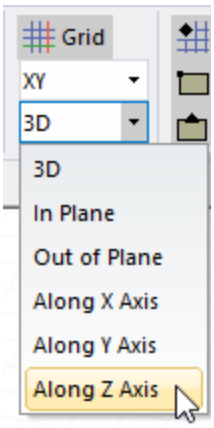
Moving the Cursor Along the Z-Axis

Use this movement mode to constrain the cursor to a moving only in the +/- Z direction relative to the reference point. Movement is constrained to the Z direction as defined by the working coordinate system, not necessarily the global Z direction. The line of motion passes through the reference point, so it may be parallel to the Z-Axis and not necessarily along the actual Z-axis.

Before or after clicking a desired drawing command, change the movement mode using one of the following methods:

- From the menu bar, click **3D Model**> **Movement Mode**> **Along Z Axis**.
- On the **Draw** ribbon tab, choose **Along Z Axis** from the **Movement mode** drop-down

menu:



The following additional methods are available only after clicking a drawing command:

- Press and hold the **Z** key while clicking the next point.
- Right-click in the Modeler window and choose **Movement Mode**> **Along Z Axis** from the shortcut menu.

The next point you select will be on the same plane as the reference point and in the positive or negative Z direction relative to the reference point.

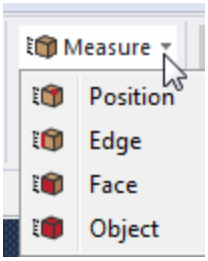
The *Measure Data* window and the coordinate text boxes in the Status Bar provide precise cursor location information.

Measure Modes

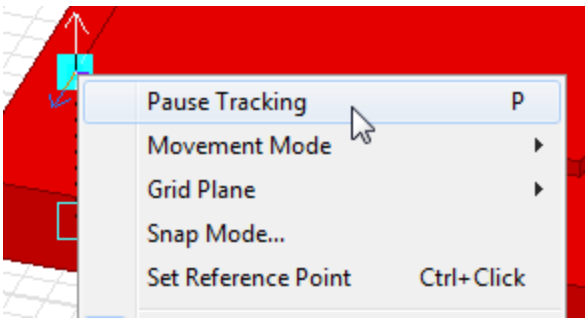
The Measure modes lets you measure the position, length, area, and volume of objects. With two faces selected, with two edges selected, or with an edge and a face selected, the Measure Mode displays the angle and distance between them. The Measure Position mode dynamically measures the distance between a reference point and the cursor location. You can pause dynamic tracking in order to copy text information from the *Measure Data* window, and then Resume dynamic updating.

1. To access a Measure mode, do one of the following:
    - From the menu bar, click **Modeler** > **Measure** > [**Position, Edge, Face,** or **Object**].
    - Right-click in the Modeler window and choose **Measure** > [**Position, Edge, Face,** or **Object**] from the short-cut menu.
    - On the **Draw** ribbon tab, choose **Position, Edge, Face,** or **Object** from the

Measure drop-down menu.

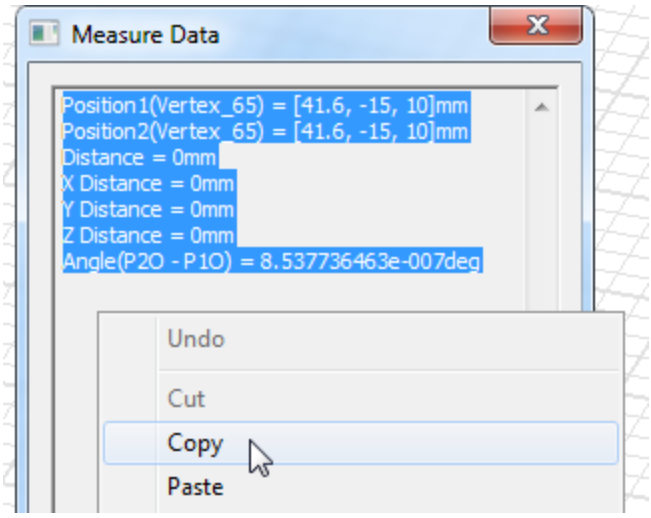


2. Select items to measure as described below under Measurement Modes and Data.
3. Optionally, to *Pause* the dynamic tracking, do one of the following:
   - Right-click in the Modeler window and select **Pause Tracking**:



   - Press **P**.

While paused, you can move the cursor without changing the *Measure Data* window contents. You can also copy and paste the data to a text file:



To resume measurement tracking, do one of the following:

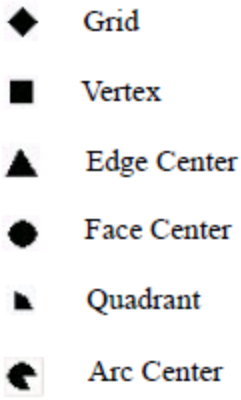
- Right-click in the Modeler window and choose **Resume Tracking**.
- Press **P** again.

4. To exit the Measure mode, do one of the following:
   - Click **Close** on the *Measure Data* window.
   - Press **Esc**.

Measurement Modes and Data:

- **Position:** Use the following procedure to measure the position of points and the distance between them:
  1. Click a snap point anywhere on the model, which becomes *Position 1* (the measurement reference point).
  2. Hover over another snap point but do not click the mouse (or **Ctrl+click** the second point to redefine the reference point but retain the measurement data).

     The following symbols indicate the available types of snap points:

     > **Note:**
     >
     > As you move the cursor, the Measure Data window displays the reference point, current cursor location, and measurement information. Clicking on a new vertex or other snap point updates the reference to the new location and resets the measurement results to zero. To resent the reference point location without clearing the measurement data, **Ctrl+click** the point.

     The *Measure Data* window gives you the following information:

     - Position1 coordinates
     - Position2 coordinates
     - Distance between Position1 and Position2

- X Distance: X-component of the Distance.
- Y Distance: Y-component of the Distance.
- Z Distance: Z-component of the Distance.
- Angle: The angle of the line connecting Position1 and Position2

- **Edge:** The *Measure Data* window gives you the following information:
  - Edge name
  - Length
  - Reference coordinate system
  - Start Vertex Position (with Vertex name)
  - End Vertex Position (with Vertex name)
  - Additional data for circles and arcs only:
    - Arc Center Position
    - Arc Radius
    - Arc Diameter
    - Plane
- **Face:** Each item listed below only appears in the *Measure Data* window when applicable to the selected face.
  - Face ID
  - Area
  - Perimeter
  - Cylinder Radius or Sphere Radius
  - Cylinder Diameter or Sphere Diameter
  - Plane (orientation of the plane – the face may be parallel to the listed plane and does not necessarily lie on it)
  - Plane Normal (normal vector for planes not aligned with a global or relative coordinate system)
  - Reference coordinate system
  - Center Position
  - No. of Loops
  - No. of Edges
  - No. of CoEdges
  - No. of Vertices
- **Object:** Each item listed below only appears in the *Measure Data* window when applicable to the selected object.
  - Volume(with *ObjectName* — zero for shell objects)
  - Area(with *ObjectName* — total surface area for solids and shells)
  - No. of Lumps

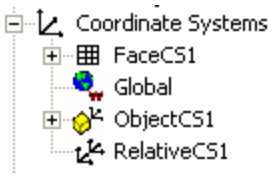
- No. of Shells
- No. of Faces
- No. of Loops
- No. of Edges (the total number of unique edges)
- No. of CoEdges (the total number of coincident edges, where adjacent faces intersect)
- No. of Vertices

Setting Coordinate Systems

The modeler has four types of coordinate systems (CS) that enable you to easily orient new objects:

- Global CS
- Relative CS
- Face CS
- Object CS

Every coordinate system 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 *Global* coordinate system (CS) is the fixed, default CS for each new project. It cannot be edited or deleted.
- A *Relative CS* 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 and/or orientation accordingly. You can define a relative CS to be *Offset* and/or *Rotated* from an existing CS.

  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 *Face CS* 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 *working CS*. 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 (as shown on Global in the figure above). The docked **Properties** window lists the CS associated with an object as the *Orientation*. By default, the orientation is Global, but if you have created the object under a different coordinate system, that CS will be shown for the orientation. You can click on the current orientation to see a drop-down menu of other orientations that you can assign for an object.

User-defined CSs are saved with the active project. When you open a project, the CS designated as working CS when you last saved it is active.

Setting the Working Coordinate System

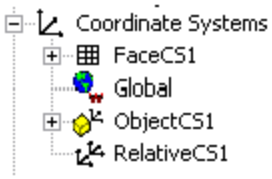
The working coordinate system (CS) is the current CS with which objects being drawn are associated. The working CS can be the Global CS or a user-defined Relative, Object, or Face CS. Select the working CS by clicking its name in the History Tree, or follow this procedure:

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. In this following figure the Global is the working CS.



   Objects that you draw hereafter will be associated with the CS you selected.

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 global

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.
- You can click **Tools> Options> General Options** to display the *Options* dialog box.

  Under the *3D Modeler> Drawing* options, choose your preferred *Relative Coordinate System Creation Mode*. The **Axis/Position** option uses the cursor or coordinate entry boxes in the Status bar to define the CS. The **Euler angle** option causes the *Set Origin, Set Euler Angles,* or *Set Origin and Euler Angles* dialog box to display (depending on the type of Offset CS you are creating). You define the origin and angles numerically in the dialog box.

  You can also press **F3** or **F4** after clicking a *RelativeCS* command to switch between the two entry modes on the fly.

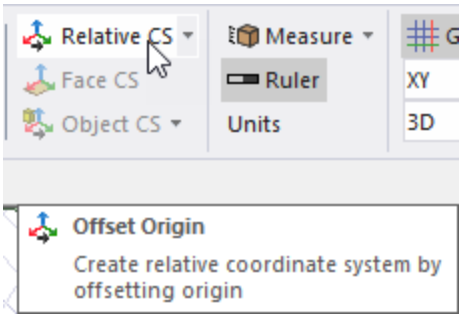
After you have created a relative CS it appears in the History Tree:

Selecting the CS in the History Tree causes the docked *Properties* window to show the CS properties.

Creating an Offset Relative CS

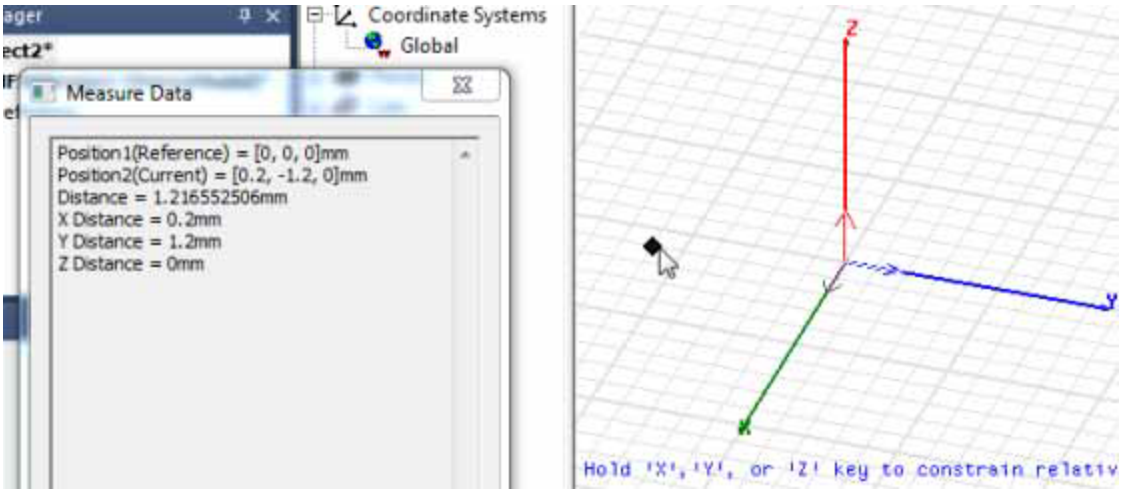
To create a relative CS with an origin that lies a specified distance from another CS's origin:

1. In the History Tree, click the CS upon which you want to base the new relative CS, making it the working CS.
2. From the menu bar, click **Modeler> Coordinate System> Create> Relative CS> Offset** or, on the **Draw** ribbon tab, click **Relative CS**. (You do not have to access the drop-down menu because *Offset* is the default option for a *Relative CS* in the ribbon:



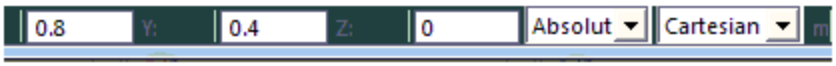
3. Depending on your *Relative Coordinate System Creation Mode* choice (**Tools> Options> General Options**, *3D Modeler> Drawing*), or whether you press **F3** or **F4**, you can select the origin in one of the following two ways:
   - **Axis/Position** option selected, or **F3** pressed – Use one of the following two coordinate entry methods:
     - Specify the coordinates graphically (click to select):

       Use the cursor to click the point. You can see the coordinate information in the *Measure Data* window.

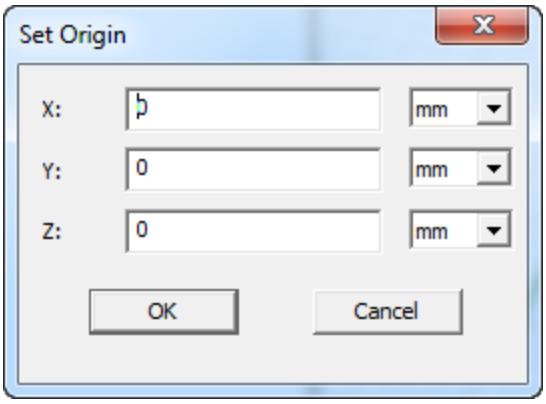
       To select a point that does not lie in the current plane, use the *Movement Mode* commands or choose a different drawing plane.

     - Specify the coordinates numerically (using the coordinate entry text boxes in the Status Bar):

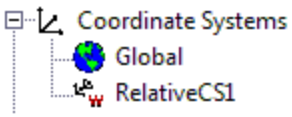
       

       a. In the Status Bar, choose the type of coordinate system from the rightmost drop-down menu (**Cartesian, Cylindrical,** or **Spherical**).
       b. Select either Relative or Absolute coordinates from the other drop-down menu.
       c. Type the CS origin coordinates in the **X, Y** and **Z** (or **dX, dY,** and **dZ**) text boxes.
   - **Euler angle** option selected, or **F4** pressed:

     In the *Set Origin* dialog box, specify the **X, Y,** and **Z** coordinates and units and click **OK**.



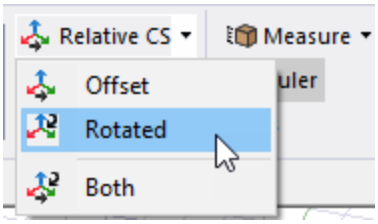
The new relative CS is created. Its origin has moved from the previous working CS, but its axes remain the same. It is listed in the History Tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.



Creating a Rotated Relative CS

To create a new relative CS with its axes rotated away from another CS's axes:

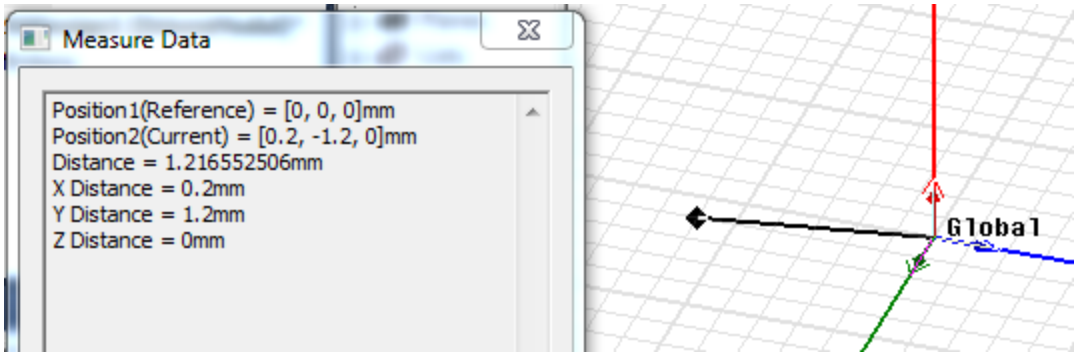
1. In the History Tree, select the CS upon which you want to base the new relative CS, making it the working CS.
2. Click **Modeler> Coordinate System> Create> Relative CS> Rotated** or, on the **Draw** ribbon tab, click **Relative CS> Rotated**:



.

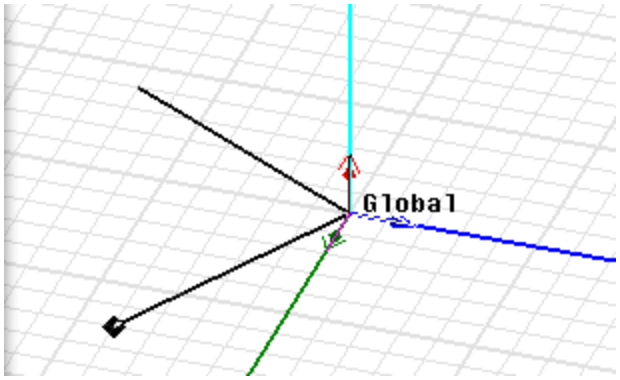
3. Depending on your *Relative Coordinate System Creation Mode* choice (**Tools> Options> General Options** – *3D Modeler> Drawing*), or whether you press **F3** or **F4**, you can specify the axes in one of the following two ways:
   - **Axis/Position** option selected, or **F3** pressed – Use one of the following coordinate entry methods:

- Specify the X-axis and XY plane graphically (click to select), as follows:
  a. Use the cursor to click a point along the desired X-axis. You can see the coordinate information in the *Measure Data* window.

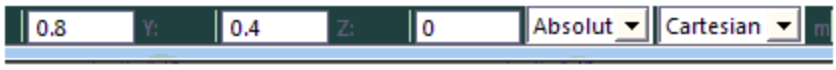


  To select a point that does not lie in the current plane, use the *Movement Mode* commands or choose a different drawing plane.

  b. In the same manner, select any point lying on the XY plan that's not on the X-axis.



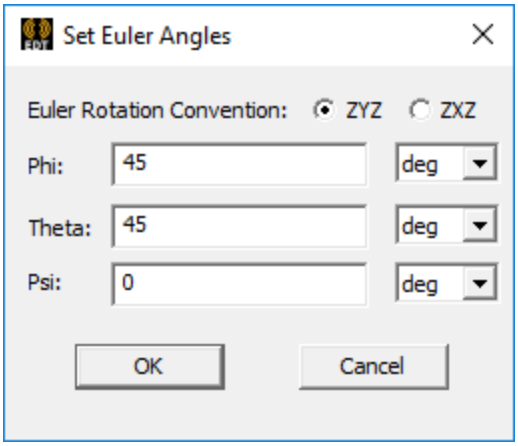
- Enter the coordinates numerically in the Status Bar, as follows:



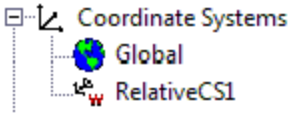
  a. In the Status Bar, choose the type of coordinate system from the rightmost drop-down menu (**Cartesian, Cylindrical,** or **Spherical**).
  b. Select either Relative or Absolute coordinates from the other drop-down menu.
  c. Type the coordinates of a point on the desired X-axis in the X, Y and Z (or dX, dY, and dZ) text boxes and press **Enter**.
  d. In the same manner, type the coordinates of a point lying on the XY plane, but not on the X-axis, and press **Enter** again.

You do not need to specify the Z-axis. It is automatically calculated to be perpendicular to the new XY plane.

- **Euler angle** option selected, or **F4** pressed:
  a. In the *Set Euler Angles* dialog box, select the preferred **Euler Rotation Convention** (**ZYZ** or **ZXZ**).
  b. Specify the **Phi, Theta,** and **Psi** values and select units for each from the drop-down menus.



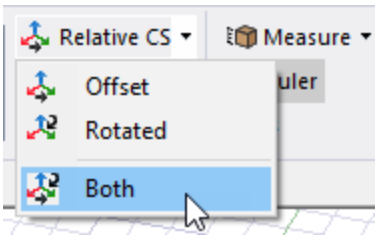
The new relative CS is created. It has the same origin as the previous working CS, but its axes are rotated. It is listed in the History Tree under *Coordinate Systems*. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its XY, YZ, and XZ planes.



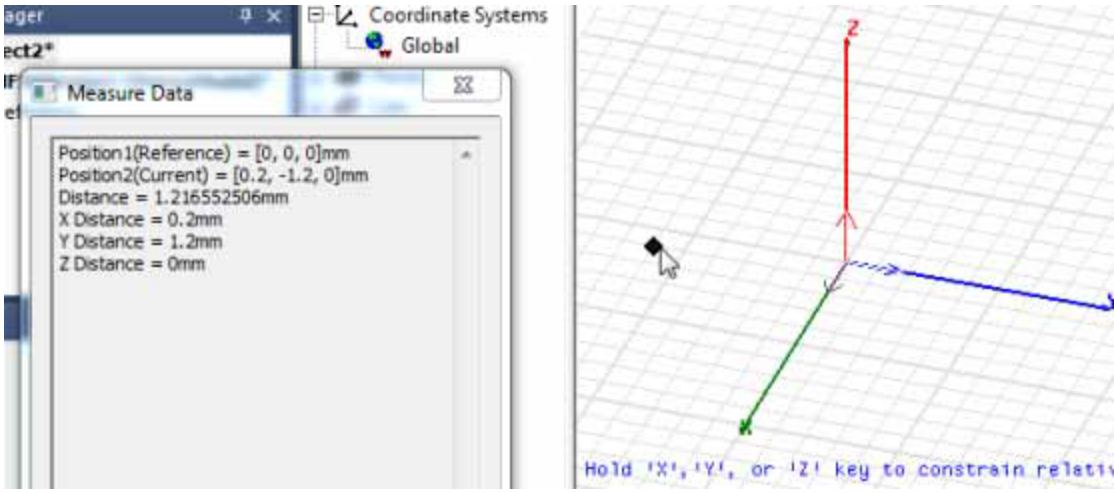
Creating an Offset and Rotated Relative CS

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. Click **Modeler> Coordinate System> Create> Relative CS>** **Both** or, on the **Draw** ribbon tab, click **Relative CS> Both**:

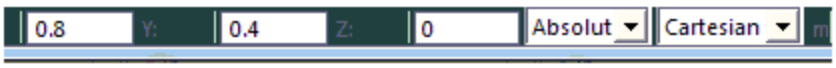


3. Depending on your *Relative Coordinate System Creation Mode* choice (**Tools> Options> General Options** – *3D Modeler> Drawing*), or whether you press **F3** or **F4**, you can specify the origin and axes in one of the following two ways:
   - **Axis/Position** option selected, or **F3** pressed – Use one of the following coordinate entry methods:
     - Specify the Origin, X-axis, and XY plane graphically (click to select), as follows:
       a. Use the cursor to click the Origin point. You can see the coordinate information in the *Measure Data* window.



       To select a point that does not lie in the current plane, use the *Movement Mode* commands or choose a different drawing plane.

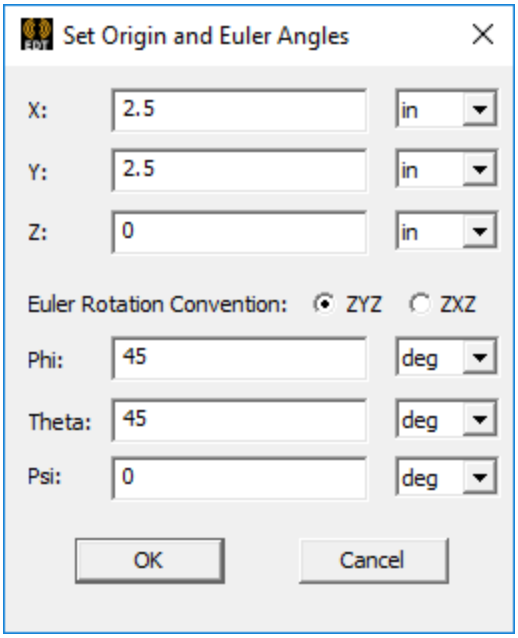
       b. In the same manner, click a point to define the X-axis.
       c. Select a third point, one lying on the XY plan that's not on the X-axis, to complete the CS definition.
     - Specify the Origin, X-axis, and XY plane numerically in the Status Bar, as follows:



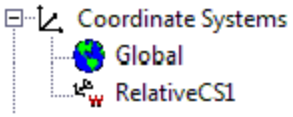
a. In the Status Bar, choose the type of coordinate system from the rightmost drop-down menu (**Cartesian, Cylindrical,** or **Spherical**).
b. Select either Relative or Absolute coordinates from the other drop-down menu.
c. Type the CS origin coordinates in the **X, Y** and **Z** (or **dX, dY,** and **dZ**) text boxes.
d. In the same manner, type the coordinates of a point on the X-axis and press **Enter**.
e. Type the coordinates of a third point, one lying on the XY plan that's not on the X-axis, and press **Enter** to complete the CS definition.

You do not need to specify the Z-axis. It is automatically calculated to be perpendicular to the new XY plane.

- **Euler angle** option selected, or **F4** pressed:
  a. In the *Set Origin and Euler Angles* dialog box, specify the **X, Y,** and **Z** coordinates of the Origin and select units for each from the drop-down menus.
  b. Select the preferred **Euler Rotation Convention** (**ZYZ** or **ZXZ**).
  c. >Specify the **Phi, Theta,** and **Psi** values and select units for each from the drop-down menus.

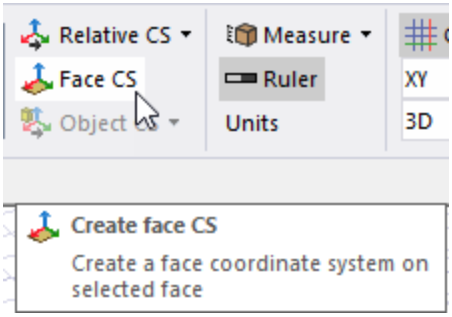


The new relative CS is created. It is listed in the History Tree under *Coordinate Systems*. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its XY, YZ, and XZ planes.



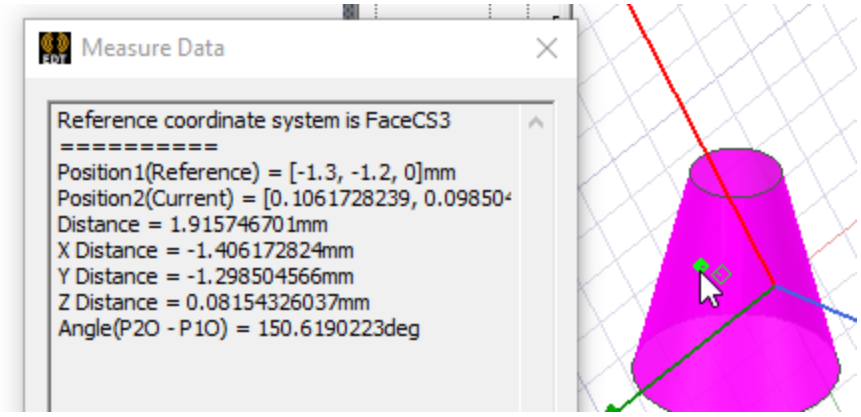
Creating a Face Coordinate System

1. Select the object face upon which you want to create the face CS. The CS face can be planar or curved.
2. Click **Modeler> Coordinate System> Create> Face CS** or, on the **Draw** ribbon tab, click the **Create Face CS** icon.



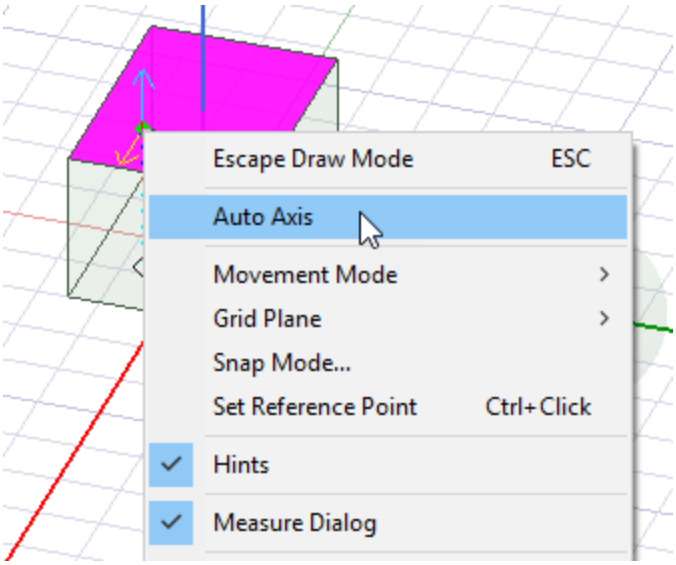
3. Select the Origin in one of the following ways:
   - Click a point on the selected face to define the Origin. If the point is not on the face, you will receive an error message.

     The *Measure Data* window shows the current cursor coordinates. The cursor shape will change to identify grid points, vertices, edge center, face center, quadrant, and arc center points, but you can click any point on the selected face.



   - Type the point's coordinates in the **X, Y,** and **Z** text boxes on the Status Bar.
4. Specify the positive X-axis direction in one of the following ways:
   - Click a point on the selected face. If the point is not on the face, you will receive an error message.

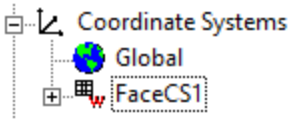
- Type the coordinates of a point that is relative to the previously selected point in the **dX, dY,** and **dZ** text boxes, where **d** is the distance from the previously selected point.



- Right-click in the Modeler window and select **Auto Axis** from the shortcut menu. The modeler selects the X-axis based on the principal curvature of the face or the dimensions of a flat face. The *axial* direction of a curved face or the long dimension of a flat face is selected as the X-axis direction.

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 perpendicular to the XZ plane.

The new face CS is listed in the History Tree under *Coordinate Systems*. It automatically becomes the working CS; objects that you draw hereafter will be 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 will affect the face CS and, in turn, affect objects dependent upon that face CS. A face CS, or any object created on it, is *not* affected by operations that occur after the face CS is created. Also see the Move CS to End command.

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 docked *Properties* window, 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 rotation operation occurs later in the History Tree).

Automatically Creating Face Coordinate Systems

You can instruct the modeler to automatically create a new face CS every time you draw on an object's face.

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

   The *Options* dialog box appears.

2. In the tree on the left side of the dialog box, select **3D Modeler> Operation**.
3. In the *Coordinate System* section, select **Automatically switch to face coordinate system**.
4. Click **OK**.

Now, when you select a face, and then click a drawing command, a new face CS will be created on the selected face. The modeler automatically sets the new face CS as the working CS, and the object you draw is oriented according to the new face CS.

> **Note:**
>
> The modeler will not automatically create a new face CS if a face CS has already been assigned to the selected face.

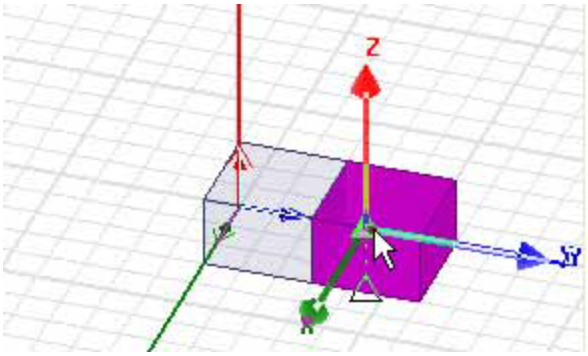
Creating an Object Coordinate System

You can create coordinate systems based on any object of solid, sheet, or wire type. The **Modeler> Coordinate System> Create> Object CS** command, and the equivalent ribbon 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
- Vertex
- Edge Center
- Face Center
- Quadrant
- Arc Center

For example, in this case, the cursor shows Edge Center triangles as valid selection points for a **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.

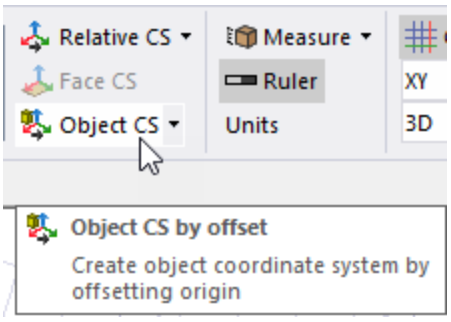
For example, suppose you create a box, then an Object CS on a face of the box, and then a cylinder on the Object CS. If you then edit the box's dimensions in the docked *Properties* window, the cylinder will move accordingly. But if you rotate the box using the **Edit> Arrange> Rotate** command, the box will move, but the cylinder will not move because the operation occurs later in the History Tree.

Creating an Offset Object CS

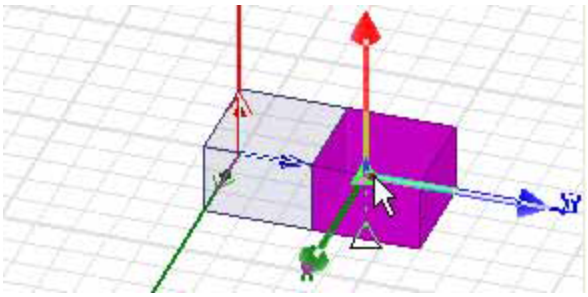
To create an Offset Object coordinate system (CS):

1. Select the working CS to use as the basis of the new Object CS.
2. Select the object to use as the basis of the new Object CS.
3. From the menu bar, click **Modeler> Coordinate System> Create> Object> Offset** or, from the **Draw** ribbon tab, click **Object CS**. (You do not have to access the *Object CS*

drop-down menu because *Offset CS by offset* is the default type of Object CS):



4. Click a point on the selected object to locate the Origin for the new CS.
    - You can select any snap point on the object (depending on the current snapping mode options) to select origin of the Object CS. When you hover the mouse over a valid point, a coordinate system preview is shown.

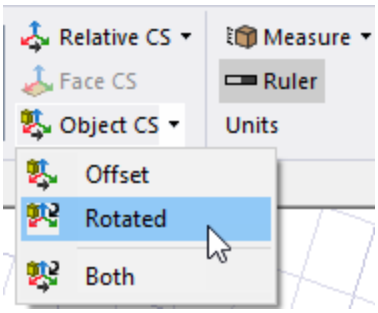


    - The point must be on the selected object.
    - When you select the Origin point, the location is validated, and an appropriate message indicates when a point is invalid. Points where a CS preview appears are always valid.
    - The X-axis {1,0,0}, Y-axis {0,1,0} and Z-axis {0,0,1} are parallel to the corresponding axes of the working CS that was active when you executed the Object CS command, but the Origin has been relocated.
    - The axis coordinates can be later edited through the docked *Properties* window or the *Properties* dialog box.

Creating a Rotated Object CS

To create a Rotated Object coordinate system (CS):

1. Select the working CS to use as the basis of the new Rotated Object CS.
2. Select the object to use as the basis of the new Rotated Object CS.
3. From the menu bar, click **Modeler> Coordinate System> Create> Object> Rotated** or, on the **Draw** ribbon tab, click **Object CS> Rotated**:



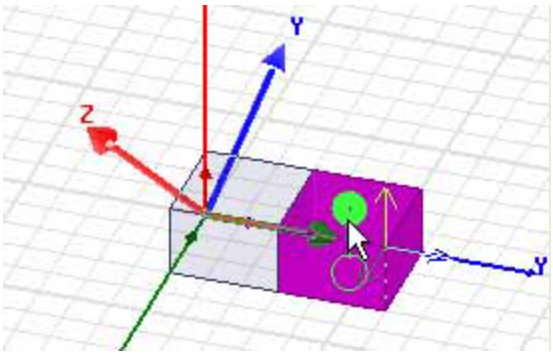
You are prompted to *Select X axis*. A preview of X axis is shown for valid selections.

4. Click a point on the selected object to locate the X-axis direction for the new CS.
   - The Origin {0,0,0} remains at the location defined for the working CS that was active when you executed the Rotated Object CS command. You can edit the coordinates later through the docked *Properties* window or the *Properties* dialog box.
   - The clicked point must be on the selected object.

   You are prompted to *Select to define XY plane*.

5. For 3D models, click a second point on the selected object to define the XY plane.

   A CS preview displays as the cursor hovers over valid selections.



In the 2D modeler, you are prompted to select only the X axis. The 2D model lies on the XY plane, and the Y-axis is 90 degrees from the X-axis.

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 to use as the basis of the new Object CS.
2. Select the object to use as the basis of the new Object CS.

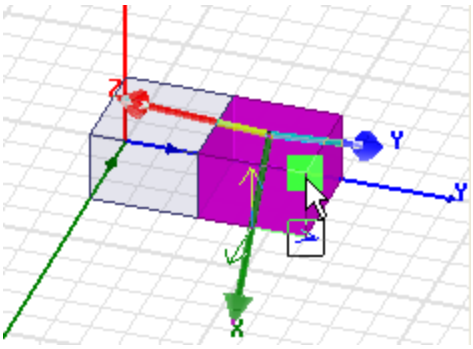
   The requirements to define this type of Object CS are a combination of *Offset* and *Rotated* Object CS definitions. You are first prompted to select the Origin (refer to the Creating an Offset Object CS page).

3. Select a point on the object to define the Origin.

   You are then prompted to define the XY plane (refer to the Creating a Rotated Object CS page.

4. For 3D models, select a second point on the object to define the XY plane.

   A CS preview appears when you hover over valid snapping points:



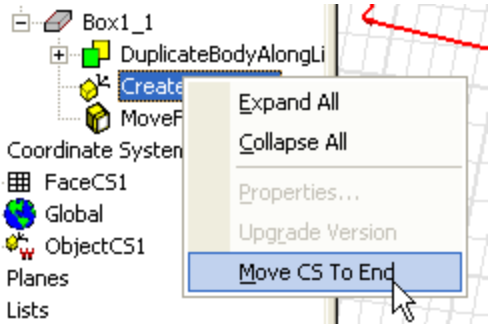
   In the 2D modeler, you are asked only to select the origin and the X axis. The 2D model lies on the XY plane, and the Y-axis is 90 degrees from the X-axis.

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 operation after a *CreateFaceCS* or *CreateObjectCS* in the History Tree:

Then, selecting a Face or Object CS enables the **Move CS to End** command in the **Modeler> Coordinate System** submenu and in the shortcut menu that appears when you right-click the operation in the History Tree:



Executing this 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, and so on).

Modifying Coordinate Systems

Keep in mind that when you edit a CS, the following will also be affected:

- All objects drawn on the CS.
- All CSs that were defined relative to that CS.
- All objects drawn on a CS that were defined relative to that CS.

There are two ways to modify a coordinate system. You can select the coordinate system in the History Tree and edit its parameters in the docked *Properties* window. This approach does not allow you to change whether the coordinate system is Absolute or Relative. Nor does it allow you to change how you express the coordinates (as Cartesian, Cylindrical, or Spherical).

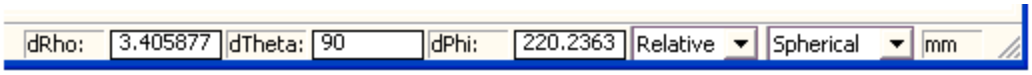
If you want to also modify the whether the coordinate system is Absolute or Relative, and to change how your express the coordinates, do the following:

1. From the menu bar, click **Modeler> Coordinate System> Edit**.

   The *Select Working CS* dialog box appears.

2. Select the CS you want to modify.
3. Click **Select**.

   This action places you in the graphical selection mode for redefining the CS (origin or X-axis) and enables the editable text boxes in the Status Bar at the bottom right area of the Ansys Electronics Desktop application.



4. You can click points to redefine the CS. Alternatively, you can specify the type of coordinate definition (Absolute or Relative) and tab between the coordinate entry text boxes in the Status Bar to numerically define new coordinates.

- 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.
- If you selected an Object CS, follow the directions for creating an object CS.

5. Select the type of coordinate system from rightmost drop-down menu in the Status Bar (Cartesian, Cylindrical, or Spherical).

   The values you specify are translated to Cartesian coordinates in the *Properties* of the revised coordinate system.

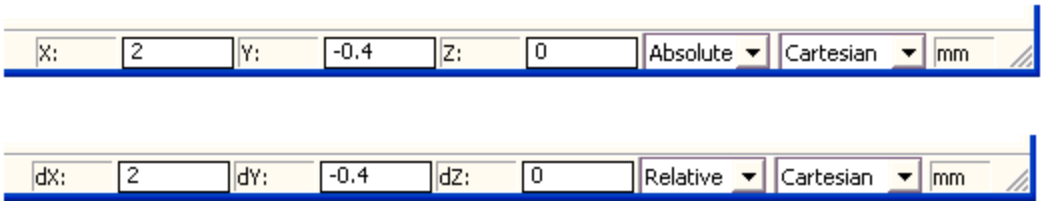
An Object CS can be edited in the same way as it was created. For example, if the CS was created in Offset mode, it will be edited in Offset mode only.

- You can edit the reference CS of an object CS through the docked *Properties* window or the *Properties* dialog box. Doing so will impact object CSs created in *Offset* or *Rotated* mode. It will not impact an object CS created in *Both* (offset and rotated) mode, as the CS fully depends on the object.
- The Origin will be editable when an Object CS was created in *Rotated* mode. Otherwise the Origin coordinates will be read-only text.
- The X Axis and Y Axis will be editable when an Object CS was created in *Offset* mode. Otherwise it will be axis coordinates will be read-only text.
- Either the X Axis or Y axis can be reversed through the docked *Properties* window or the *Properties* dialog box. When the X Axis is reversed, the Y Axis does not change and vice versa.
- When origin or axis properties are defined from snapping points, the point ID is listed in the properties instead of the coordinates. Some examples of snapping point IDs are: *Vertex_10, Face_7 center, Edge_9 midpoint, Edge_8 quadrant,* and *Edge_17 arc center*.

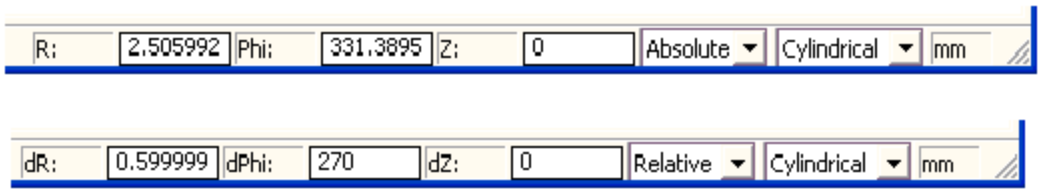
Expressing Cartesian, Cylindrical, or Spherical Coordinates

You have choices for expressing coordinates, specifically as Cartesian, Cylindrical, or Spherical by using the editable fields in the Status Bar. In each case, you can also specify whether to enter the coordinates as Absolute or Relative to a reference point. Click the first text box, or press **Tab** to begin entering values or variables. **Tab** to the next text box or **Ctrl+Tab** to the previous one. Press **Enter** when you are done. After you enter values or variables in the text boxes, they are evaluated as Cartesian for the coordinate system *Properties*, regardless of the way they were defined.

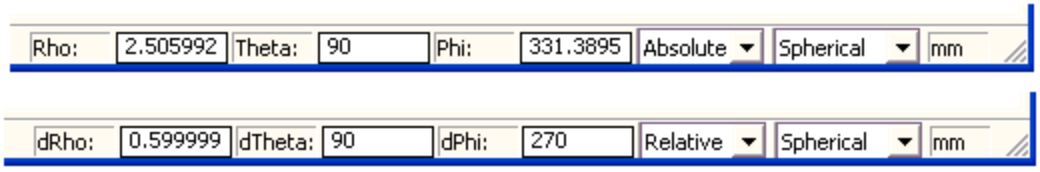
- **Cartesian:** Coordinates represent the point's distance from the origin in the x, y, and z directions. The values are specified in the **X, Y,** and **Z** text boxes.



- **Cylindrical:** Specify the point's radius, measured from the origin, is the **R** text box, the angle from the x-axis in the **Theta** text box, and the z-distance from the origin in the **Z** text box.



- **Spherical:** Specify 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.



Deleting Coordinate Systems

1. In the *Coordinate System* branch of the History Tree, click the name of the CS you want to delete.
2. Delete the CS using one of the following methods:
   - Press **Delete**.
   - From the menu bar, click **Edit**> ✕ **Delete**.
   - On the **Desktop** ribbon tab, click ✕ **Delete**.

   The CS will be deleted and, all objects drawn on it will be deleted. Furthermore, any CS that was dependent upon the deleted CS will be deleted, and any objects that were drawn on the dependent CS will also be deleted.

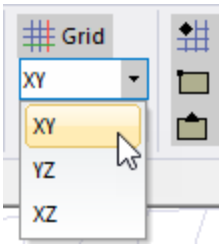
Setting the Drawing Plane

The *Drawing plane* options are available from a drop-down menu on the *Draw* ribbon tab and from the *Grid Plane* submenu of the *Modeler* menu. The XY, YZ, and XZ drawing planes

correspond to the active coordinate system, not the global planes (unless *Global* is the currently active CS).

To set the drawing plane, do one of the following:

- On the **Draw** ribbon tab, select **XY, YZ,** or **XZ** from the **Drawing plane** drop-down menu:



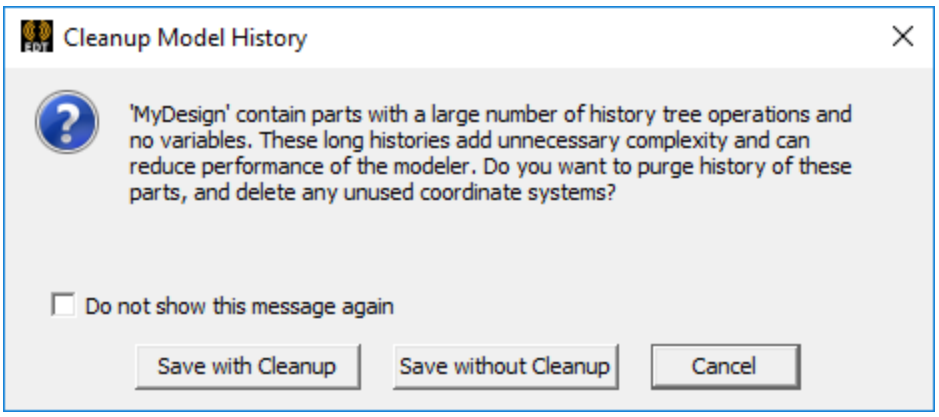
- From the menu bar, click **Modeler> Grid Plane>** and select one of the planes **XY, YZ,** or **XZ**.

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



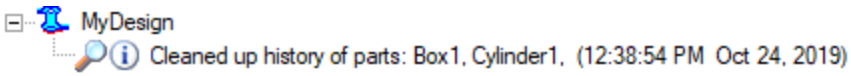
The number of actions that triggers this prompt can be set in 3D Modeler Operations Options.

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:



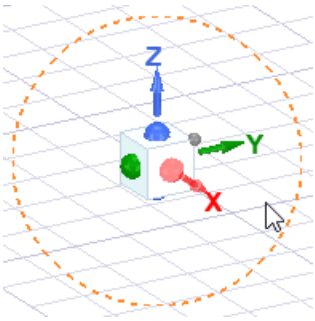
9 - Modifying the 3D Model View

You can modify the view of objects in the 3D **Modeler** window without changing their actual rotation or position 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 view orientation and to quickly switch to predefined or user-defined view orientations. Additionally, you can choose different object visualization options, show/hide objects, and customize the window background.

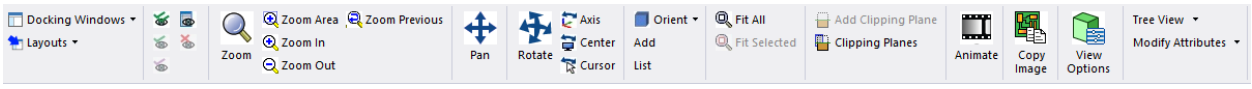
What do you want to do?

Change the View	Show or Hide Design Objects
Change Object Visualization	Change the Background

In addition to various menu bar, ribbon, and shortcut menu commands, you can manipulate the view orientation of 3D models and 3D plots 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.

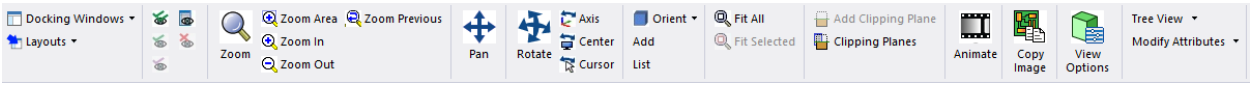


Change the View

- Using the Orientation Gadget
- Changing the Model View with Alt+Double-Click Areas in the Modeler Window
- Apply an Orientation to the Current View

- Set the View Options
- Rotate the View
- Pan the View
- Zoom In or Out
- Zoom to Selected Excitation
- Zoom In or Out Using a Mousewheel
- Fit Contents in the View Window
- Use Clip Planes

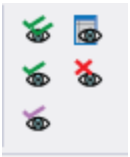
You can also access many of the Modify View features via the **View Ribbon**. The following image has hot links for each area.



Show or Hide Design Objects

- Show or Hide Objects
- Showing Only Selected Objects in All or Active Views

You can also access the Show or Hide features by using the following icons on the **View, Draw,** or **Model** ribbon tabs:



Change Object Visualization

- Render Objects as Wireframes, Flat-Shaded, or Smooth-Shaded Solids
- Set the Enhanced Display of Material Color and Transparency
- Set the Surface Visualization
- Modify the View Orientation
- Modify the Lighting
- Set the Projection View

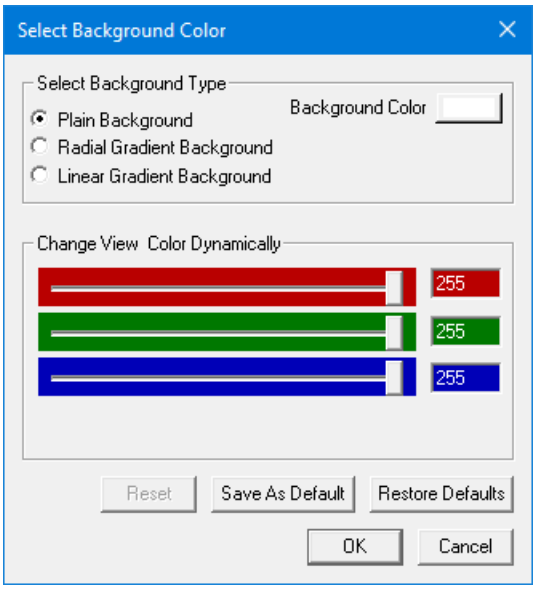
Change the Background

Setting the Background Color

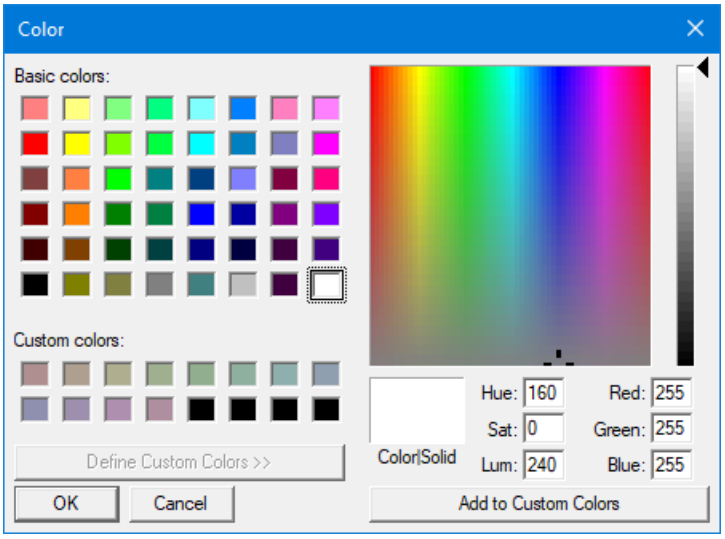
To set the color of the background in the view window:

1. Click **View** > **Modify Attributes** > **Background color** or, on the **View** tab of the ribbon, click **Modify Attributes** and select **Background Color...** from the drop-down menu.

   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.



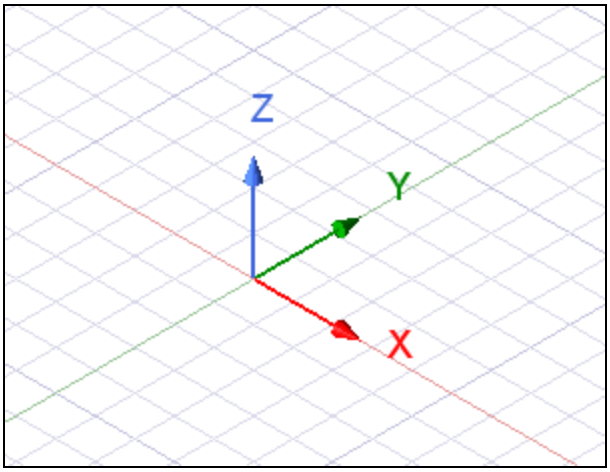
- Use the RGB sliders under **Change View Color Dynamically** to specify the color's red, green, and blue values.

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. Then click **Bottom Color** and select a color from the **Color** palette.
      - 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 *3D Modeler* windows in either the current project or future projects.
6. Click **OK**.

The background color you set will be saved with the design. New background color settings assigned to other designs after this point, including new default settings, will not affect this design.

Modifying the Coordinate System Axes View

The coordinate system axes displays the x, y, z orientation from the origin point for the current working coordinate system. The axes use an XYZ = RGB coloring scheme.



By option, you can also use a triad which 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.

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** > **General Options**.

   This command displays the *Options* dialog box.
2. Expand the **3D Modeler** branch of the options tree.
3. Expand the **Display** sub-branch.
4. Select the **General** subgroup.
5. Select **Show orientation of selected objects**.
6. Click **OK** to close the dialog box.

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.

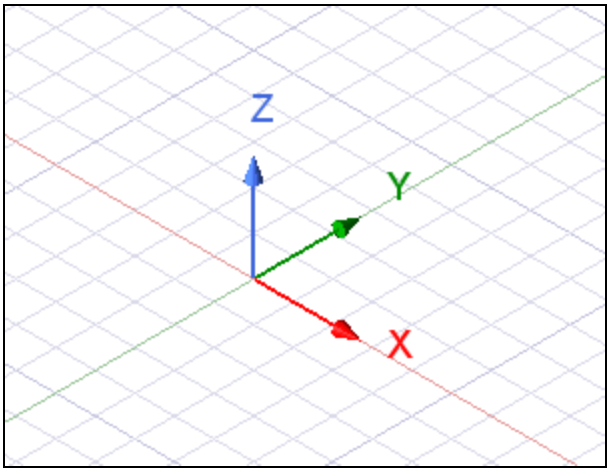
To show the triad:

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.

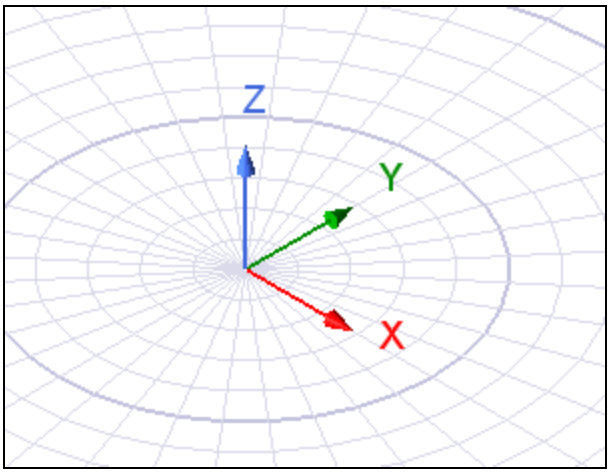
Choosing Grid Settings

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.



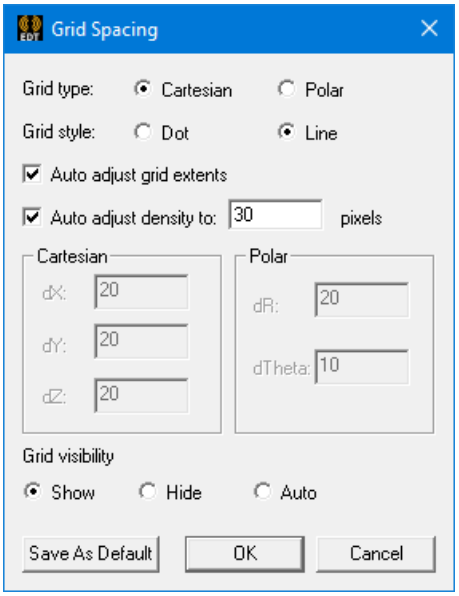
Grid spacing is set according to the current project's drawing units. You can control the following aspects of the grid:

- Type (rectangular or circular)
- Style (dots or lines)
- Grid Extent (minimal for existing objects or as a plane)
- Density
- Spacing
- Visibility
- Snap settings
- Grid plane

Setting the Grid Type

1. Click **View** > **Grid Settings**.

   The **Grid Spacing** dialog box appears.



2. Select a grid type for the active Modeler 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.

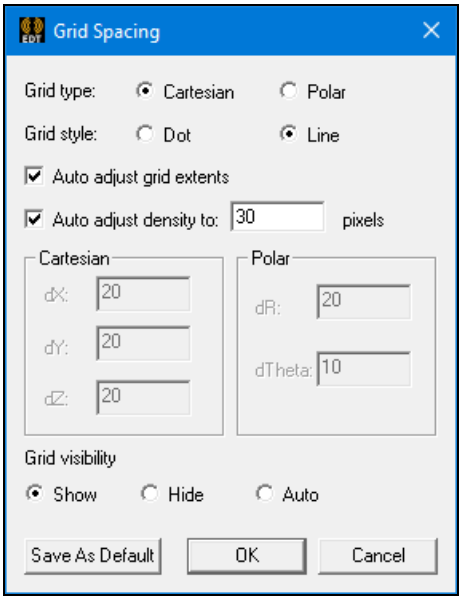
For Cartesian grids, you define a coordinate by specifying its distance from the origin along each axis in the **X**, **Y**, and **Z** text boxes or its relative distance from the previously selected point in the **dX**, **dY**, and **dZ** text boxes.

For polar grids, you 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** dialog box appears.



2. Select one of the following grid styles for the active view window:

   Dot Displays each grid point as a dot.

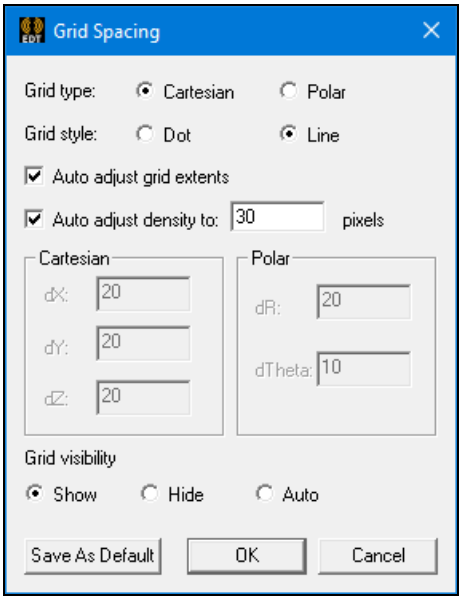
   Line Displays lines between grid points.

3. To show a Minimal grid around existing objects, check **Auto adjust grid extends**. To display a grid as a plane, deselect **Auto adjust grid extends**.

Setting the Grid Density and Spacing

1. Click **View** > **Grid Settings**.

   The **Grid Settings** dialog box appears.



2. If you want to change the density of the grid in the active Modeler 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.

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 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.

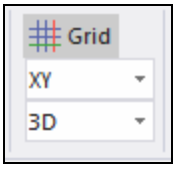
    If you selected a polar grid, type the values for **dR** and **dTheta**. **dR** represents the difference between each radius. **dTheta** is the difference between angles.

    The distance between grid points will increase and decrease proportionately as you zoom in and out in the active Modeler window.

4. Click **OK**.

Setting the Grid's Visibility

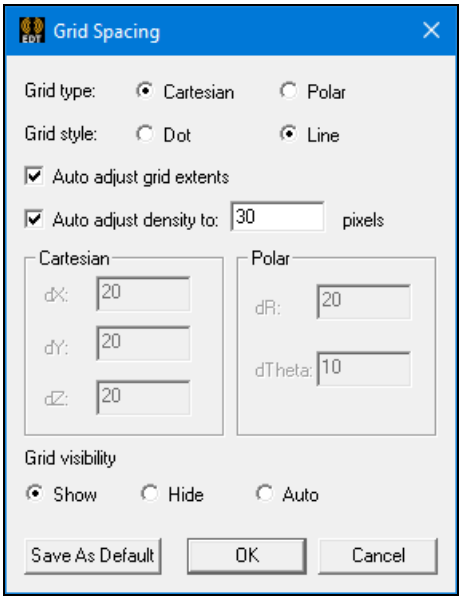
To hide the grid in the Modeler window, on the **Draw** ribbon tab, click the Grid toolbar icon. Click it again to show the grid.



Alternatively:

1. Click **View** > **Grid Settings**.

   The **Grid Spacing** dialog box appears.

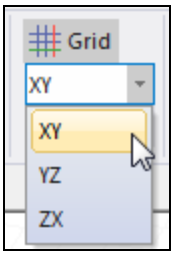


2. Select **Show** to make the grid always visible in the active **Modeler** window. Select **Hide** to make the grid always invisible in the active **Modeler** window. Select **Auto** so the grid appears only while you are drawing an object in the **Modeler** window.

Setting the Grid Plane

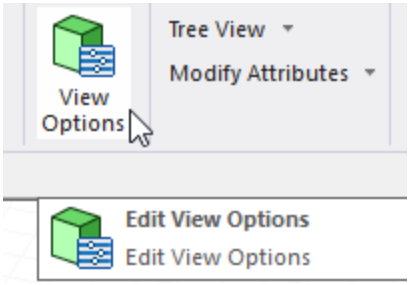
To specify the plane on which you want to display the grid in the active view window, do one of the following:

- Click **Modeler** > **Grid Plane** and then select a grid plane: **XY**, **YZ**, or **XZ**.
- Click a grid plane on the drop-down menu under Grid on the **Draw** ribbon tab:

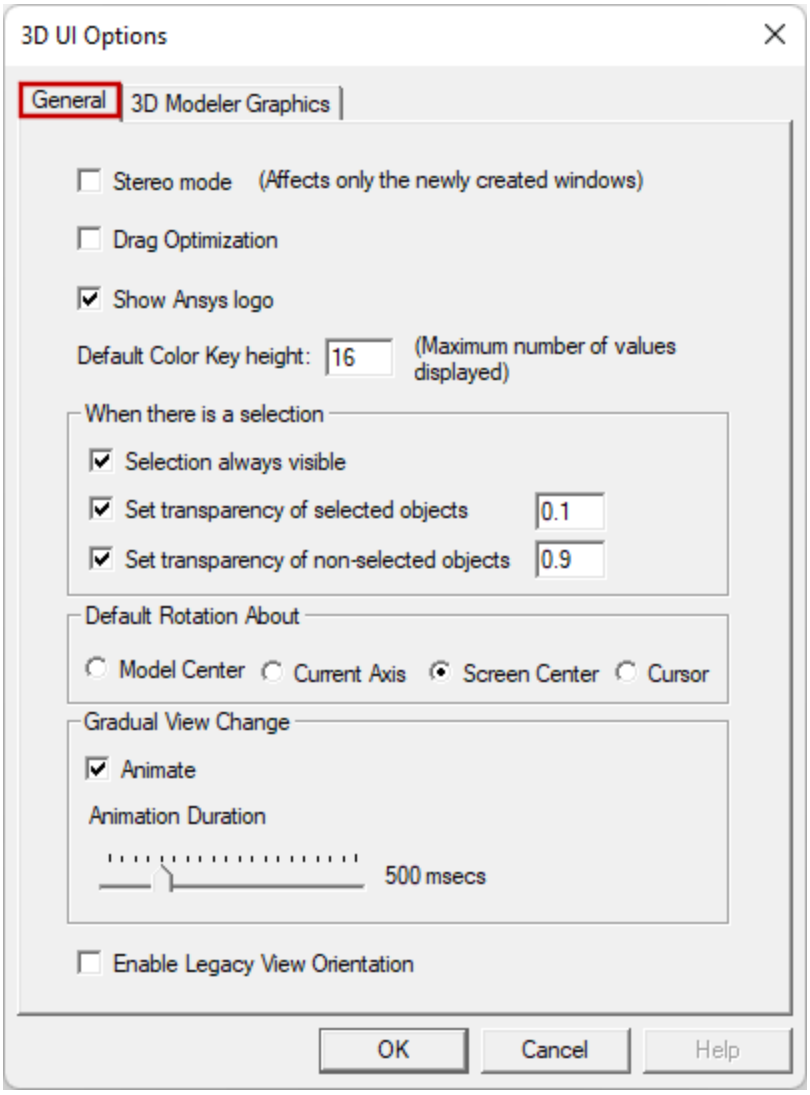


View Options: 3D UI Options

To access View Options, use the **View** > **Options** command, or click **View Options** on the **View** ribbon.



The **3D UI Options** window appears.



The **General** tab lets you set defaults for the following view options:

- Stereo Mode (default, disabled) – This option is only applicable to systems with support for stereographic viewing for greater 3D realism.
- Drag Optimization (default, disabled)
- Show Ansys logo (default, enabled)
- Default Color Key height (default = 16)

When there is a selection option:

- Selection always visible (default, enabled)
- Set transparency of selected objects
- Set transparency of non-selected objects.

Default Rotation About options:

- Model Center
- Current Axis
- Screen Center (default)
- Cursor

  When you select one of these as the default, the **View** > **Rotate** selection menu changes to show that the **Alt+drag** combination attaches to your selection.

Gradual View Change settings:

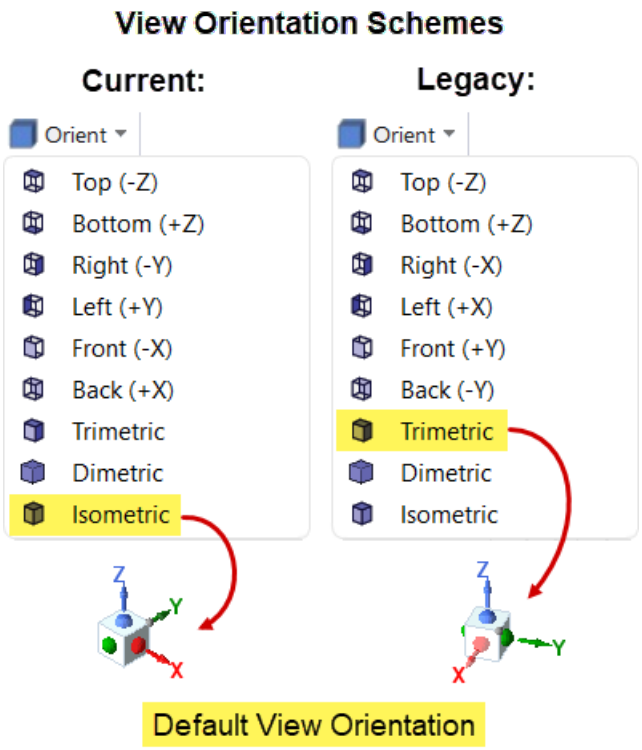
- Animate
- Animation Duration

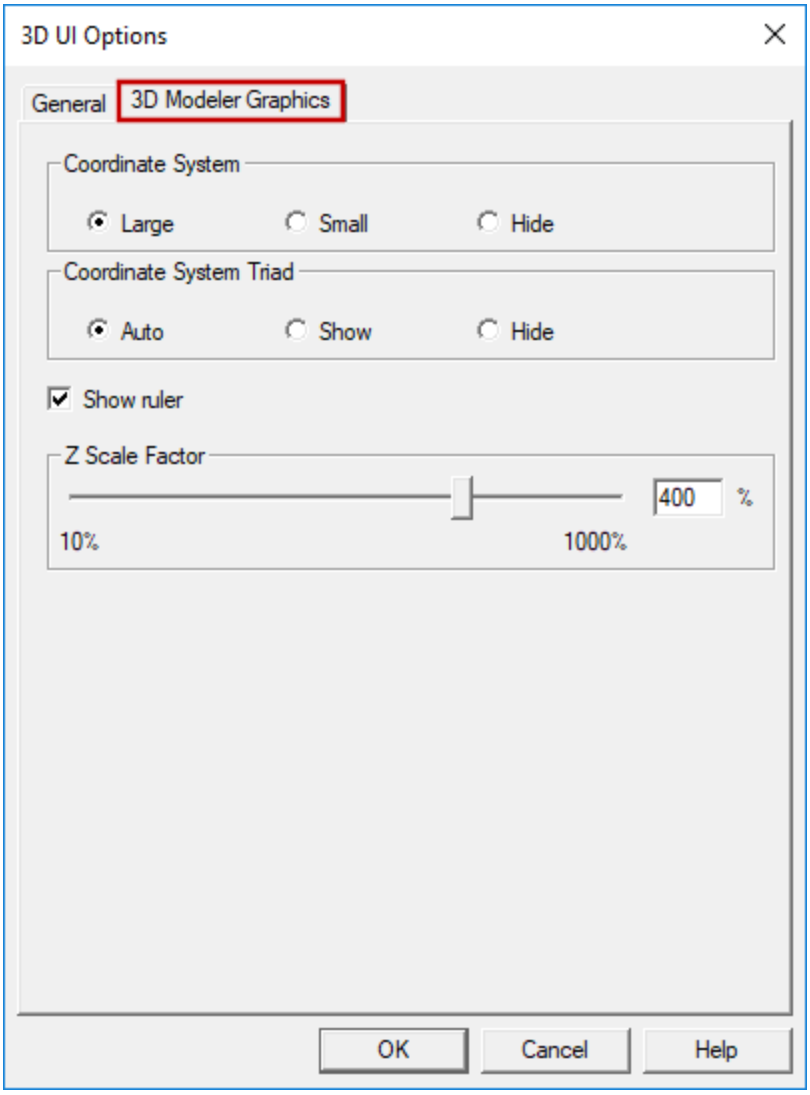
  For operations such as Fit Selection, Fit All, and Zoom, these settings determine whether Electronics Desktop animates the transition between images and how smoothly it does so.

Enable Legacy View Orientations option:

- When selected, the view orientations scheme and the axis directions for named views follow the legacy conventions (software version 2023 R2 or older):
- When cleared, the view orientations scheme and the axis directions for named views follow the current view orientation scheme

The following view is a comparison of the two options:





The **3D Modeler Graphics** tab lets you set the following options which are invoked for the next Project and design created.

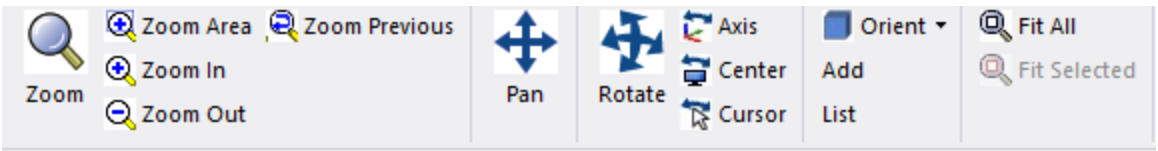
Settings	Description of Choices
Coordinate System	• **Large** (extends to the window edges) • **Small** (centered, not extending toward edges with smaller arrows) • **Hide** (do not display)
Coordinate System Triad (A secondary, smaller coordinate system appearing near the lower right corner of the Modeler window)	• **Auto** • **Show** • **Hide**
Show Ruler (To indicate the scale of the model, based on current units. The ruler appears at the lower center of the Modeler window)	Select to enable; clear to disable
Note: You can also toggle the ruler visibility by clicking **Ruler** on the **Draw** ribbon tab or by right-clicking in the *Modeler* window and choosing **View** > **Ruler** from the shortcut menu.	
Z Scale Factor	The default is 400%. You can use the slider or the text field to specify a different value.

Panning the View

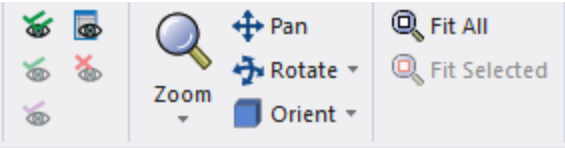
To move (pan) the view:

1. Click **View** > **Interaction** > **Pan** or click the **Pan** icon on the **View, Draw,** or **Model** ribbon tab.

   View ribbon tab:



   Draw or **Model** ribbon tabs:



2. Click and drag the mouse in the direction you want to pan the view.

   The view will pan until you release the left mouse button.

3. To exit **Pan** mode, click **Pan** on the **View** menu again or press **Esc**.

> **Tip:**
>
> Alternatively, pan the view using one of the following methods:
>
> - Hold down **Ctrl** as you click and drag using the **middle mouse button**.
> - If you chose to select *Enable Legacy View Navigation* in the General > User Interface options, you can also use the legacy combination **Shift** + **drag** (with the left mouse button) to pan the view.
> - Press **Ctrl** + ***arrow keys*** to pan vertically or horizontally. The movement is 10 screen pixels per press of the arrow key.
> - In addition to the menu bar and ribbon tabs, you can access *Pan* from a shortcut menu. Right-click in the Modeler window and click **View** > **Pan**. Click the command again or press **Esc** to terminate panning.

Zooming in and Out

You can magnify (zoom in) or shrink (zoom out) the contents in the view window using the following methods:

- Hot Keys (that work without the mouse)
- Hot Keys (used in conjunction with clicking and dragging the mouse)
- **Zoom In** and **Zoom Out** commands, which produce 5% zoom increments
- **Zoom Previous** to undo recent zoom commands, as described below
- Zoom into a Rectangular Area
- Zoom in or out Using a Mouse Wheel
- Zoom to a Selected Boundary or 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 **Ctrl** + "**+**"

  If using the "=\+" key (not the "+" key on the numeric keypad) do **not** press the *Shift* key.

  The view zooms in 5 percent.

To zoom out using hotkeys:

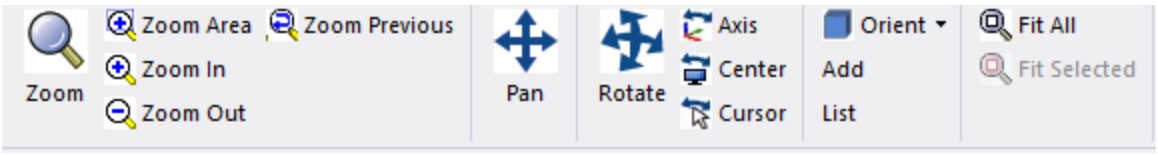
- Press **Ctrl** + "**-**"

  The view zooms out 5 percent.

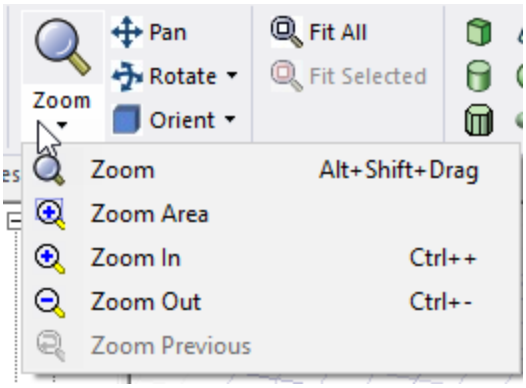
To zoom by dragging the mouse.

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** commands.



   Draw or **Model** ribbon tabs with **Zoom** icon, and drop-down menu for Zoom commands:



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**.

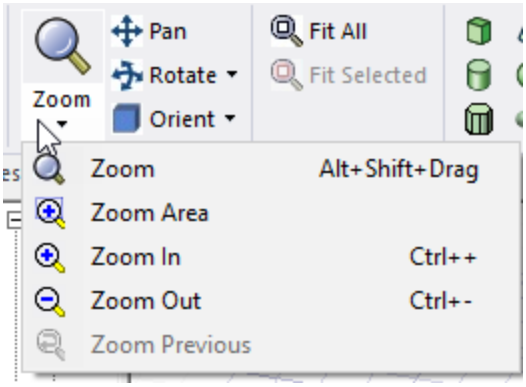
Zoom In and Zoom Out Commands

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.

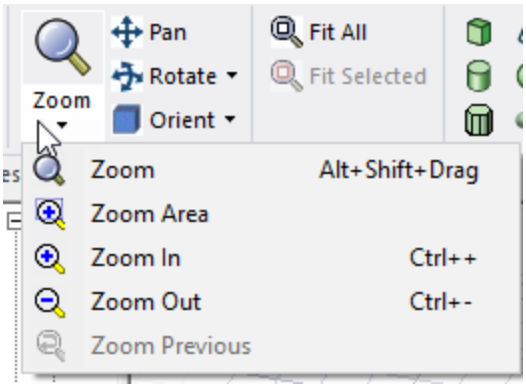
   When zooming on a view of model objects the absolute size of the model does not change.

   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

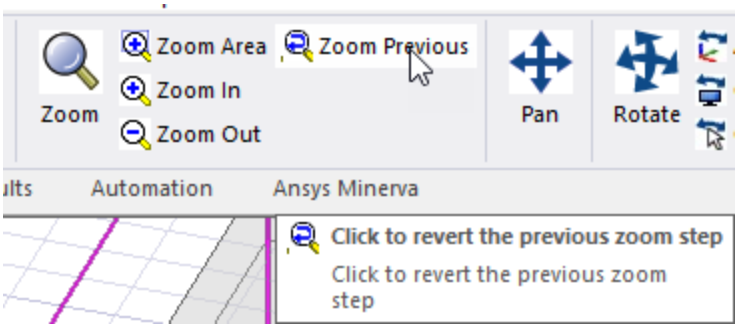
The **Zoom Previous** command is not enabled until at least one **Zoom** command has been used.



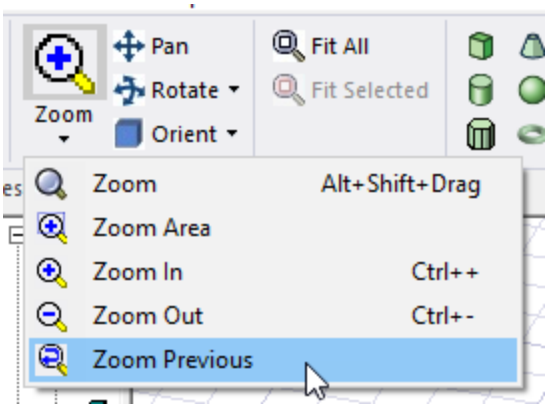
It operates on any existing sequence of **Zoom** commands. That is, it undoes the last previous **Zoom** command, and then, another exists, the one before that. To use **Zoom Previous**.

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 **Shift** as you click and drag using the **middle mouse button**. Moving the cursor upward zooms in and moving it downward zooms out.
- If you chose to select *Enable Legacy View Navigation* in the General > User Interface options, you can also use the legacy mouse-button and hotkey for zooming. Hold down **Alt** + **Shift** as you click and drag (using the left mouse button).
- Roll the mouse wheel. Rolling the top of the wheel forward (away from you) moves the model view further away from you (zooming out). Conversely, rolling the top of the wheel rearward (towards you) zooms in.
- Right-click in the Modeler window and then click **View** > **Zoom** on the shortcut menu. Click and drag using the left mouse button to zoom in or out. Repeat the command or press **Esc** to cancel zooming.

Note:

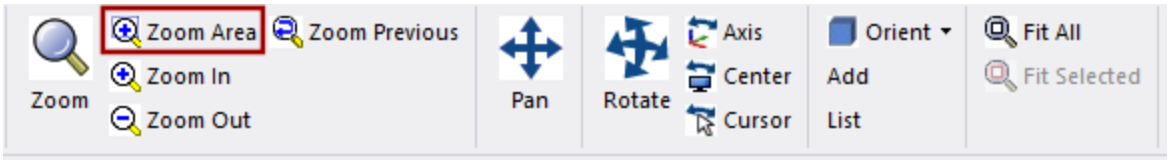
If zooming is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel** > **3D Settings** > **Manage 3D Settings** > **Global Settings** > **Global Presets: Workstation App - Dynamic Streaming**

For more information about graphics cards, navigate to the appropriate Windows or Linux Ansys Installation Guide and search for **OpenGL**.

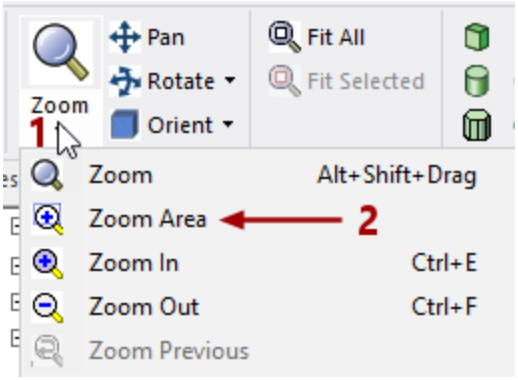
Zooming into a Rectangular Area

Zoom to expand a specified rectangular area to fill the view window, as follows:

1. Use one of the following methods to begin the *Zoom Area* mode:
   - Using the menu bar, click **View** > **Interaction** > **Zoom Area**.
   - From the **View** ribbon tab, click **Zoom Area**:



   - From the **Draw** or **Model** ribbon tab, click **Zoom** > **Zoom Area**:



   - Right-click in the Modeler window and choose **View** > **Zoom Area** from the shortcut menu.
2. Click and drag the mouse to draw a rectangle enclosing the area that you want to zoom into.

When you release the mouse button, the rectangular area will come close to filling the display area of the window.

The absolute size of the zoomed geometry does not change, only the scaling of the model view is affected.

3. To end the **Zoom Area** mode, click the command again or press **Esc**.

   The rotate and zoom drop-down menu icons on the ribbon update dynamically to indicate the current rotate/zoom mode. These ribbon buttons are also "sticky." That is, the last active rotate or zoom mode becomes the button default until a different command is chosen or the application is restarted. Default commands are restored the next time you launch the Ansys Electronics Desktop program.

   For 2D or 3D reports, axis labels and tick marks are adjusted after the zoom operation is complete.

Zooming in or out Using a Mouse Wheel

You can zoom the active model view in and out by rotating the mouse wheel. This action works in the Modeler window, for 2D and 3D plots, and more. The cursor location is the center point whenever using the mouse wheel for zooming, and this point remains fixed. The grid and ruler, when visible, are adjusted dynamically as you zoom in and out.

Rotating the wheel forward, away from you, moves the model away from you (zooming out). Rotating the wheel backward, towards you, pulls the model closer (zooming in).

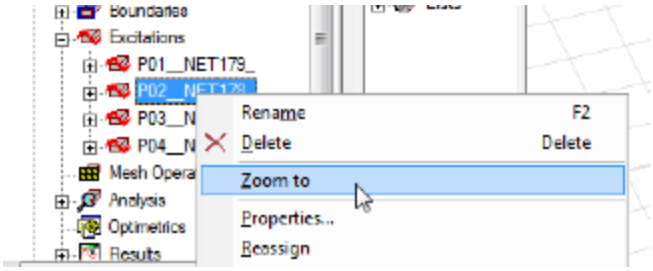
> **Note:**
>
> If Zooming is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel** > **3D Settings** > **Manage 3D Settings Global Settings** > **Global Presets: Workstation App - Dynamic Streaming**
>
> For more information about graphics cards, navigate to the appropriate Windows or Linux Ansys Installation Guide and search for **OpenGL**.

Zoom to a Selected Boundary or Excitation

Zoom to is an option that can enlarge the view of a boundary or excitation. This command can be very useful for locating small items with boundaries or excitations assigned or to inspect any problem areas.

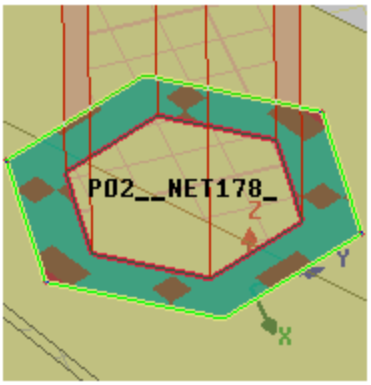
You can use the **Zoom to** command as follows.

1. If needed, expand the *Boundary* or *Excitation* branch in the Project Manager to see the previously assigned conditions.
2. Right-click the boundary or excitation of interest to highlight it in the model view and to access the shortcut menu, as shown below:



3. Select **Zoom to**.

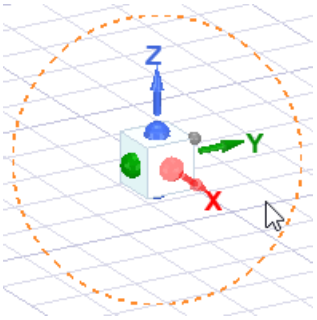
   Notice that the command instantly increases magnification, filling the screen with the object(s) or face(s) to which the boundary or excitation is assigned, as shown in the following figure:



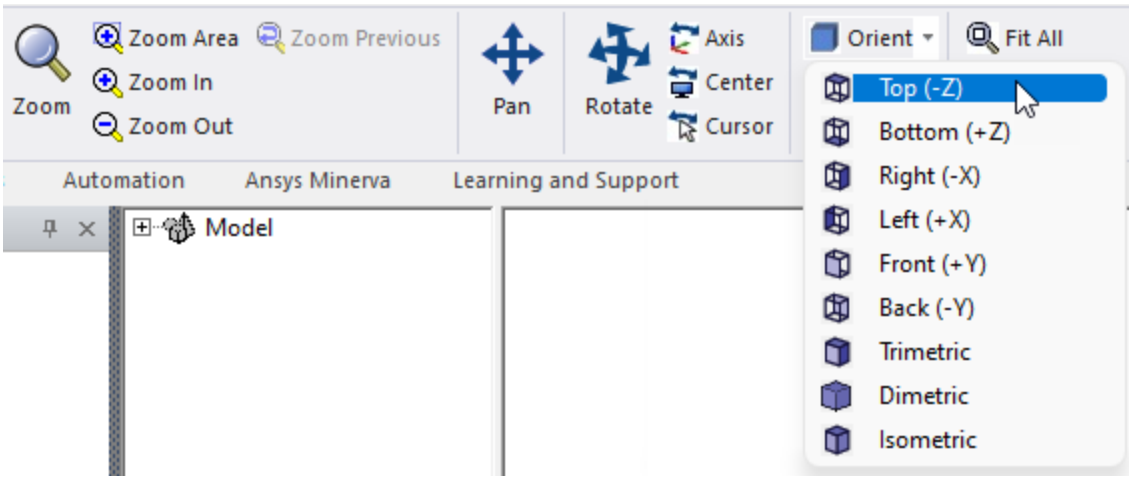
Modifying the View Orientation

There are several places you can access commands to change the view orientation in a *Modeler* or *3D Plot* window:

- Use the View Orientation Gadget:



- Select the **Orient** drop-down Menu on the **View** ribbon tab and choose one of the predefined view orientations:



  The **Orient** drop-down menu can also be located on the **Draw** and **Model** ribbon tabs.

  The sign and letter in parentheses that follows each standard view indicates the viewing direction, not the model side. For example, consider a simple box aligned with the global planes. The right side of the box is the +X side, but the viewing direction is -X (you look leftward to see the right side of an object).

- 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 the 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.

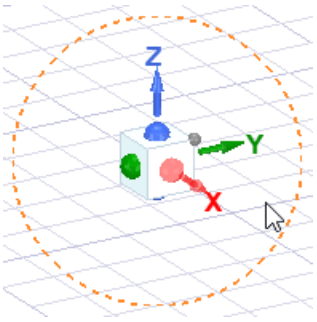
1. Select a default orientation or create a new orientation.
2. Click **Apply to View** for the selected view to appear in the window.
3. Optionally, click the **Reset View Orientation** button to restore the selected view to the original angles.
4. Click **Make Default** if you want the selected viewing direction to be the initial viewing direction when a **3D Modeler** window is opened, either in the current project or future projects.
5. Click **Close** to dismiss the dialog box.

   The orientation you set will be saved with the design. New orientations assigned to other designs after this point will not affect this orientation.

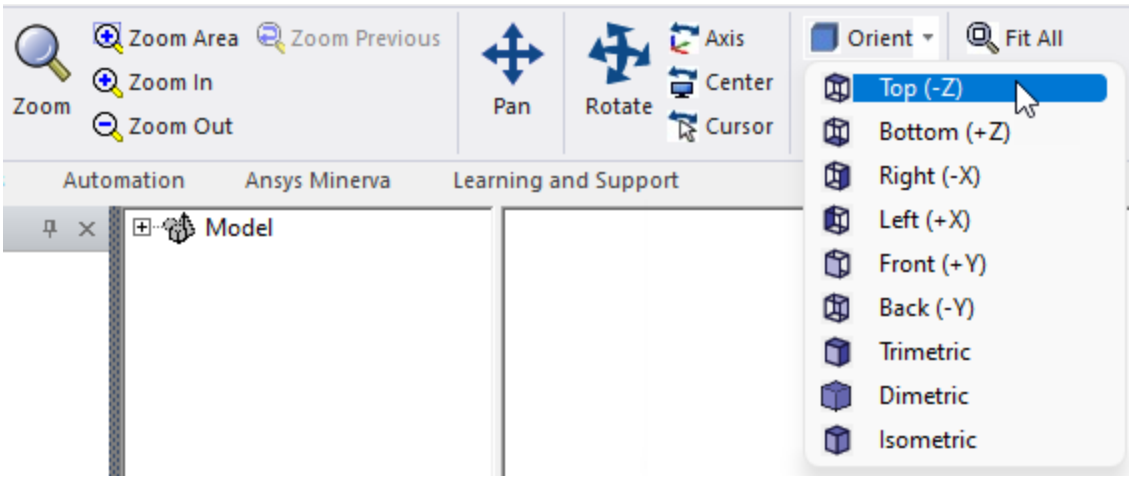
Apply an Orientation to the Current View

Apply view orientation settings using any one of several available methods:

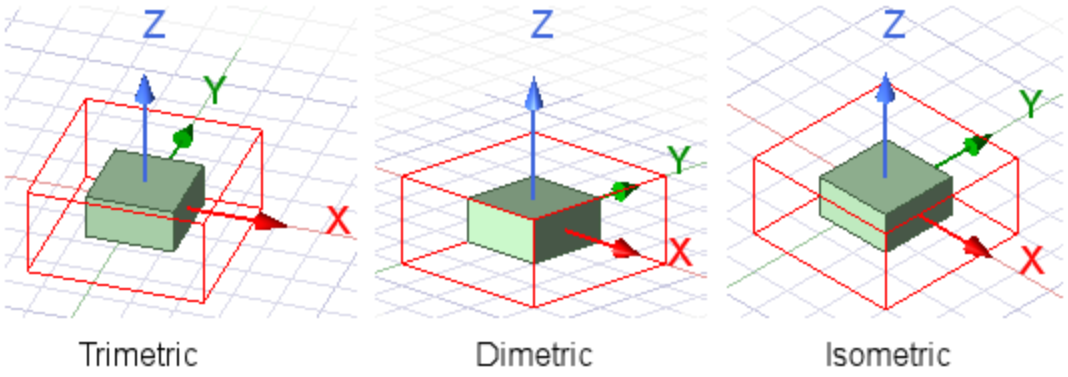
- Use the Orientation Gadget:



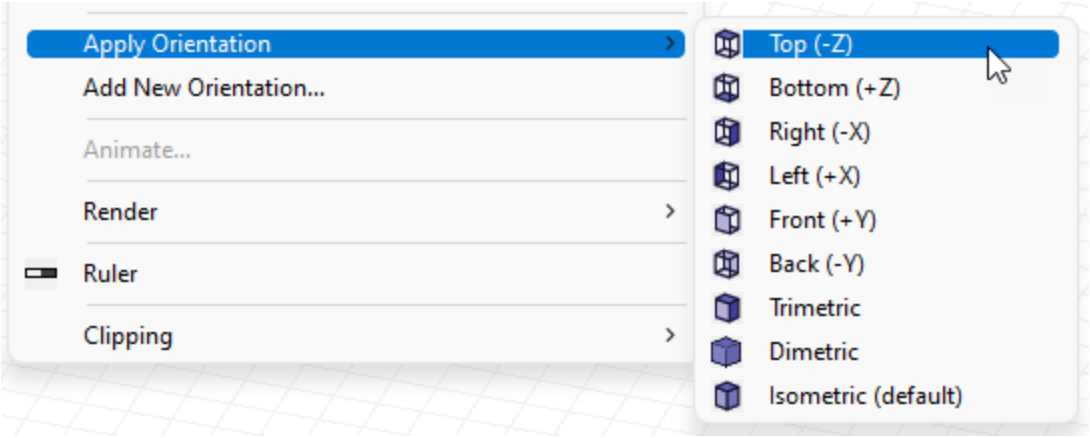
- Use the commands in the **Orient** drop-down menu located on the **View, Draw,** or **Model** ribbon tabs:



Possible orientations are Top, Bottom, Right, Left, Front, Back, Trimetric, Dimetric, and Isometric. These last three differ as shown below:



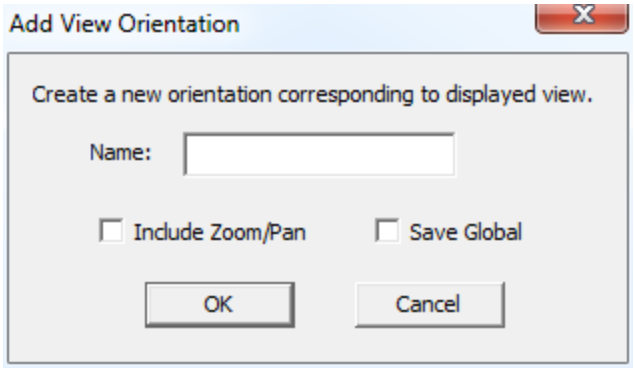
- Right-click in the Modeler window. In the shortcut menu that appears, go to the **View** > **Apply Orientation** submenu:



In addition to the standard views, any custom views that you have defined will appear in this submenu.

After applying an orientation, you can click **View** > **Undo View : *orientation***.

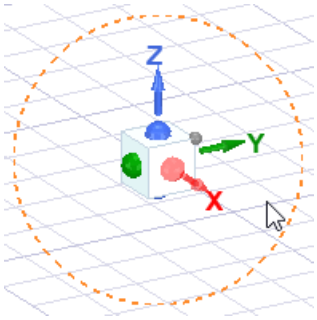
You can also right-click in the Modeler window and choose **View** > **Add New Orientation** to create a custom viewpoint.



The **Name** that you assign will appear in the **View** > **Apply Orientation** submenu within the Modeler window's shortcut menu. You can specify whether to **Include Zoom/Pan** (in addition to the viewpoint orientation) and whether to **Save Global**, which makes the custom orientation available for all projects.

Using the Orientation Gadget

The **Orientation Gadget** is a tool for quickly manipulating the view orientation of 3D models and 3D plots graphically. It is a convenient alternative to using the **Orient** drop-down menu on the **View** ribbon tab. The Orientation Gadget is enabled by default and appears near the lower left corner of the display area in *Modeler* and *3D Plot* windows:



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) axes, indicating the axis positive directions for the currently active coordinate system. 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):

- **Click the large dot at the center of any of the six cube faces:**

  This action rotates the view orientation 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, which is the *Left* view (+Y viewing direction).

  > **Note:**
  >
  > If the current viewing direction corresponds to one of the global or active user-defined coordinate system axes (±X/Ux, ±Y/Uy, or ±Z/Uz), clicking the axis label or dot at the center of the Orientation Gadget (on the face that is towards you) reverses the view orientation. So, with a single click of the mouse, you can switch from front-to-back, right-to-left, and top-to-bottom (or the inverses of these view orientation changes).

- **Click on one of the gadget axes:**

  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 (-X, -Y, or -Z viewing direction, respectively).

- **Click the small gray dot at the cube's +X, +Y, +Z corner:**

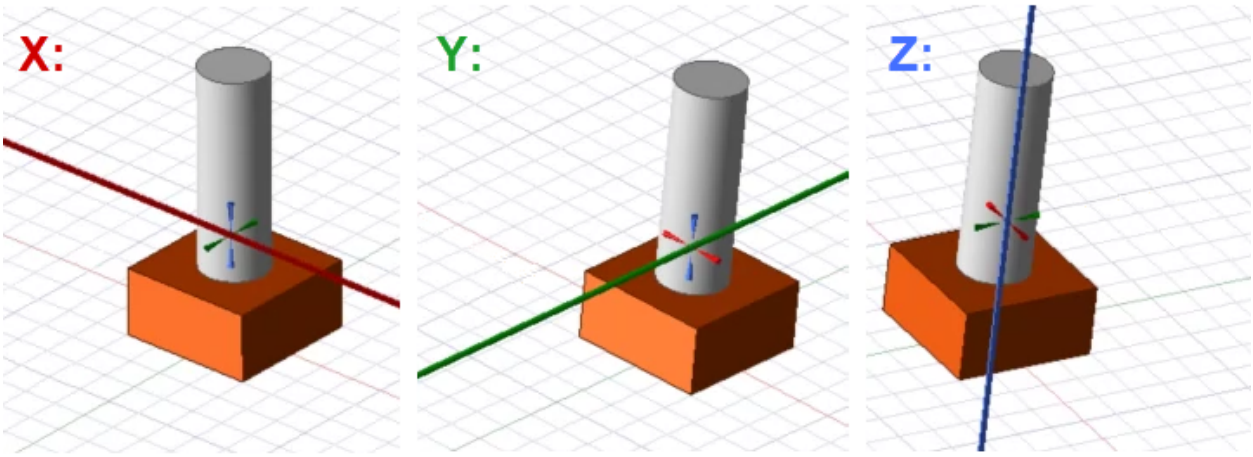
  This action produces the standard *Isometric* view orientation.

- **Click on or very close to the cube (but not on an axis or dot) and drag the mouse:**

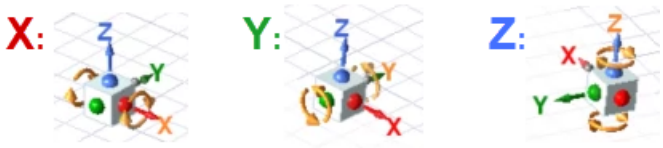
This action freely rotates the view orientation. The rotation center depends on the **Default Rotation About** option selected in the *3D UI Options* dialog box. The behavior is equivalent to clicking and dragging in the display area using the middle button / wheel but is performed with the left mouse button. Additionally, you can constrain rotation to be about any of the three active coordinate system axes by pressing one of the following keys while clicking and dragging the mouse:

- **X + click and drag** to rotate the view orientation about the active **X** axis.
- **Y + click and drag** to rotate the view orientation about the active **Y** axis.
- **Z + click and drag** to rotate the view orientation about the active **Z** axis.

During constrained rotation, a heavy red (X), green (Y), or blue (Z) axis line appears in the Modeler window to provide a clear visual indication of the axis of rotation. These axes pass through the center of rotation, and this center is marked by crosshairs. When dragging the view orientation gadget to rotate the view, the center of rotation is the centerpoint of the drawing canvas. The following image shows the appearance of the rotation axes and center markers:



Additionally, a pair of gold, circular arrows appear on the view orientation gadget indicating the axis of constrained rotation:



While the same constrained rotation functionality for 3D models is also available for 3D reports, the red, green, and blue axes do not appear in 3D report windows during axis-constrained rotation. However, the pair of circular, gold arrows appear on the view

orientation gadget in both the Modeler and 3D Report windows to indicate the axis of constrained rotation.

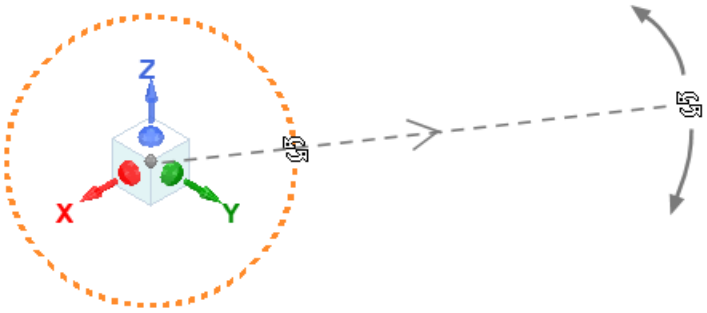
For more information and tips about the usage of this feature, see Rotating or Spinning the View.

- **Orange dashed circle:**

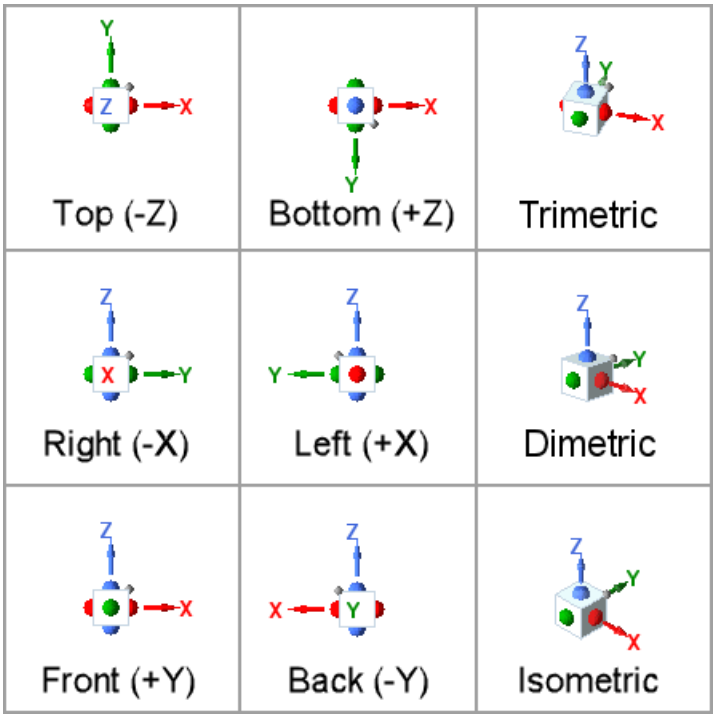
  This circle is used to rotate the current view about an axis normal to the screen. That is, the view rotation is constrained to the plane of the screen regardless of the current viewing direction. The line weight of the dashed orange circle becomes heavier when the cursor is touching it, indicating a correct clicking point. There are three actions, as follows:

  - **Click anywhere on the circle:** The model rotates **90 degrees clockwise**.
  - **Ctrl+click anywhere on the circle:** The model rotates **90 degrees counterclockwise**. This action and the preceding one are particularly useful for changing the in-plane positive axis directions for standard top, bottom, left, right, front, or back views.
  - **Click and drag:** Freely rotates the model in either direction. Release the mouse button when the desired orientation is reached. While the mouse button remains pressed, the cursor changes to a rotation icon:

    You do not have to keep the cursor on the circle while dragging. You can often achieve more precise rotations with the cursor further away from the orientation gadget. Moving the cursor in a radial direction away from the center of the orientation gadget produces minimal rotation. Then, move it tangentially for precise rotation. The greater the distance from the gadget, the finer the rotational control.



The appearance of the Orientation Gadget for each of the standard view orientation is as shown in the following image:

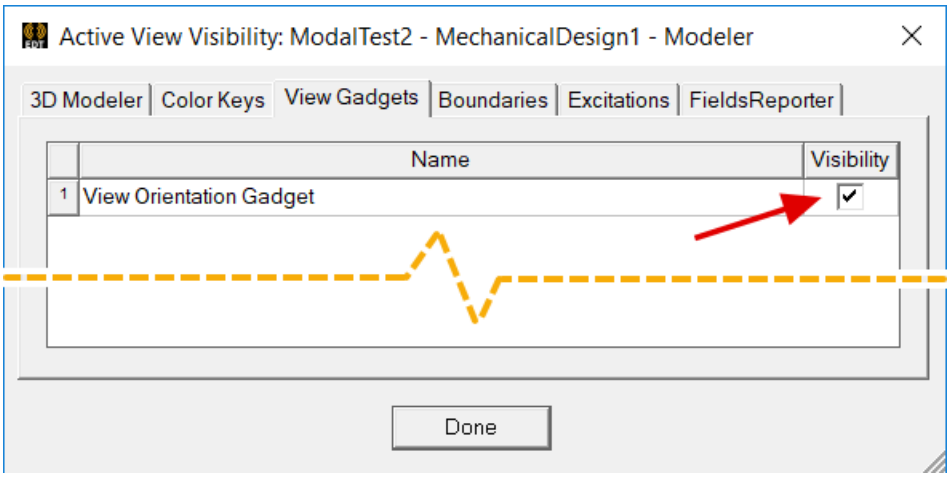


> **Note:**
>
> The orientation, colored dots, and axes of the Orientation Gadget correspond to the active coordinate system, whether **Global** or **user-defined**. All standard views are defined in the same manner as they are for the global CS, but Ux, Uy, and Uz directions are used instead of global X, Y, and Z.

Controlling Visibility of the Orientation Gadget

You can enable or disable the Orientation Gadget visibility 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.

Rotating or Spinning the View

You can rotate the view relative to the Model Center, the Screen Center, the Current Axis (that is, the active coordinate system origin), or the Cursor location. You can activate the rotation modes via the menu bar, ribbon tabs, or shortcut menu. When a *Rotation* command is active, you can click and drag using the **left mouse button** to manipulate the model viewpoint.

Rotation Using Only the Mouse: You can use the **middle mouse button** to rotate the view *without* selecting a *Rotation* command. From the menu bar, click **View** > **Options** to access the *General* tab of the *3D UI Options* dialog box. Here, you can choose the desired **Default Rotation About** option. The view rotates about the **Screen Center** by default. The other available options are rotation about the **Model Center**, **Current Axis** (active coordinate system origin), or **Cursor** location. This setting also controls the rotation center for the **Spin** command.

Rotation using Only the Keyboard: You can use the key combination **Alt** + ***arrow keys*** to rotate the model view about a vertical or horizontal axis. Similar to using the middle mouse button, the center of rotation, through which the vertical or horizontal rotation axis runs, is determined by the **Default Rotation About** selection in the *3D UI Options* dialog box.

For each active *Rotation* command, or when using the middle mouse button, the behavior differs depending on where you click and drag, as follows:

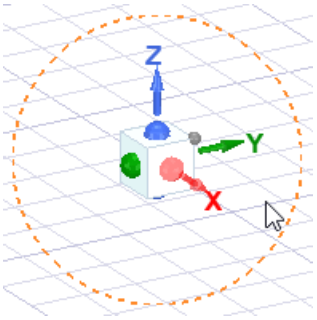
- **Click and drag the cursor near any border of the Modeler window's display area:**

  The view is constrained to rotate about an axis that is perpendicular to the screen. The location of the axis, which is the rotation centerline, depends on the active rotation mode (Current Axis, Model Center, Screen Center, or Cursor) or, when no command is active, the *Default Rotation About* option previously described.

- **Click and drag the cursor within the interior of the display area (not close to a border):**

In this case, the view rotates freely about any axis, depending on the direction you drag and the relative positions of the cursor and center of rotation. The rotation center depends on the active rotation mode (Current Axis, Model Center, Screen Center, or Cursor) or, when no command is active, the *Default Rotation About* option previously described.

- **Click and drag on or near the View Orientation Gadget to freely rotate the view:**

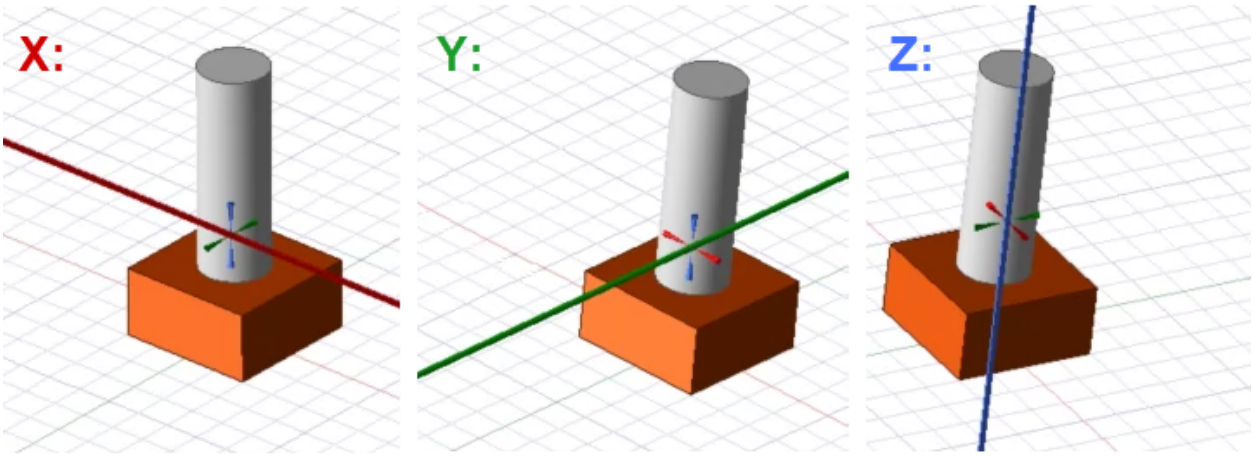


- **Constrained rotation about the X, Y, or Z axis**:

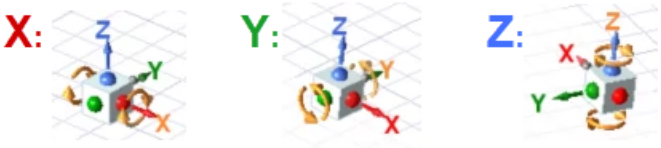
  Use the following hotkeys to constrain rotation of the model viewpoint to be about a coordinate system axis (global or user-defined). These hotkeys work whether you perform the rotation using the **middle mouse button**, any of the **Rotate** commands available on the ribbon (under the *View, Draw,* or *Model* tabs), or any of the **Rotate** commands in the menu bar (**View** > **Interactions** > **...**).

  - **X + click and drag** to rotate the view orientation about the active **X** axis.
  - **Y + click and drag** to rotate the view orientation about the active **Y** axis.
  - **Z + click and drag** to rotate the view orientation about the active **Z** axis.

  During constrained rotation, a heavy red (X), green (Y), or blue (Z) axis line appears in the Modeler window to provide a clear visual indication of the axis of rotation. These axes pass through the center of rotation, and this center is marked by crosshairs. When using the middle mouse button, the center of rotation is the centerpoint of the drawing canvas. For menu and ribbon commands, you can choose alternative rotation centerpoints (model center, cursor location, or origin), which are covered further down in this help topic. The following image shows the appearance of the rotation axes and center markers:



Additionally, a pair of gold, circular arrows appear on the view orientation gadget, indicating the axis of constrained rotation:



While the same constrained rotation functionality for 3D models is also available for 3D reports, the red, green, and blue axes do not appear in 3D report windows during axis-constrained rotation. However, the pair of circular, gold arrows appear on the view orientation gadget in both the Modeler and 3D Report windows to indicate the axis of constrained rotation.

This feature provides a very convenient means of achieving the ideal model viewpoint for setup operations and for presentation purposes, giving you much more control than free rotation.

Tip:

- For best results, drag the cursor in a roughly circular path around the rotation centerpoint. The model orientation will follow the cursor movement, with the angle of rotation matching the angle of the cursor's arc about the same centerpoint.
- Hold down the X, Y, or Z key throughout the constrained rotation operation. The axial rotation constraint is canceled as soon as you release the key, though the rotation mode remains active. Depending on how you initiated rotation, cancel it by either releasing the middle mouse button, pressing **Esc**, or clicking the command again.
- During an active constrained rotation operation, you can release the current hotkey and press one of the other hotkeys to choose a different rotation axis, without first canceling the operation.

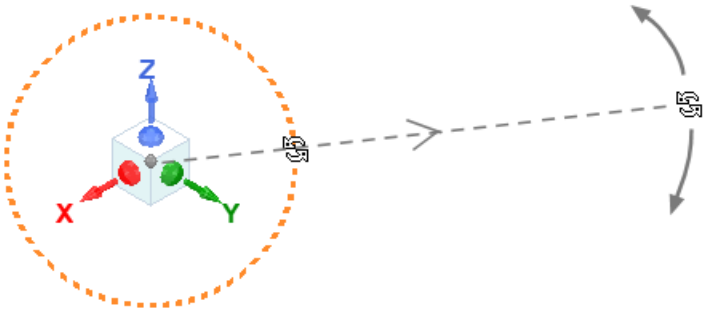
- **Rotating the view about an axis normal to the screen:**

  This orange dashed circle that appears around the view orientation gadget when the cursor is nearby is used to rotate the current view about the screen-normal axis. That is, the 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. There are three actions, as follows:

  - **Click anywhere on the circle:** The model rotates **90 degrees clockwise**.
  - **Ctrl + click anywhere on the circle:** The model rotates **90 degrees counterclockwise**. This action and the preceding one are particularly useful for changing the in-plane positive axis directions for standard top, bottom, left, right, front, or back views.
  - **Click and drag:** Freely rotates the model in either direction. Release the mouse button when the desired orientation is reached. While the mouse button remains pressed, the cursor changes to a rotation icon:

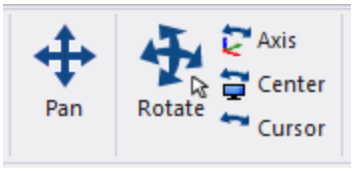
    You do not have to keep the cursor on the circle while dragging. You can often achieve more precise rotations with the cursor further away from the orientation gadget. Moving the cursor in a radial direction away from the center of the orientation gadget produces minimal rotation. Then, move it tangentially for precise

rotation. The greater the distance from the gadget, the finer the rotational control.



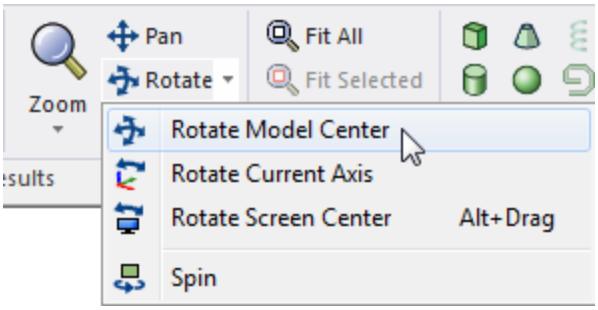
To rotate the view about the **Model Center**:

1. Choose the *Model Center* rotation mode using one of the following four methods:
   - From the **View** ribbon tab, choose **Rotate**:



> **Note:**
>
> The *Center* icon on the *View* ribbon tab is for rotation about the *screen center*, not the model center.

   - From the **Draw** or **Model** ribbon tab, choose **Rotate** > **Rotate Model Center**.



> **Tip:**
>
> After selecting a rotation mode, the selected option becomes the default operation for the remainder of the application session or until a different rotation option is selected. To repeat the last rotation command, you do not have to access the drop-down menu; just click **Rotate**:

- From the menu bar, click **View** > **Interaction** > **Rotate Model Center**.
- Right-click in the Modeler window and choose **View** > **Rotate** > **Rotate Model Center** from the shortcut menu.

2. Click and drag the mouse in the direction you want to rotate the view.
3. To exit the rotation mode, click the command again (per step 1) or press **Esc**.

Current Axis option – To rotate the view about the origin of the currently active coordinate system axes:

1. Choose the *Current Axis* rotation mode using one of the following four methods:
   - From the **View** ribbon tab, click **Axis**.
   - From the **Draw** or **Model** ribbon tab, choose **Rotate** > **Rotate Current Axis**.
   - From the menu bar, click **View** > **Interaction** > **Rotate Current Axis**.
   - Right-click in the Modeler window and choose **View** > **Rotate** > **Rotate Current Axis** from the shortcut menu.
2. Click and drag the mouse in the direction you want to rotate the view.
3. To exit the rotation mode, click the command again (per step 1) or press **Esc**.

To rotate the view around the **Screen Center**:

1. Choose the *Screen Center* rotation mode using one of the following four methods:
   - From the **View** ribbon tab, click **Center**.
   - From the **Draw** or **Model** ribbon tab, choose **Rotate** > **Rotate Screen Center**.
   - From the menu bar, click **View** > **Interaction** > **Rotate Screen Center**.
   - Right-click in the Modeler window and choose **View** > **Rotate** > **Rotate Screen Center** from the shortcut menu.
2. Click and drag the mouse in the direction you want to rotate the view.
3. To exit the rotation mode, click the command again (per step 1) or press **Esc**.

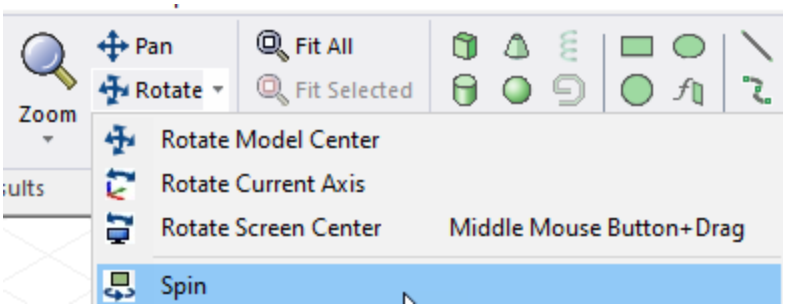
To rotate the view around the **Cursor** location:

1. Choose the *Cursor* rotation mode using one of the following three methods:
   - From the **View** ribbon tab, click **Cursor**.
   - From the menu bar, click **View** > **Interaction** > **Rotate Cursor**.
   - Right-click in the Modeler window and choose **View** > **Rotate** > **Rotate Cursor** from the shortcut menu.
2. Click and drag the mouse in the direction you want to rotate the view.
3. To exit the rotation mode, click the command again (per step 1) or press **Esc**.

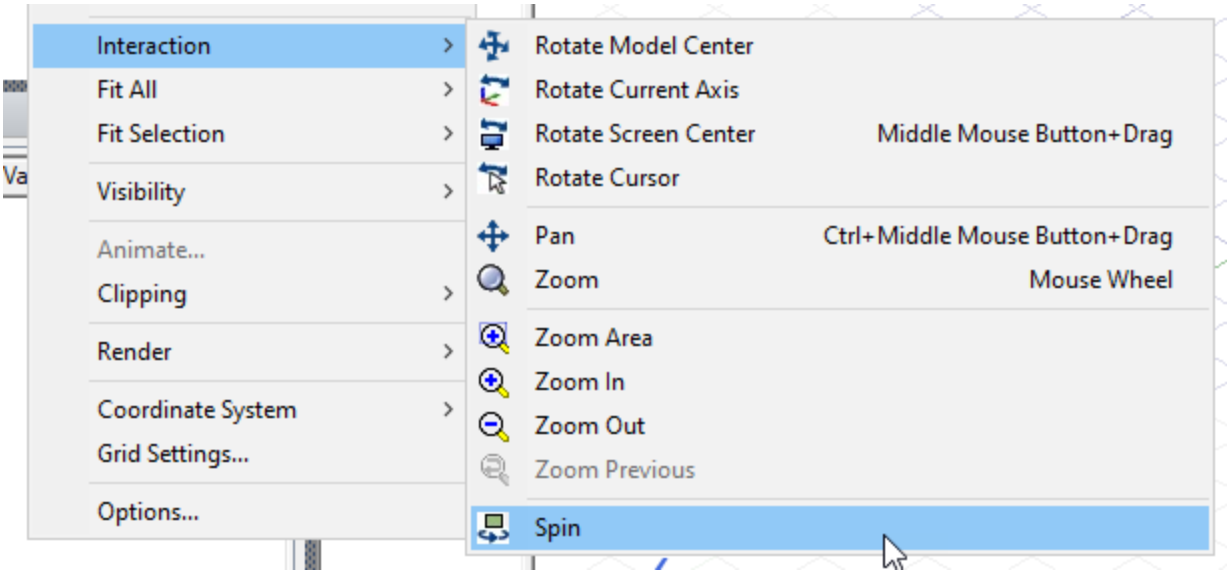
Spin Mode: This tool is useful for capturing animations (for presentation purposes) that show a 360 degree view of the model geometry, mesh, or simulation results.

To **Spin** the model view (that is, to keep it rotating after you release the mouse button):

1. Choose the *Spin* mode using one of the following two methods:
   - From the **Draw** or **Model** ribbon tab, choose **Rotate** > **Spin**.



   - From the menu bar, click **View** > **Interaction** > **Spin**.

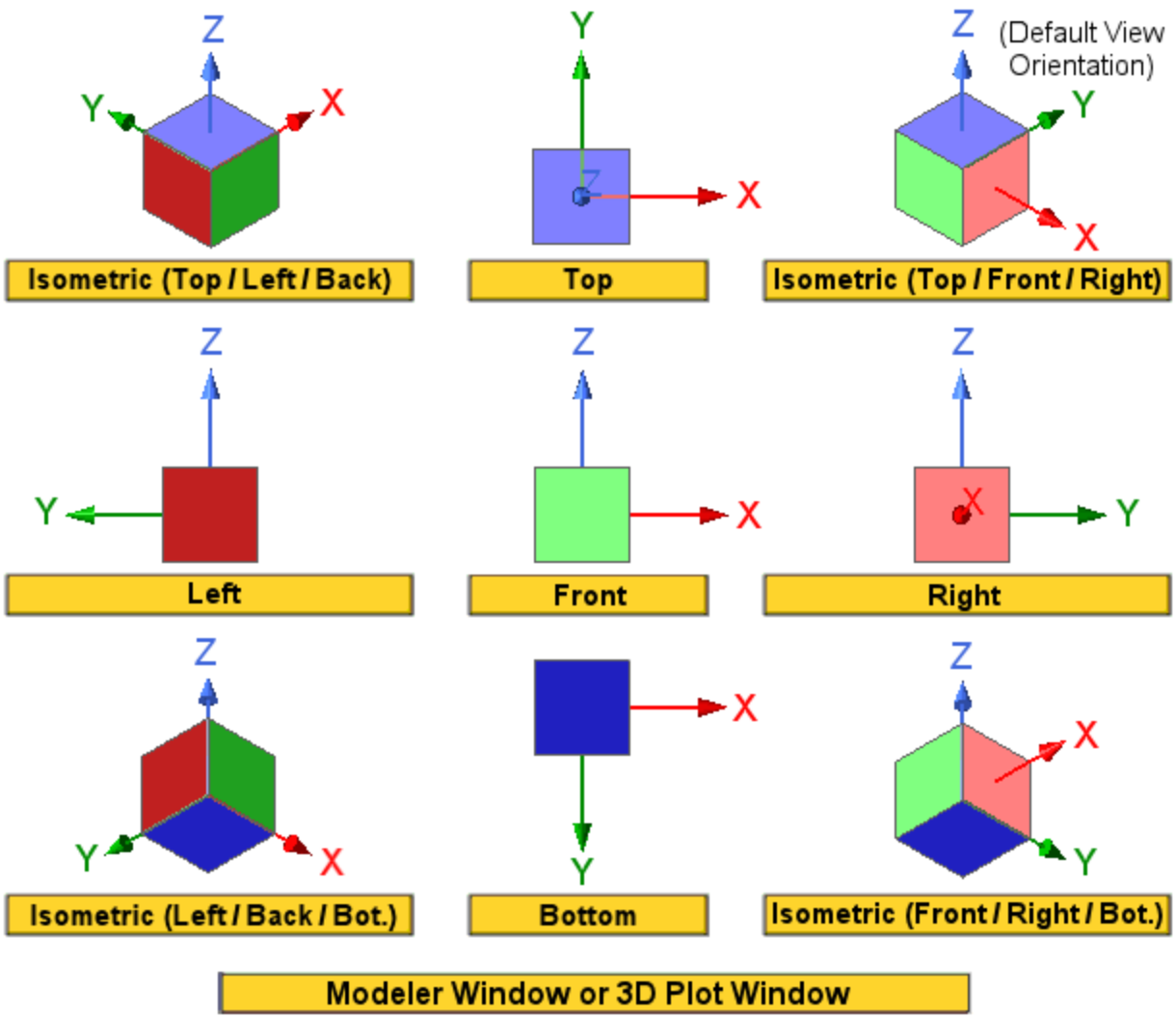


2. Click and drag the mouse in the direction you want to spin the view. Release the mouse button while still dragging.
    - The model will continue to spin in the specified direction. The spinning speed will be proportional to how fast you were dragging the cursor when you released the mouse button.
    - The center of rotation for the *Spin* mode is controlled by the *Default Rotation About* option selected in the 3D UI Options dialog box.
    - The *X, Y,* and *Z* keys also work as modifiers for the *Spin mode* (for constrained spinning about one of the active coordinate system axes).
3. To exit the *Spin* mode, click the command again (per step 1) or press **Esc**.

> **Note:**
>
> If rotation response is slow, especially for complex models:
>
> - For some graphics cards, you can improve the performance by setting **NVIDIA Control Panel** > **3D Settings** > **Manage 3D Settings** > **Global Settings** > **Global Presets: Workstation App - Dynamic Streaming**.
>
>   For more information about graphics cards, navigate to the appropriate Windows or Linux Ansys Installation Guide and search for **OpenGL**.

Changing the Model View with Alt+Double-Click Areas

In the following figure, the colored boxes show the nine view orientations you can obtain by using **Alt + double-click** in the corresponding areas of the *Modeler* window. Each face is a unique color and is named according to the standard view orientation names. The **Alt + double-click** view navigation method is also applicable to report windows containing *3D Plots* (such as a 3D Polar Gain plot).



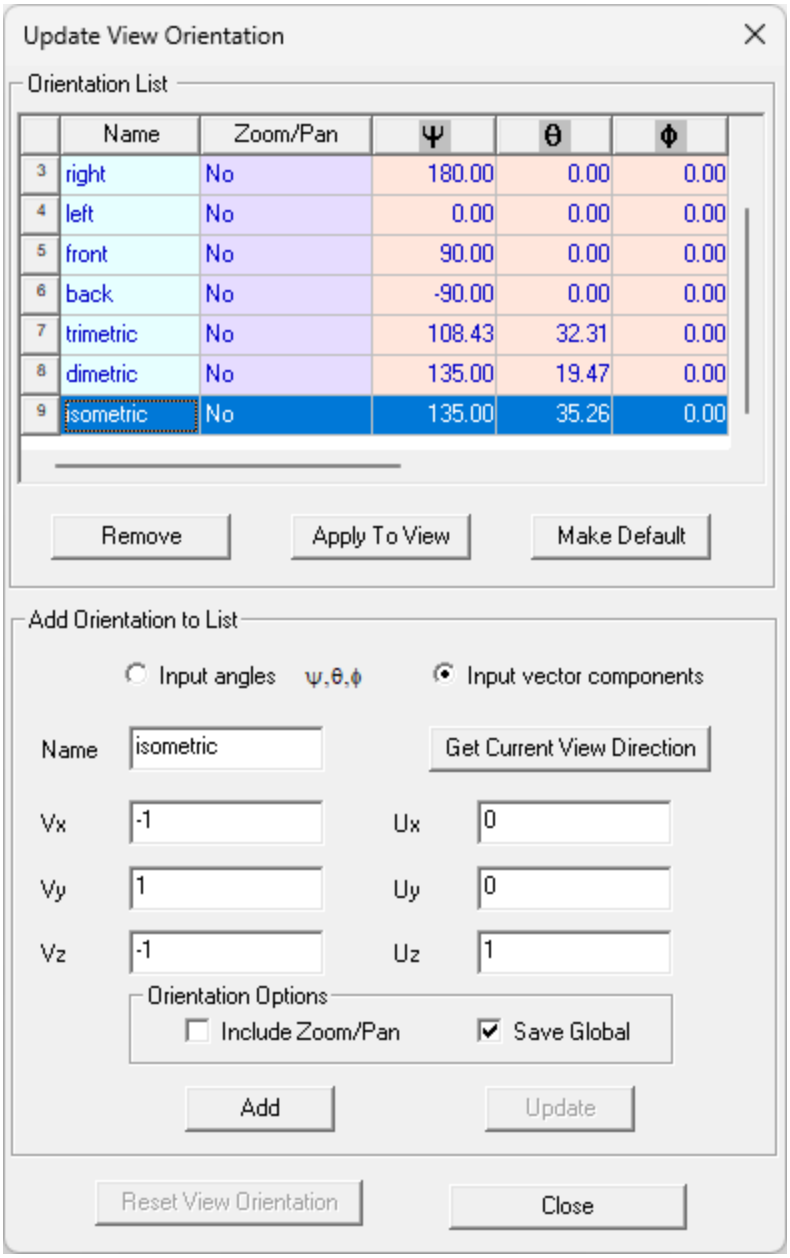
The *Back, Trimetric,* and *Dimetric* standard view orientations are not available via *Alt* + *double-click* zones. For these three views, use other available ribbon commands or shortcut menus. For user-defined views, use the *Update View Orientation* dialog box or the Modeler window's shortcut menu (*Right-click* > *View* > *Apply Orientation*).

Applying a Default View Orientation

To apply a default viewing direction to the active view window:

1. Access the *Update View Orientation* dialog box using one of the following two methods:
   - From the **View** ribbon tab, click **List**.

     No icon is associated with this command.
   - From the menu bar, click **View** > **Modify Attributes** > **Orientation List**.



2. Click one of the orientation names listed in the orientation list.
3. Click one of the orientation names listed in the viewing directions list.
4. Click **Apply**.

   The viewing direction will be applied to the active view window.

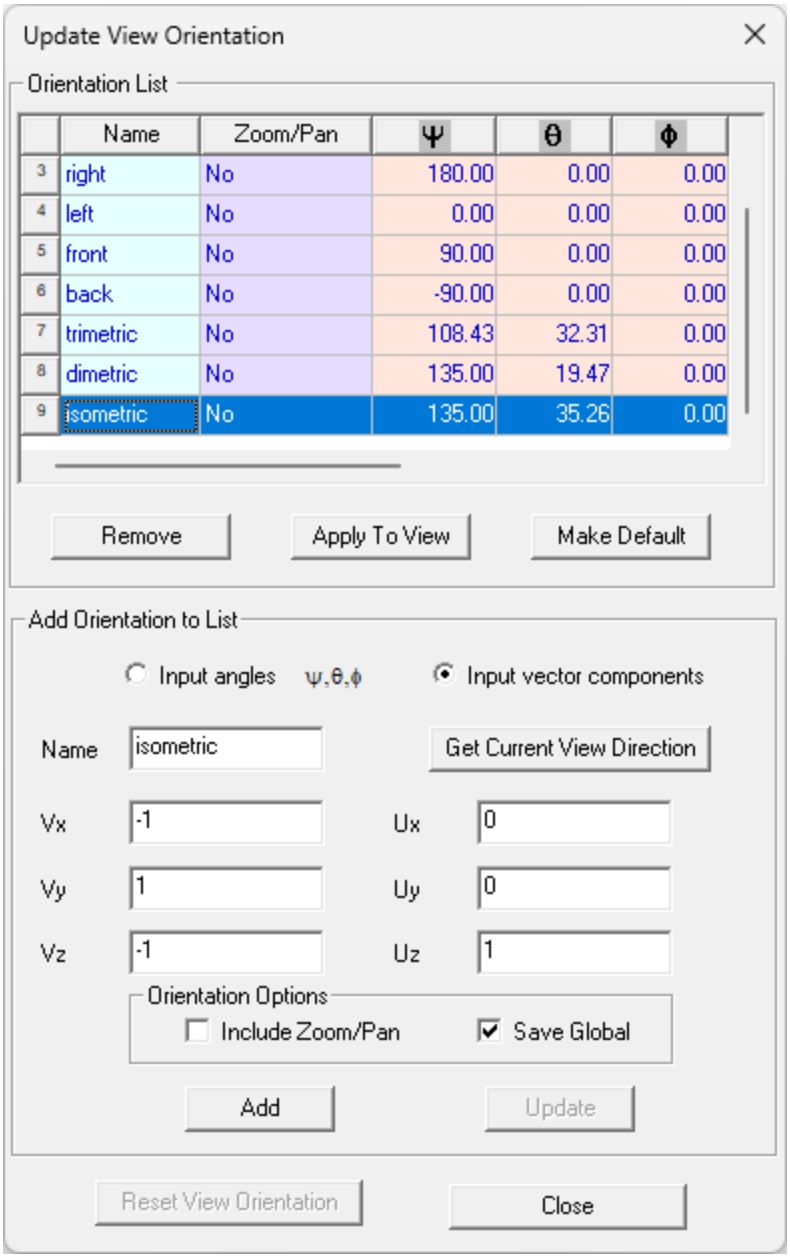
   You can use the **Reset View Orientation** button to restore the view to the original angle.
5. Click **Close**.

Applying a New View Orientation

To create and apply a new viewing direction to the active view window:

1. Access the *Update View Orientation* dialog box using one of the following two methods:
   - From the **View** ribbon tab, click **List**.

     No icon is associated with this command.
   - From the menu bar, click **View** > **Modify Attributes** > **Orientation List**.



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, phi** and **theta** text boxes.
3. Type a name for the new orientation in the **Name** text box.
4. Click **Add/Edit**.

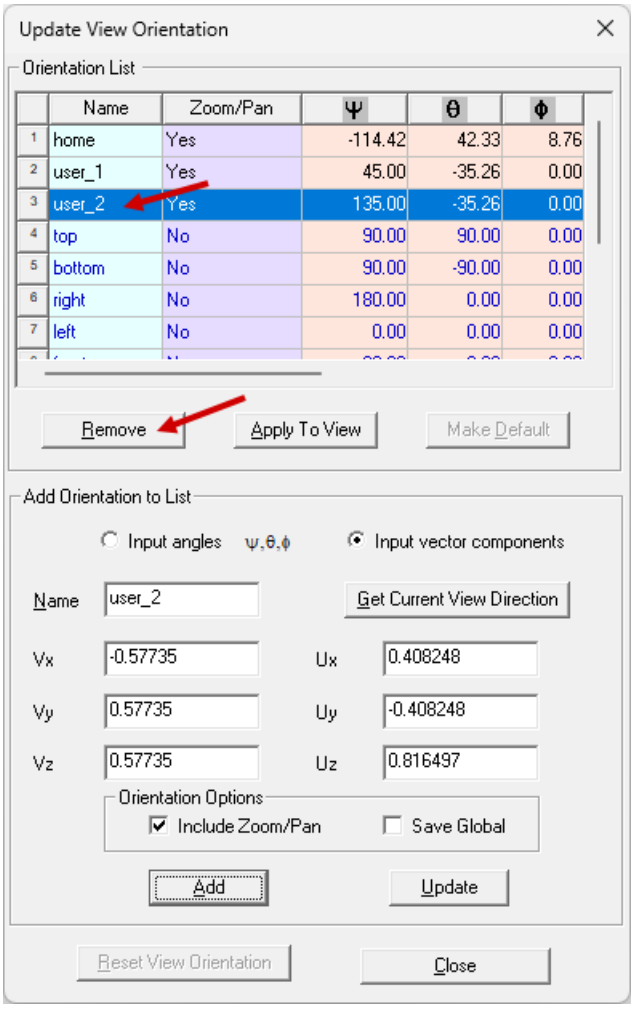
   The new orientation is added to the list of viewing directions.
5. Click **Make Default** if you want the new viewing direction to be the initial viewing direction when a **3D Modeler** window is opened in the current project or future projects.
6. Click **Close**.

Removing an Orientation

To remove a viewing direction from the list in the orientation settings dialog box:

1. Access the *Update View Orientation* dialog box using one of the following two methods:
   - From the **View** ribbon tab, click **List**.

     No icon is associated with this command.
   - From the menu bar, click **View** > **Modify Attributes** > **Orientation List**.



2. Click the viewing orientation you want to delete from the list of names.
3. Click **Remove**.

   The viewing direction is removed from the list.

> **Note:**
>
> This operation cannot be undone.

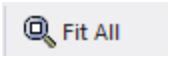
Fitting Objects in the View Window

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** or use the equivalent icons in the **Draw** or **Model** ribbon tabs.



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.

> **Tip:**
>
> Alternatively, fit all objects in the active view window using one of the following methods:
>
> - Press **Ctrl+D**
> - Right-click in the Modeler 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.

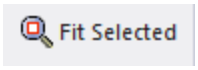
Fitting Selected Objects in a View Window

To fit one or more selected objects, 3D Components, or User Defined Models in the Modeler window or traces selected in the Reporter:

1. When you are working on a model view, select the objects 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.**

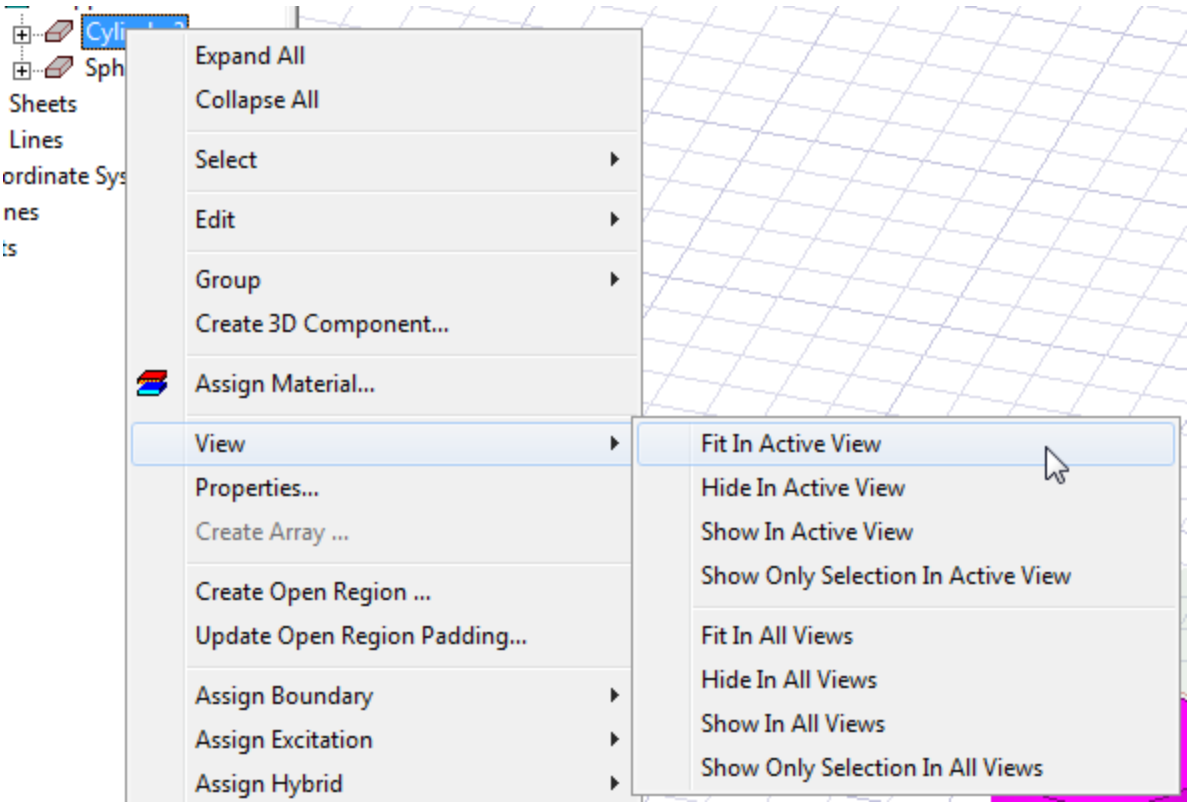
     > **Tip:**
     >
     > Alternatively, fit the selection in the active view window by clicking **View** > **Fit Selection** on the shortcut menu.

   You can also use the Fit Selected icon in the **Draw** and **Model** ribbon tabs:



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 shortcut menu.
2. Select **View** > **Fit in Active View** or **Fit in All Views**:



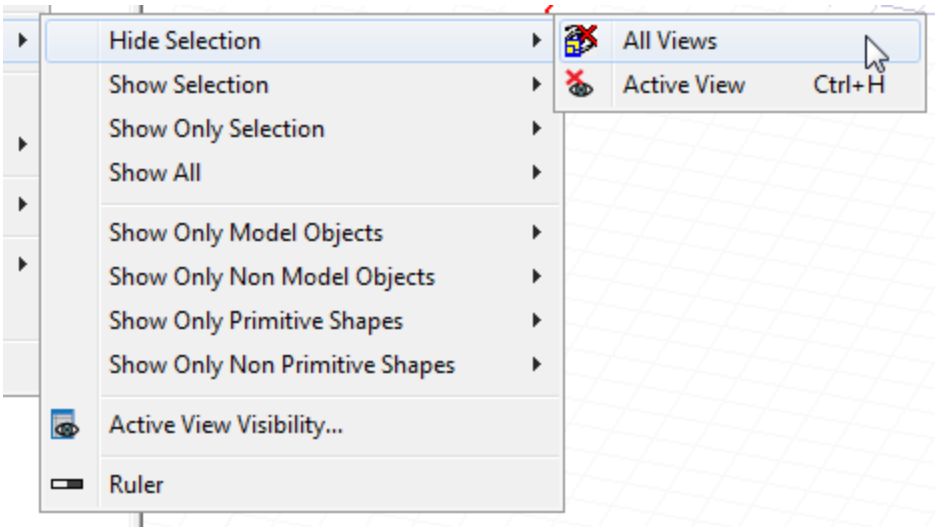
.The view adjusts to fit the select objects.

Hiding Objects from View

To hide selected objects.

1. Select the object you want to hide from view. This enables the menu commands and **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** icon in the **View, Draw,** or **Model** ribbon tabs to hide selected objects in 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 choose to Show only selected objects in all or active views, effectively hiding all unselected objects.

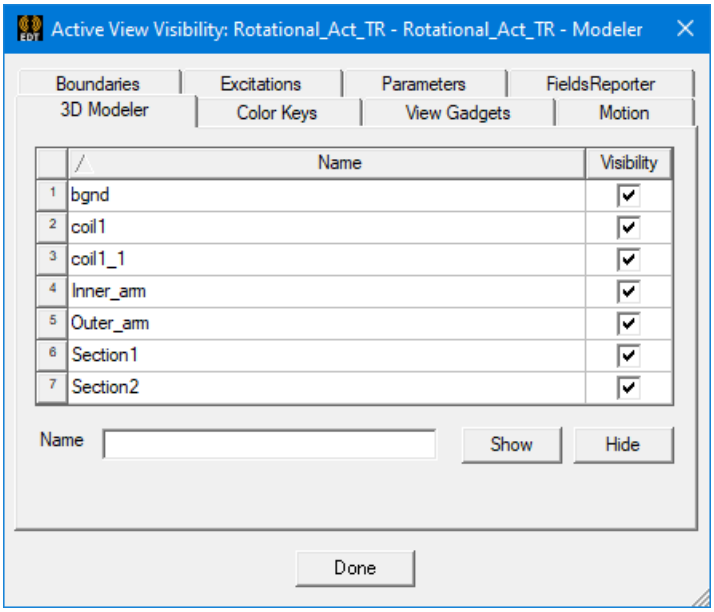
> **Note:**
>
> Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

Object visibility is saved with the project.

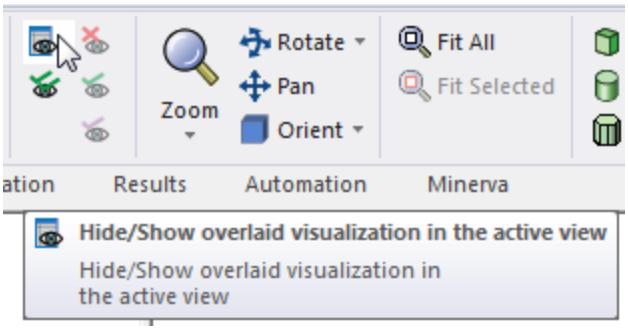
Showing Objects

You can show or hide objects in several ways, using the **Active View Visibility** dialog box for individual objects, or **Visibility** menu selections for All Objects, Only Selection, Model Objects, Non-model Objects, Primitive Shapes, or Non-primitive shapes.

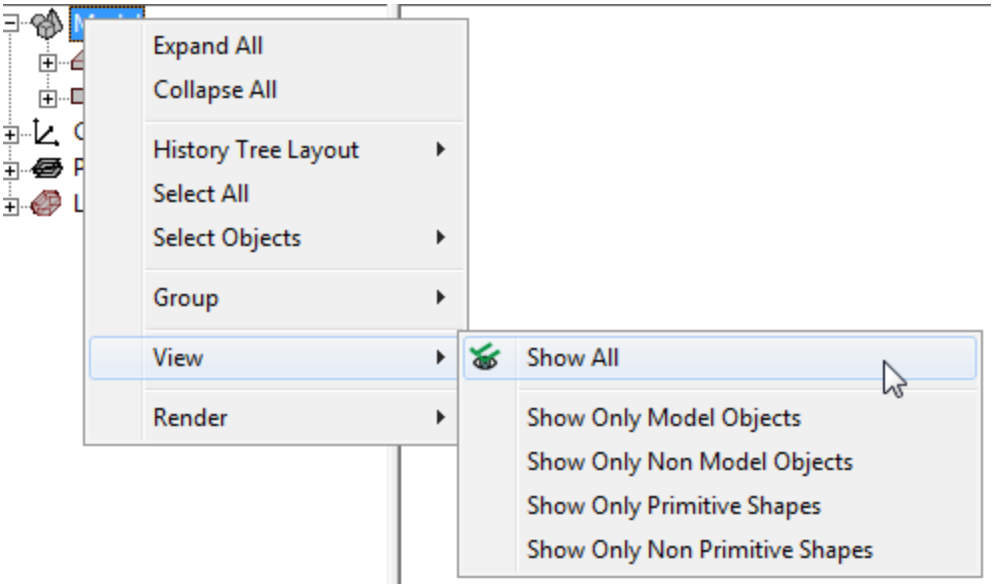
The **Active View Visibility** dialog box contains tabs for 3D Modeler, Color Keys, Array Setup, Boundaries, Excitations, and Fields Reporter. Select the tab for the objects you want to show or hide, and select or deselect **Visibility**.



You can access the dialog box from the menu bar (**View** > **Visibility** > **Active View Visibility...**) or by clicking the Hide/Show visualization icons in the **View, Draw,** or **Model** Ribbon tabs.



The **Visibility** menus are available on the **View** menu and as shortcut menus in the *Modeler* window and in the History Tree when you right-click the **Model** icon:

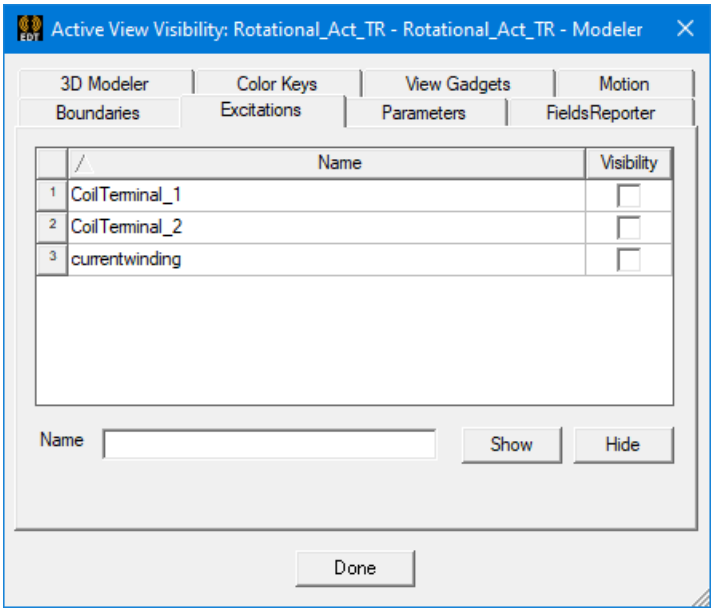


To show *one or more objects* that are currently hidden by using the *Active View Visibility* dialog box:

1. Click **View** > **Visibility** > **Active View Visibility...**, or click the visibility icons 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 box contains tabs for 3D Modeler objects: Color Key objects, Boundaries, Excitations, and Fields Reporter objects.



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 box 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.

> **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.

To show *all objects* that are currently hidden by using the **Visibility** menu:

1. Click **View** > **Visibility** > **Show All** or use the shortcut menu in the Modeler window or History Tree 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.

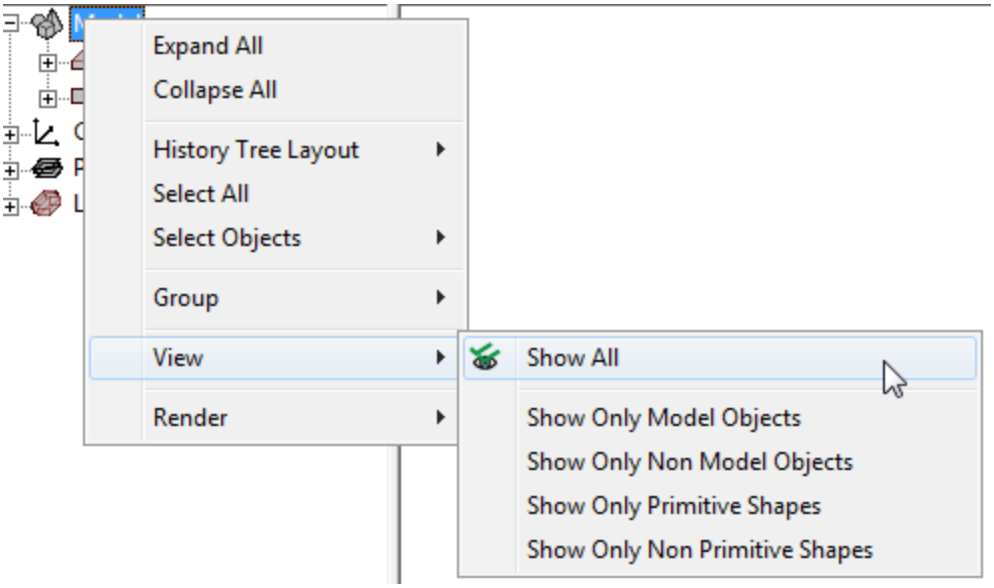
To show *selected objects* that are currently hidden:

1. Select the object. Hidden items are selected once the node corresponding to them is clicked in the history tree.
2. Click **View** > **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 use the **View, Draw,** or **Model** ribbon tab icons to **Show selected objects in active views**. Or, open the **Active View Visibility** dialog box.

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 **Hide/Show** icons on the **View,Draw,** and **Model** ribbon tabs and the **View** menu selections.
2. You can right-click to display the shortcut menu and select **View** > **Show All**.



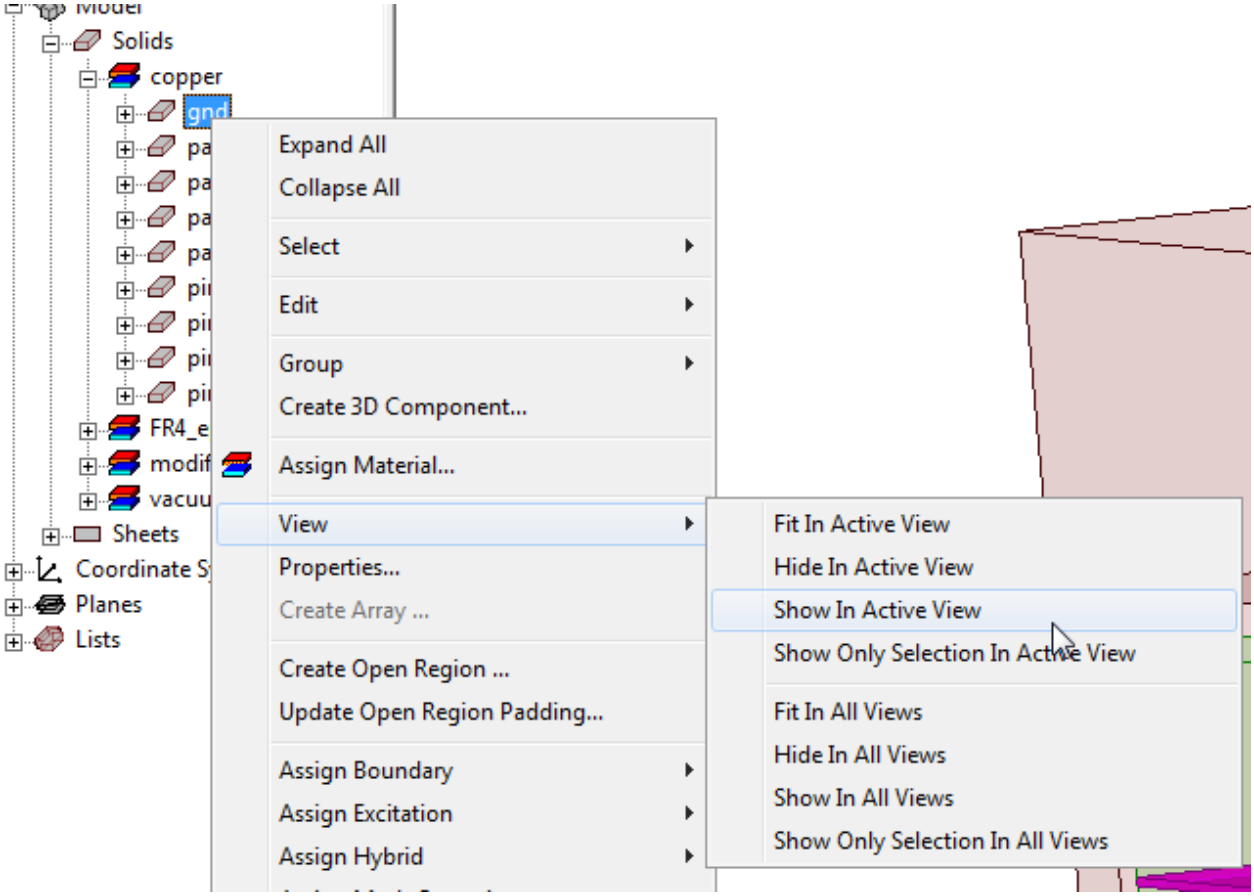
Object visibility is saved with the project.

Showing Only Selected Objects in All or Active Views

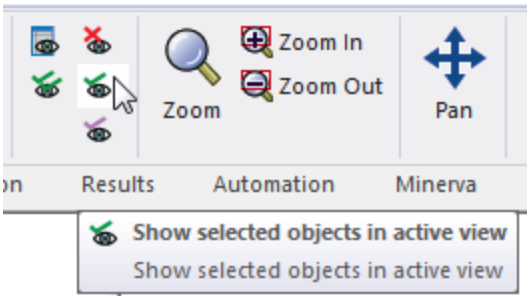
To show only *selected objects*:

1. Select the object from the Project Manager 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 an object in the History Tree and use the shortcut menu:
3. You can right-click to display the shortcut menu and select **View** > **Show in Active View, Show in All Views,** or **Show Only Selection In Active View**.



You can also use the **View/Draw/Model** ribbon icons to **Show selected objects in all views** and **Show selected objects in active views**.



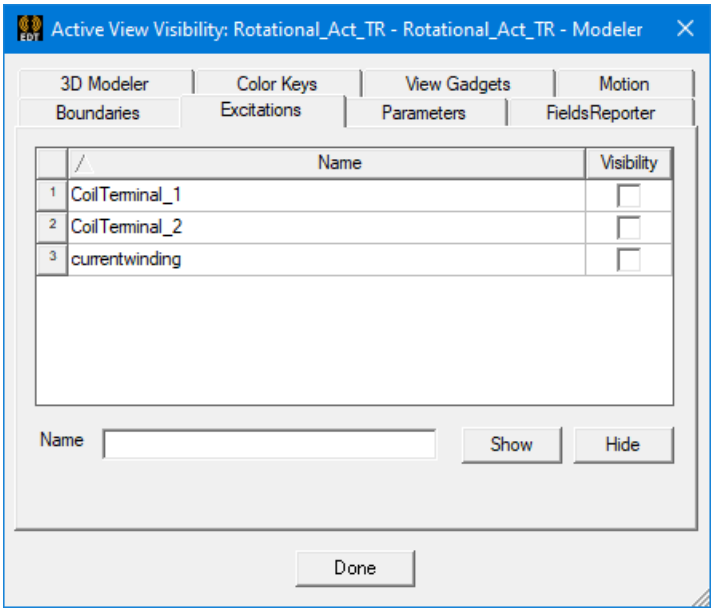
The selected objects reappear.

Active View Visibility Dialog Box

If there are many objects, it may be easier to show or hide objects using the **Active View Visibility** dialog box.

1. Click **View** > **Visibility** > **Active View Visibility**, or click the equivalent **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.

Rendering Objects as Wireframes or Solids

To render (display) *all* objects in the view window as wireframe outlines, flat-shaded solids, or smooth-shaded solids:

1. Click **View**> **Render** and click one of the following:
    - **Wireframe**:

      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 to display a wireframe view.

    - **Smooth Shaded**:

      The objects in the view window are displayed as shaded objects with smooth edges.

      You can also use the **F7** key to display a smooth shaded view.

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 **Attribute** tab of the docked **Properties** window, select **Display Wireframe**.

Setting the Default View Rendering Mode

To set a default rendering mode for all objects created in the active design and in future designs:

1. Click **Tools** > **Options** > **Modeler Options**.
2. Click the **Display** tab.
3. Select one of the following options from the **Default View Render Mode** drop-down menu:
    - **Wireframe**.

      The objects in the view window will be displayed as skeletal structures, enabling you to see all sides of the objects at one time.

    - **Smooth Shaded**.

      The objects in the view window will be displayed as shaded objects with smooth edges.

4. Click **OK**.

   The rendering mode will be applied to all new objects you create.

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:

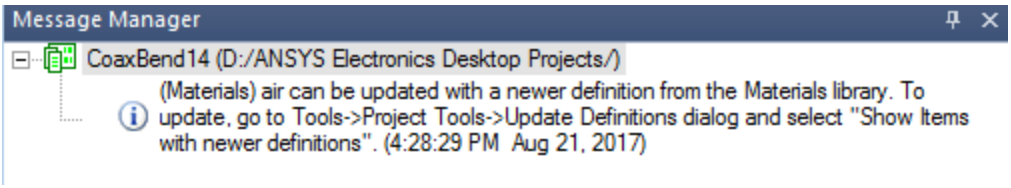
1. Click **View** > **Render** > **Enhanced Display** or press **F8**.

Enhanced Display overrides part 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 part attribute color and transparent values are applied.

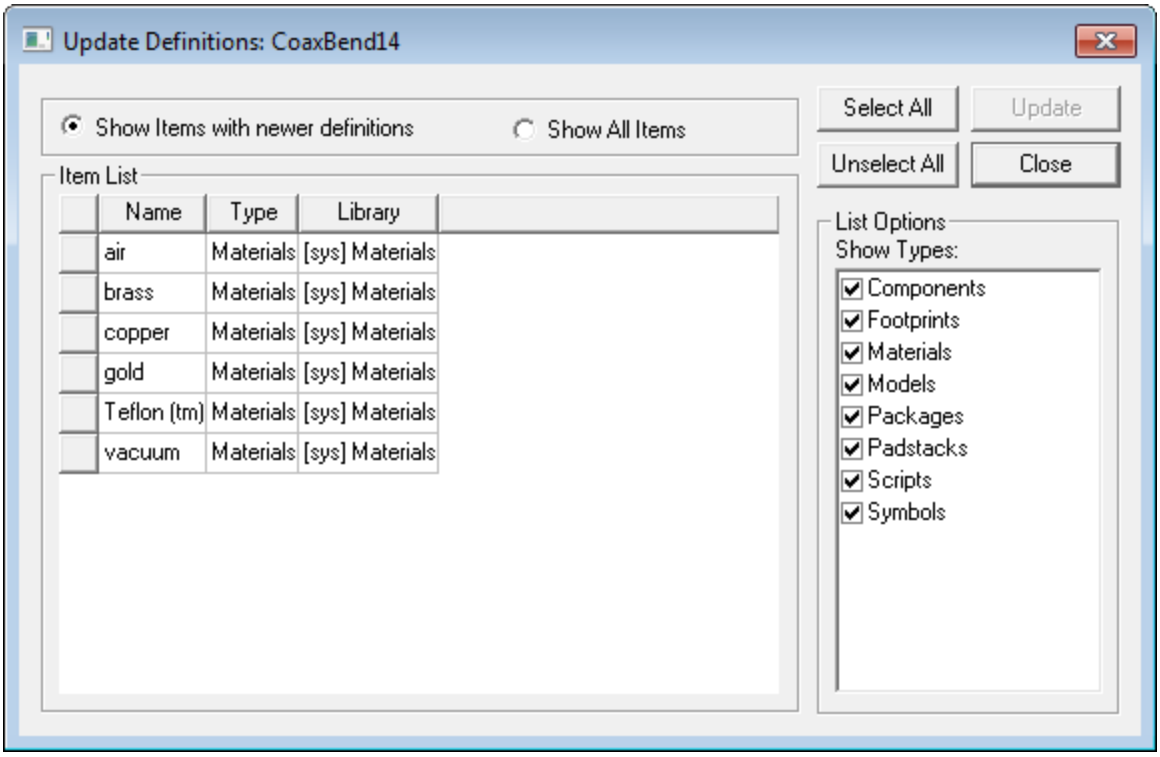
It is possible for the view to be in the “Enhanced Display” mode but some of the parts in the model could have “Use Material Appearance” unchecked. In this case you can edit the **Color** or **Transparent** values, or select **Use Material Appearance** in the docked **Properties** window, but they are not applied to the active view visualization. This resembles how the **Wire Frame** setting is handled currently.

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 the material appearance from the assignment is used.

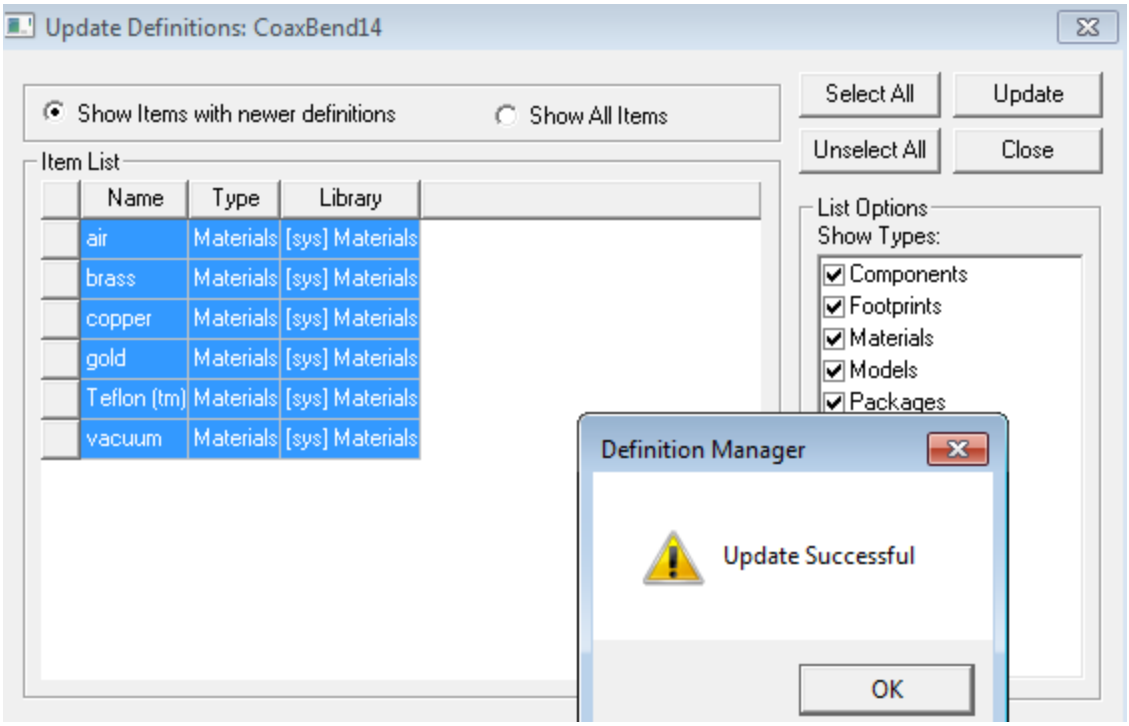
When you are working with a legacy project and use **View** > **Render** > **Enhanced Display**, if there are updates to the materials definitions for appearance, you will receive an info message informing you that the material definition was updated.



If you want to update, click **Tools** > **Project Tools** > **Update Definitions** to open the **Update Definitions** dialog box.



Here you can multi-select rows of materials to update them. In case of updates available from multiple libraries, you can choose the library location. Selecting material rows enables the **Update** button. Selecting **Update** then produces a message on a successful update.



Setting the Curved Surface Visualization

You can 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. The default gives satisfactory results (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 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.

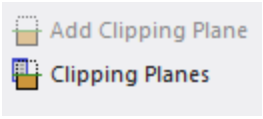
5. The **Save As Default** button lets you Save any values you change to the drop-down menus for the fields.
6. The **Restore Default**s button lets you return to the original values. Any values you provided through **Save As Default** remain on the drop-down menus for the fields for surface and normal deviations.
7. Click **Apply** to apply the current values to the active modeler window, and **Close** or **Cancel** to close the dialog box without changing settings.

Using Clip Planes

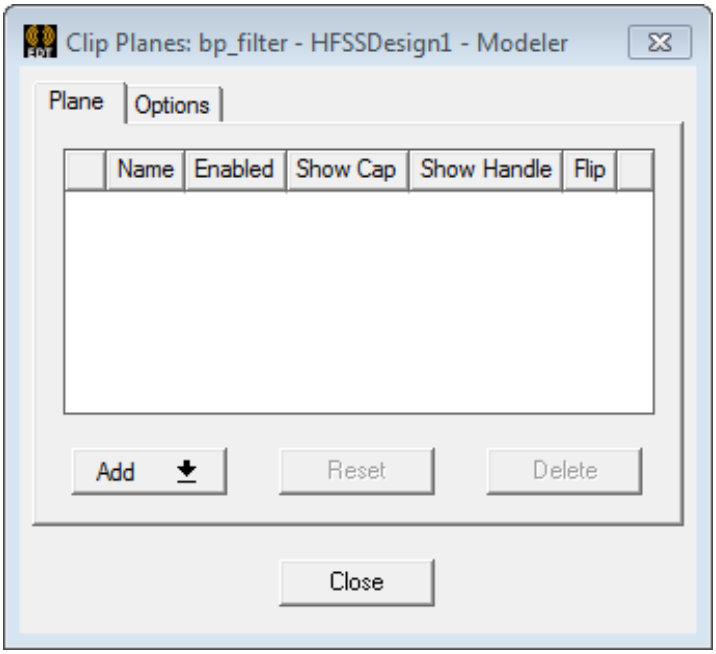
The **View** > **Clipping** > **Clip Plane** command lets you define a clip plane that you can use to interactively make any desired cut-away view of a model. If you use **Edit** > **Copy Image** or **Modeler** > **Export**> **[image format]** with the clip plane active, the image shows the clipped plane. 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:

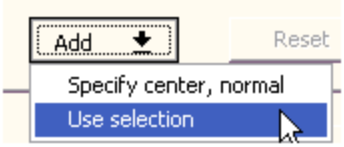
1. Click **View** > **Clipping** > **Clip Plane** or click the **Clipping Planes** icon on the **View** ribbon tab.



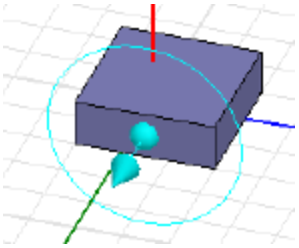
This displays the **Clip Plane** dialog box with the **Plane** tab selected.



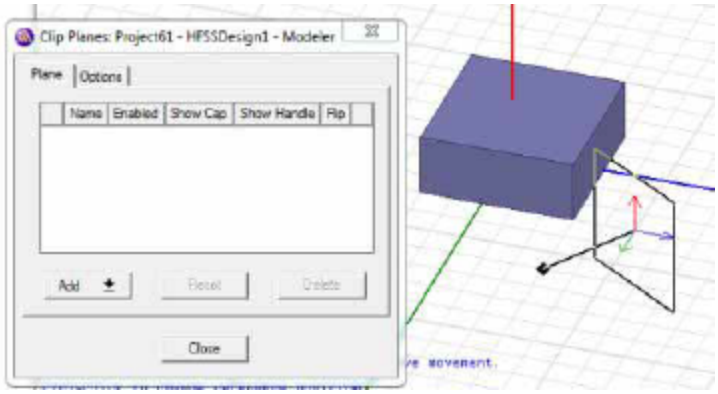
2. The **Add** button contains a drop-down menu with choices for **Specify center, normal**, and **Use selection**. If you want to use a selection, you must first select a face or an existing cut plane.



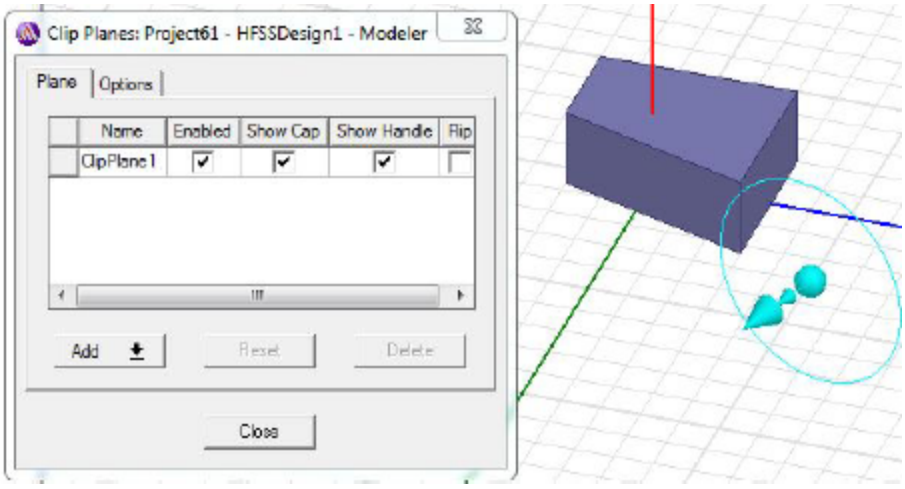
- If you first select a face or cut plane, and then click **Add** > **Use Selection**, the clip plane is added on that face.



- If you select **Add** > **Specify center, normal**, this launches a Measure dialog box and enters a mode for you to click to define the start location (shown as a triad).
  a. When you move the cursor, a rectangle represents the clip plane, and a vector the current direction.

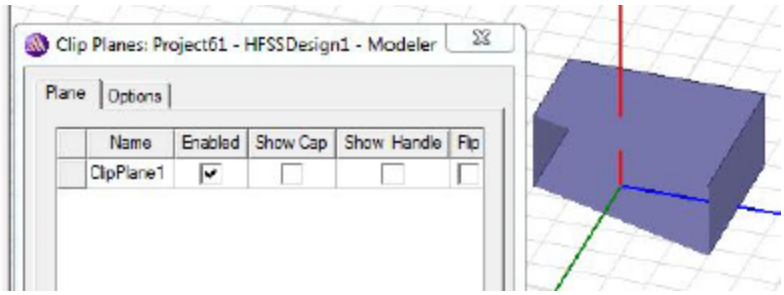


  b. Click again to set the reference position.

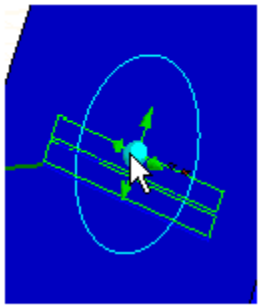


  After the second click, the clip plane is active. The handle is visible as a circle with a sphere at the center, and an arrow pointing the normal for the plane.

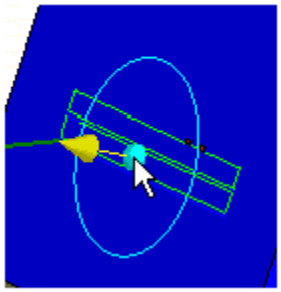
The **Clip Planes** dialog box shows the clip plane name, that it is enabled, the cap (which is the plane surface), and the handle. **Flip selection** lets you reverse the direction of the clip plane. If you deselect **Show cap** and/or **show handle**, they disappear from the display.



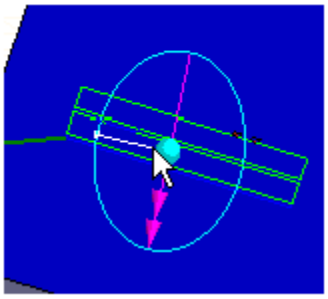
3. With **Show Handle** enabled, you can use the handle to manipulate the location and orientation of the clip plane. The handle changes appearance and function relative to the position of the cursor. Dragging the cursor makes use of the current function.



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 of the **Clip Planes** dialog box 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 Lighting

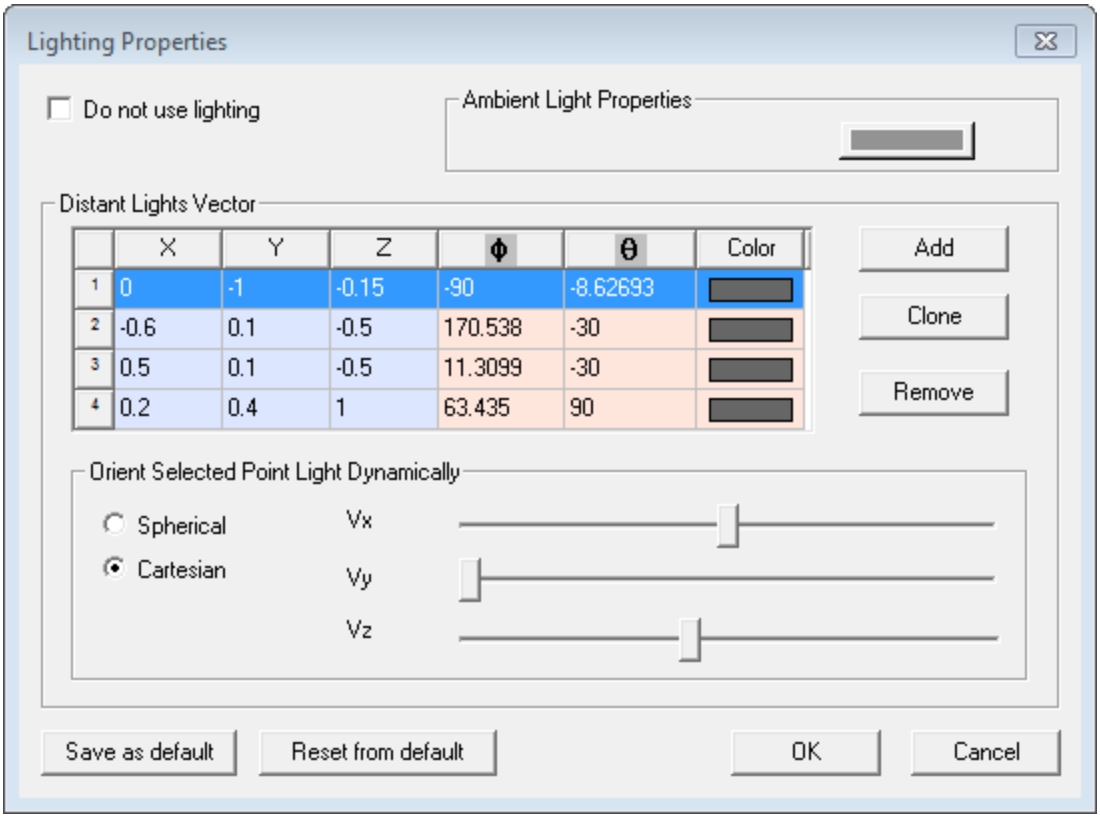
You have the option to emit the following types of light on a design:

- *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.

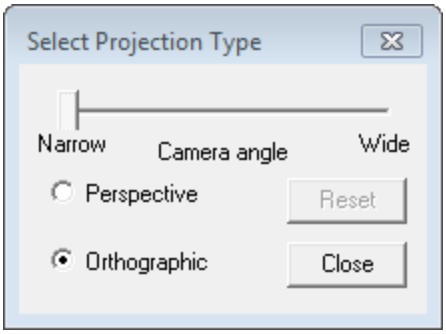
The lighting settings will be saved with the design. New lighting applied to other designs after this point, including new default settings, will not affect these lighting settings.

Setting the Projection View

To modify the projection of model objects (the camera angle) in the view window:

1. Click **View**> **Modify Attributes**> **Projection** or, on the **View** tab of the ribbon, click **Modify Attributes** and select **Projection** from the drop-down menu.

   The *Select Projection Type* dialog box appears:



2. Select **Perspective** to enable the slider to change the 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 than 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* dialog box closes. The last view you specified in the projection window remains visible in the view window.

The projection view you set will be saved with the design. New projection views assigned to other designs after this point will not affect this projection setting.

10 - Working with Boundaries in Q3D Extractor and 2D Extractor

Depending on the model, you can assign different types of boundaries.

- For 2D Extractor, see: Finite Conductivity Boundary.
- For Q3D Extractor, see: Infinite Ground Plane Boundary and Thin Conductor Boundary.

Please note the following:

- An infinite ground plane in a Q3D design is an ideal model. The infinite ground plane serves as a return path and makes all terminals in other nets well-referenced.
- A finite ground plane net without terminals is ignored in DC RL analysis and is treated as a floating conductor in AC RL analysis.
- In cases using a finite ground as a return path, add terminals on the ground plane according to the real port setup.
- To approximate an infinite ground using a finite ground, add reference terminals on the finite ground. These reference terminals should be placed close to terminals on other nets. The RL values with a finite ground can then be obtained from Reduce Matrix by grounding the finite ground net.

Assigning a Boundary

To assign a boundary:

1. In the Modeling Workspace, select the appropriate geometry for the boundary.
2. Assign the boundary one of three ways:
   - Right-click in the Modeling Workspace and select **Assign Boundary** > [**Boundary Type**].
   - Right-click **Boundaries** in the **Project Manager** and select **Assign Boundary** > [**Boundary Type**].
   - Select [**Q3D Extractor/2D Extractor**] > **Boundary** > **Assign** > [**Boundary Type**].

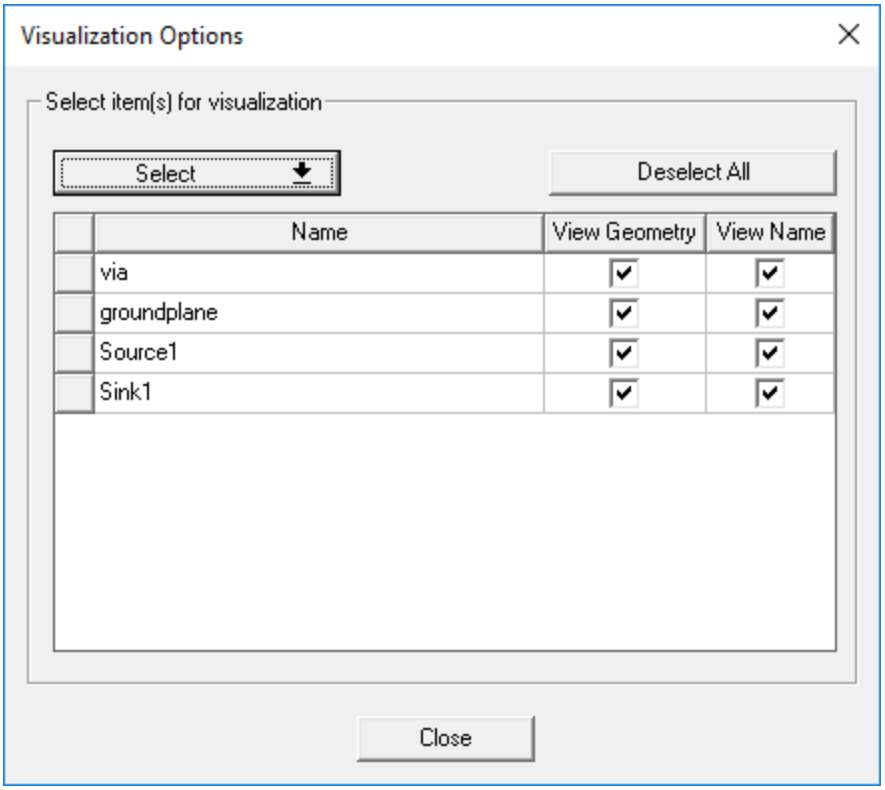
   A boundary-specific dialog box appears.
3. Enter the appropriate parameters and click **OK**.

Showing and Hiding Boundaries

To show or hide boundaries:

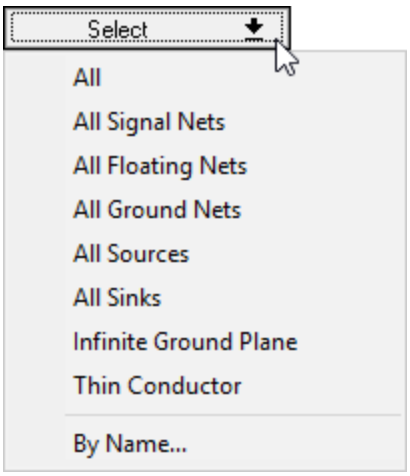
1. Select [**Q3D Extractor/2D Extractor**] > **Boundary** > **Visualization**.

   The **Visualization Options** dialog box appears.



2. Select an object to see it highlighted in the Modeling Workspace.
3. Select the check boxes for geometry and names you want visible. Deselect the check boxes or those you want to hide.

4. If desired, click **Select** to select by type:



   By Name allows you to find objects using regular expressions.

5. Click **Close**.

> **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.

Reassigning a Boundary

To reassign a previously assigned boundary:

1. In the Modeling Workspace, select the new object to which you want the boundary reassigned.
2. Right-click the boundary in the **Project Manager** and select **Reassign**.

   The boundary is removed from the original object and reassigned to the new object.

Deleting Boundaries

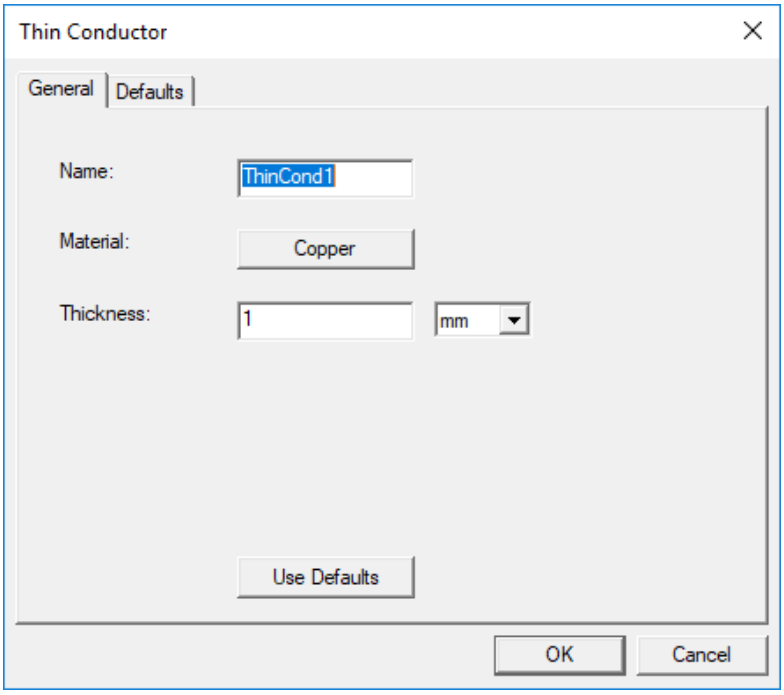
To delete a boundary, right-click the boundary in the **Project Manager** and select **Delete**.

To delete all boundaries, do one of the following:

- Select [**Q3D Extractor/2D Extractor**] > **Boundary** > **Delete All**.
- Right-click **Boundaries** in the **Project Manager** and select **Delete All**.

Assigning Thin Conductor Boundaries in Q3D Extractor

In Q3D Extractor, you can assign a **Thin Conductor** boundary. See: Working with Boundaries.



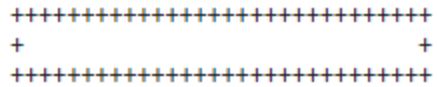
In the **Thin Conductor** dialog box, you are asked to specify the thin conductor **Name**, **Material**, and **Thickness**.

This boundary replaces a thin 3D conductor with a 2D sheet. The 2D sheet can be any curved surface without limitations.

> **Note:**
>
> Thin 3D conductors (aspect ratio > 1000) significantly slow down the mesh generation. This creates a problem for the solvers (ill-conditioned matrices), and results in long run times or failures. The benefits are less memory usage and shorter run times.

The solver assumes that the charges are on the 2D sheet instead of on the top, bottom, and boundary surfaces.

For example:



is modeled as:

++++++++++++++++++++++++++++++

where "+" are the charges.

For large aspect ratios, 3D results agree with the 2D thin conductor approach. For smaller aspect ratios, it is a tradeoff between performance and accuracy.

Capacitance and Conductance (CG) Solver

Thickness is not used by the capacitance solver.

AC Inductance and Resistance (AC IR) Solver

Like the capacitance solver, the current flows on the 2D sheet instead of on the top, bottom and boundary surfaces. For the resistance calculation, half of the surface impedance is used. This is equivalent to half of the current flowing on the top and the other half flowing on the bottom surface. The thickness is used in the inductance solver.

DC Inductance and Resistance (DC IR) Solver

The DC current flows on the 2D sheet. The thickness is used by the solver to apply the proper correction.

Assigning a Thin Conductor Boundary on a 2D Object

To assign a thin conductor boundary on a 2D object, you must select an object on the XY plane. Thin conductors can only be assigned on 2D objects (face not allowed), and you can manually assign a net on that 2D object. You can also auto identify the net, and the object will be detected as a net.

Assigning Infinite Ground Plane Boundaries in Q3D Extractor

In Q3D Extractor, you can assign an **Infinite Ground Plane** boundary. See: Working with Boundaries.

In the **Infinite Ground** dialog box, you can specify a **Name** for the infinite ground plane.

To assign an infinite ground plane boundary, you must select a face or object on the XY plane.

> **Important:**
>
> Only one infinite ground plane is allowed in a design.

> **Important:**
>
> Objects should not lie on both sides of ground plane, or intersect the ground plane.

> **Note:**
>
> When infinite ground planes are present, CG results may differ slightly from releases prior to 2022 R2.

Assigning Contact Resistance Boundaries in Q3D Extractor

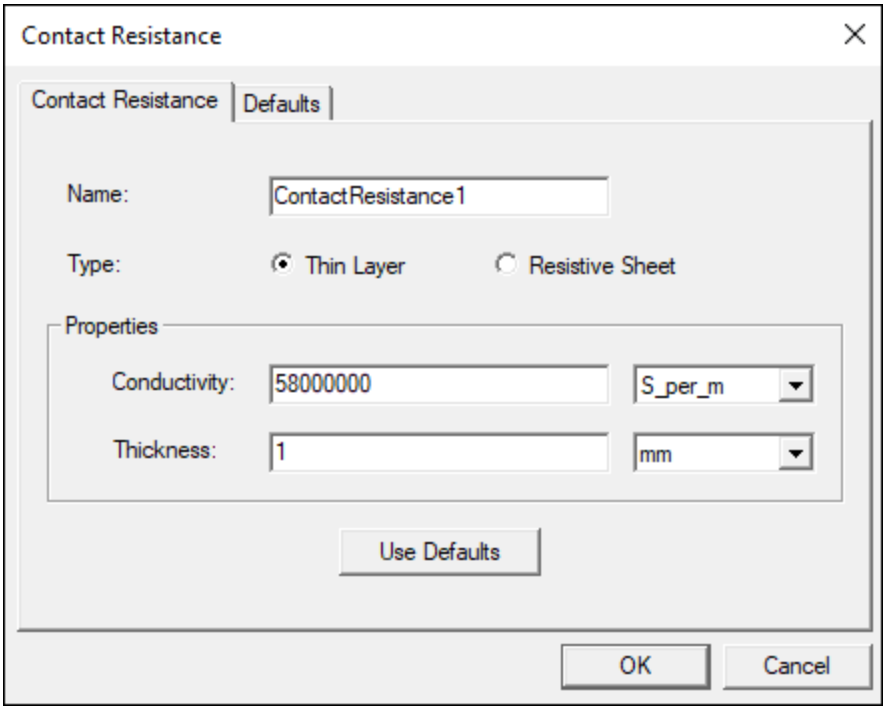
Contact Resistance boundaries in Q3D Extractor can be modeled as either Thin Layer or Resistive Sheet.

- For Thin Layer boundaries, the oxidation layer is modeled by having a face/sheet assigned a finite thickness (in meters) and a conductivity (in S/m). The solver takes the face and extrudes it to build a virtual prism with the given thickness. It builds the stiffness matrix for the prism and stamps it to the global matrix, then computes the potential. This is similar to Thin Layer conduction in the Maxwell DC conduction solver.
- For Resistive Sheet boundaries, the oxidation layer is modeled by having a face/sheet assigned a resistance (in Ohm). The solver imposes conductance per unit area between the potentials on the two sides of the sheet, which is 1/(resistance * contact area). This is similar to the Resistive Sheet modeling in Maxwell's magnetostatic and Eddy solvers.

When working with boundaries, assign a **Contact Resistance** boundary from the **Contact Resistance** window.

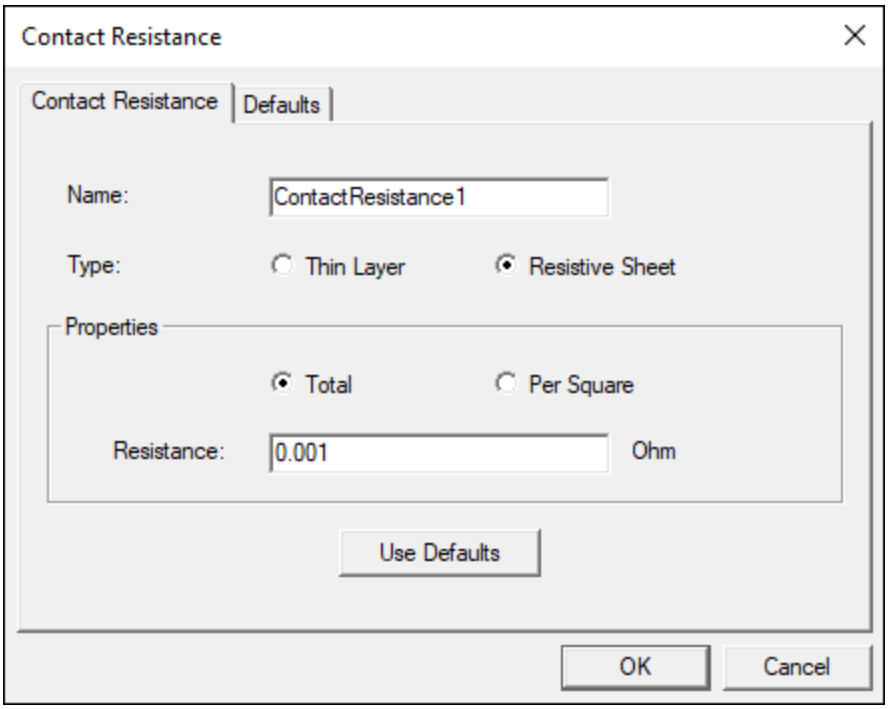
The window contains two tabs:

- **Contact Resistance** – assign either a **Thin Layer** or **Resistive Sheet** boundary.



For **Thin Layer**, specify:

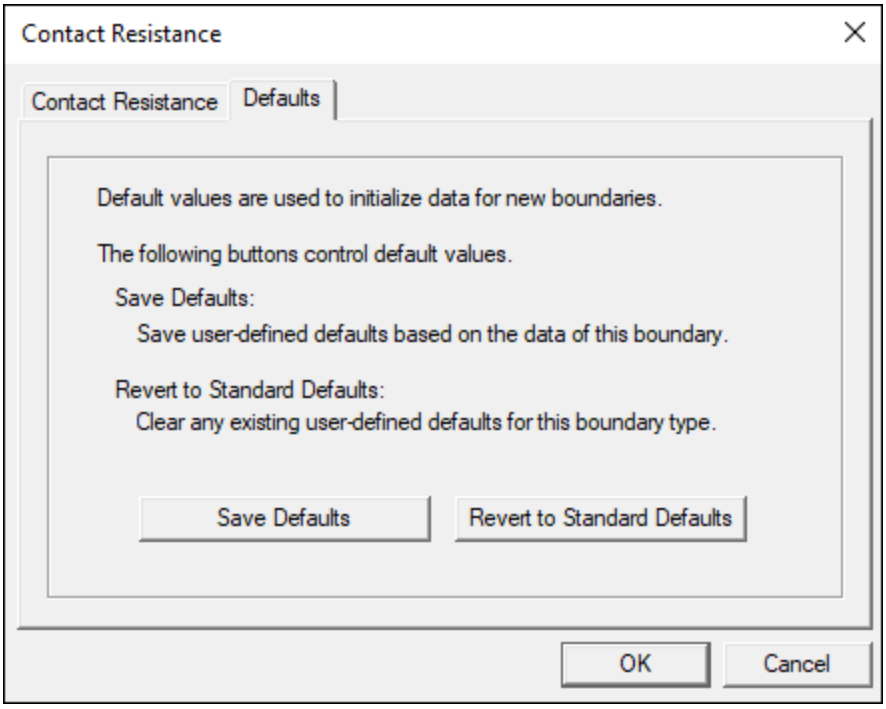
- **Name** – a name for the boundary.
- **Conductivity** – the conductivity of the thin conducting layer.
- **Thickness** – the thickness of the thin conducting layer.



For **Resistive Sheet**, specify:

- **Name** – a name for the boundary.
- **Total or Per Square** – select how the resistance value should be interpreted. Total indicates that the resistance value provided is the total resistance across all assignments of this boundary. Per Square indicates that the resistance value provided is the resistance of a unit square in SI for the material used in the assigned resistive sheets.
- **Resistance** – the resistance value in Ohms or Ohms/square, depending on whether Total or Per Square is selected above.

- **Defaults** – allows you to either save settings as defaults or restore default settings.



Additional Considerations

When a validation check is performed, Q3D Extractor checks for the following:

- The resistance/conductivity value should be greater than 0.
- The resistance/conductivity value should not be dependent on a variable.
- The assigned faces cannot overlap.

During the solution process, Q3D Extractor checks for the following:

- The sheet must have conductors touching both of its faces.
- The sheet cannot be smaller than the contacting surface.
- More than two solids sharing the same edge within the contact resistance boundary is not allowed.

The solution will be also invalidated if the resistance value is changed.

Assigning Finite Conductivity in 2D Extractor

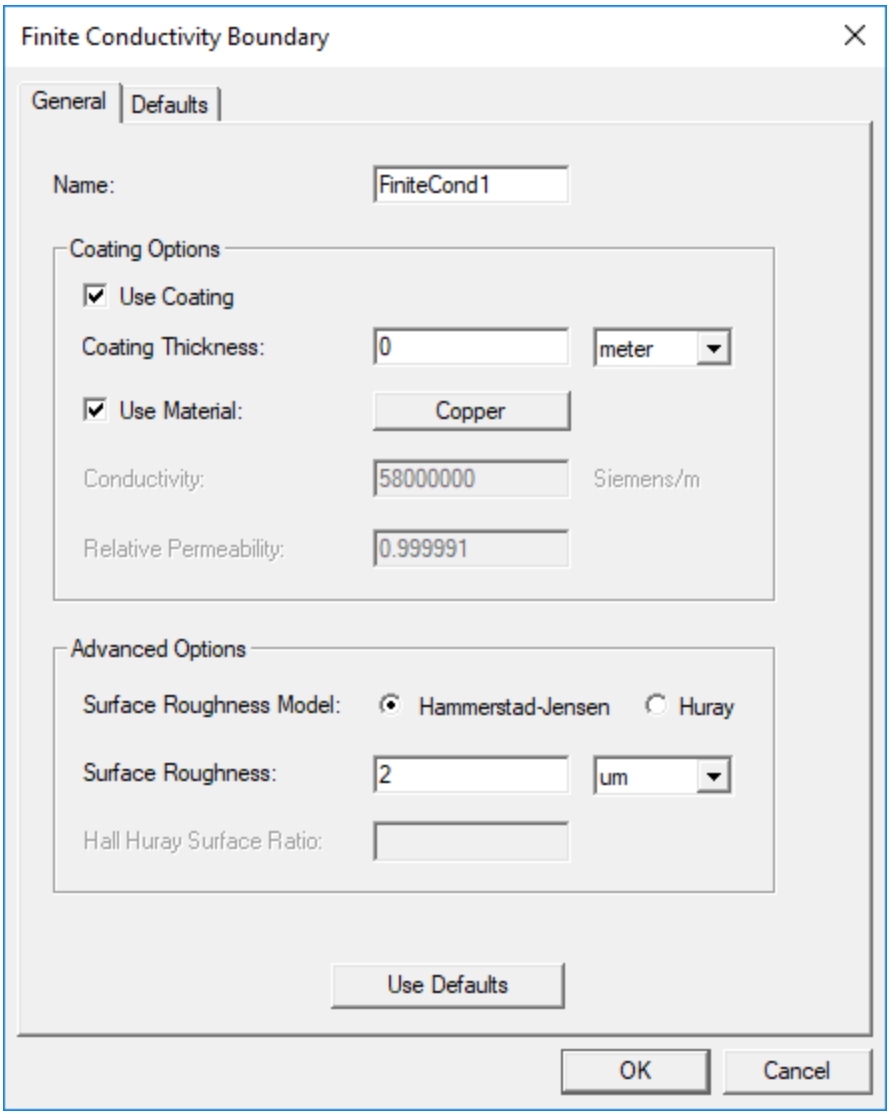
In 2D Extractor, you can select an *edge* and assign a **Finite Conductivity** boundary.
See: Working with Boundaries.

Note:

Finite conductivity boundary can only be assigned to the following conductor types: Signal Line, Non-Ideal Ground, Reference Ground or Floating Line.

Note:

Finite Conductivity Boundaries utilize an Impedance Boundary Condition formulation to solve. For finite conductivity boundaries to be considered during a simulation, a conductor must be assigned as **Solve on Boundary** or **Automatic**, as described in Working with Conductors.



In the **Finite Conductivity** dialog box, you are asked to specify the **Name**, **Coating Options**, and **Advanced Options**.

- **Use Coating** – If you want to add a layer on top of the base object, select **Use Coating** and specify a **Coating Thickness**.

  When this is deselected, the the project is solved using the **Conductor Thickness** specified while assigning conductors.

- **Use Material** – If you want to select an existing material for the coating, select **Use Material**.

  Otherwise, you will need to specify the **Conductivity** and **Relative Permeability**.

- **Surface Roughness Model** – Select either **Hammerstad-Jensen** or **Huray**.

  For **Hammerstad-Jensen**, you must specify the **Surface Roughness** and unit of measure.

  For **Huray**, you must specify a **Nodule Radius** and **Surface Ratio**. The **Nodule Radius** describes the radius of copper spheres that model the surface roughness, while the **Surface Ratio** is a unitless quantity defined by the parameter sr:

  $$sr = 4\pi r^2 \frac{N}{A}$$

  where:

  A is the area of a unit cell

  N is the number of nodules per cell (modeled as spheres)

  r is the radius of a typical nodule

  > **Note:**
  >
  > For more information about the complex-value-based, causal, modified Huray model used in the Ansys Electronics Desktop software, see the Huray Surface Roughness Model topic.

Impedance Boundary Conditions in 2D Extractor

For 2D Extractor models, every conductor set to **Solve on Boundary** or **Automatic** must have Impedance Boundary Conditions (IBCs) if it is one of the following types:

- Signal Line
- Non Ideal Ground

- Reference Ground
- Floating Line

You can define Impedance Boundary Conditions (*except for on perfect conductors*).

If Impedance Boundary Conditions are not specified on all edges of the conductor, the software creates implicit conditions on those edges.

The implicit IBC parameters are:

- **Surface Roughness** – this is taken to be equal to zero.
- **Coating Thickness** – this is taken to be equal to zero. For implicit processes, the solution is solved using the conductor thickness specified.
- **Coating Material** – the same material is used as the base object.

Huray Surface Roughness Model

Surface roughness can increase conductor power losses by more than two times the loss associated with smooth conductors. In Ansys Electronics Desktop, a modification of the standard real-valued Huray surface roughness model is implemented to accurately account for these losses in both the frequency and time domains.

This modification fixes a causality problem with the original Huray model by adding some frequency-dependent reactance to the roughness-enhanced surface impedance. The purpose is to provide a complex-valued causal analytic model that exactly matches the loss results of the original model but is also suitable for time domain computations.

The original Huray model was limited to calculations in the frequency domain and did not account for the effects of roughness on the phase of the transmitted signal. These limitations have been eliminated. The enhanced causal model not only provides accurate broadband modeling of the losses in very rough copper foils (typically used in PCB manufacturing) but also accurately predicts increased phase delay due to the surface roughness.

Essentially, the smooth surface impedance (Z_{smooth}) is increased by a complex “Huray” factor (H_c), a function of frequency, given by the following equation (1):

$$H_c\left(j\omega\right) = 1 + \frac{K}{1+\left(\frac{j2\omega}{\omega_0}\right)^{-1/2}}$$

where:

- ω is the frequency in radians/second
- ω_0 is a corner frequency represented by the equation (2):

$$\omega_0 = \frac{2}{a^2 \mu_r \mu_0 \sigma}$$

- K is a derived constant used to simplify the equation (3):

$$K = \frac{3}{2} \cdot SR = \frac{6\pi a^2 N}{A_f}$$

in which:

- *SR* is the Hall-Huray surface ratio:

$$\frac{4\pi a^2 N}{A_f}$$

- *a* is the radius of a spherical nodule
- μ_r is the relative permeability of the conductor
- μ_0 is the permeability of free space

$$\mu_0 = 4\pi \cdot 10^{-7}\ \mathrm{H/m}$$

- σ is the electrical conductivity in Siemens/m
- *N* is the number of nodules
- A_f is a unit cell area

Note:

a and *SR* are the two parameters you must enter in the **Finite Conductivity Boundary** dialog box when you select the Huray surface roughness model:

For more information about this complex causal Huray model (its necessity, background, derivation, proof of accuracy, and consistency with the original real-valued model), see Reference 6 below.

Once the Huray surface roughness factor is determined, the rough surface impedance (Z_{rough}) is determined by the equation (4):

$$Z_{rough} = H_c \cdot Z_{smooth}$$

References:

1. P.G Huray, S.G Pytel, S.H Hall, F. Oluwafemi, R.I. Mellitz, D. Hua, and P. Ye, "Fundamentals of a 3-D "Snowball" Model for Surface Roughness Power Losses", 11th Annual IEEE SPI Proceedings, May 13 – 16, 2007.
2. S.H Hall, S.G. Pytel, P.G Huray, D. Hua, A. Moonshiram, G. Brist, and E. Sijercic, "Multi-GHz, Causal Transmission Line Modeling Methodology with a Hemispherical Surface Roughness Approach", IEEE Transactions on Microwave Theory and Techniques, December 2007 pp 2614 – 2624.
3. S.G Pytel, P.G Huray, A. Moonshiram, S.H Hall, R.I Mellitz, G. Brist, F. Oluwafemi, H.M Meyer, L. Walker, and M. Garland, "Analysis of Copper Treatments and the Effects on Signal Propagation", 58th Annual IEEE ECTC, May 26 – 30, 2008, pp 1144 – 1149.
4. S.G Pytel, "Multi-gigabit data signaling rates for PWBs including dielectric losses and effects of surface roughness", PhD. Dissertation, University of South Carolina, 2007.
5. P.G Huray, O. Oluwafemi, J. Loyer, E. Bogatin, and X. Ye; "Impact of Copper Surface Texture on Loss: A Model That Works", DesignCon 2010, February 1 - 4, 2010.
6. J. Eric Bracken, Ansys, Inc., "A Causal Huray Model for Surface Roughness", DesignCon 2012, January 2012.

11 - Working with Conductors in 2D Extractor

2D Extractor allows you to define the following types of conductors:

- Signal Line
- Non-Ideal Ground
- Reference Ground (aka. Ideal Ground)
- Floating Line
- Surface Ground

All conductor types (except for Surface Ground) can only be assigned to conducting, non-touching, and non-overlapping bodies. Surface Ground conductors are only allowed in a closed solution type, can only be assigned to the bounding box, and can only be assigned to a non-conducting object.

> **Important:**
>
> To pass the validation check, make sure that you follow these rules:
>
> - Conductors can only be assigned to "bodies", any other entity type is not allowed.
> - All conductor types except Surface Ground must be assigned to conducting object.
> - No conductor assignment is allowed on touching bodies.
> - You cannot assign multiple conductor assignments on same body.

Assigning Conductors

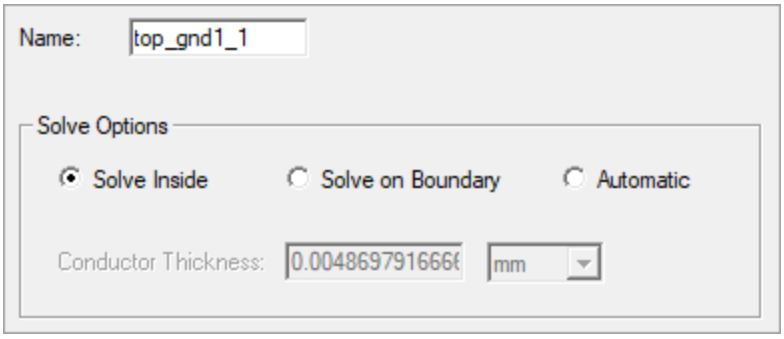
To assign a conductor:

1. In the Modeling Workspace, select the appropriate object(s) for the conductor(s).
2. Assign the conductor one of three ways:
   - Right-click in the Modeling Workspace and select **Assign Conductor** > [**Conductor Type**].
   - Right-click **Conductors** in the **Project Manager** and select **Assign Conductor** > [**ConductorType**].
   - Select [**2D Extractor**] > **Conductor** > **Assign** > [**Conductor Type**].

   A conductor-specific dialog box appears. See: Conductor Types.

3. Enter the appropriate parameters.

> **Note:**
>
> There are no parameters for a **Surface Ground** conductor.

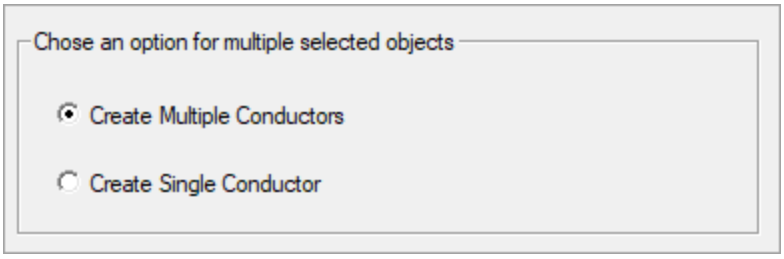


- **Solve Inside** – forces the solver to always solve inside and ignores any IBC in the design. This might lead to failure of RL solve for very high frequencies because of the failure of accurate field approximations at very small skin depth regions. Mesh operation assignments are highly recommended for high frequency simulations.
- **Solve on Boundary** – uses the IBC formulation for accurate field solutions at very high frequencies. The solver does not solve the fields inside the conductors. This might lead to failure of RL solve for low frequencies because no mesh refinement is performed inside the conductors during adaptive simulation. *This option must be selected for finite conductivity boundary settings (for example, surface roughness) to be considered*.
- **Automatic** – allows the solver to blend **Solve Inside** and **Solve on Boundary** for the nominal solution and the frequency sweep. The solver automatically determines upper and lower frequency limits for blending. For frequencies between these limits, the solver blends both methods via weighting factors. For frequencies below the lower limit, the solver uses **Solve Inside** only. For frequencies above the upper limit, the solver uses **Solve on Boundary** only.

Important:

- **Automatic** is the best option for simulating problems with a wide frequency spectrum. **Solve on Boundary** gives incorrect results at very low frequencies, while **Solve Inside** gives incorrect results at very high frequencies.
- **Solve on Boundary** and **Automatic** require that you specify Conductor Thickness. The value must be greater than 0. The default conductor thickness is obtained by dividing the conductor 's area by its perimeter (A/p). When selecting multiple conductors, the default value is the average conductor thickness.

4. If you selected multiple objects, you must select one additional setting:



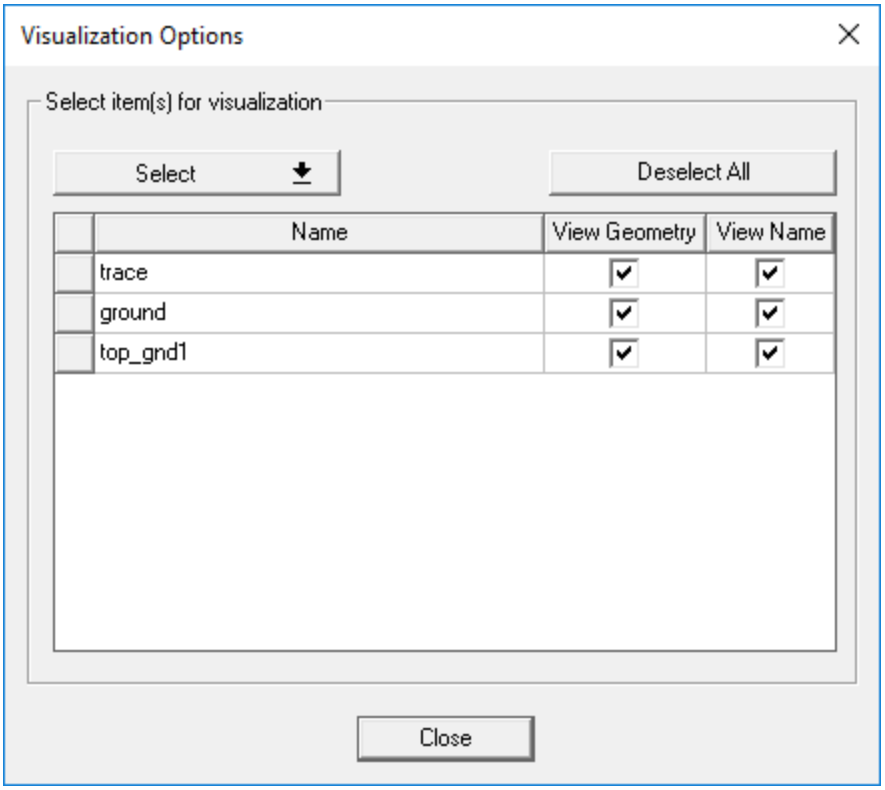
   - **Create Multiple Conductors** – creates a separate conductor for each selected object.
   - **Create Single Conductor** – creates a single conductor, joined in parallel.
5. Click **OK**.

Showing and Hiding Conductors

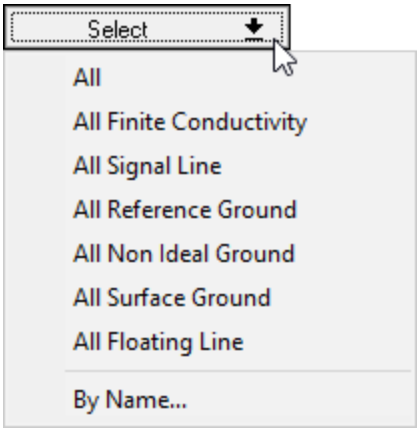
To show or hide conductors:

1. Select [**2D Extractor**] > **Conductor** > **Visualization**.

   The **Visualization Options** dialog box appears.



2. Select an object to see it highlighted in the Modeling Workspace.
3. Select the check boxes for geometry and names you want visible. Deselect the check boxes or those you want to hide.
4. If desired, click **Select** to select by type:



By Name allows you to find objects using regular expressions.

5. Click **Close**.

Zooming to a Conductor

The **Zoom To** command zooms the view in the Modeling Workspace in or out to show the selected conductor. This can be useful in looking at problem areas.

To zoom to a conductor, right-click the conductor in the **Project Manager** and select **Zoom To**.

Reassigning a Conductor

To reassign a previously assigned conductor:

1. In the Modeling Workspace, select the new object to which you want the conductor reassigned.
2. Right-click the conductor in the **Project Manager** and select **Reassign**.

   The conductor is removed from the original object and reassigned to the new object.

Deleting Conductors

To delete a conductor, right-click the conductor in the **Project Manager** and select **Delete**.

To delete all conductors, do one of the following:

- Select [**2D Extractor**] > **Conductor** > **Delete All**.
- Right-click **Conductors** in the **Project Manager** and select **Delete All**.

Conductor Types

You can add five types of conductors:

- **Signal Line** – to model signal carrying transmission lines.
- **Non-Ideal Ground** – to model switching noise.
- **Reference Ground** – to indicate return paths for current flow.
- **Floating Line** – to model objects with a charge but no current.
- **Surface Ground** – to model a perfectly conducting grounded object surrounding a transmission line.

In a transmission line problem, there must be at least one signal line and one grounded reference (a reference ground for an open solution type, or a surface ground for a closed solution type).

2D Extractor only computes matrix entries for objects declared as signal lines or non-ideal grounds. However, the matrices are affected by all conductors, since they have an effect on the field solutions from which the circuit parameters are extracted.

Signal Line Conductors

You must declare an object to be a signal line if it represents a signal-carrying transmission line in the structure being analyzed. Signal lines are included in the parametric matrices.

Matrix Computations

Any number of conductors can be identified as signal lines in the 2D Extractor. For each conductor you define, another row is added to the requested parameter matrices. For example, if six lines are defined as signal lines and you requested admittance and impedance, the software generates a 6x6 CG matrix and a 6x6 RL matrix for the structure.

Treatment During the Solution Process

- If admittance has been requested, zero volts or one volt is applied to each signal line in sequence as intermediate matrix solutions are generated.
- If impedance has been requested, zero amps or one amp is applied to each signal line in sequence as intermediate matrix solutions are generated. The current return path is assumed to be through the reference or surface ground in the problem.

Automatic Assignment

You can automatically assign signal line conductors so that all conducting objects are assigned as Signal Line conductors. To do so, select **2D Extractor** > **Conductor** > **Auto Assign Signals**.

> **Important:**
>
> - To pass validation, you must also define a Reference Ground in the design.
> - If you have some pre-defined conductors in your design, you will receive a warning that existing conductors assignments will be deleted if you proceed.

Non-Ideal Ground Conductors

All conductors, including grounds, have a certain amount of impedance, and currents that pass through conductors cause voltage drops across that impedance. This voltage drop in a ground conductor, which occurs when the signal is switching from high to low or vice versa, is known as switching noise, or "ground bounce." It induces cross-talk on other conductors in the transmission line.

Use non-ideal grounds to model switching noise in circuits. Non-ideal grounds are nonzero impedance grounded conductors whose voltages are allowed to fluctuate during a circuit simulation.

Non-ideal grounds are included in the parametric matrices. During matrix computations, they are treated exactly like signal lines. (Non-ideal grounds are provided to help you determine which conductors are signal lines and which are ground references). The sink point of a non-ideal ground is connected to ground and its source point connected to a source, allowing all other points along its length to float, rather than being forced to zero as in an ideal ground.

Reference Ground Conductors

You must declare any object connected to ground as a reference ground. Reference grounds are excluded from the parameter matrices. However, they do affect the field solution results. During the general field solution:

- If admittance have been requested, all points on a reference ground are set to zero volts.
- If impedance have been requested, all points on a reference ground are set to -1 amps.

During matrix computations, reference grounds are treated in the same way that they were in the general field solution. In addition to serving as return paths for currents applied to matrix computations, they are the return paths for other currents in the circuit.

Floating Line Conductors

Identify floating conductors as floating lines. Floating lines can have a charge applied to them but have no current flowing through them. They are excluded from the parameter matrices, but affect the field solution results. For this reason, be sure to identify any floating line that exists in your structure.

Floating conductors are not included in the parameter matrix solutions because they are not within the circuit current loop. However, they indirectly affect the inductance and capacitance matrix solutions because they affect the energy results that are part of the field solutions.

Surface Ground Conductors

You must declare a closed dielectric object whose surface is connected to a common ground as a surface ground. Surface grounds are treated as shields and are excluded from the parameter matrices. However, they do affect the field solution results. Surface grounds are zero impedance grounded conductors that serve as a return path for all conductors within the ring.

Typically, this type of conductor is used in situations where the transmission line is surrounded by a grounded object, such as the outside ring in the shielded transmission line. In such a case, declare the outside ring to be a surface ground.

> **Note:**
>
> If the outer ring were instead defined as a reference ground, the entire ring would be modeled as a solid conductor.

However, as a surface ground, only the outer surface of the object is assumed to be conductive. The system allows you to later define the material characteristics of any material inside the ring that is not occupied by other objects.

Setting Toggle Options for Conductors in 2D Extractor

If needed, you can toggle a selected conductor to another type. This is particularly useful when you have automatically assigned signal lines and want to toggle one of the signal lines to reference ground.

You cannot assign a Surface Ground conductor for the open solution type. If you want to toggle any conductor to Surface Ground and you have an open problem, you have to switch the design type to closed when prompted by 2D Extractor. Only when the solution type is switched is the conductor toggled.

To toggle a conductor:

1. Right-click a specific conductor in the **Project Manager** and select **Toggle**.
2. Select another conductor type from the submenu.

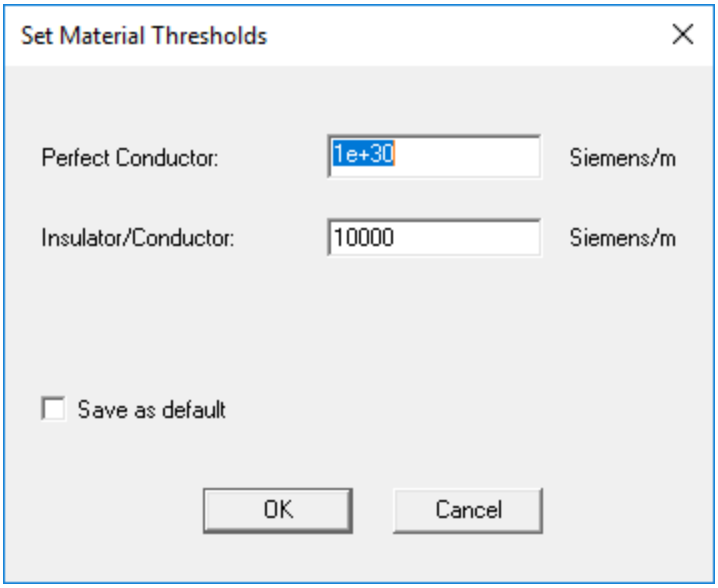
Setting Material Thresholds in 2D Extractor

2D Extractor uses the conductivity threshold to determine which objects are considered conductors. Objects with conductivity equaling or exceeding the threshold are considered conductors, and those with lower conductivity are considered non-conductors.

To set conductivity threshold values:

1. Select **2D Extractor** > **Conductors** > **Set Material Thresholds**.

   The **Set Material Thresholds** dialog box appears.

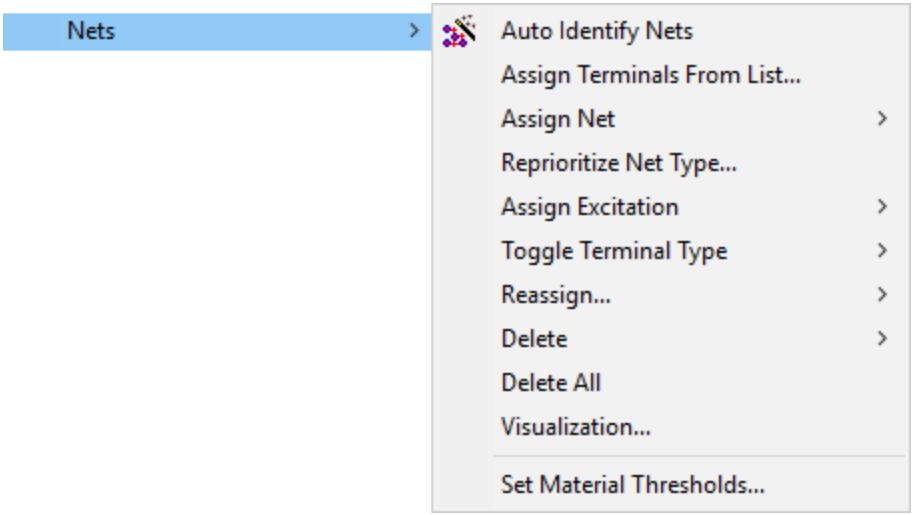


2. In the **Perfect Conductor** field, enter a value. Any conductor having conductivity above or equal to this threshold is considered a perfect conductor in the design.
3. In the **Insulator/Conductor** field, enter a value. Any object having conductivity equal to or greater than this conductivity is considered a conducting object in the design.
4. If desired, use the **Save as default** check box to make these settings your default.
5. Click **OK**.

12 - Working with Nets and Terminals in Q3D Extractor

A *net* is a collection of touching conductor objects separated by non-conducting materials or by the background material. A *terminal* can be a source or a sink.

Nets and Terminals can be modified using the **Q3D Extractor** > **Nets** submenu.



> **Important:**
>
> Nets can be assigned only to conductive materials. If you want to include a drawn/imported object as part of a net, you must assign it a conducting material.

Assigning Nets and Terminals

Nets can be assigned manually or automatically.

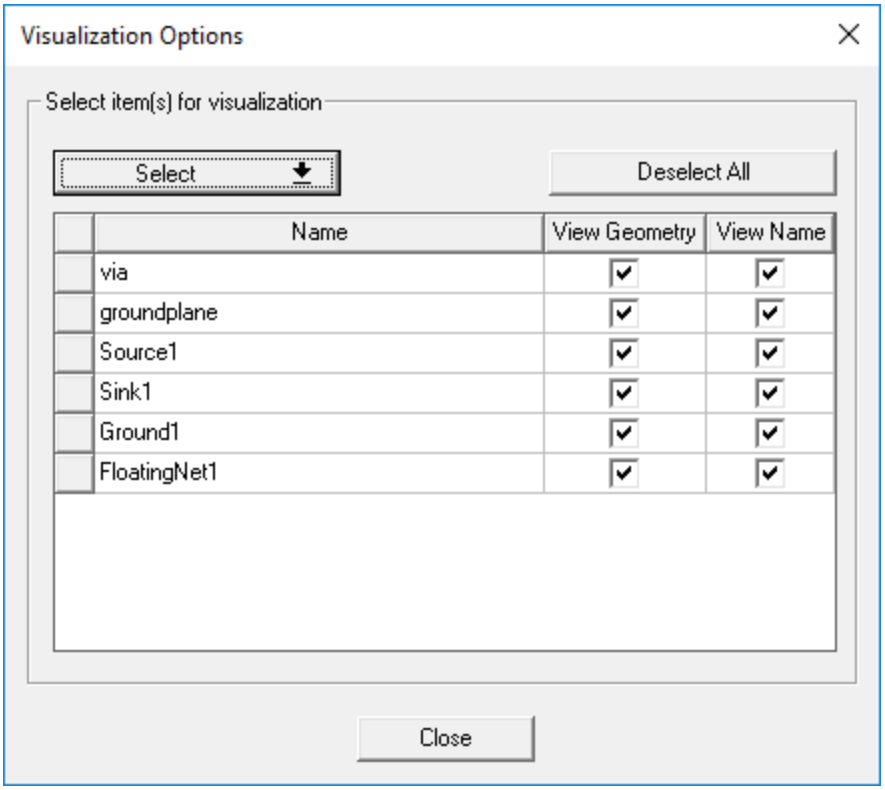
Terminals can be assigned from a list.

Showing and Hiding Nets and Terminals

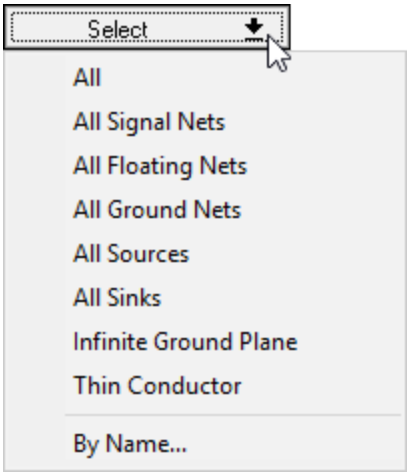
To show or hide nets or terminals:

1. Select **Q3D Extractor** > **Nets** > **Visualization**.

   The **Visualization Options** dialog box appears.



2. Select an object to see it highlighted in the Modeling Workspace.
3. Select the check boxes for geometry and names you want visible. Deselect the check boxes or those you want to hide.
4. If desired, click **Select** to select by type:



By Name allows you to find objects using regular expressions.

5. Click **Close**.

Zooming to Nets and Terminals

The **Zoom To** command zooms the view in the Modeling Workspace in or out to show the selected net or terminal. This can be useful in looking at problem areas.

To zoom to a net or terminal, right-click it in the **Project Manager** and select **Zoom To**.

Reassigning Nets and Terminals

To reassign a previously assigned net or terminal:

1. In the Modeling Workspace, select the new object you want to be a net or terminal.
2. Right-click the net or terminal in the **Project Manager** and select **Reassign**.

   The net or terminal is removed from the original object and reassigned to the new object.

Deleting Nets and Terminals

To delete a net or terminal, right-click it in the **Project Manager** and select **Delete**.

To delete all *nets*, do one of the following:

- Select **Q3D Extractor** > **Nets** > **Delete** > **All Nets**.
- Right-click **Nets** in the **Project Manager** and select **Delete** > **All Nets**.

To delete all *terminals*, do one of the following:

- Select **Q3D Extractor** > **Nets** > **Delete** > **All Terminals**.
- Right-click **Nets** in the **Project Manager** and select **Delete** > **All Terminals**.

To delete all nets *and* terminals, do one of the following:

- Select **Q3DExtractor** > **Nets** > **Delete All**.
- Right-click **Nets** in the **Project Manager** and select **Delete All**.

Assigning Nets Manually

Manually assigning nets is useful if you do not want to include an object made of conductive material in the matrices.

> **Warning:**
>
> Perfect electric conductors (PECs) and non-PECs cannot be part of the same net.

To manually assign a net:

1. Select the 3D conductive object to which you want to assign a net.
2. Assign the net one of three ways:

   - Right-click in the Modeling Workspace and select **Assign Net** > [**Net Type**].
   - Right-click **Nets** in the **Project Manager** and select **Assign Net** > [**Net Type**].
   - Select [**Q3D Extractor**] > **Nets** > **Assign** > [**Net Type**].

   A dialog box appears prompting you to name the net.
3. Enter a **Name** for the net.
4. Click **OK**.

   Nets appear in the **Project Manager** under **Nets**.

Assigning Nets Automatically

If you have two conducting objects touching each other, Q3D Extractor treats them as one combined object. Q3D Extractor can auto identify nets by finding touching conductors

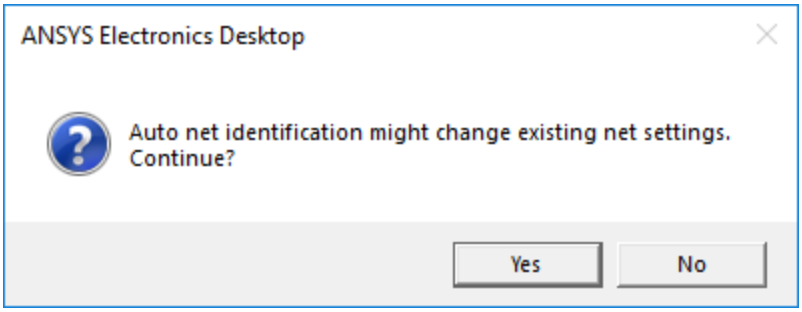
> **Important:**
>
> - Unconnected multilump solids cannot be assigned as nets.
> - Perfect electric conductors (PECs) and non-PECs cannot be part of the same net.
> - Automatically identifying nets may undo any changes you've made to net assignments.

To automatically identify nets:

1. Select one of the following:

   - Right-click in the Modeling Workspace and select **Auto Identify Nets**.
   - Right-click **Nets** in the **Project Manager** and select **Auto Identify Nets**.
   - Select [**Q3D Extractor**] > **Nets** > **Auto Identify Nets**.

   A prompt warns you that your changes may be undone:



2. Click **Yes** to continue.

Nets appear in the **Project Manager** under **Nets**.

Usually, the default conductor assignments generated by this command do not need to be changed.

After auto identifying, the name of the net depends on the number of objects:

- If the net consists of a single object, the object's name is used as the net name.
- When a multi-object net is auto identified, an automatically generated name is given to the new net.

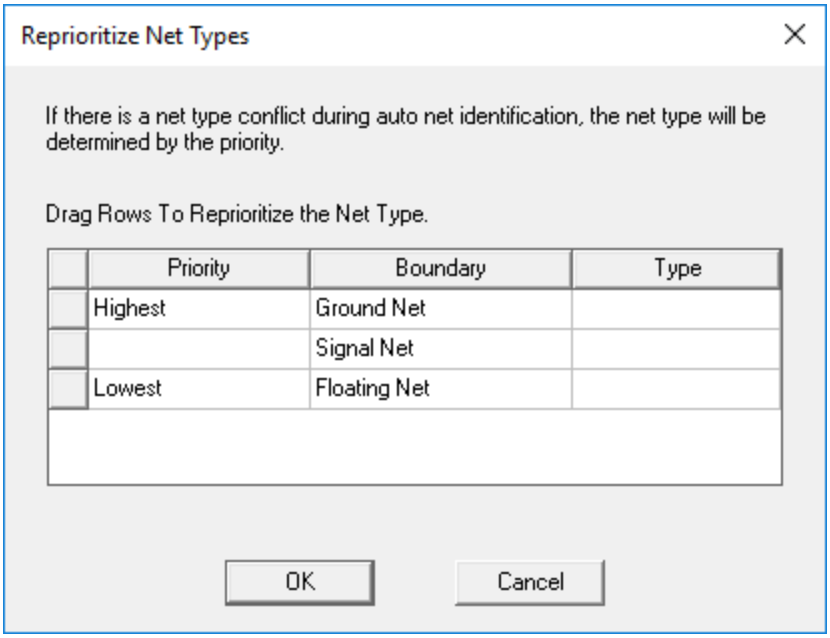
Reprioritizing Net Types

A net type can have a higher priority than other net types, and you can change the priority levels. Reprioritizing nets solves priority conflicts during automatic net identification.

To reprioritize net types:

1. Select **Q3D Extractor** > **Nets** > **Reprioritize Net Type**.

   The **Reprioritize Net Types** dialog box appears.



2. Drag rows to reprioritize. The top row has highest priority and the bottom row has lowest.
3. Click **OK**.

These settings will apply when you auto identify nets.

Setting Material Thresholds in Q3D Extractor

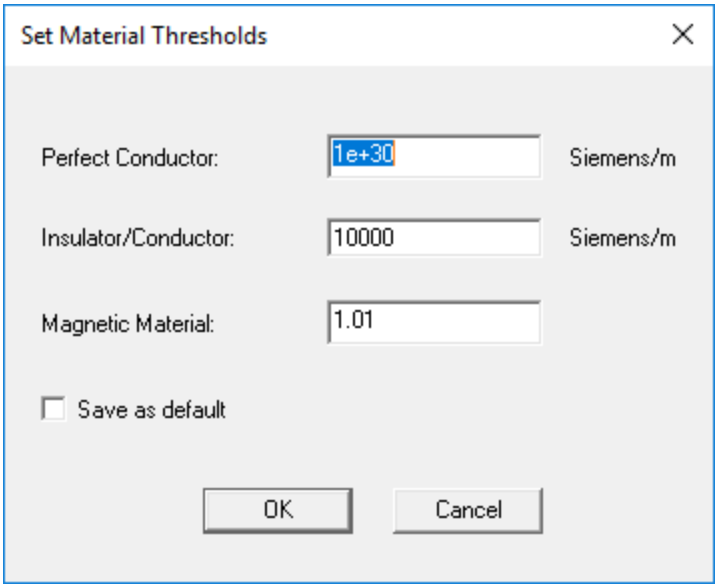
Materials thresholds tell Q3D Extractor how to handle conductive and magnetic materials.

The conductivity thresholds are used while identifying nets to determine which objects are considered conductors. Objects with conductivity equaling or exceeding the threshold are considered to be conductors, and those with lower conductivity are considered to be non-conductors. Materials that fall between the two thresholds are treated as normal conductors that can carry current throughout the volume of the material.

To set material thresholds:

1. Click **Q3D Extractor** > **Nets** > **Set Material Threshold**.

   The **Set Material Thresholds** dialog box appears.



2. In the **Perfect Conductor** field, enter a value. Any conductor having conductivity above or equal to this threshold is considered a perfect conductor in the design. These are treated as having infinite conductivity and surface current only.
3. In the **Insulator/Conductor** field, enter a value. Any object having conductivity equal to or less than this conductivity is considered an insulator in the design. These are treated as having no conductivity and no current carrying capability.
4. In the **Magnetic Material** field, enter a value. Any object having permeability equal to or greater than this is considered a magnetic object in the design.
5. If desired, select the **Save as default** check box to make these settings your default.
6. Click **OK**.

   The material thresholds are applied.

Assigning Terminals (Sources and Sinks)

You can assign sources and sinks to any 2D or 3D object.

If you want to generate a capacitance solution, you do not need to assign any sources or sinks. However, for resistive or inductive solutions, any net that you want to include in the inductance/resistance matrix solution must have a single sink defined, along with at least one source.

Before assigning either a source or a sink, you must select a face of the 3D object or select a 2D object that is touching a 3D object.

A whole face of a 3D object must be chosen to apply an excitation. You cannot cut or use boolean operations to split a planar face of a body into multiple pieces and set the excitation on

one piece(s). If an excitation should be applied to part of a larger plane face, a 2D sheet body of the needed size can be created and the excitation can be applied to that body. It is necessary for an excitation to be on a complete face or sheet body to ensure the mesh outlines are correct.

To assign a source or sink to an object:

1. Perform one of the following actions:
   - In the **Modeling Workspace**, right-click an object and select **Assign Excitation** > [**Source**/**Sink**].
   - In the **Project Manager**, right-click an object and select **Assign** > [**Source**/**Sink**].
   - Select an object and click **Q3D Extractor** > **Nets** > **Assign Excitation** > [**Source**/**Sink**].

   A dialog box prompts you to name the source/sink.
2. Enter a name for the source/sink.
3. Click **OK**.

To assign a source or sink from a list of objects:

1. Access the **Assign Terminals from List** dialog box one of two ways:
   - Select **Q3D Extractor** > **Nets** > **Assign Terminals from List**.
   - In the **Project Manager**, right-click **Nets** and select **Assign Terminals from List**.

   The **Assign Terminals from List** dialog box appears.

2. Select an object.

   You can also use the **Filter** field to search for objects by regular expression.

3. Use the **Terminals** drop-down menu to assign either **Source** or **Sink**. Click **Clear Assignment** to remove a Source or Sink and return the object to **Unassigned** state.
4. Click **OK**.

Toggling Terminal Type

By default, Q3D Extractor's AC RL and DC RL solvers use Constant Voltage (equipotential) terminals. For AC-RL calculations, a constant voltage is held over the perimeter of the terminal face. For DC-RL calculations, a constant voltage is held over the area of the terminal face.

However, in some applications, the properties of these terminals can lead to unwanted or non-physical inductance values.

An alternate form of excitation, the Uniform Current, enforces the condition that the current along the perimeter of a terminal is uniformly distributed for AC-RL calculations, and that the current over the area of a terminal is uniformly distributed for DC-RL calculations.

For AC-RL calculations, both Constant Voltage and Uniform Current terminals use the following procedure for extraction:

1. For excitation of the structure, one terminal is excited with 1V while all other terminals are kept at 0V.
2. Q3D Extractor solves the system equation:

   $$\mathbf{Z} * \mathbf{I} = \mathbf{V}$$

   and calculates inductance using the terminal equation:

   $$\mathbf{Y} = \mathbf{I}/\mathbf{V}$$

   with Y being the inverse of jωL.

For DC-RL calculations, see: Extracting DC Inductance from the Field Solution.

The differences between Constant Voltage and Uniform Current are described below.

Constant Voltage

- The potential on the terminal perimeter is constant for AC calculations, and constant on the terminal area for DC calculations.
- The current distribution is non-uniform on the perimeter of the terminal, or on the terminal area.
- Introducing additional terminals will change the path of the current, and self-inductance entries may also change.

Uniform Current

- Forces a uniform current distribution on the perimeter of the terminals for AC calculations, or on the area of the terminals for DC calculations to approximate those terminals as point ports.
- For terminals comprising multiple disjoint facets, each facet has its own constant current density different from all other facets. This is for the scenario of a terminal defined over a pingroup.
- The voltage on the terminal is not constant, and voltage (1V or 0V) will be observed at one point only.
- Due to the uniform current distribution, the self-inductance is usually higher.

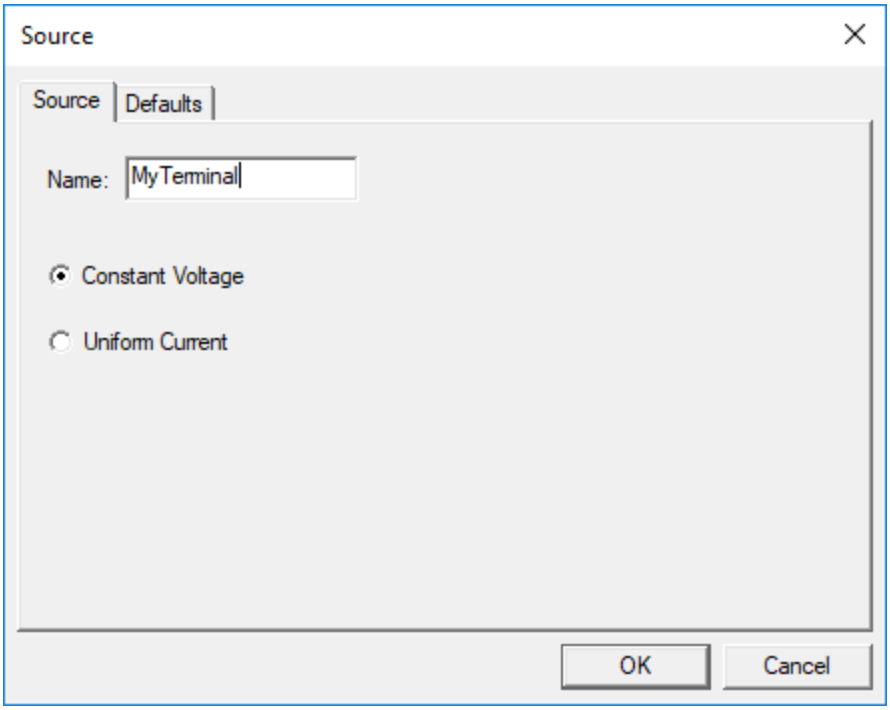
Changing a Terminal's Type

You can change a terminal's type to either Constant Voltage or Uniform Current, as appropriate. Q3D Extractor can support having both types in a single design.

To change the type of a single terminal:

1. From the **Project Manager**, double-click a sink or source terminal.

   The [**Source** / **Sink**] window appears.



2. Select either **Constant Voltage** or **Uniform Current**.
3. Click **OK**.

Changing All Terminals

To change all terminals in the design to one type, perform one of the following:

- Select **Q3D Extractor** > **Nets** > **Toggle Terminal Type** > **All to** [**Constant Voltage** / **Uniform Current**].
- Right-click in the modeling workspace and select **Toggle Terminal Type** > **All to** [**Constant Voltage** / **Uniform Current**].
- In the **Project Manager**, right-click a terminal and select **Toggle** > **All Terminals to** [**Constant Voltage** / **Uniform Current**].

13 - Solution Setups in Q3D Extractor and 2D Extractor

The **Solve Setup** window allows you to specify how the solution will be computed.

You can define more than one setup per design, and 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.

Adding a Solution Setup

To add a solution setup to a design:

1. Select a design from the Project Manager.
2. Open the **Solve Setup** window one of three ways:
   - Select **2D Extractor** > **Analysis Setup** > **Add Solution Setup**.
   - Right-click **Analysis** in the Project Manager and select **Add Solution Setup**.
   - Copy and paste an existing sweep solution setup, then double-click it to change its properties.

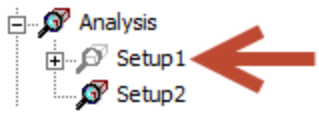
     > **Note:**
     >
     > Copying and pasting only works on sweep setups.
3. Select the appropriate settings. See: detailed instructions for Q3D Extractor and 2D Extractor.
4. Click **OK**.

Enabling and Disabling a Solution Setup

You can enable or disable a setup two ways:

- In the **Solve Setup** window, select or deselect the **Enabled** check box.
- In the **Project Manager**, right-click a setup and select **Enable Setup** or **Disable Setup**.

Disabled setups appear grayed out in the **Project Manager**:



Deleting a Solution Setup

You can delete a setup from the **Project Manager** two ways:

- Select the setup and press **Delete**.
- Right-click the setup and select **Delete**.

Duplicating a Solution Setup

You can duplicate an existing setup by right-clicking it and selecting **Copy** (or by using **Edit** > **Copy**), then selecting **Analysis** in the Project Manager and pasting the setup. By default, the copy is named Setup*n*, where *n* increments with each new setup. You can then edit each copy to make desired changes.

Specifying Solution Settings in Q3D Extractor

The **Solve Setup** window allows you to specify how the solution will be computed.

You can define more than one setup per design, and each solution setup includes:

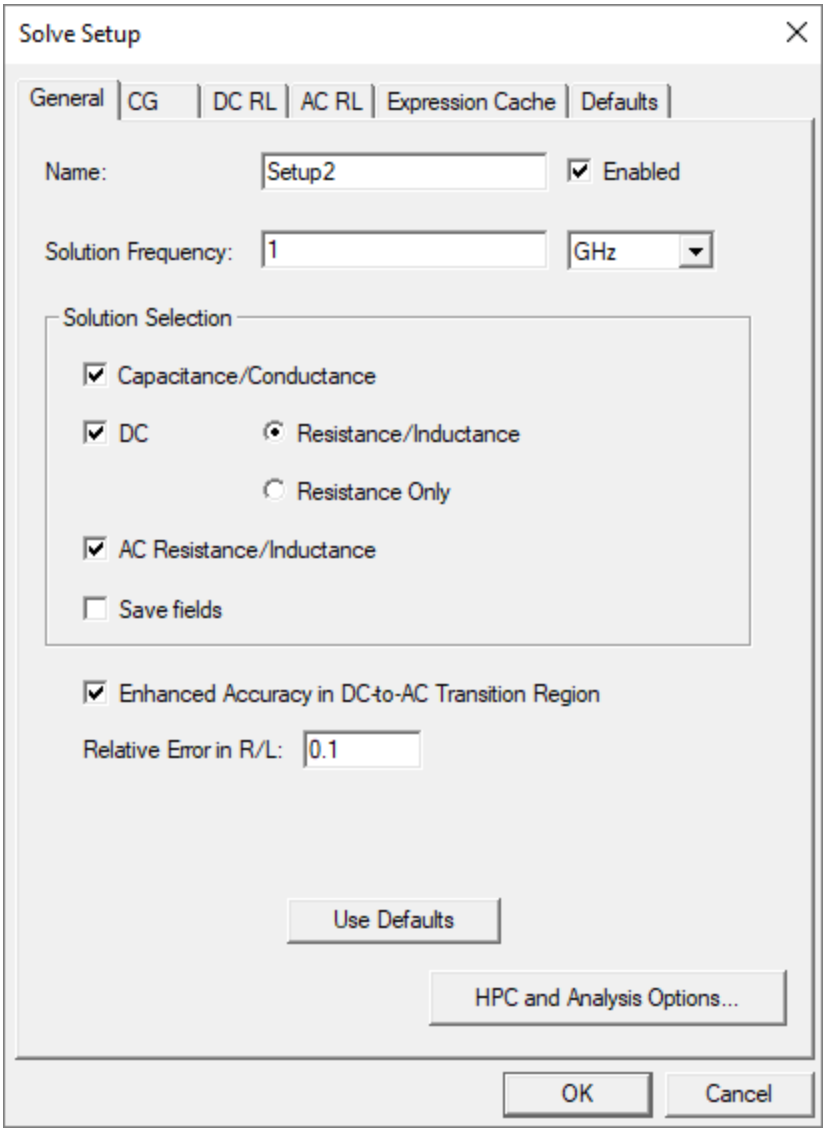
- General data about the solution's generation
- Adaptive mesh refinement parameters if you want the mesh to be refined iteratively in areas of highest error

To add a solution setup to a design:

1. Select a design from the Project Manager.
2. Open the **Solve Setup** window one of three ways:
    - Select **Q3D Extractor** > **Analysis Setup** > **Add Solution Setup**.
    - Right-click **Analysis** in the Project Manager and select **Add Solution Setup**.
    - Copy and paste an existing sweep solution setup, then double-click it to change its properties.

      > **Note:**
      >
      > Copying and pasting only works on sweep setups.

    The **Solve Setup** window appears, on the **General** tab.



Note:

To display the **Compute Dynamic Fields** check box that enables Far Field calculations, enable the beta option.

3. Enter a **Name** for the solution.
4. Select the **Enabled** check box to enable the solution. Deselect it to make the solution inactive.
5. Specify the **Solution Frequency** and unit of measure.

6. In the **Solution Selection** area, use the check boxes to select the solution type(s).

   By default, **Capacitance/Conductance**, **DC Resistance/Inductance**, and **AC Resistance/Inductance** are selected.

   You can click **Use Defaults** to restore the default settings.

7. If desired, click **HPC and Analysis Options** to change HPC and Analysis settings.
8. Select the relevant tab(s) to make additional settings:
   - **CG** – settings for mesh generation for capacitance and conductance.
   - **DC RL** – settings for mesh generation for DC inductance and resistance.
   - **AC RL** – settings for mesh generation for AC inductance and resistance.
   - **Expression Cache** – list of expressions (including post-processing variables) available for use in convergence for adaptive analysis.
   - **Defaults** – options to save the current settings as the defaults for future solution setups or to revert the current settings.

   > **Note:**
   >
   > Tabs appear and disappear based on which solutions are selected.

9. In order to save disk space, the **Save Fields** check box is disabled by default. To enable fields post processing or workbench coupling, select it.

   > **Note:**
   >
   > This setting cannot be applied to sweeps, as each frequency in a sweep contains a flag not to save fields.

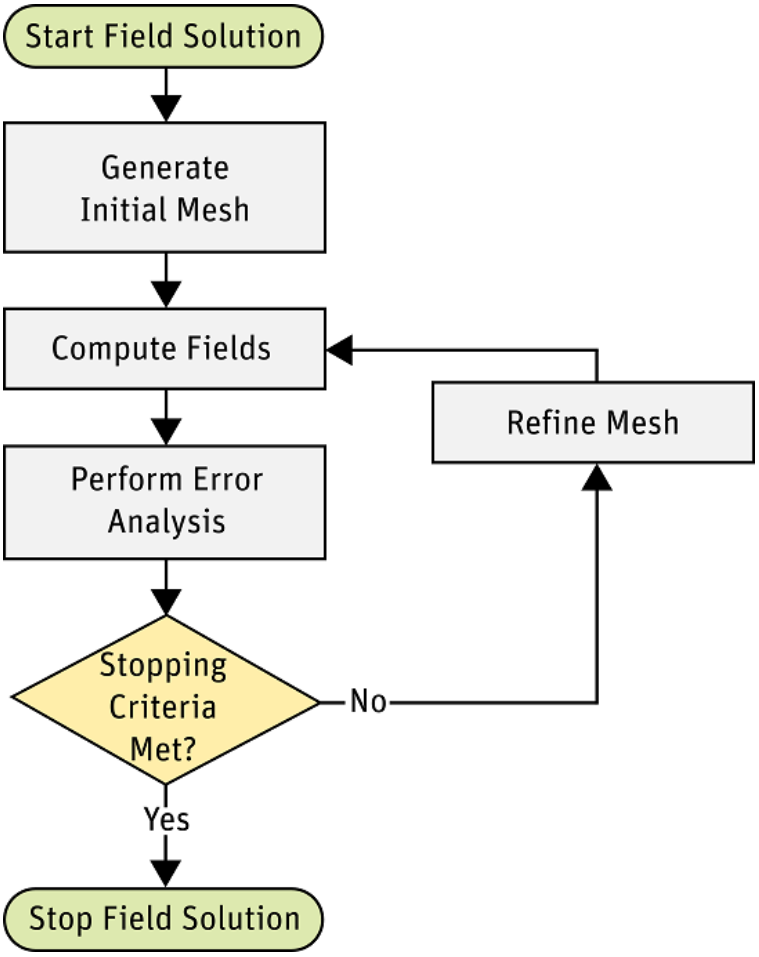
10. If you are performing transition region analysis, enable **Enhanced Accuracy in DC-to-AC Transition Region** and specify the relative error.

    > **Note:** This option is only available if you have enabled both DC and AC analysis.

11. Click **OK**.

Overview: Analysis Parameters in Q3D Extractor

During adaptive analysis the system iteratively refines the starting mesh in order to reduce the size of individual elements in areas of high error. This improves accuracy.



When an adaptive analysis is performed:

- Q3D Extractor generates a field solution using the specified mesh.
- Q3D Extractor computes the desired parameters and the change in the matrix from the previous pass (the matrix delta). It then compares the delta with the **Percent error**. When the matrix delta is less than the specified **Percent error** value, the field solution process stops.
- Q3D Extractor performs an error analysis in each triangle (capacitance and AC analysis) or tetrahedron (DC analysis) in the mesh. Elements with the highest energy error are then refined (broken down into smaller elements), producing a more accurate solution in these areas.
- Another solution is generated using the refined mesh, and Q3D Extractor repeats this process until the stopping criteria are satisfied.

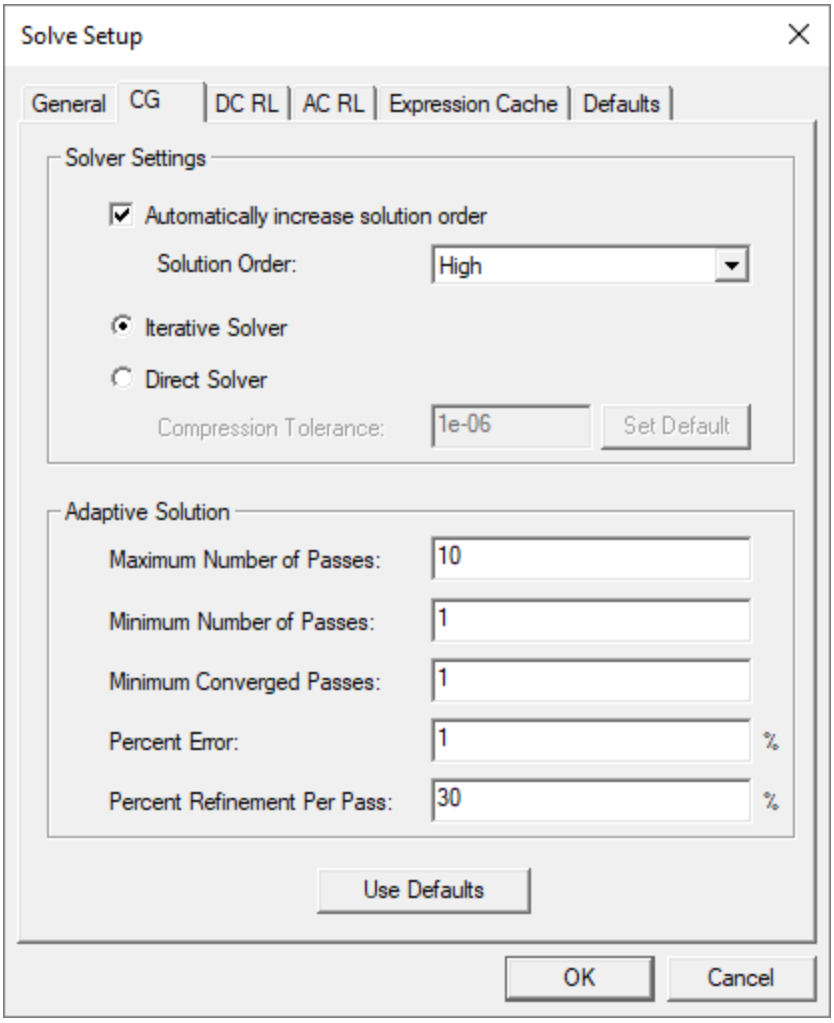
Setting Capacitance and Conductance (CG) Analysis Parameters in Q3D Extractor

To set up a capacitance and conductance (CG) analysis, first add a solution setup.

Then, from the **Solve Setup** window:

1. Select the **CG** tab.

   The **CG** options display.



2. In the **Solver Settings** area:
   - Specify the **Automatically increase solution order** setting.

Solution order refers to the accuracy level. The solver will give accurate results at the **Normal** setting for most applications. The **High**, **Higher**, and **Highest** options offer greater accuracy at the expense of speed and memory.

- Select either **Iterative Solver** or **Direct Solver**.
  - **Iterative Solver** – The Iterative Solver is generally the fastest option. However, it uses FMM compression, which may lead to slow convergence of GMRES iterations for poorly conditioned matrices.
  - **Direct Solver** – The Direct Solver has a higher setup time and uses more memory than the Iterative Solver. In the setup phase of the direct solver, an LU factorization of the FMM-compressed MoM system of equations is performed. In the solution phase, each right hand side is solved using forward (L) and backward(U) substitutions.The Direct Solver converges quickly for poorly conditioned matrices and should only be used in cases requiring longer solution times (e.g., cases with extreme geometries and cases with more than 500 nets).

    By default, the Direct Solver **Solution Tolerance** is *1e-6* for Normal or High solution order, *1e-7* for Higher, and *ie-8* for Highest. Putting a larger tolerance requires less setup time and memory, while putting a tighter tolerance requires more setup time and memory but leads to rapid convergence and faster solution time.

3. In the **Adaptive Solution** area, specify values for the following:
   - **Maximum Number of Passes** – the maximum number of mesh refinement cycles that you would like Q3D Extractor 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 convergence criteria are reached.

     > **Important:**
     >
     > 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 Q3D Extractor requesting more memory than is available or taking excessive time to compute solutions.

   - **Minimum Number of Passes** – the minimum number of mesh refinement cycles. Q3D Extractor will not stop analysis until after this number of passes has been completed.

- **Minimum Converged Passes** – the minimum number of passes that must meet convergence criteria before the adaptive analysis will stop.
- **Percent Error** – the desired solution accuracy. Smaller values produce more accurate but slower solutions; larger values produce less accurate but faster solutions.
- **Percent 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. For example, entering 10 causes the mesh to increase approximately 10 percent each pass. If your mesh consisted of 1000 elements, the tetrahedra or triangles would be refined so that 100 new elements are added to the mesh. The default value is 30% and you can you generally accept the default value.

4. Click **OK**.

Temperature-Dependent CG Analysis

Q3D Extractor's CG solver can intake temperature-dependent dielectric materials.

The process generally involves:

1. Adding a CG setup, as described above.
2. Updating a dielectric material definition to include a thermal modifier.
3. Adding a parametric setup for temperature analysis using the variable $temp_var.
4. Viewing data by design variation. Each temperature is a design variation.

Distributed Memory CG Solutions (Beta)

Q3D Extractor's Distributed Memory CG Solver (MPI-FMM) can take advantage of cloud computing platforms in order to simulate very large designs. By redistributing the mesh so that each MPI task handles only a partial mesh, and by parallelizing tasks, MPI-FMM allows Q3D Extractor to handle designs containing 100 million+ mesh elements.

For more information, see: Distributed Memory CG Solutions.

Infinite Ground Planes

> **Note:**
>
> When infinite ground planes are present, CG results may differ slightly from releases prior to 2022 R2.

Distributed Memory CG Solutions

Q3D Extractor's Distributed Memory CG Solver (MPI-FMM) can take advantage of cloud computing platforms in order to simulate very large designs. By redistributing the mesh so that each MPI task handles only a partial mesh, and by parallelizing tasks, MPI-FMM allows Q3D Extractor to handle designs containing 100 million+ mesh elements.

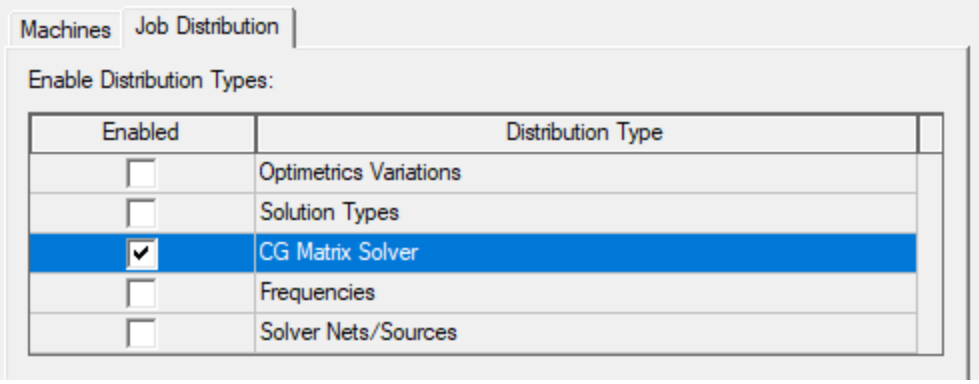
Important:

- Ansys recommends using Distributed Memory CG Solutions for applications that are large in size and cannot fit on a single node.
- Using too many tasks for smaller meshes will slow down the solver.
- For best performance, use a smaller number of cores per task (ideally 4).
- CG Matrix Solver cannot be enabled simultaneously with the Solver Nets/Sources Distribution Type option.

Submitting a Distributed Memory CG Solution

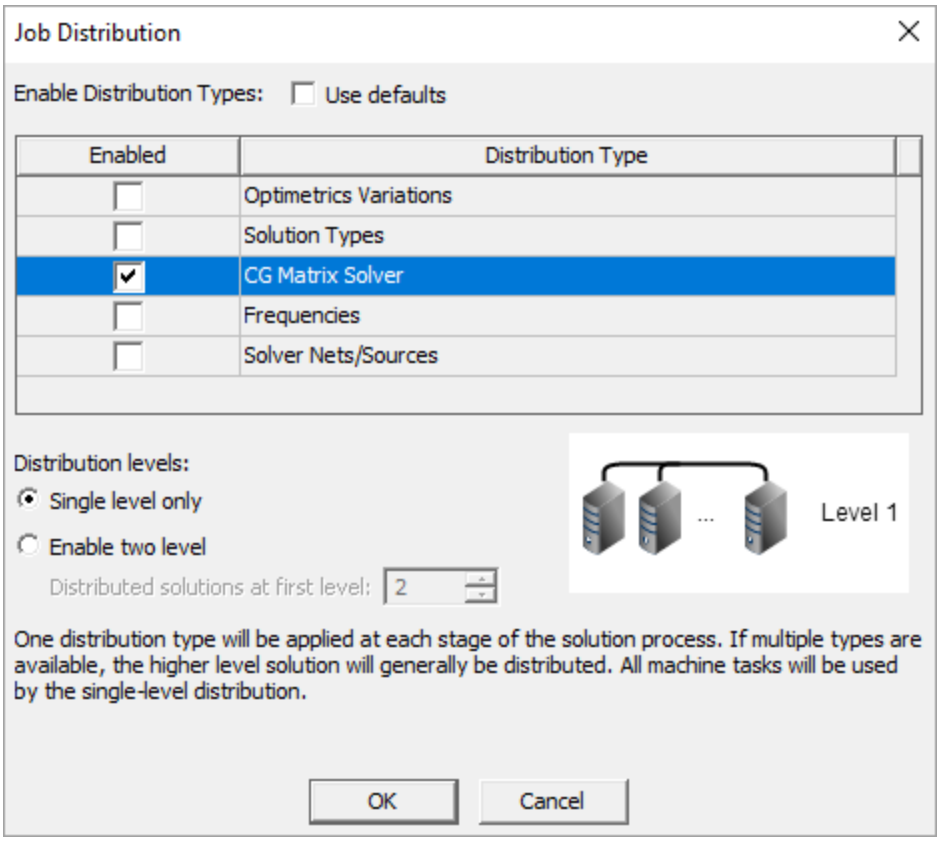
To use the MPI-FMM solver:

1. Create a CG solution setup.
2. Prepare to submit an HPC job.
3. Enable **CG Matrix Solver** one of two ways:
    - From the solution setup's **General** tab:
        - Click **HPC and Analysis Options**.
        - Select an HPC configuration and click **Edit** to launch the **Analysis Configuration** window.
        - From the **Job Distribution** tab, select the check box next to **CG Matrix Solver** to enable it.



- Click **OK**.

- From the **Submit Job** window:
  - Select the **Compute Resources** tab.
  - Ensure **Use Automatic Settings** is disabled.
  - In the **Job Distribution** area, click **Modify** to launch the **Job Distribution** window.
  - Select the check box next to **CG Matrix Solver** to enable it.



  - Click **OK**.

> **Important:**
>
> CG Matrix Solver *cannot* be enabled simultaneously with Solver Nets/Sources.

4. Submit the job.

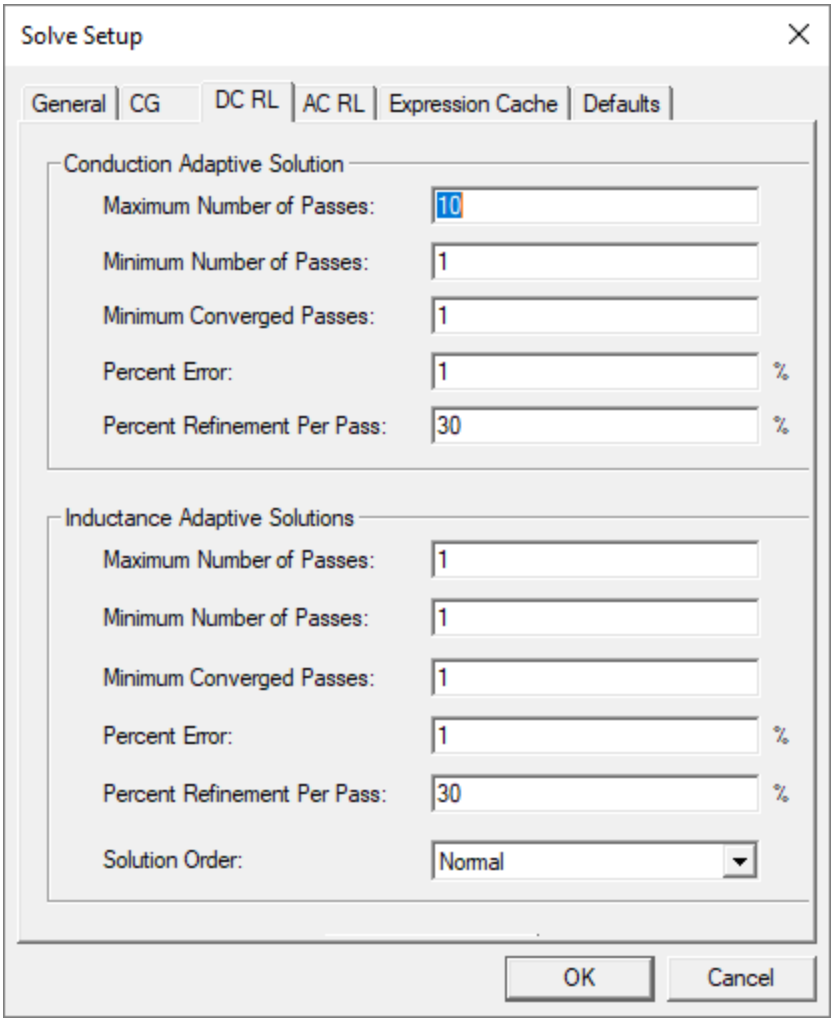
Setting DC Resistance and Inductance (RL) Analysis Parameters in Q3D Extractor

To set up a DC Resistance and Inductance (RL) analysis, first add a solution setup.

Then, from the **Solve Setup** window:

1. Select the **DC RL** tab.

   The **DC RL** options display.



2. In the **Conductance Adaptive Solution** and **Inductance Adaptive Solutions** areas, specify values for the following:

- **Maximum Number of Passes** – the maximum number of mesh refinement cycles that you would like Q3D Extractor 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 convergence criteria are reached.

  > **Important:**
  >
  > 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 Q3D Extractor requesting more memory than is available or taking excessive time to compute solutions.

- **Minimum Number of Passes** – the minimum number of mesh refinement cycles. Q3D Extractor will not stop analysis until after this number of passes has been completed.
- **Minimum Converged Passes** – the minimum number of passes that must meet convergence criteria before the adaptive analysis will stop.
- **Percent Error** – the desired solution accuracy. Smaller values produce more accurate but slower solutions; larger values produce less accurate but faster solutions.
- **Percent 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. The default value is 30%.

  > **Note:**
  >
  > Magnetic materials (conducting and non-conducting) affect inductance in the DC RL solution. If magnetic materials are present, and you run more than one inductance adaptive solution, Q3D Extractor will do adaptive solve on both surface and volume meshes. If magnetic materials are present in a model, it is recommended to solve for five or more inductance passes.

- **Solution Order** – refers to the accuracy level. The solver will give accurate results at the **Normal** setting for most applications. The **High**, **Higher**, and **Highest** options offer greater accuracy at the expense of speed and memory.

3. Click **OK**.

Important:

For designs containing Pulse Width Modulated (PWM) voltages and currents (common in power electronics), an advanced workflow is required to accurately estimate losses in bus bars, connectors, and PCBs. See: Electrothermal Flow for Power Electronics.

Setting AC Resistance and Inductance (RL) Analysis Parameters in Q3D Extractor

Q3D Extractor's AC RL solves quasi-magnetostatic problems using either Adaptive Cross Approximation (ACA) or Multilevel Fast Multipole Method (MLFMM). Both are effective and accurate methods of solving surface integral equations discretized via Method of Moments (MoM).

- ACA is best for solving small problems, and is the default solver.
- MLFMM is more efficient in both RAM consumption and simulation time for large problems. MLFMM is best used in applications with more than 1 million triangles.

  Important:

  The ACA solver *must* be used in the following cases:

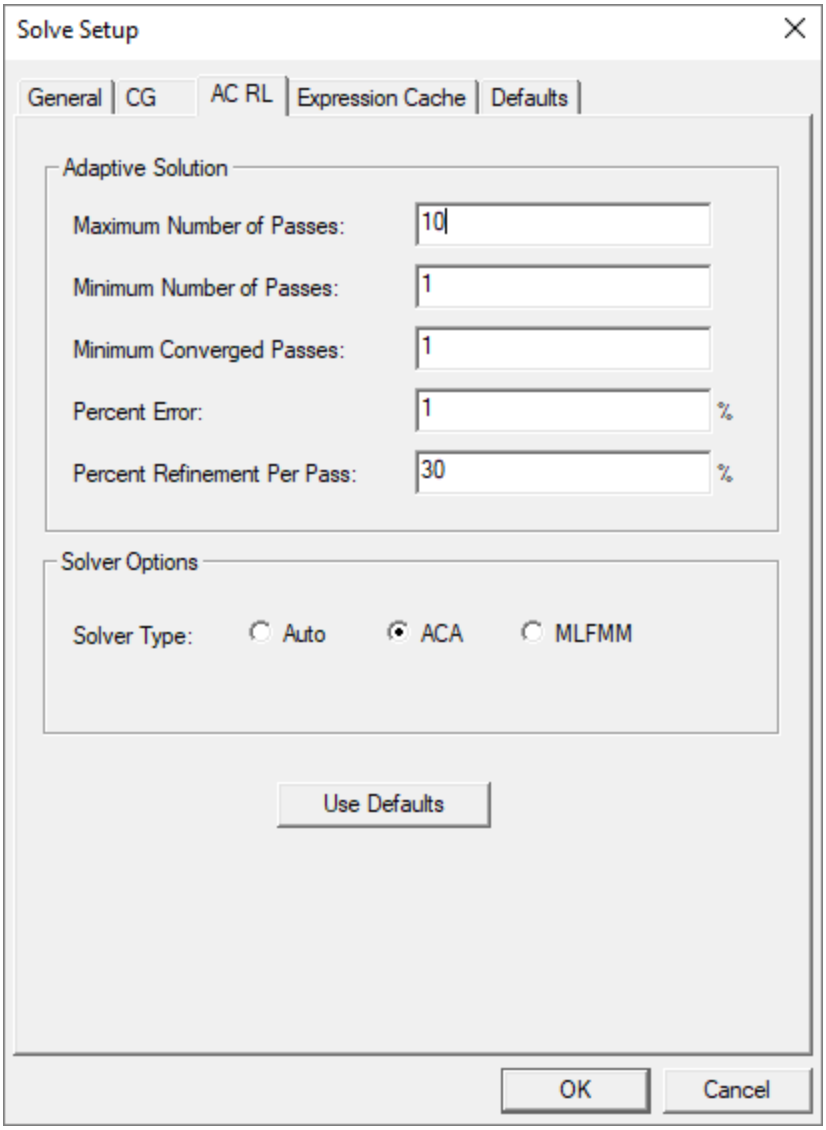
  - For models with an infinite ground plane.
  - For models containing magnetic materials.
  - When Enhanced Accuracy in DC-to-AC Transition Region is enabled.

To set up an AC RL analysis, first add a solution setup.

Then, from the **Solve Setup** window:

1. Select the **AC RL** tab.

   The **AC RL** options display.



2. In the **Adaptive Solution** area, specify values for the following:

    - **Maximum Number of Passes** – the maximum number of mesh refinement cycles that you would like Q3D Extractor 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 convergence criteria are reached.

> **Important:**
>
> 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 Q3D Extractor requesting more memory than is available or taking excessive time to compute solutions.

- **Minimum Number of Passes** – the minimum number of mesh refinement cycles. Q3D Extractor will not stop analysis until after this number of passes has been completed.
- **Minimum Converged Passes** – the minimum number of passes that must meet convergence criteria before the adaptive analysis will stop.
- **Percent Error** – the desired solution accuracy. Smaller values produce more accurate but slower solutions; larger values produce less accurate but faster solutions.
- **Percent Refinement Per Pass** – the *maximum* growth rate, per pass. This determines how many triangles are added at each iteration of the adaptive refinement process. The triangles with the highest error are then refined. The default value is 30%.

> **Note:**
>
> A growth rate lower than the maximum occurs when there are not enough high error elements to warrant adaption. This reduces memory usage without sacrificing accuracy.

> **Note:**
>
> Magnetic conducting materials only impact resistance, not inductance since the AC solution assumes a fully developed skin effect (surface current).

3. In the **Solver Options** area, select one of the following:
   - **Auto** – When selected, Q3D Extractor automatically decides whether to use ACA or MLFMM, depending on the mesh size. ACA will be automatically selected for any problems with fewer than 300,000 triangles, as well as for any designs with infinite ground, magnetic materials, or transition region solve enabled.
   - **ACA** – Adaptive Cross Approximation.
   - **MLFMM** – Multilevel Fast Multipole Method.

4. Click **OK**.

Transition Region RL Analysis

For RL frequency sweeps, Q3D Extractor calculates RL values at a DC point and at an AC point (very high frequency), and uses the blending method to obtain the frequency sweep between DC and AC values. The method is fast and provides satisfying results for most applications. However, for some applications (such as power electronics), the transition region (between DC and AC) RL values are of more value.

For these scenarios, Q3D Extractor has an option to enable enhanced accuracy in the DC to AC transition region. With this option enabled, Q3D Extractor computes RL values at some points in the transition region (between KHz and MHz), and combines these values with the values at DC and AC points for an accurate frequency sweep. Q3D Extractor solves the field inside and outside a conductor and enforces the boundary condition on the conductor surface to provide accurate RL values at the transition region frequencies.

To perform transition region analysis:

1. From the **Solve Setup** window, enable both AC RL and DC RL simulations, and select **Enhanced Accuracy in DC-to-AC Transition Region**.
2. The **Relative Error in R/L** field determines accuracy. In most cases, the default value (0.1) is sufficient. If a more accurate frequency sweep is desired, 0.01 or 0.001 can be used and Q3D Extractor will solve more points in the transition region. However, using a smaller error percentage can cause the transition region solver to consume a great deal of time and memory.
3. Run the AC and DC simulations.

   Q3D Extractor collects the DC, AC and transition region RL values and uses the transition region interpolator to plot R/L.

4. Generate reports or export s-parameter data.

Electrothermal Flow for Power Electronics

When simulating the bus bars, connectors, and PCBs of power electronics converters, most cases deal with Pulse Width Modulated (PWM) voltages and currents. In these cases, Q3D Extractor alone cannot generate DC equivalent circuits that:

- Accurately represent power losses, and
- Satisfy the current balance in accordance with Kirchhoff's first law.

Therefore, an advanced workflow is required to accurately estimate losses in bus bars, connectors, and PCBs for power electronics. There are two choices:

- Time Domain Workflow using Maxwell A-Phi solver
- Frequency Domain Workflow using Q3D Extractor or SIwave (for PCBs)

Harmonic Loss Calculation

Q3D's Harmonic Loss calculation is a post-processing field for the DC RL simulation. It is based on Parseval's Power Theorem, which states that the power of a signal is equal to the sum of square of the magnitudes of various harmonic components present in the discrete spectrum.

Power loss density is calculated for the real and imaginary parts of the net current at each frequency separately.

The skin and proximity effects on losses are not simulated directly. Instead, they are approximated by scaling the dissipated power of a net with the AC resistance value from the matrix at each frequency. This approach cannot provide high accuracy in the power loss density distribution for cases in which geometry parts with different conductivity values are combined in one net.

Frequency Domain Workflow using Q3D Extractor

For one period of power converter output frequency, perform the following:

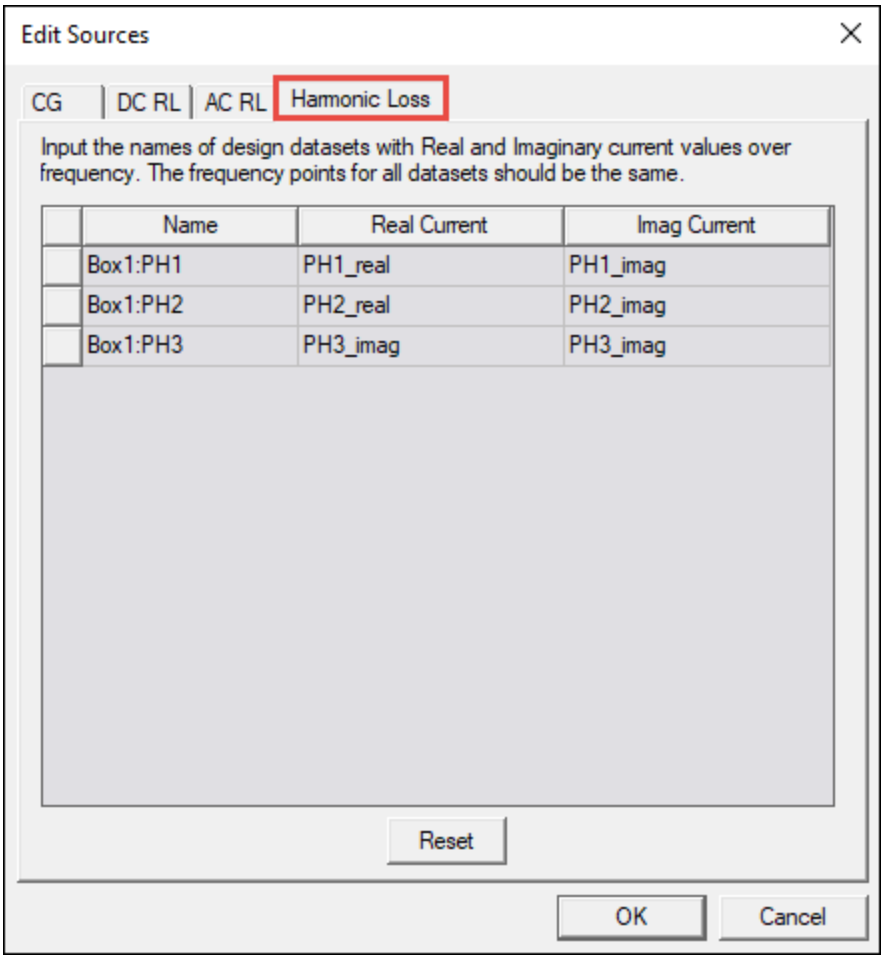
1. Set up a Q3D Extractor design. Ensure that **DC RL**, **AC RL**, and **Save Fields** are enabled.
2. Add a Frequency Sweep to the solution setup. Ensure that the type is **Interpolating**, and that the frequency points cover the entire range of interest, including DC point.

   > **Tip:**
   >
   > For higher accuracy of the power loss density distribution, consider applying additional mesh operations (Inside Selection > Length-Based).

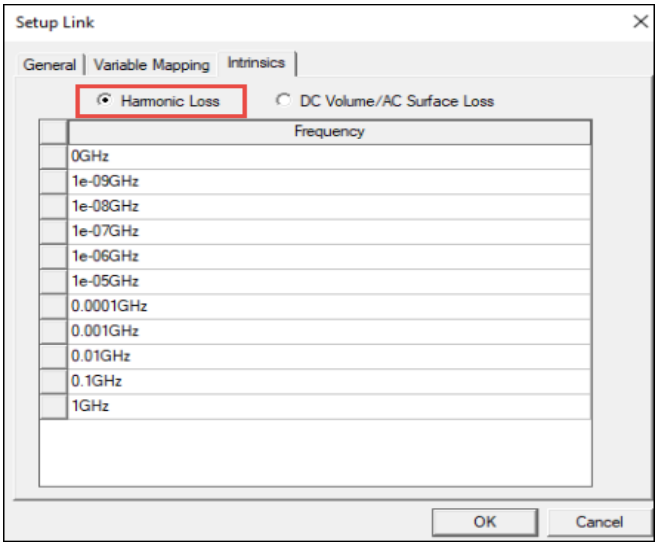
3. Create Design Datasets for frequency-dependent real and imaginary currents at each source either manually or by performing the following steps:
   a. Create a Circuit design, for which the Q3D model is a part. If using Ansys Circuit or Ansys Twin Builder, you can include the Q3D parasitic model using **Dynamic Link**.
   b. Run a Circuit Transient analysis.
   c. After Transient simulation reaches steady state for one period of fundamental frequency, record the current values corresponding to each source in the Q3D Extractor model.
   d. Perform FFT on all recorded currents, plotting real and imaginary parts separately.
   e. Export real and imaginary currents vs. frequency for all Q3D sources as datasets, ensuring that the frequency points are the same in all datasets.
   f. In the Q3D Extractor design, import the datasets as Design Datasets.

> **Note:**
>
> Optionally, you can enable and use the beta option **Push Excitations to Q3D from Circuit**.

4. In the Edit Sources window, select the **Harmonic Loss** tab and set real and imaginary current values for the corresponding sources.



5. Run the Q3D simulation.
6. Select the geometry and right-click **Field Overlays** in the **Project Manager**. Select **DC R/L Fields** > **Harmonic_Loss_Density**. Total loss value can be calculated using the Fields Calculator.
7. If coupling with a thermal solver inside Electronics Desktop (i.e., EM Loss link to Icepak), open the **Setup Link** dialog box and ensure that **Harmonic Loss** is selected on the **Intrinsics** tab before analysis.



Specifying Solution Settings in 2D Extractor

The **Solve Setup** window allows you to specify how the solution will be computed.

You can define more than one setup per design, and each solution setup includes:

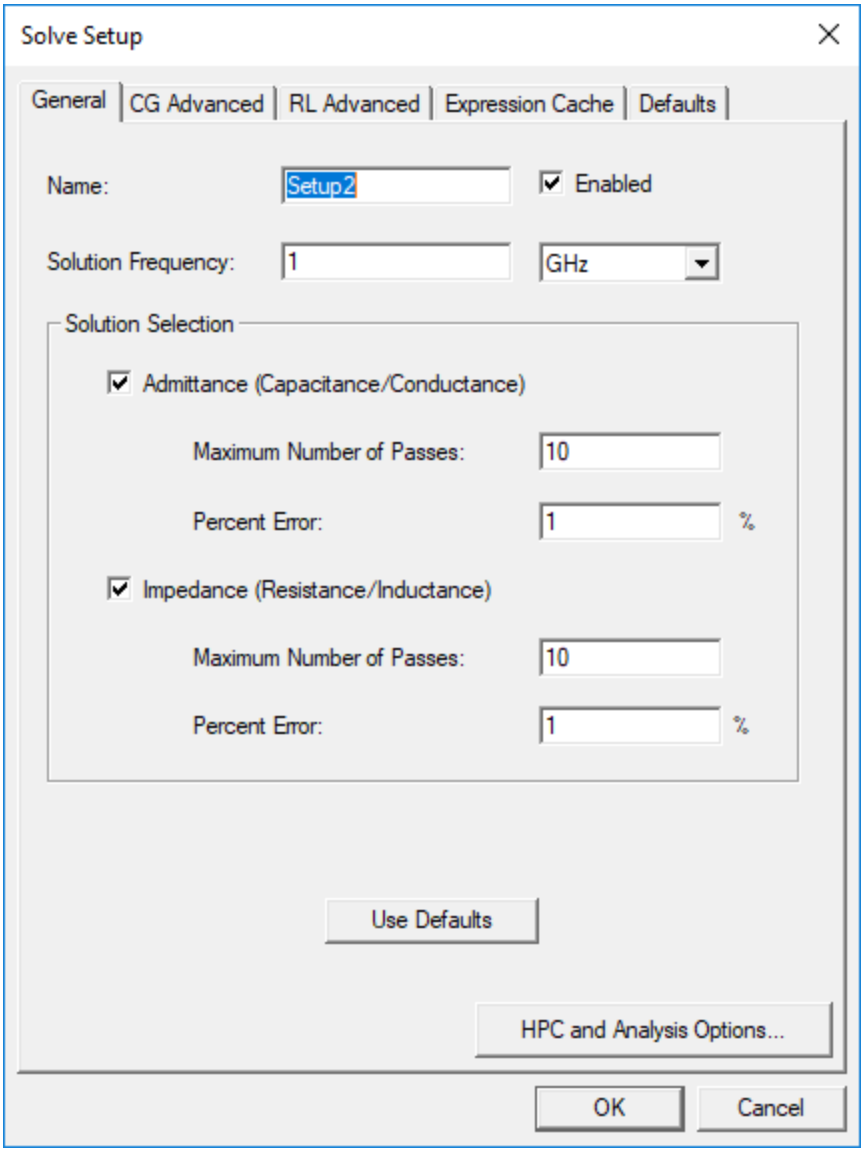
- General data about the solution's generation
- Adaptive mesh refinement parameters if you want the mesh to be refined iteratively in areas of highest error

To add a solution setup to a design:

1. Select a design from the Project Manager.
2. Open the **Solve Setup** window one of three ways:
   - Select **2D Extractor** > **Analysis Setup** > **Add Solution Setup**.
   - Right-click **Analysis** in the Project Manager and select **Add Solution Setup**.
   - Copy and paste an existing sweep solution setup, then double-click it to change its properties.

     > **Note:**
     >
     > Copying and pasting only works on sweep setups.

   The **Solve Setup** window appears, on the **General** tab.



3. Enter a **Name** for the solution.
4. Select the **Enabled** check box to enable the solution. Deselect it to make the solution inactive.
5. Specify the **Solution Frequency** and unit of measure.
6. In the **Solution Selection** area, use the check boxes to select the solution type(s).

   By default, **Admittance (Capacitance/Conductance)** and **Impedance (Resistance/Inductance)** are selected.

   You can click **Use Defaults** to restore the default settings.
7. If desired, click **HPC and Analysis Options** to change HPC and Analysis settings.

8. Select the relevant tab(s) to make additional settings:
   - **CG Advanced** – settings for mesh generation for capacitance and conductance.
   - **RL Advanced** – settings for mesh generation for inductance and resistance.
   - **Expression Cache** – list of expressions (including post-processing variables) available for use in convergence for adaptive analysis.
   - **Defaults** – options to save the current settings as the defaults for future solution setups or to revert the current settings.

   > **Note:**
   >
   > Tabs appear and disappear based on which solutions are selected.

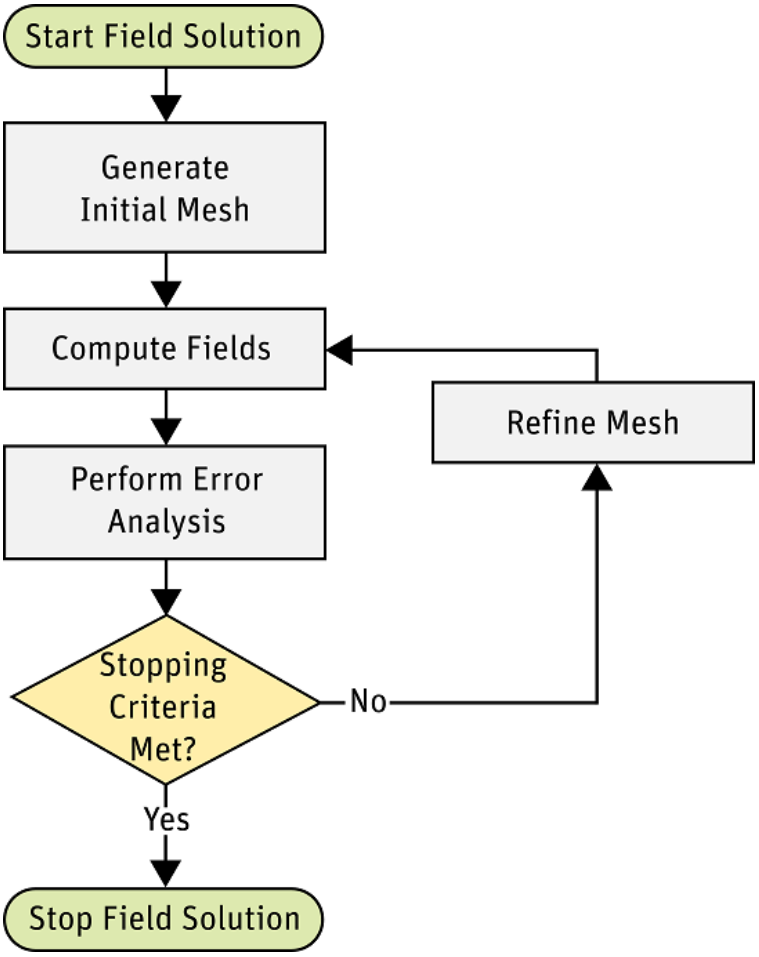
9. In order to save disk space, the **Save Fields** check box is disabled by default. To enable fields post processing or workbench coupling, select it.

   > **Note:**
   >
   > This setting cannot be applied to sweeps, as each frequency in a sweep contains a flag not to save fields.

10. Click **OK**.

Overview: Analysis Parameters in 2D Extractor

During adaptive analysis the system iteratively refines the starting mesh in order to reduce the size of individual elements in areas of high error. This improves accuracy.



When an adaptive analysis is performed:

- 2D Extractor generates a field solution using the specified mesh.
- 2D Extractor computes the desired parameters and convergence parameters (Total Energy, Delta Energy, Energy Error, Total Loss, Delta Loss, and Loss Error). It then compares the computed convergence parameters with the target **Percent error**. The field solution process stops when it is less than the specified **Percent error** value.
- 2D Extractor performs an error analysis in each triangle in the mesh. When the error targets are not satisfied, the mesh is refined. Elements with the highest energy error are broken down into smaller elements during the refinement process, producing a more accurate solution in these areas.
- Another solution is generated using the refined mesh, and 2D Extractor repeats this process until the stopping criteria are satisfied.

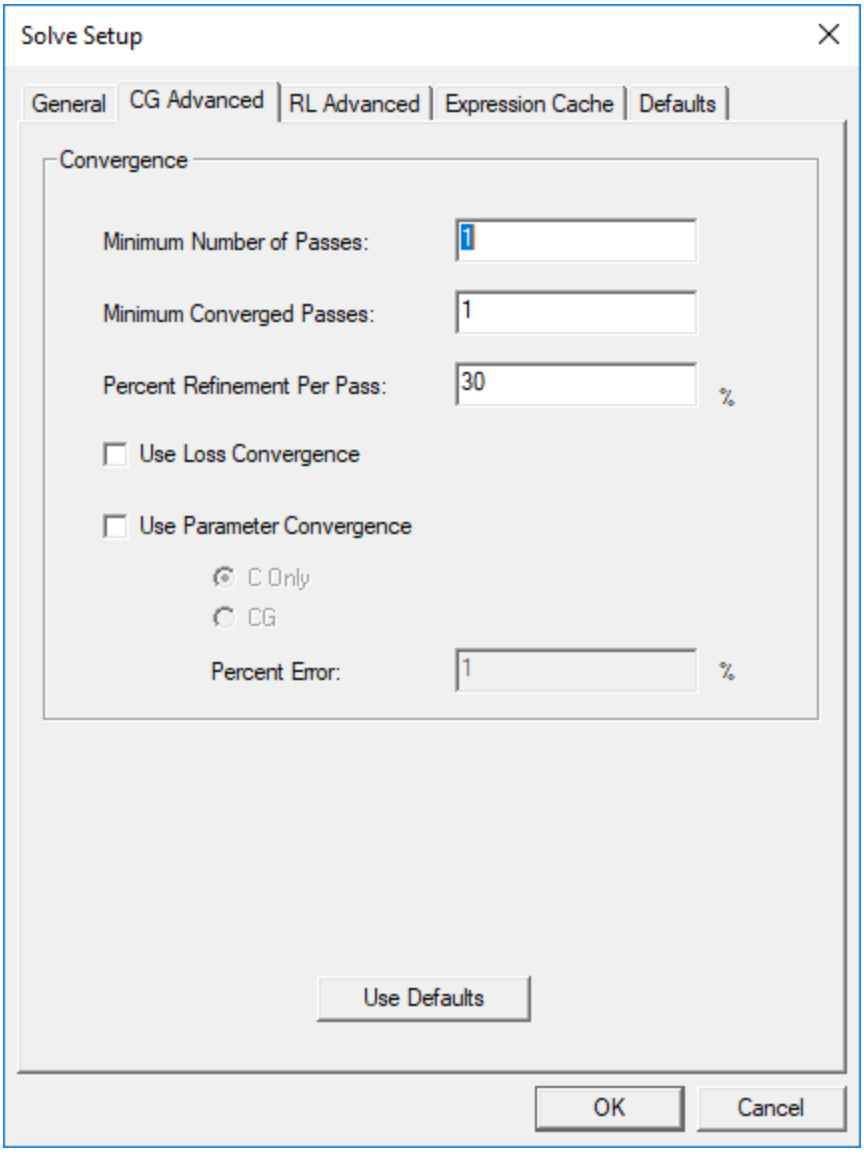
Setting Capacitance and Conductance (CG) Analysis Parameters in 2D Extractor

To set up a capacitance and conductance (CG) analysis, first add a solution setup.

Then, from the **Solve Setup** window:

1. Select the **CG Advanced** tab.

   The **CG Advanced** options display.



2. In the **Convergence** area, specify values for the following:

    - **Minimum Number of Passes** – the minimum number of mesh refinement cycles. 2D Extractor will not stop analysis until after this number of passes has been completed.
    - **Minimum Converged Passes** – the minimum number of passes that must meet convergence criteria before the adaptive analysis will stop.
    - **Percent 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. For example, entering 10 causes the mesh to increase approximately 10 percent each pass. If your mesh consisted of 1000 elements, the tetrahedra or triangles would be refined so that 100 new elements are added to the mesh. The default value is 30% and you can you generally accept the default value.
    - **Use Loss Convergence** – select to use loss in the convergence criteria.
    - **Use Parameter Convergence** – select to use CGRL matrices in calculating convergence. Parameter convergence is determined based on the values of previous pass matrix and the current matrix. The maximum delta of a matrix entry is calculated by comparing entries of previous matrix and current matrix. The final delta is given by: Final delta (%) = 100 X (maxdelta / max value).

      If the final delta is less than the target delta, then the pass is considered converged with respect to parameter convergence.

      Select **C Only** for a lossless matrix, or **CG** for a lossy matrix.

        - **Percent Error** – the desired solution accuracy. Smaller values produce more accurate but slower solutions; larger values produce less accurate but faster solutions.

3. Click **OK**.

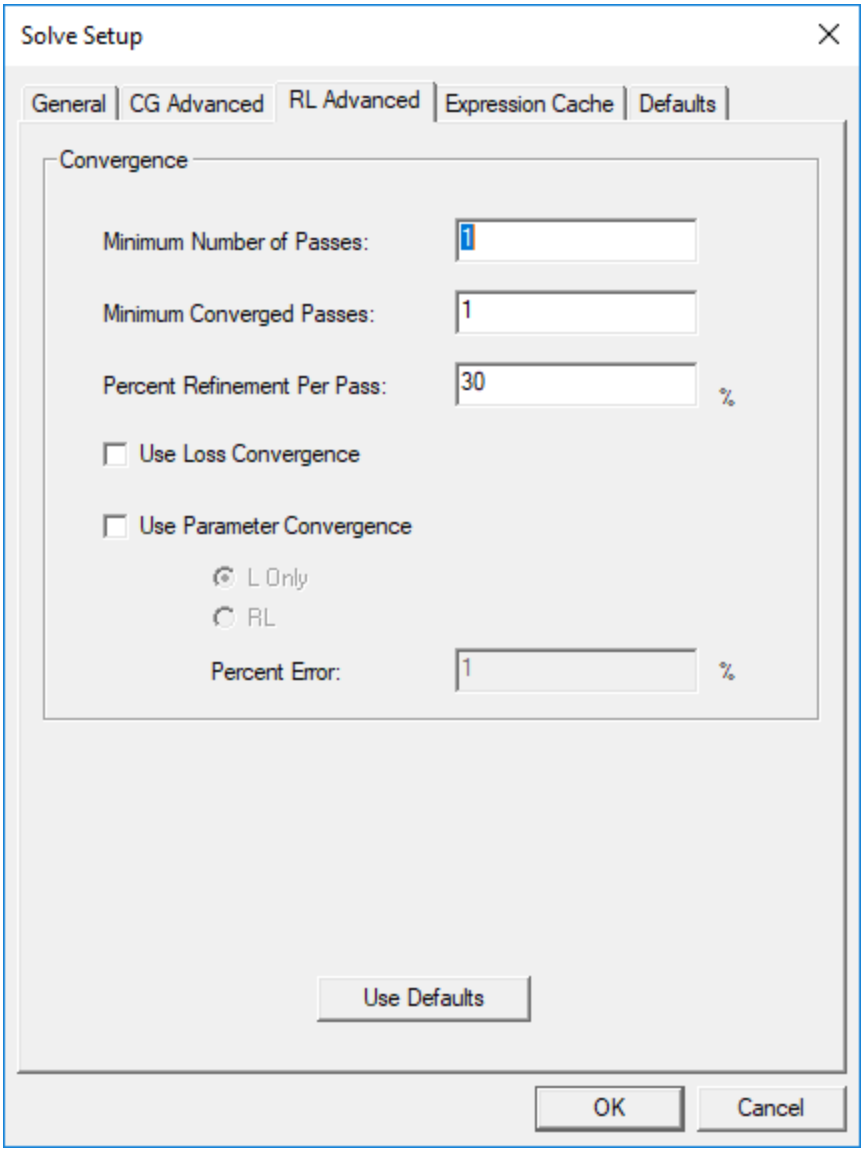
Setting Resistance and Inductance (RL) Analysis Parameters in 2D Extractor

To set up a Resistance and Inductance (RL) analysis, first add a solution setup.

Then, from the **Solve Setup** window:

1. Select the **RL Advanced** tab.

    The **RL Advanced** options display.



2. In the **Convergence** area, specify the following:

   - **Minimum Number of Passes** – the minimum number of mesh refinement cycles. 2D Extractor will not stop analysis until after this number of passes has been completed.
   - **Minimum Converged Passes** – the minimum number of passes that must meet convergence criteria before the adaptive analysis will stop.
   - **Percent 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. The default value is 30%.
   - **Use Loss Convergence** – select to use loss in the convergence criteria.

- **Use Parameter Convergence** – select to use RL matrices in calculating convergence. Parameter convergence is determined based on the values of previous pass matrix and the current matrix. The maximum delta of a matrix entry is calculated by comparing entries of previous matrix and current matrix. The final delta is given by: Final delta (%) = 100 X (maxdelta / max value).

  If the final delta is less than the target delta, then the pass is considered converged with respect to parameter convergence.

  Select **L Only** for a lossless matrix, or **RL** for a lossy matrix.

  - **Percent Error** – the desired solution accuracy. Smaller values produce more accurate but slower solutions; larger values produce less accurate but faster solutions.

3. Click **OK**.

> **Note:**
>
> Epsilon is assumed to be zero for all objects in the design since displacement current terms (omega * epsilon) are not included during the simulation.

Modifying the Solution Order in Q3D Extractor

Q3D uses the Fast Multipole Method (FMM) to accelerate the solution of large problems. The potential due to a distant collection of sources is approximated by a multipole expansion. A higher order results in higher accuracy, but increases the CPU time for the solution.

By default, the solution order is Normal, but you can increase the accuracy for problems where high accuracy is essential.

To modify the solution order:

1. Open the **Solve Setup** window one of three ways:
   - From the **Project Manager**, right-click **Analysis** and select **Add Solution Setup**.
   - From the **Project Manager**, double-click an existing solution setup.
   - Select **Q3D Extractor** > **Analysis Setup** > **Add Solution Setup**.

   The **Solve Setup** window appears, on the **General** tab.

2. Select the **DC RL** tab.

> **Note:**
>
> If the **DC RL** tab is not visible, ensure that **DC** is selected on the **General** tab.

3. Use the **Solution Order** drop-down menu to select the solution order (Normal, High, Higher, or Highest).
4. Click **OK**.

Specifying Expressions for Adaptive Convergence

You can specify additional convergence criteria through the use of expressions and output variables.

To set expressions as **Convergence** criteria:

1. In the **Project Manager**, expand **Analysis**.
2. Double-click the setup you want to modify.

   The **Solve Setup** window appears.

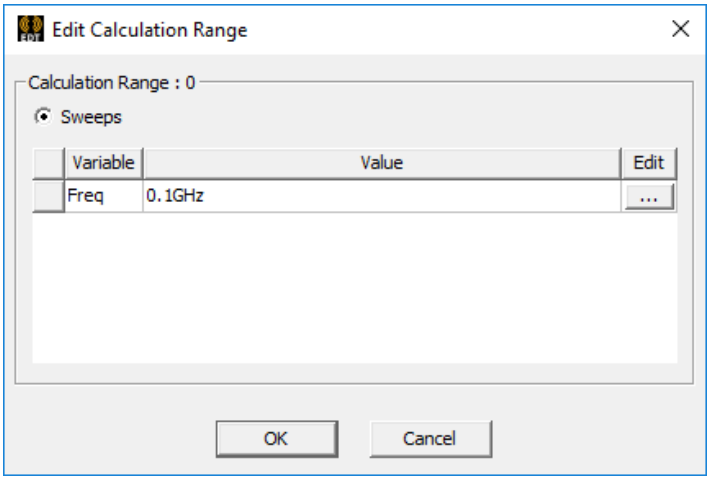
3. Select the **Expression Cache** tab.

   The **Expression Cache** tab lists any expressions you have added, in a table.

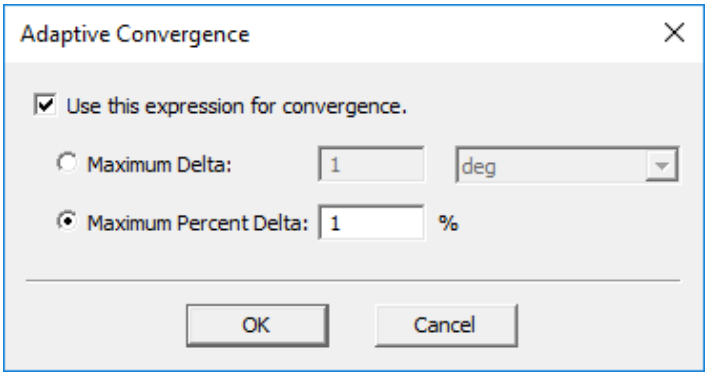
	Title	Expression	Context	Intrinsics	Convergence
	C_groundplane_groundplane_1	C(groundplane,groundplane)	Original	Freq='0.1GHz'	None
	atanh_Volume_groundplane__1	atanh(Volume(groundplane))	Original	Freq='0.1GHz'	1 %
	cos_ACR_via_Source1_via_Source1__1	cos(ACR(via:Source1,via:Source1))	Original	Freq='0.1GHz'	None

   You can edit the **Title**, **Context**, **Intrinsics**, and **Convergence** directly in the table:

   - The **Context** column allows you to select the original or a reduced matrix.
   - The **Intrinsics** column contains a button that opens an **Edit Calculation Range** dialog box. If the button reads "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 that you can select.

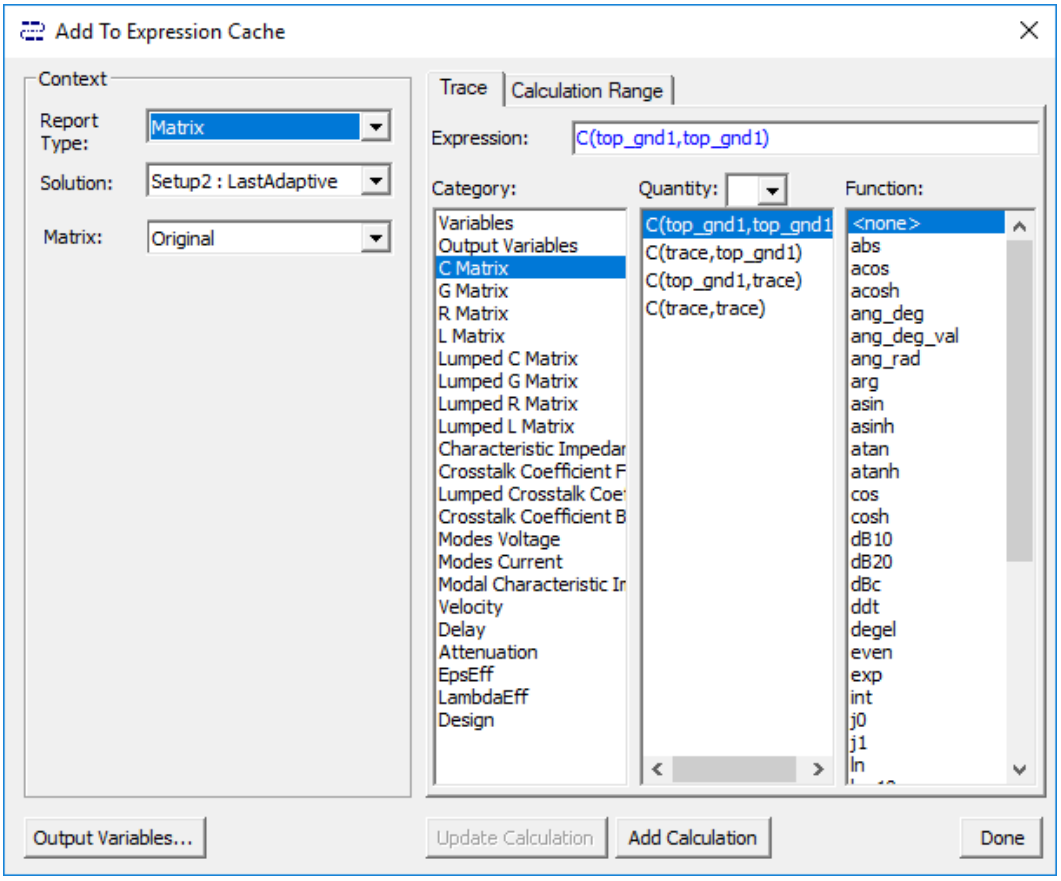


- The **Adaptive Convergence** column contains a button that opens the Adaptive Convergence dialog box. From there, you can specify whether to use the expression for convergence, and specify a Maximum Delta.



4. Click **Add**.

   The **Add to Expression Cache** window appears.



5. Use the drop-down menus to specify the **Context**.
6. On the **Trace** tab, select from the **Category**, **Quantity** and **Function** lists to create expressions.

   Your **Category** selection affects which Quantities and Functions display.

   If you have defined one or more output variables, you can see them listed as Quantities by clicking **Output Variables**.

7. On the **Calculation Range** tab, you can view and edit the values of available sweep variables.

   For example:

	Variable	Value	Edit
	Freq	0.1GHz	...

8. When you are finished adding expressions, click **Done**.
9. Click **OK** to exit the **Solve Setup** window.

Frequency Sweeps in Q3D Extractor

To generate a solution across a range of frequencies, add a frequency sweep to the solution setup. Q3D Extractor performs the sweep after the adaptive solution, if one is defined. If an adaptive solution is not requested, the sweep is the only solution generated. You can also disable a sweep and run only the adaptive solution (or a ports-only solution) without the sweep, then later reactivate the sweep.

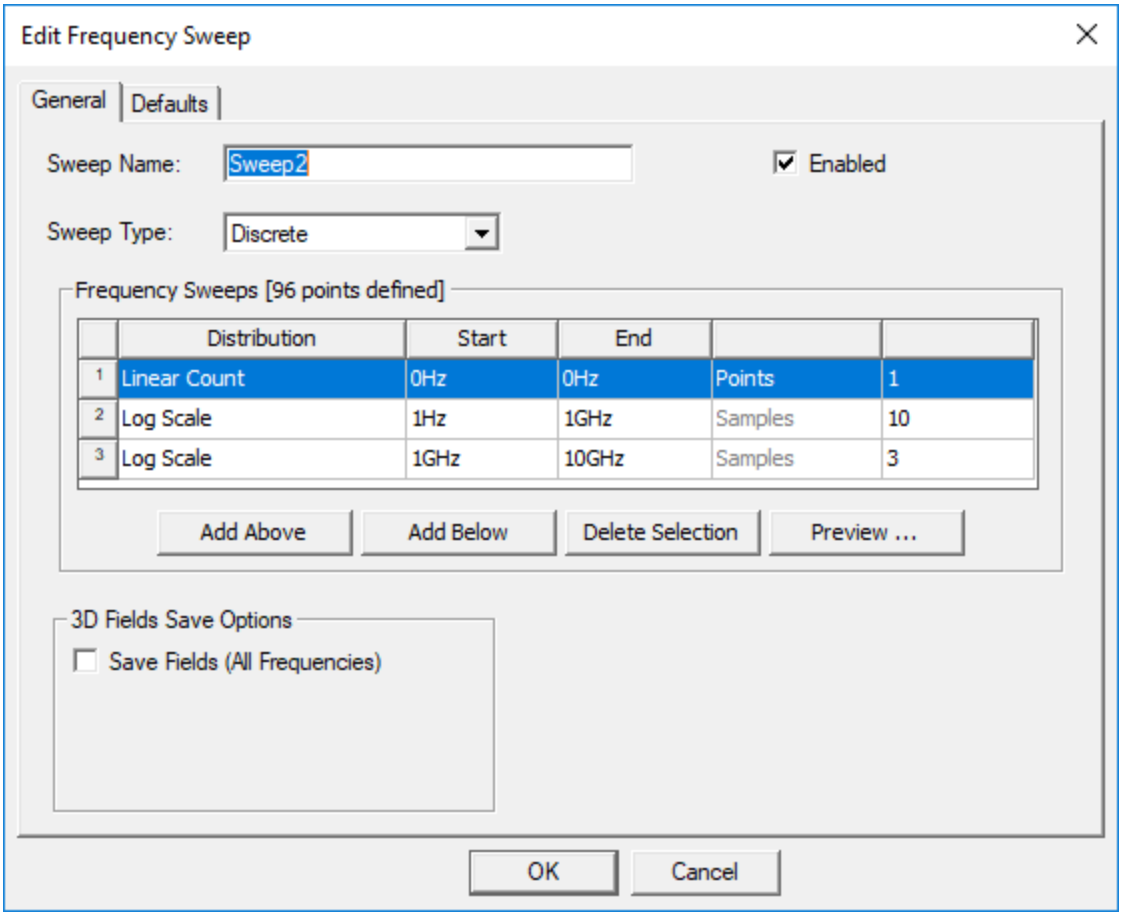
> **Important:**
>
> Q3D Extractor's blending algorithm assumes that DC inductance is always higher than AC inductance. If DC inductance values are smaller than AC inductance values, the sweep will fail and give constant curves.

Adding a Frequency Sweep

To add a frequency sweep:

1. Open the **Edit Frequency Sweep** window one of two ways:
   - Select **Q3D Extractor** > **Analysis Setup** > **Add Frequency Sweep** and select the solution setup to which the sweep applies.
   - In the **Project Tree**, right-click the setup to which the sweep applies and select **Add Frequency Sweep**.

   The **Edit Frequency Sweep** window appears.



> **Note:**
>
> The available tabs and options change with the selected sweep type.

2. Specify the sweep parameters:

   - **Sweep Name** – enter a name for the sweep.
   - **Sweep Type** – select discrete or interpolating:

     - **Discrete Sweep** – Generates field solutions at specific frequency points in a frequency range. Best when only a few frequency points are necessary to accurately represent the results in a frequency range.
     - **Interpolating Sweep** – Estimates a solution for an entire frequency range. Best when the frequency range is wide and the frequency response is smooth. When you select Interpolating sweep, the **Interpolation** tab appears, where you can specify the maximum number of solutions, minimum number of solutions, and error tolerance. See Step 4 below.

> **Note:**
>
> The information specified for an Interpolating sweep dictates the amount of information that can be viewed on a post-processing plot.

- **Enabled** – select to enable the sweep or clear the check box to disable it.
- **Frequency Sweeps** – 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:

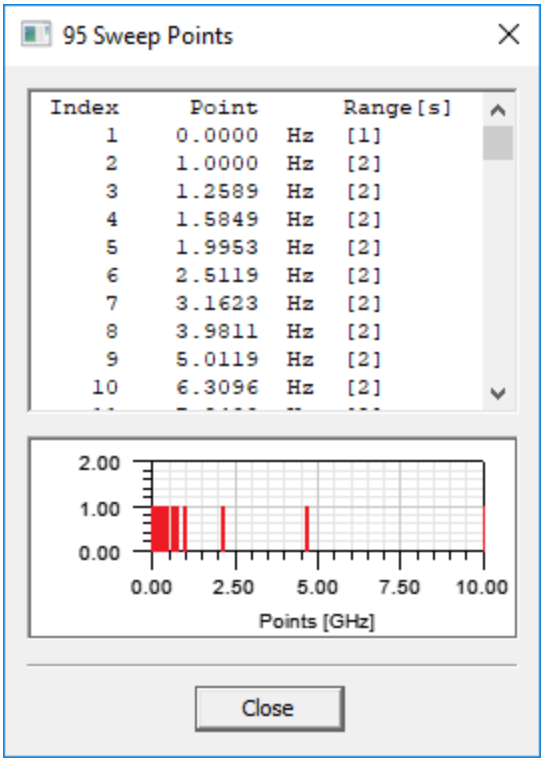
  - **Linear Step** – A linear range of frequency points for which you specify a constant step size. Q3D Extractor will solve the frequency point at each step in the specified frequency range, including the start and stop frequencies. For example, specifying 10 for the start frequency, 20 for the stop frequency, and 2.5 for the step size instructs Q3D Extractor to compute a solution for frequencies of 10, 12.5, 15, 17.5, and 20.
  - **Linear Count** – A linear range of frequency points for which you specify the number (count) of points within the frequency range. Q3D Extractor will divide the frequency range into the count you specify and solve each frequency point in the count.
  - **Log Scale** – A range of points for which you specify the start, end, and number of samples. Q3D Extractor assigns the sampled points using intervals based on a logarithmic scale.
  - **Single Point** – Individual frequency points. You can insert specific frequency points that you want to solve in the frequency range after you have added uniform frequency points to solve.*For Discrete sweeps only*.

  You can add multiple sweep types to the same setup.

  Specify the **Start** and **End** frequency for each sweep, as well as any additional options for that sweep type.

- **3D Fields Save Options** – for discrete sweeps only, you can select **Save Fields (all frequencies)** to save the calculated 3D field solutions associated with the chosen frequencies.

3. Click **Preview** to view a preview of the sweep, as currently defined:



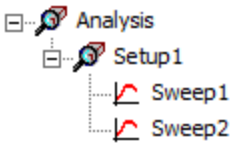
4. If you are running an **Interpolating** sweep, select the **Interpolation** tab to specify additional settings:

   - **Max Solutions** – the maximum number of solutions that will be solved for the frequency range. The default is 50.
   - **Error Tolerance** – the maximum relative difference allowed between two successive interpolation solutions. The default is 0.01 percent, which is usually satisfactory.
   - **Minimum Solutions** – the minimum number of converged solutions that will be solved for the frequency range. For example, if this value is three, then once the sweep reaches convergence it simulates at two extra frequencies. This resembles the minimum number of converged adaptive passes in a regular simulation. Setting a minimum number of solutions can eliminate non-physical S-parameter spikes. The default value is 0.
   - **Minimum Number of Sub Ranges** – the sub range number acts as an initial condition on the sweep to force initial even breakup of the null range into subranges. The end points and middle of each subrange will be solved. This controls the points at which the interpolating sweep is broken up and prevents redundant effort caused by neighboring interpolating sweeps solving the same

point. For example, the 1GHz to 4GHz and the 4GHz to 9 GHz sweeps do not both solve the 4 GHz datapoint. The default value is 1.

5. If desired, select the **Defaults** tab to save your settings as the default or change your settings to the current default settings.
6. Click **OK** to add the sweep.

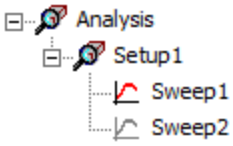
   The sweep appears in the Project Tree:



Enabling and Disabling a Frequency Sweep

To disable a Frequency Sweep without deleting it, right-click the sweep in the Project Tree and select **Disable Sweep**.

Disabled sweeps appear grayed out in the Project Tree (see Sweep2 in this example):



To enable a disabled Frequency Sweep, right-click the sweep and select **Enable Sweep**.

Deleting a Frequency Sweep

To delete a sweep, right-click it in the Project Tree and select **Delete**.

Duplicating a Frequency Sweep

You can duplicate an existing frequency sweep by right-clicking it and selecting **Copy** (or by using **Edit** > **Copy**), then selecting the solution setup and pasting the sweep. By default, the copy is named Sweep*n*, where *n* increments with each new sweep. You can then edit each copy to make desired changes.

Frequency Sweeps in 2D Extractor

To generate a solution across a range of frequencies, add a frequency sweep to the solution setup. 2D Extractor performs the sweep after the adaptive solution, if one is defined. If an

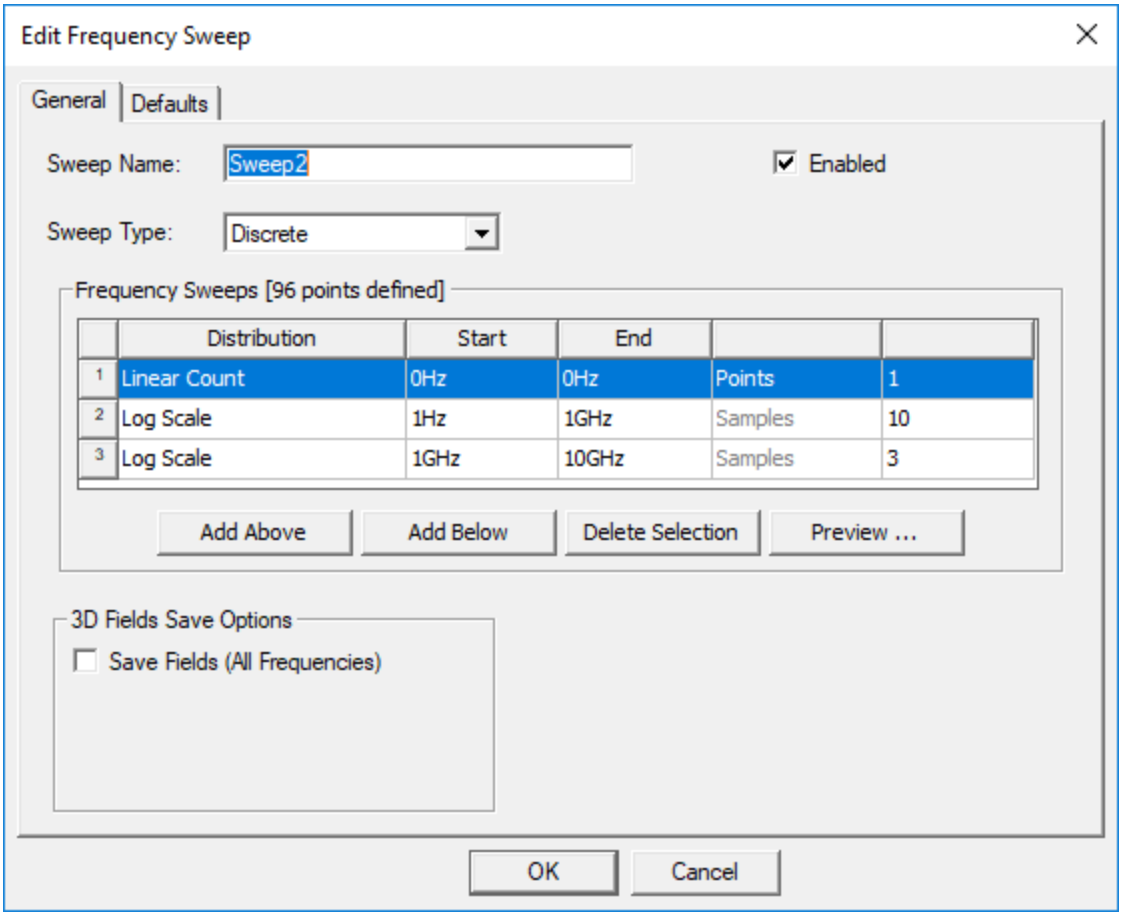
adaptive solution is not defined, the sweep is the only solution generated. You can also disable a sweep, so that only the adaptive solution runs, then later reactivate the sweep definition.

Adding a Frequency Sweep

To add a frequency sweep:

1. Open the **Edit Frequency Sweep** window one of two ways:
   - Select **2D Extractor** > **Analysis Setup** > **Add Frequency Sweep** and select the solution setup to which the sweep applies.
   - In the **Project Tree**, right-click the setup to which the sweep applies and select **Add Frequency Sweep**.

   The **Edit Frequency Sweep** window appears.



> **Note:**
>
> The available tabs and options change with the selected sweep type.

2. Specify the sweep parameters:

   - **Sweep Name** – enter a name for the sweep.
   - **Sweep Type** – select discrete or interpolating:

     - **Discrete Sweep** – Generates field solutions at specific frequency points in a frequency range. Best when only a few frequency points are necessary to accurately represent the results in a frequency range.
     - **Interpolating Sweep** – Estimates a solution for an entire frequency range. Best when the frequency range is wide and the frequency response is smooth. When you select Interpolating sweep, the **Interpolation** tab appears, where you can specify the maximum number of solutions, minimum number of solutions, and error tolerance. See Step 4 below.

       > **Note:**
       >
       > The information specified for an Interpolating sweep dictates the amount of information that can be viewed on a post-processing plot.

   - **Enabled** – select to enable the sweep or clear the check box to disable it.
   - **Frequency Sweeps** – 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:

     - **Linear Step** – A linear range of frequency points for which you specify a constant step size. 2D Extractor will solve the frequency point at each step in the specified frequency range, including the start and stop frequencies. For example, specifying 10 for the start frequency, 20 for the stop frequency, and 2.5 for the step size instructs 2D Extractor to compute a solution for frequencies of 10, 12.5, 15, 17.5, and 20.
     - **Linear Count** – A linear range of frequency points for which you specify the number (count) of points within the frequency range. 2D Extractor will divide the frequency range into the count you specify and solve each frequency point in the count.
     - **Log Scale** – A range of points for which you specify the start, end, and number of samples. 2D Extractor assigns the sampled points using intervals based on a logarithmic scale.

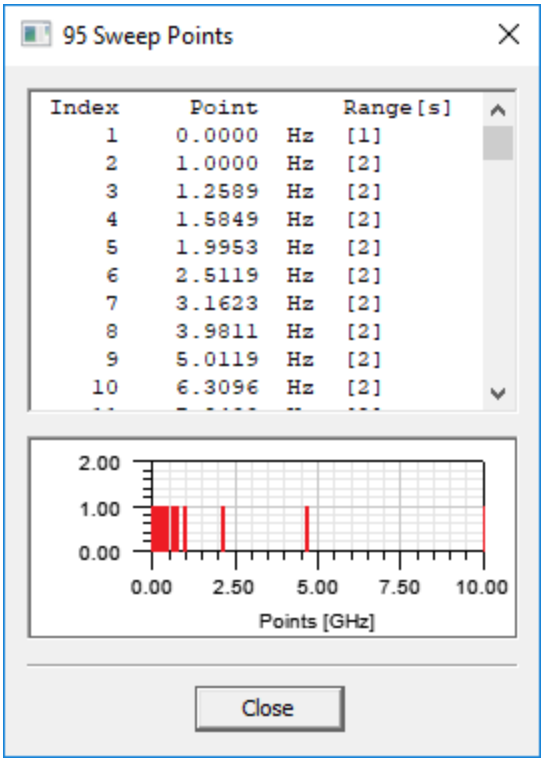
- **Single Point** – Individual frequency points. You can insert specific frequency points that you want to solve in the frequency range after you have added uniform frequency points to solve.*For Discrete sweeps only*.

You can add multiple sweep types to the same setup.

Specify the **Start** and **End** frequency for each sweep, as well as any additional options for that sweep type.

- **3D Fields Save Options** – for discrete sweeps only, you can select **Save Fields (all frequencies)** to save the calculated 3D field solutions associated with the chosen frequencies.

3. Click **Preview** to view a preview of the sweep, as currently defined:



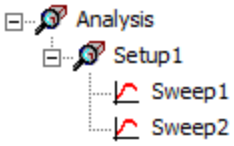
4. If you are running an **Interpolating** sweep, select the **Interpolation** tab to specify additional settings:

- **Max Solutions** – the maximum number of solutions that will be solved for the frequency range. The default is 50.
- **Error Tolerance** – the maximum relative difference allowed between two successive interpolation solutions. The default is 0.01 percent, which is usually satisfactory.

- **Minimum Solutions** – the minimum number of converged solutions that will be solved for the frequency range. For example, if this value is three, then once the sweep reaches convergence it simulates at two extra frequencies. This resembles the minimum number of converged adaptive passes in a regular simulation. Setting a minimum number of solutions can eliminate non-physical S-parameter spikes. The default value is 0.
- **Minimum Number of Sub Ranges** – the sub range number acts as an initial condition on the sweep to force initial even breakup of the null range into subranges. The end points and middle of each subrange will be solved. This controls the points at which the interpolating sweep is broken up and prevents redundant effort caused by neighboring interpolating sweeps solving the same point. For example, the 1GHz to 4GHz and the 4GHz to 9 GHz sweeps do not both solve the 4 GHz datapoint. The default value is 1.

5. If desired, select the **Defaults** tab to save your settings as the default or change your settings to the current default settings.
6. Click **OK** to add the sweep.

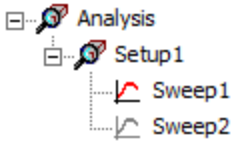
   The sweep appears in the Project Tree:



Enabling and Disabling a Frequency Sweep

To disable a Frequency Sweep without deleting it, right-click the sweep in the Project Tree and select **Disable Sweep**.

Disabled sweeps appear grayed out in the Project Tree (see Sweep2 in this example):



To enable a disabled Frequency Sweep, right-click the sweep and select **Enable Sweep**.

Deleting a Frequency Sweep

To delete a sweep, right-click it in the Project Tree and select **Delete**.

Duplicating a Frequency Sweep

You can duplicate an existing frequency sweep by right-clicking it and selecting **Copy** (or by using **Edit** > **Copy**), then selecting the solution setup and pasting the sweep. By default, the copy is named Sweep*n*, where *n* increments with each new sweep. You can then edit each copy to make desired changes.

14 - Reducing Matrices in Q3D Extractor and 2D Extractor

This section contains information about the following:

Using Reduce Matrix Operations in Q3D Extractor

When you generate a solution, a row and column are generated in the capacitance matrix for each net in the model, while a row and column are generated in the inductance and resistance matrices for each source terminal in the model.

Reducing a model 's capacitance, inductance, and resistance matrices enables you to quickly and easily vary the excitations and connections of the conductors in your model without having to modify the conductor and source assignments and generate a new solution. In addition, it lets you model designs that cannot otherwise be simulated in Q3D Extractor — such as conductors connected in parallel or series. The reduced matrices can then be used to generate SPICE equivalent subcircuits for the model via the **Make SPICE File** command, and can also be input to other circuit simulation packages.

Reduce Matrix operations allow you to modify conductor definitions and generate new matrices containing fewer elements. Operations for Q3D Extractor include:

- Move Sink
- Add Sink
- Join in Series
- Join in Parallel
- Join Selected Terminals
- Float Net
- Ground Net
- Float Terminal
- Float at Infinity
- Return Path
- Change Frequency

Reduced matrices allow you to:

- Simulate series and parallel connections between nets or terminals.
- Model floating nets and floating terminals in a structure.
- Examine the effects of grounding different nets in a structure.

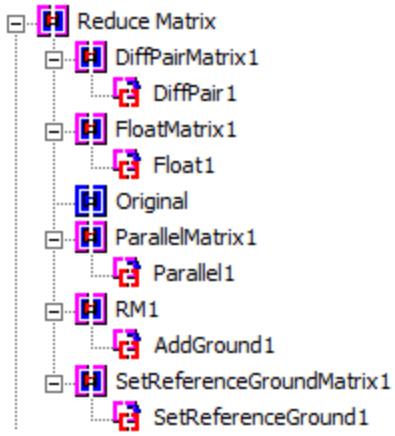
- For capacitance matrices, model the case where the charge at infinity is zero (that is, infinity is not held at zero potential).
- For AC problems, find the resistance at different frequencies.
- For inductance and resistance matrices:
- Change the location of current sink points within conductors.
- Add current sink points to multi-terminal conductors.
- Specify that current returns through a conductor in the model.

Based on these modifications, Q3D Extractor generates new matrices without computing new field solutions for any selected parameters that already have field solutions. This saves computing time and resources and allows you to easily optimize ground assignments when you generate a SPICE equivalent circuit.

> **Important:**
>
> Before you solve, remember that you must assign *all* signal-carrying conductors to nets.

You can select any object in the Modeler window and right-click, then select **Assign Reduce Matrix**. If the selected objects are valid terminals for the reduce matrix operation, they are considered for reduction; otherwise, they are ignored.

After reduction, reduced matrices and their **Reduce Matrix** operations appear in the **Project Manager**:



You can right-click the original or a reduced matrix in the project tree to see more options.

You can also right-click a Reduce Matrix operation and click **Select Assignment** to select the objects to which it applies.

Duplicating a Reduced Matrix

To duplicate a **Reduce Matrix** operation, right-click it in the **Project Manager** and select **Duplicate**.

Deleting a Reduced Matrix or Reduce Matrix Operation

To delete a reduced matrix or a **Reduce Matrix** operation, right-click it in the **Project Manager** and select **Delete**.

> **Important:**
>
> You cannot delete the original matrix!

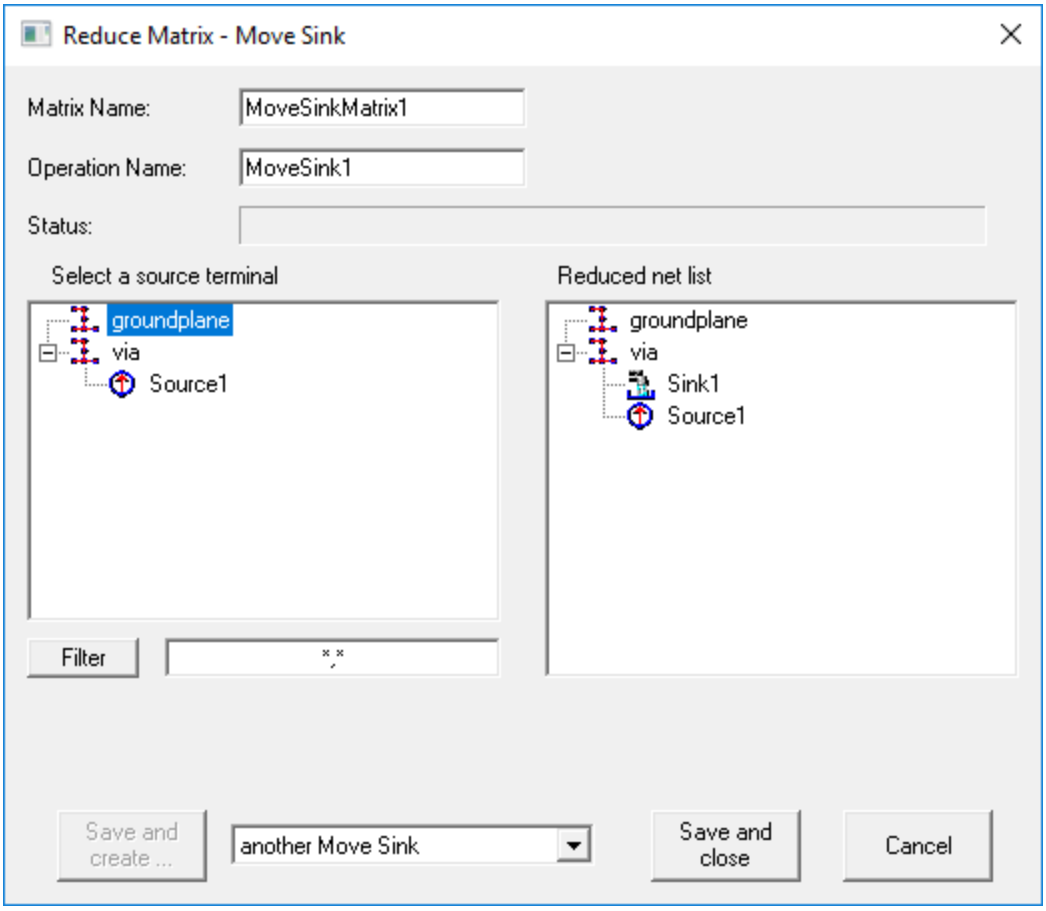
Using Reduce Matrix > Move Sink

The **Move Sink** reduction switches the locations of a source terminal and a sink terminal. It lets you vary the placement of sink terminals in a conductor without having to change the terminal assignments and generate a new solution. Instead, the existing inductance and resistance matrices are modified to take the swapped source and sink terminals into account.

To switch the locations of a current source and sink point:

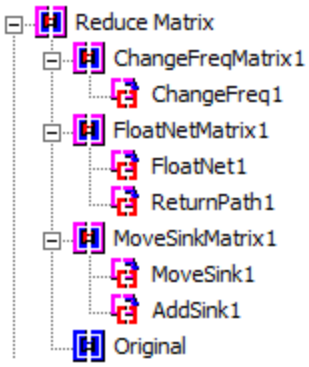
1. Click **Q3D Extractor** > **Reduce Matrix** > **Move Sink**.

   The **Reduce Matrix - Move Sink** window appears, with a list of source terminals in the left pane.



2. Select a source to move. As soon as you click the source terminal, it is removed from the terminal list and added to the **Reduced Net List**.
3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

   Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Add Sink

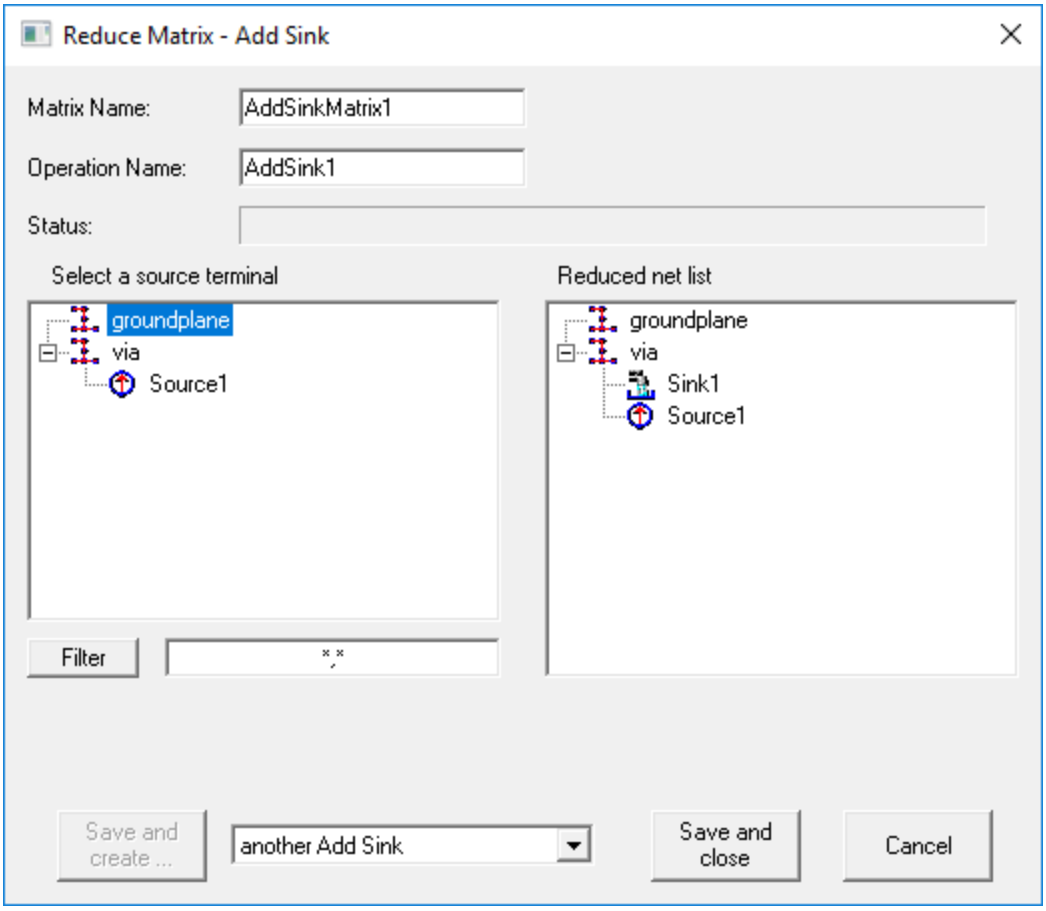
The **Add Sink** reduction changes one or more source terminals in a net into sink terminals. When you add a sink point, you assume that the new sink point has been connected via a short circuit to the original sink point.

> **Important:**
>
> If you convert all the source terminals in a net to sink terminals, you have short-circuited the net, but it still has some effect on the inductance matrices.

To convert a source terminal into a sink terminal:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Add Sink**.

   The **Reduce Matrix - Add Sink** window appears, with a list of source terminals in the left pane.

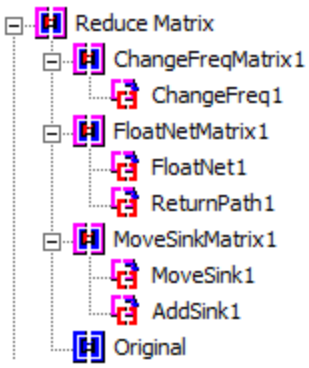


2. Select a source to convert into a sink. As soon as you click the source terminal, it is removed from the terminal list and added to the **Reduced Net List**.

> **Tip:**
>
> Press **Ctrl**+**A** to select all sources.

3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Join in Series

The **Join in Series** reduction lets you connect the sink terminals of one or more conductors in series with a source terminal on a different conductor. The connected conductors are then treated as if they were a single net. The rows and columns that represent their contributions to the matrix are replaced with a single row and column representing the contribution of this new object.

For capacitance and conductance CG) matrices, conductors in series are considered to be connected by ideal wires. Because of this, they are assumed to be at the same voltage — therefore the series connection produces the same capacitance result as a parallel connection.

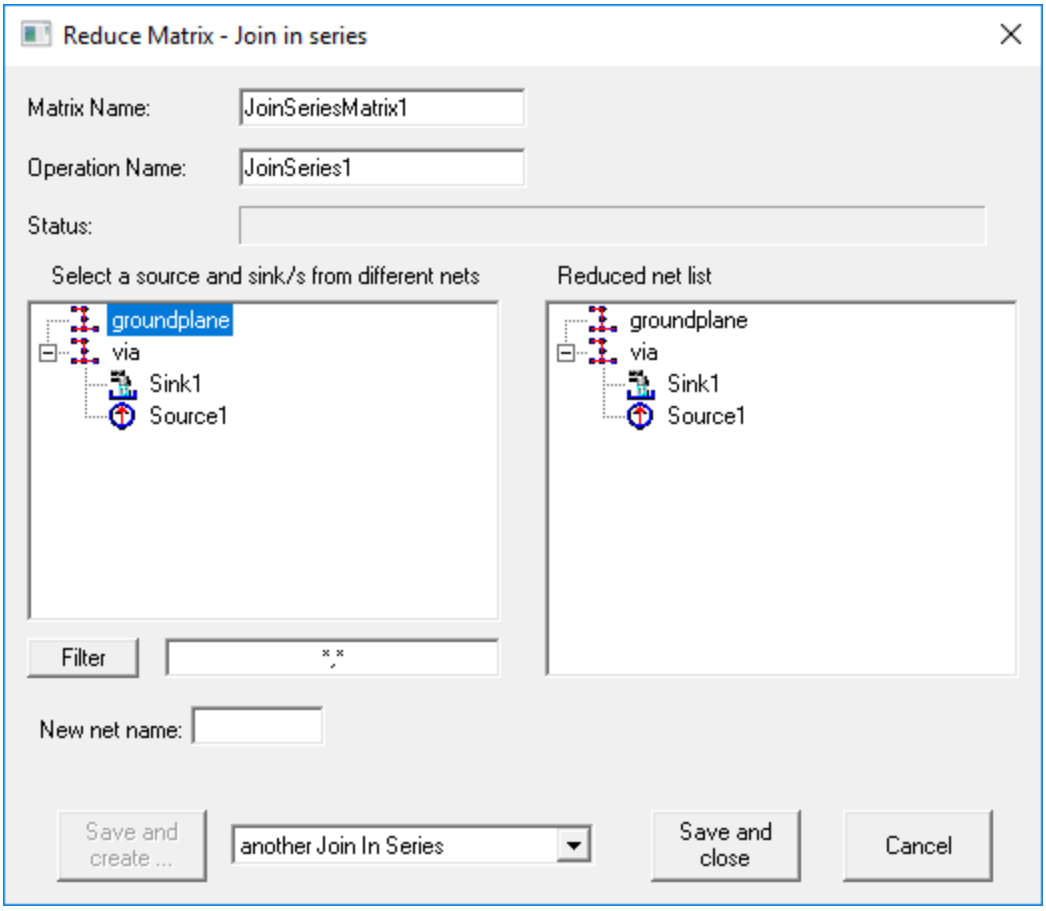
In inductance and resistance (IR) matrices, the currents leaving the selected sink terminals are considered to be forced into the selected source terminal. Because the connections between the source and sink terminals are ideal, the exact order of connection is not important.

This matrix reduction operation is used to model the inductance and resistance of a closed loop in Q3D Extractor. The software directly solves for *partial* inductance and resistance matrices, which only include the effects of currents in the section of the loop that is being modeled. Matrix reduction allows you to calculate loop inductances and resistances.

To set up series connections between the conductors or terminals in a structure:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Join in Series**.

   The **Reduce Matrix - Join in Series** window appears, with a list of sources and sinks in the left pane.

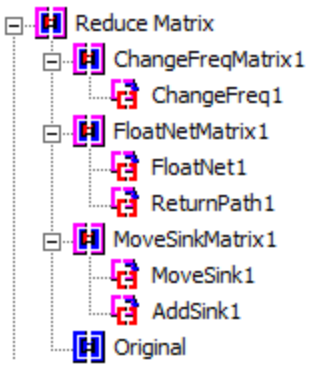


2. Select the source terminal and sink terminal(s) you want to join. As soon as you click a source or sink, it is removed from the terminal list and added to the **Reduced Net List**.

> **Tip:**
>
> Press **Ctrl**+**A** to select all sources/sinks.

3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Join in Parallel

The **Join in Parallel** reduction lets you connect two or more source terminals, which are then treated as if they were a single object. The rows and columns that represent their contributions to the matrix are replaced with a single row and column representing the contribution of this new object.

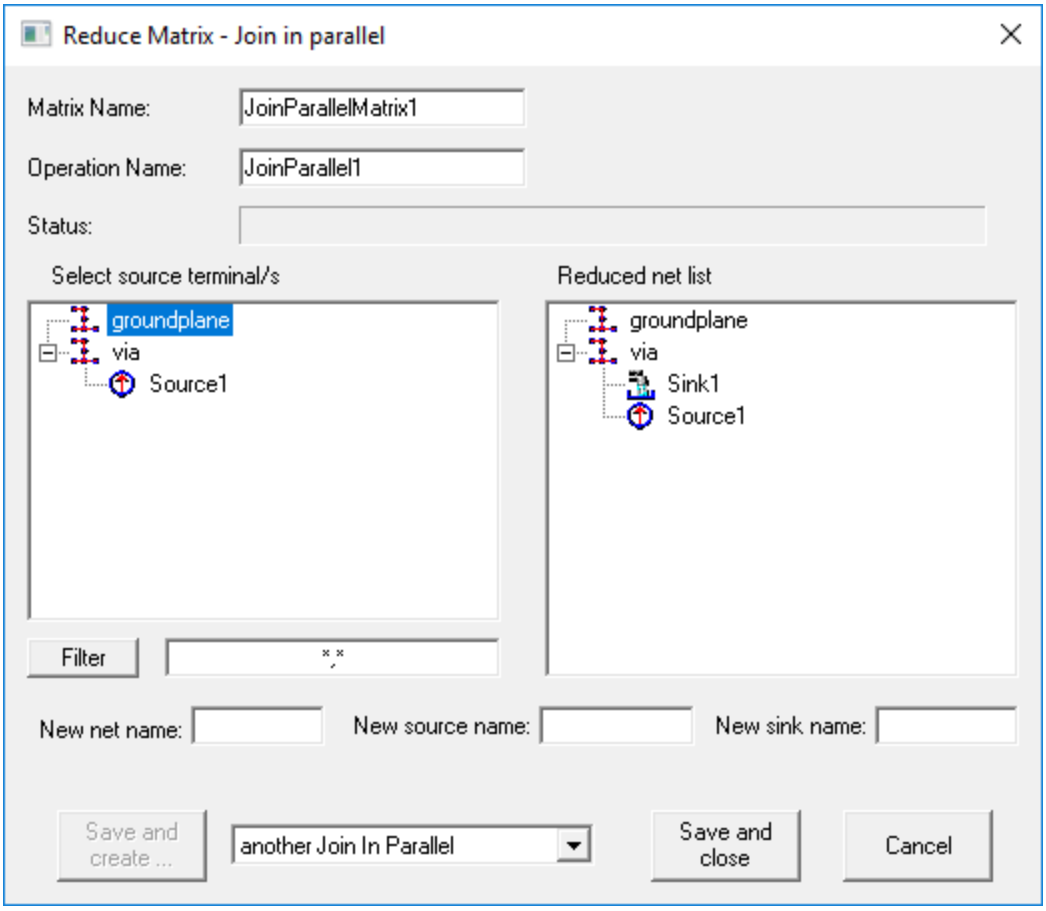
In capacitance (CG) matrices, all parallel conductors are considered to be at the same voltage. As a result, there is no mutual or coupling capacitance between them. (Alternatively, you can think of the coupling capacitance as being shorted out since both of its terminals are at the same potential.)

In inductance and resistance (IR) matrices, source terminals in parallel conductors are considered to be connected, and therefore are dependent. The corresponding sink terminals are also considered to be connected. Parallel reduction is based on the impedance of the conductors being reduced.

To set up parallel connections between the conductors in a structure:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Join in Parallel**.

   The **Reduce Matrix - Join in Parallel** window appears, with a list of sources in the left pane.



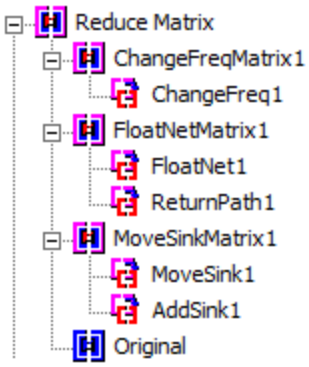
2. Select the terminals you want to join. As soon as you click a source, it is removed from the terminal list and added to the **Reduced Net List**.

> **Tip:**
>
> Press **Ctrl**+**A** to select all terminals.

3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Enter a **New net name**, **New source name**, and **New sink name**. These names should be unique.
5. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.

- **Save and Close** to save the current operation and close the window.
- **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



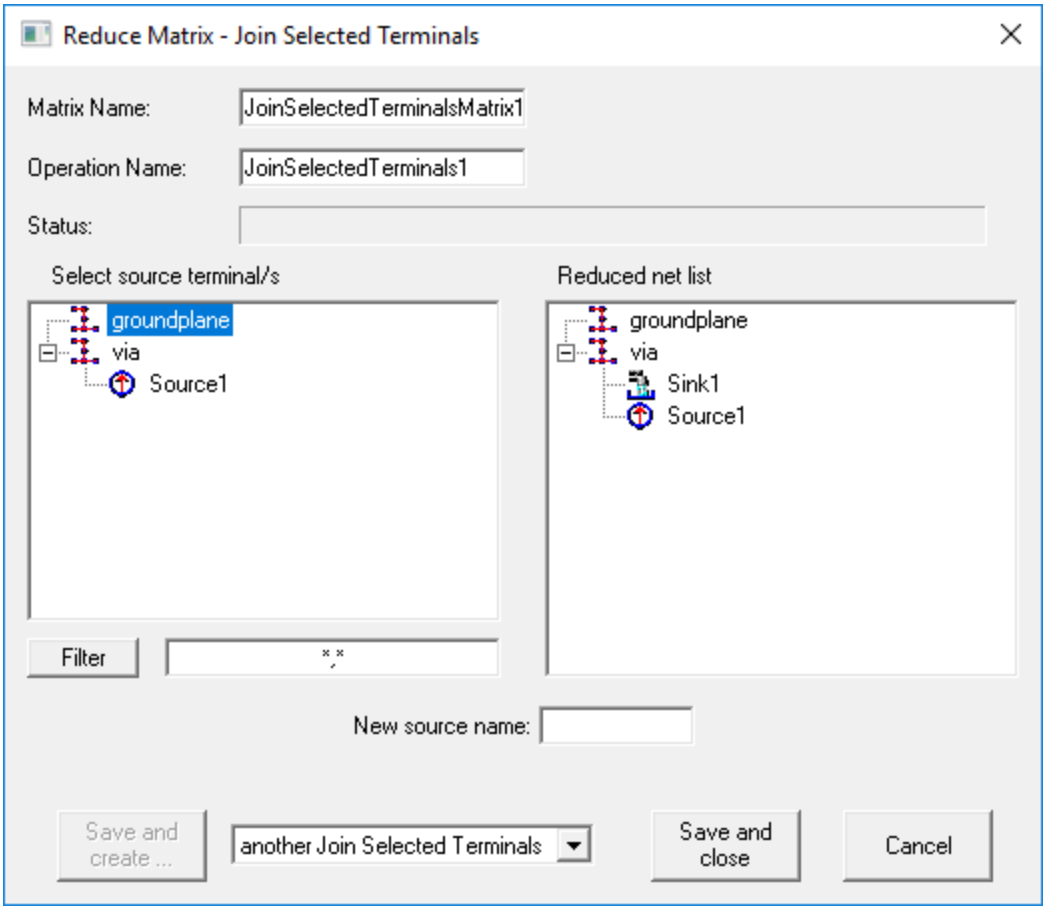
Using Reduce Matrix > Join Selected Terminals

The **Join Selected Terminals** reduction lets you connect two or more source terminals which are then treated as if they were a single object. This operation does not unite sources across nets.

To join terminals:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Join Selected Terminals**.

   The **Reduce Matrix - Join Selected Terminals** window appears, with a list of sources in the left pane.

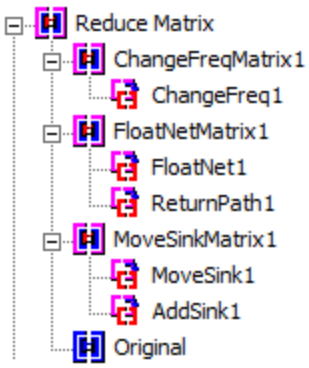


2. Select the terminals you want to join. As soon as you click a source, it is removed from the terminal list and added to the **Reduced Net List**.

> **Tip:**
>
> Press **Ctrl**+**A** to select all terminals.

3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Enter a **New source name**. This name should be unique.
5. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Float Net

The **Float Net** reduction lets you model the presence of floating conductors in a structure.

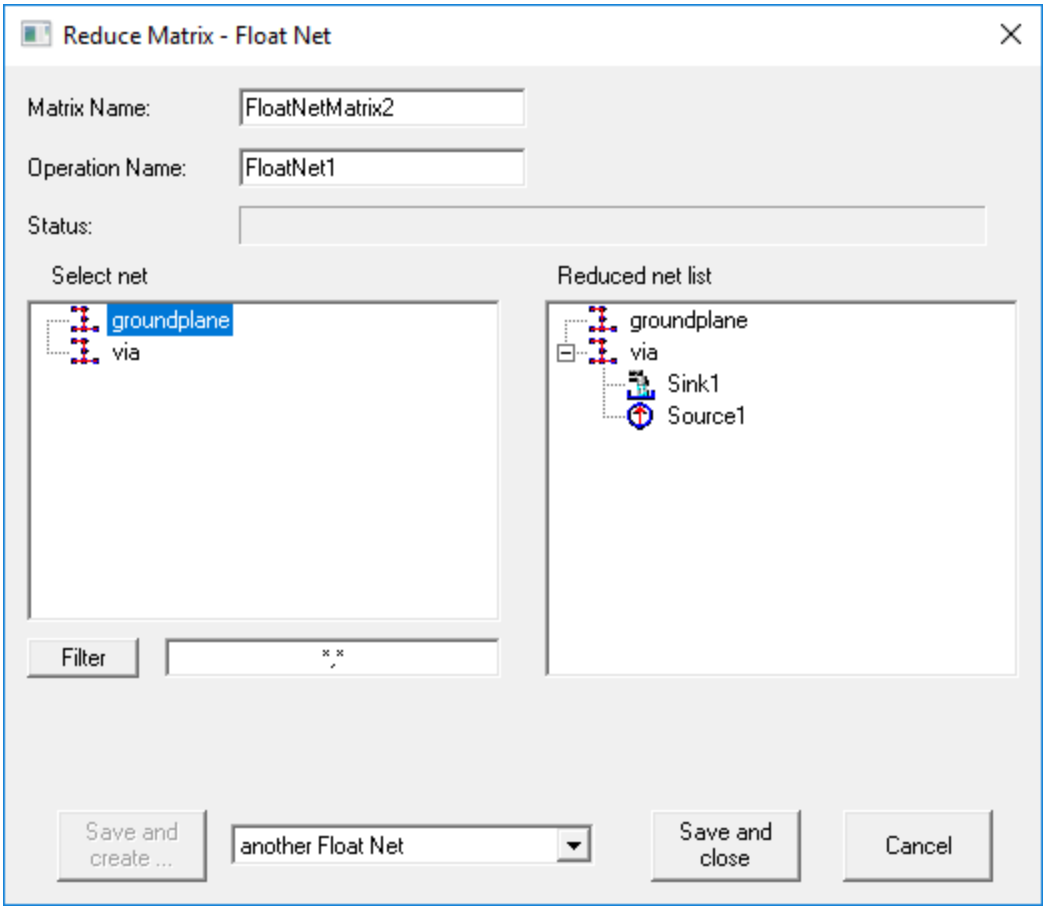
In inductance and resistance matrices, the total charge in a floating net is constrained to zero amps. This simulates an open circuit.

In a capacitance matrix, a floating net is considered to have a total zero charge. The potential is the same everywhere on the net because it is considered to be a perfect conductor. The potential of the floating net lies somewhere between the potential of the surrounding sources and ground. The exact value depends on the source excitations and on the model 's geometry.

To define which nets in a structure are floating:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Float Net**.

   The **Reduce Matrix - Float Net** window appears, with a list of nets in the left pane.

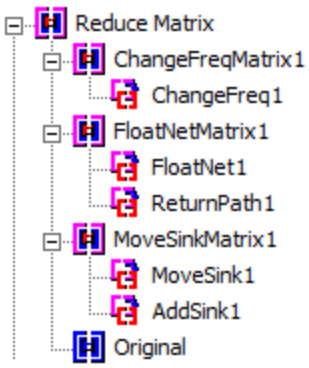


2. Select the nets you want to float. As soon as you click a net, it is removed from the terminal list and added to the **Reduced Net List**.

> **Important:**
>
> You cannot select all nets.

3. If you have a long list of nets, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Ground Net

The **Ground Net** reduction grounds the specified nets in your model. If you have objects not included in the matrix, they are already treated as grounded objects.

In capacitance (CG) matrices, grounded conductors are considered to have a potential of zero volts. The ground reference at infinity is also considered to have a potential of zero volts.

Grounded conductors in capacitance computations can be modeled in three ways:

- Through matrix reduction.
- By explicitly setting a conductor's voltage to zero.
- By assigning a conducting material to an object, and then not declaring it to be a conductor (that is, by defining it to be a passive conductor). These conductors are automatically set to zero volts during capacitance computations.

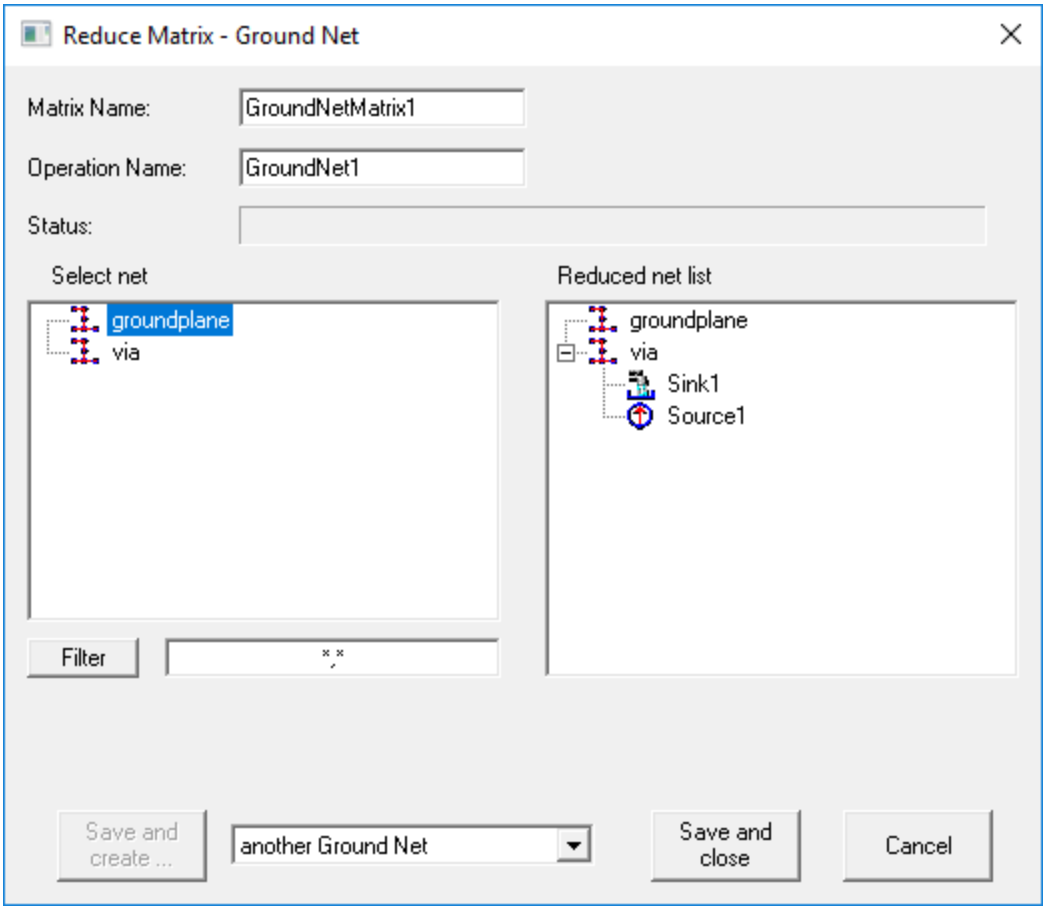
For inductance and resistance (IR) matrices, when a conductor is grounded, its source terminals are connected to its sink terminal. Current can flow in the newly created loops due to fields induced by the surrounding conductors. The fields produced by this loop affect voltage drops observed in surrounding conductors. Thus, the inductance matrix of the remaining conductors is altered by this operation.

Matrix reduction is the only way to model grounded conductors in inductance and resistance computations.

To define which nets in a structure are grounded:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Ground Net**.

   The **Reduce Matrix - Ground Net** window appears, with a list of nets in the left pane.

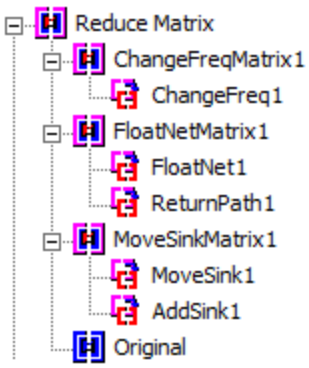


2. Select the nets you want to ground. As soon as you click a net, it is removed from the terminal list and added to the **Reduced Net List**.

> **Important:**
>
> You cannot select all nets.

3. If you have a long list of nets, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



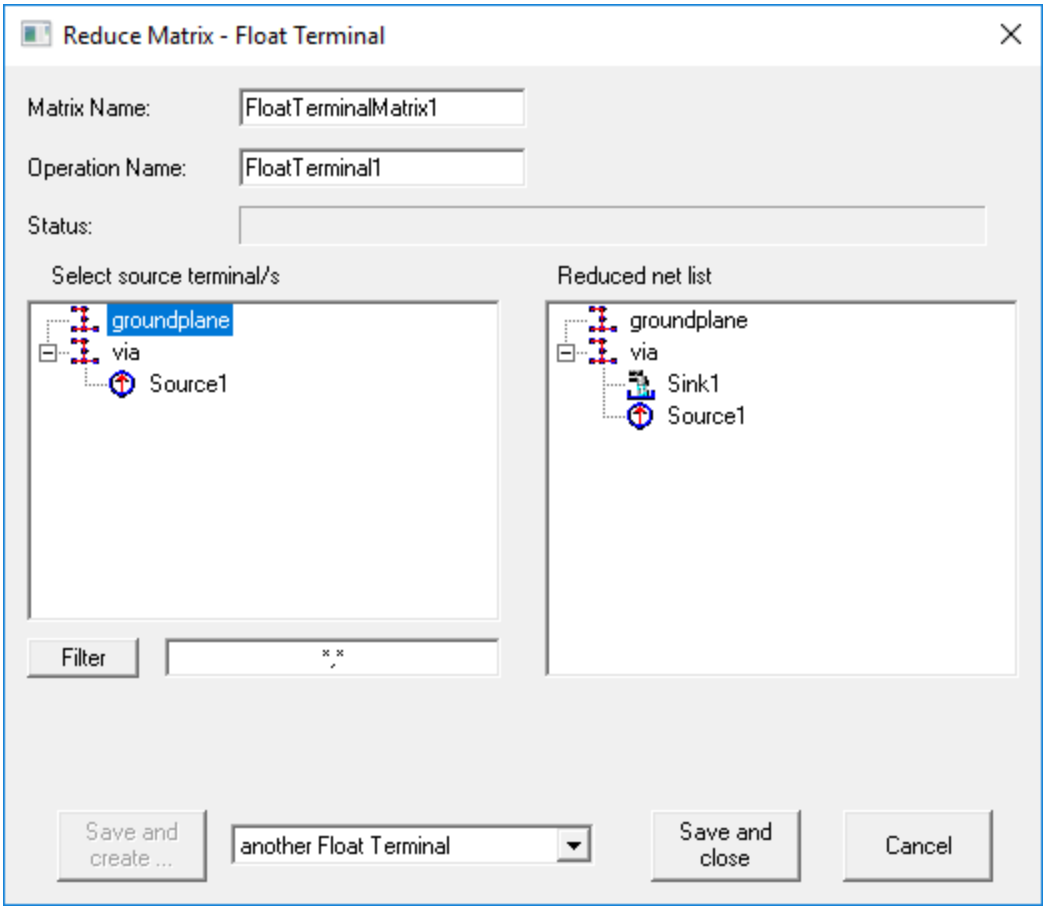
Using Reduce Matrix > Float Terminal

The **Float Terminal** operation allows you to "turn off" the current in a particular source terminal. The effect on the inductance and resistance (IR) matrices is to eliminate the row and column associated with the selected source terminal. The other entries of the matrix are unaffected.

To float a source terminal:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Float Terminal**.

   The **Reduce Matrix - Float Terminal** window appears, with a list of terminals in the left pane.

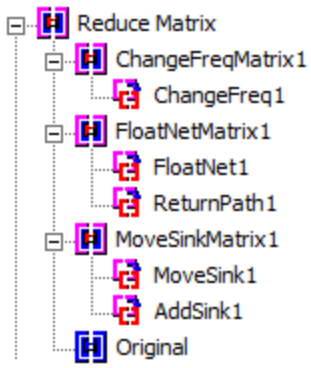


2. Select the terminal(s) you want to float. As soon as you click a terminal, it is removed from the terminal list and added to the **Reduced Net List**.

> **Important:**
>
> You cannot select all source terminals.

3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Float at Infinity

The **Float at Infinity** reduction lets you model the case where the charge at infinity is zero, causing the potential at infinity to float, enabling you to use a local ground as a reference.

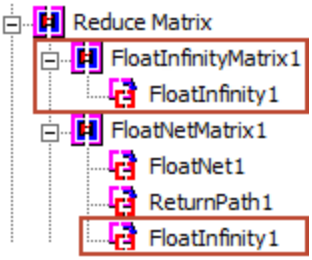
When computing capacitance, Q3D Extractor uses the voltage at infinity as a reference, and assumes that this voltage is equal to zero. This implies that the charge inside the problem region is balanced by the charge at infinity—that is, that there is charge induced at infinity.

However, in some cases, you may want to use a grounded net in the model as a reference. To do this, Q3D Extractor changes the capacitance values to reflect a local reference—modeling the case where no field lines leak from the solution region. The potential at infinity is not fixed at zero, and depends on the voltages inside the model.

To define a capacitance matrix with a floating voltage at infinity:

1. In the **Project Manager**, select either the original matrix or a previously reduced matrix.
2. Click **Q3D Extractor** > **Reduce Matrix** > **Float at Infinity**.

   The **Project Manager** displays the floated matrix either as a top-level matrix or under a previously reduced matrix:



Using Reduce Matrix > Return Path

The **Return Path** reduction lets you select a net to serve as a return path for current, enabling you to approximate the loop inductance of a closed conduction path.

If you identify a net as a current return path, the total current exiting from the sink terminals on the remaining conductors returns inside the selected conductor. This total current is forced into the sink terminal of the return path conductor and re-emerges from the source terminal. It is as if the return conductor was connected via an ideal wire to the source conductors, creating a closed current loop—only part of which is explicitly modeled. Current flow in the conductor specified as the return path is automatically reversed.

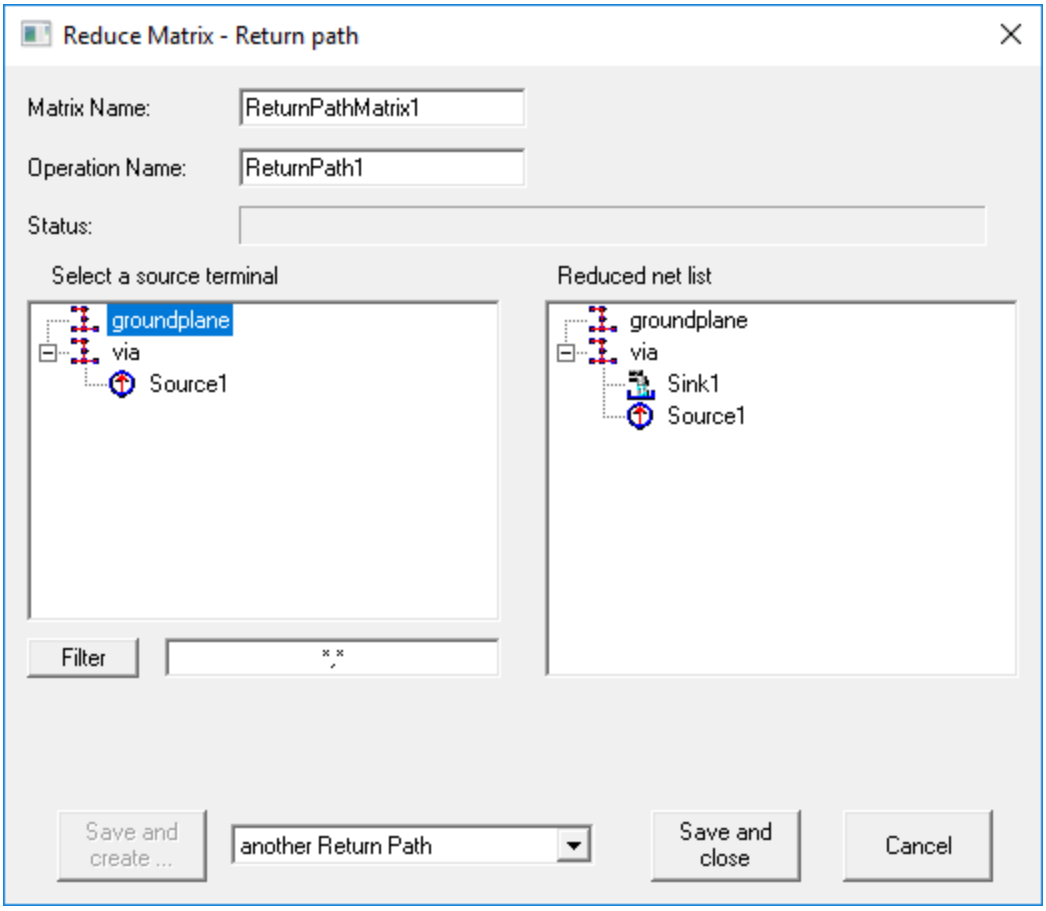
Specifying a return path does not simulate the effect of the full current loop. The effect of returning current on the partial inductance and resistance matrices is confined to the region being modeled and does not include the rest of the loop.

This matrix reduction operation is used to model the inductance and resistance of a closed loop in Q3D Extractor. The software directly solves for *partial* inductance and resistance matrices, which only include the effects of currents in the section of the loop that is being modeled.

To define a source terminal as a current return path:

1. Click **Q3D Extractor** > **Reduce Matrix** > **Return Path**.

   The **Reduce Matrix - Return Path** window appears, with a list of terminals in the left pane.

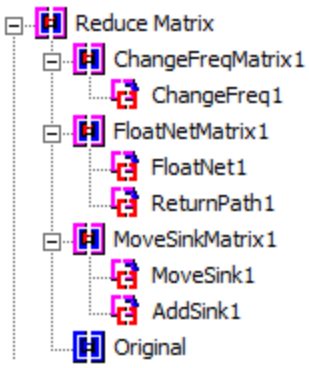


2. Select the source terminal you want to specify as a return path. As soon as you click a terminal, it is removed from the terminal list and added to the **Reduced Net List**.

> **Important:**
>
> You can only select one source terminal.

3. If you have a long list of terminals, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



Using Reduce Matrix > Change Frequency

Use this operation to change the frequency at which the AC resistance matrix is computed. **Change Frequency** scales the resistance matrix by a factor of:

$$\sqrt{\frac{f_{new}}{f_{old}}}$$

where:

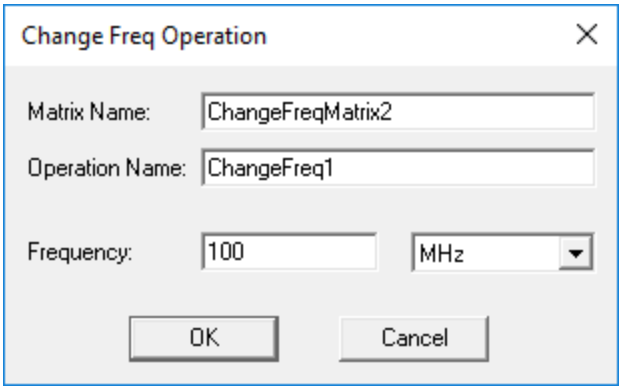
- f_{new} is the new frequency (which you will be asked to supply).
- f_{old} is the solution frequency for the matrix you are reducing.

> **Important:**
>
> **Change Frequency** only affects the AC resistance matrix. It has no effect on any other solution matrices.

To change the frequency of the model excitation:

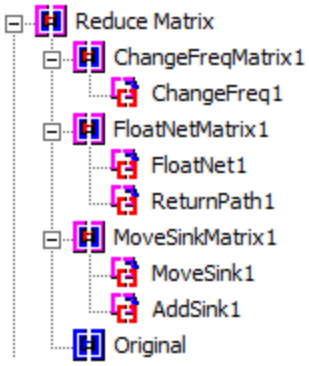
1. In the **Project Manager**, expand **Reduce Matrix** and select the matrix you want to reduce.
2. Click **Q3D Extractor** > **Reduce Matrix** > **Change Frequency**.

   The **Change Freq Operation** dialog box appears.



3. Enter a **Frequency** and select a unit of measure from the drop-down menu.
4. Click **OK**.

   Operations that have been performed appear in the **Project Manager**:



bo

Using Reduce Matrix Operations in 2D Extractor

Reduce Matrix operations allow you to modify conductor definitions and generate new matrices containing fewer elements. Operations for 2D Extractor include:

- Add Ground
- Set Reference Ground
- Float
- Parallel
- Diff Pair

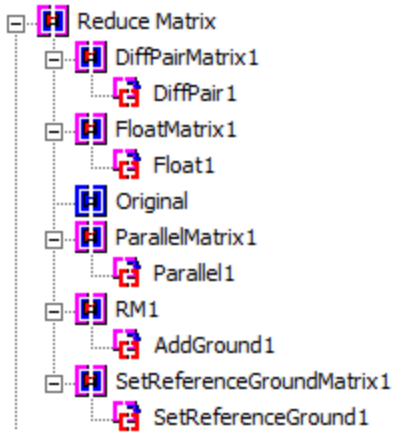
Reduced matrices allow you to:

- Simulate parallel connections between conductors.
- Model floating conductors in a structure.
- Examine the effects of grounding in a structure.
- Set up differential pairs.

Based on these modifications, 2D Extractor generates new matrices without computing new field solutions for any selected parameters that already have field solutions. This saves computing time and resources and allows you to easily optimize ground assignments when you generate a SPICE equivalent circuit.

You can select any object in the Modeler window and right-click, then select **Assign Reduce Matrix**. If the selected objects are valid terminals for reduce matrix operation, they are considered for reduction; otherwise, they are ignored.

After reduction, reduced matrices and their **Reduce Matrix** operations appear in the **Project Manager**:



You can right-click the original or a reduced matrix in the project tree to see more options.

You can also right-click a Reduce Matrix operation and click **Select Assignment** to select the objects to which it applies.

Duplicating a Reduced Matrix

To duplicate a **Reduce Matrix** operation, right-click it in the **Project Manager** and select **Duplicate**.

Deleting a Reduced Matrix or Reduce Matrix Operation

To delete a reduced matrix or a **Reduce Matrix** operation, right-click it in the **Project Manager** and select **Delete**.

> **Important:**
>
> You cannot delete the original matrix!

Using Reduce Matrix > Add Ground

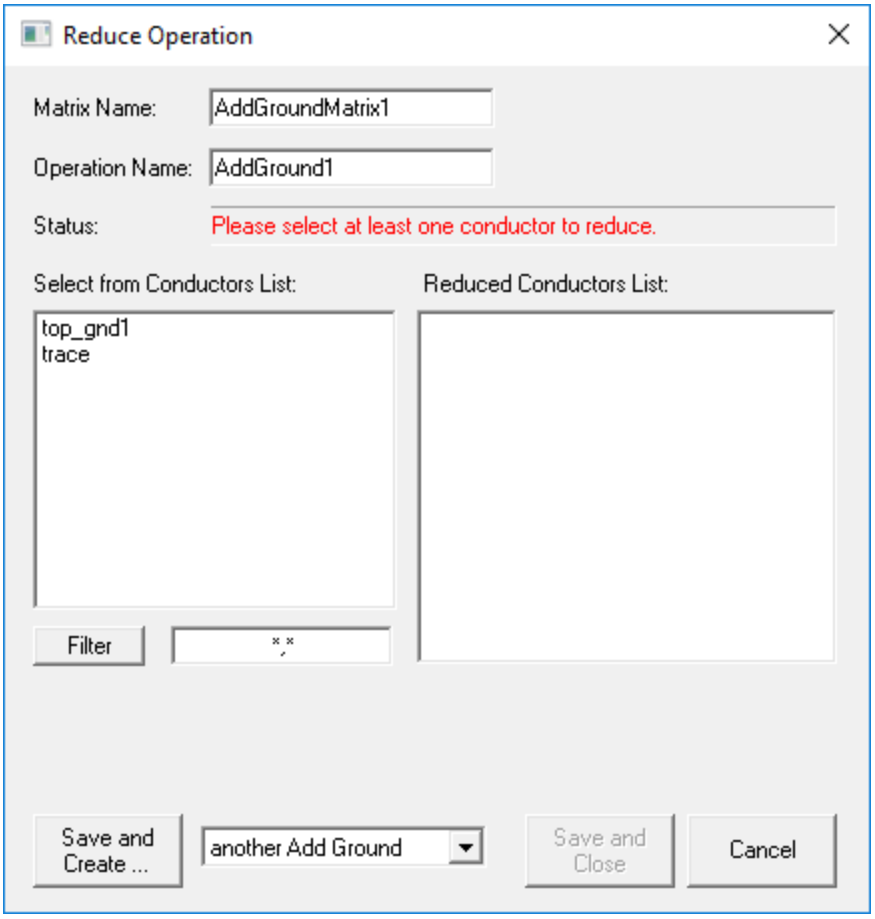
This type of reduction grounds the specified nets in your model. If you have objects not included in the matrix, they are already treated as grounded objects.

When a conductor is grounded, its source terminals are connected to its sink terminal. Current can flow in the newly created loops due to fields induced by the surrounding conductors. The fields produced by this loop affect voltage drops observed in surrounding conductors. Thus, the inductance matrix of the remaining conductors is altered by this operation.

To define which conductors in a structure are grounded:

1. Click **2D Extractor** > **Reduce Matrix** > **Add Ground**.

   The **Reduce Operation** window appears, with a list of the model's conductors in the left pane.

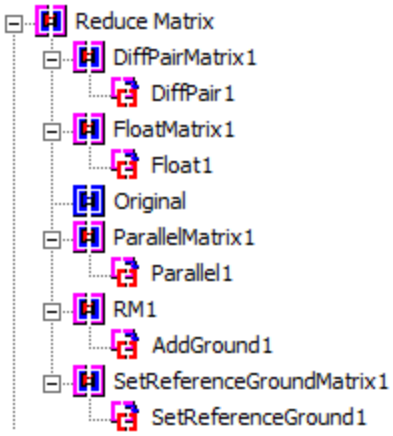


2. Select a conductor to ground. As soon as you click the conductor, it is removed from the conductors list and added to the **Reduced Conductors List**.

> **Note:**
>
> You cannot select all conductors.

3. If you have a long list of conductors, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



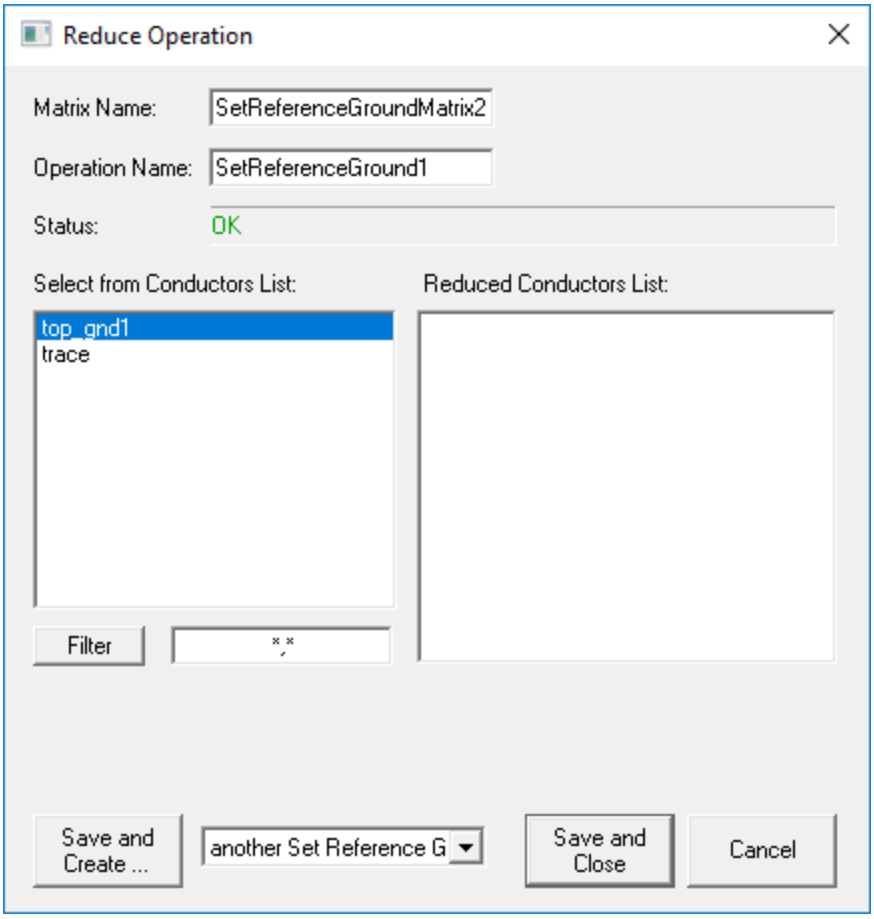
Using Reduce Matrix > Set Reference Ground

This type of reduction replaces the existing return path or reference ground in the design with the newly selected conductor.

To set the grounding:

1. Click **2D Extractor** > **Reduce Matrix** > **Set Reference Ground**.

   The **Reduce Operation** window appears, with a list of the model's conductors in the left pane.



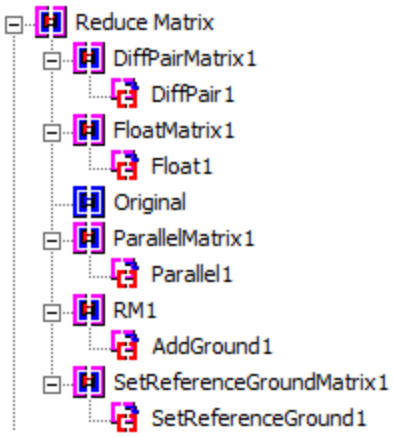
2. Select the conductors you want to ground. As soon as you click the conductor, it is removed from the conductors list and added to the **Reduced Conductors List**.

> **Note:**
>
> You can select an already reduced conductor. However, you cannot select all conductors.

3. If you have a long list of conductors, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.

- **Save and Close** to save the current operation and close the window.
- **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



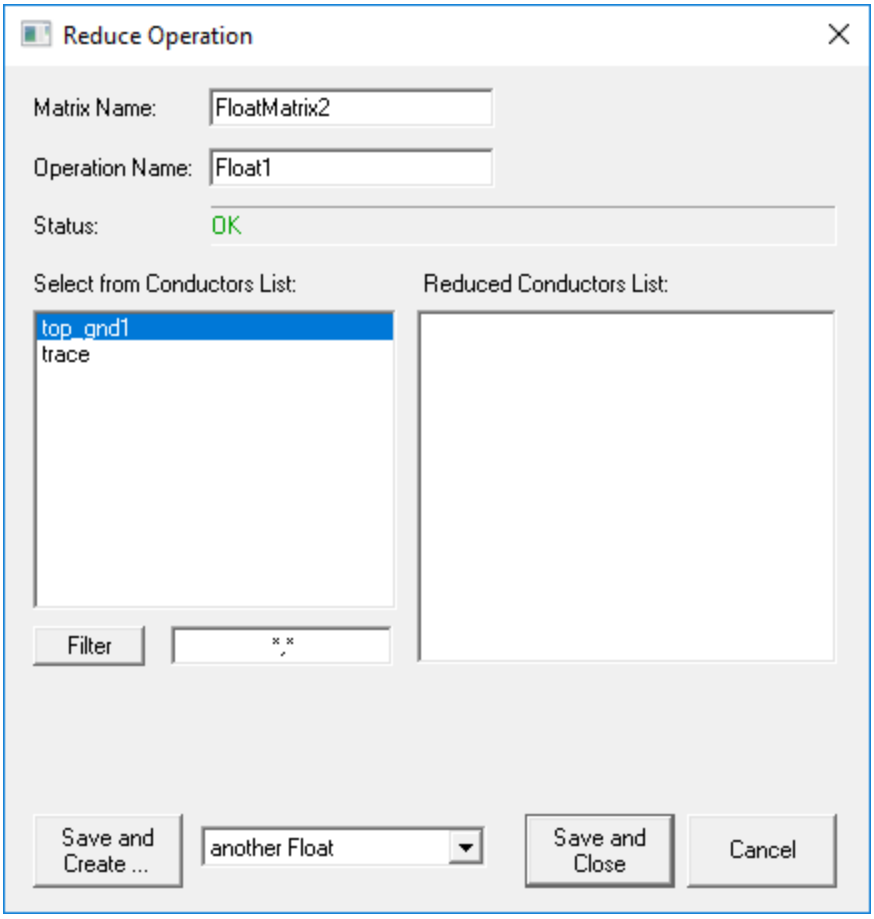
Using Reduce Matrix > Float

This type of reduction lets you model the presence of floating conductors in a structure.

To define which conductors are floating:

1. Click **2D Extractor** > **Reduce Matrix** > **Float**.

   The **Reduce Operation** window appears, with a list of the model's conductors in the left pane.



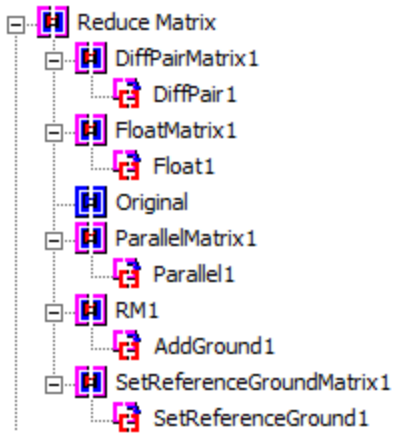
2. Select a conductor to float. As soon as you click the conductor, it is removed from the conductors list and added to the **Reduced Conductors List**.

> **Note:**
>
> You can select a conductor that has already been reduced. However, you cannot select all conductors.

3. If you have a long list of conductors, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.

- **Save and Close** to save the current operation and close the window.
- **Cancel** to close the window without making changes.

Operations that have been performed appear in the **Project Manager**:



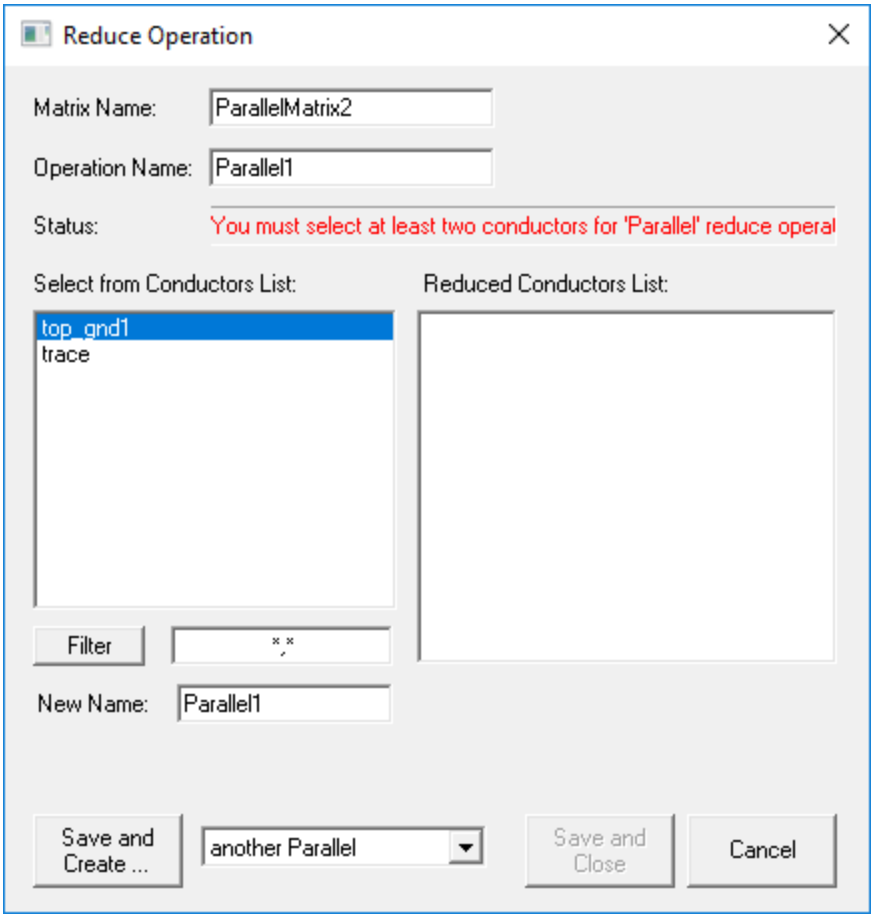
Using Reduce Matrix > Parallel

Parallel reduction lets you connect two or more source terminals in parallel, which are then treated as if they were a single object. The rows and columns that represent their contributions to the matrix are replaced with a single row and column representing the contribution of this new object.

To set up parallel connections between the conductors in a structure:

1. Click **2D Extractor** > **Reduce Matrix** > **Parallel**.

   The **Reduce Operation** window appears, with a list of the model's conductors in the left pane.



2. Select two or more conductors to connect in parallel. As soon as you click a conductor, it is removed from the conductors list and added to the **Reduced Conductors List**.

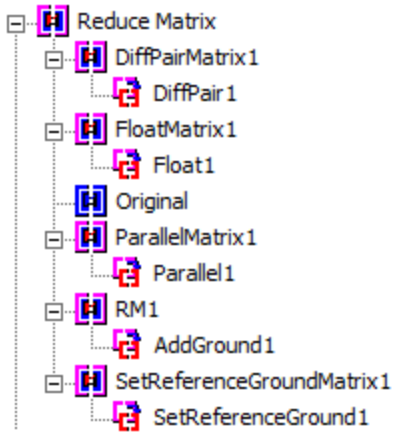
> **Note:**
>
> You cannot select a conductor that has already been reduced.

3. If you have a long list of conductors, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Enter a **New Name**. Since these conductors are treated as a single conductor in the reduced matrix, their names will be replaced with the one you specify here.

> **Note:**
>
> The **New Name** must be unique.

5. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

   Operations that have been performed appear in the **Project Manager**:



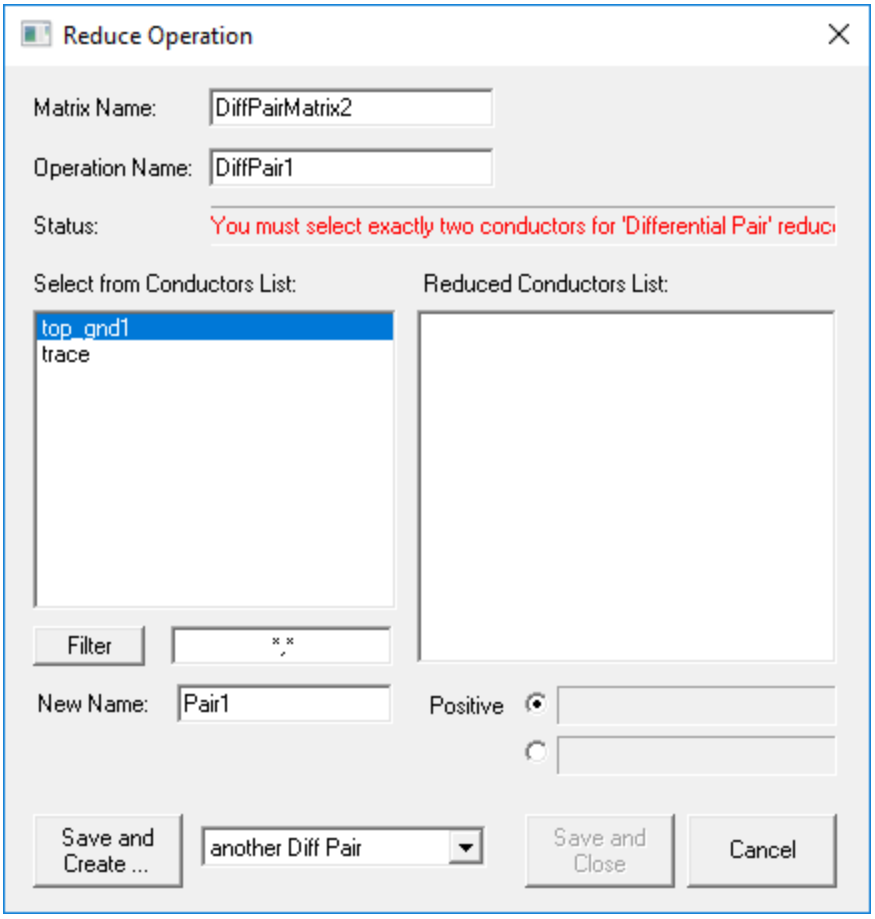
Using Reduce Matrix > Diff Pair

Common mode and differential mode entries in matrices are generated for each differential pair.

To set up differential connections between the conductors in a structure:

1. Click **2D Extractor** > **Reduce Matrix** > **Diff Pair**.

   The **Reduce Operation** window appears, with a list of the model's conductors in the left pane.



2. Select two conductors to create a differential pair. As soon as you click a conductor, it is removed from the conductors list and added to the **Reduced Conductors List**.

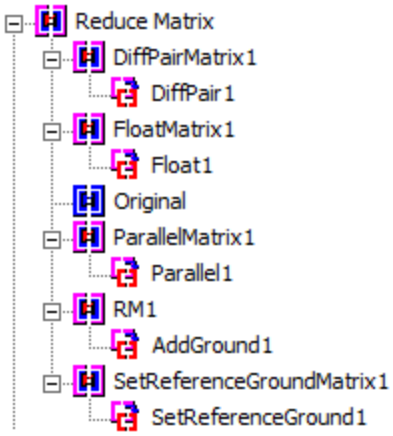
> **Note:**
>
> Only parallel-reduced conductors can be used to create differential pair operation.

3. If you have a long list of conductors, you can filter the list by entering search criteria and clicking **Filter**. Use multiple search criteria with a comma separator (for example, "Redsource*, BlueSource*").
4. Enter a **New Name** for the pair. The positive reference will be called "<new name>.cm" and the negative reference will be "<new name>.df".

> **Note:**
>
> The **New Name** must be unique.

5. Click one of the following:
   - **Save and Create** to save and leave the Reduce Operation window open to add an additional operation. You must select the next operation from the drop-down menu before clicking this button.
   - **Save and Close** to save the current operation and close the window.
   - **Cancel** to close the window without making changes.

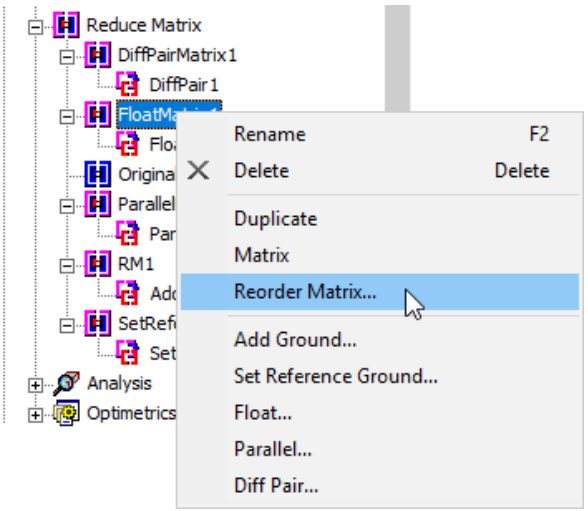
   Operations that have been performed appear in the **Project Manager**:



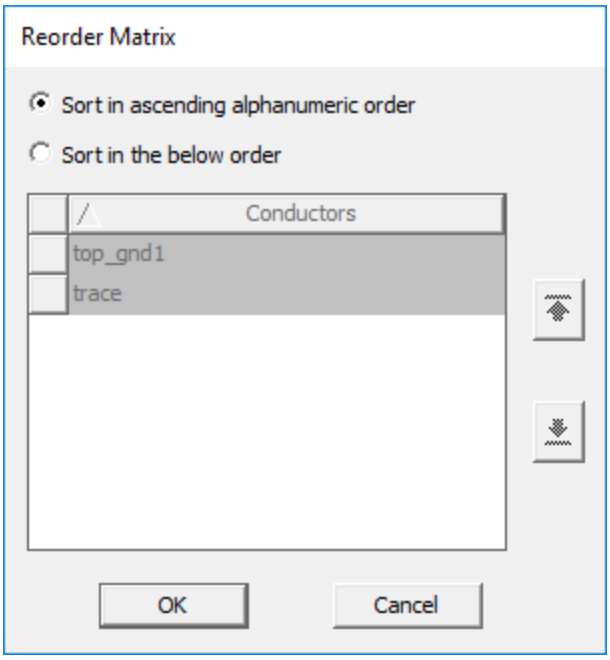
Reordering Matrices

To reorder any matrix:

1. In the **Project Manager**, expand **Reduce Matrix**.
2. Right-click a matrix and select **Reorder Matrix**.



The **Reorder Matrix** dialog box appears.



For 2D Extractor, this window contains only one tab. For Q3D Extractor, it contains tabs for RL and CG. Both tabs have the same two options.

3. Select either **Sort in ascending alphanumeric order** or **Sort in the below order**.

4. If you selected **Sort in the below order**, use the up and down arrows to rearrange the selected conductors, nets, or terminals; or drag-and-drop the items.
5. Click **OK**.

15 - 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 Q3D Extractor. For example, you can parameterize the model geometry or material properties. You can then perform the following types of analyses on your nominal design:

optiSLang	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.
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 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.
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.
DesignXplorer	An optimization tool for studying a range of design variations, used with Design of Experiments. This permits you to manage an Optimetrics simulation from the Ansys Workbench.
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 can be computed.
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.

Note:

Sensitivity, Statistical, Design Xplorer, and some Optimizers have been placed in legacy mode because they are no longer under active development. Go to **Tools** > **Options** > **General Options** > **Miscellaneous** and toggle the **Enable Legacy Optimetrics Tools** check box to show or hide these legacy tools.

Note:

Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

The HPC and Analysis Options dialog can be accessed from the setup dialog for each type of Optimetrics analysis.

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 Q3D Extractor 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. You can create a parametric setup before defining variables but 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 within which optimal conditions will occur.

Setting Up a Parametric Analysis

A *parametric setup* specifies all of the design variations that Optimetrics drives the software to solve. A parametric setup is made up of one or more variable sweep definitions, which are a set

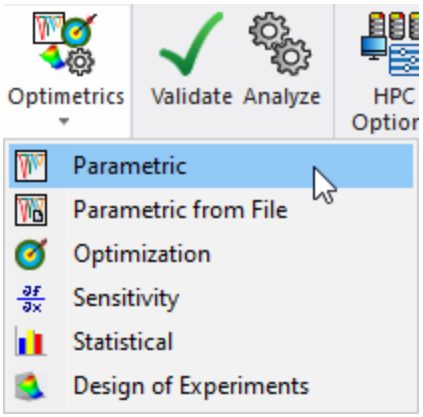
of variable values within a range that you want the software to solve when you run the parametric setup.

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. Click **Q3D Extractor**> **Optimetrics Analysis**> **Add Parametric** .
   - Alternatively, right-click **Optimetrics** in the Project Manager and then click **Add** > **Parametric** on the shortcut menu.
   - Select the **Simulation** tab in the ribbon, and select **Parametric** from the drop-down menu under the Optimetrics icon:



   The **Setup Sweep Analysis** dialog box appears.

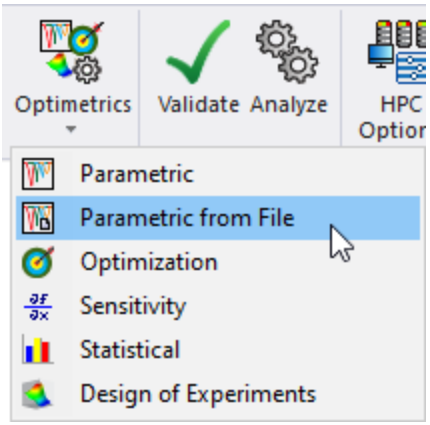
2. Add a variable sweep definition.

   After you define a parametric sweep, a shortcut menu becomes available when you right-click the setup name in the Project Manager.

> **Note:**
>
> Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

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 **Q3D Extractor** > **Optimetrics Analysis** > **Add Parametric from File** command, or with the **Simulation** tab selected, use the drop-down menu under the Optimetrics icon and select **Parametric from File**:



These methods open a file browser for a comma delimited file (.csv) or a tab delimited .txt file.

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. When unit not present, SI unit is assumed. The following rows provide the variable values and units, where SI is assumed if not specified. 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 [ ].

After you have imported a valid file, the detailed information regarding sweep values can be found at the lower portion of the Parametric **Setup Sweep Analysis** dialog**Sweep Definitions** tab (Operation/Description panel). It is treated as: first row as single point. then each time a new sweep point is added.

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 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. Click **Q3D Extractor** > **Optimetrics Analysis** > **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** drop-down menu of the **Add/Edit Sweep** dialog.
3. Click the variable for which you are defining the sweep definition from the **Variable** drop-down menu.

   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. Click **OK**.

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 value. The model is solved 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.

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, 3 design variations are solved. 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.

Creating Parametric Animations

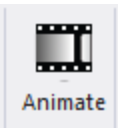
Parametric animations can be based on parametric sweeps. Following is the general procedure for creating an animation of a plot based a parametric sweep.

Prerequisites

- For animation of a field plot/overlay plot over a geometric variable, you must define and solve at least one optimetric sweep with same variable values and Save Fields selected before you can create a parametric animation of a field solution. See Creating Geometry Animations.
- For animation of geometry and VRT plots, solving parametric setup is not required.
- Animations can be based on time variables and objects whose location is linked to the time variable. See the animated example below.

Procedure:

1. Right-click in the **Modeler** window, and click **View**> **Animate...**, or select the **View** tab of the ribbon, and click the **Animate** icon.

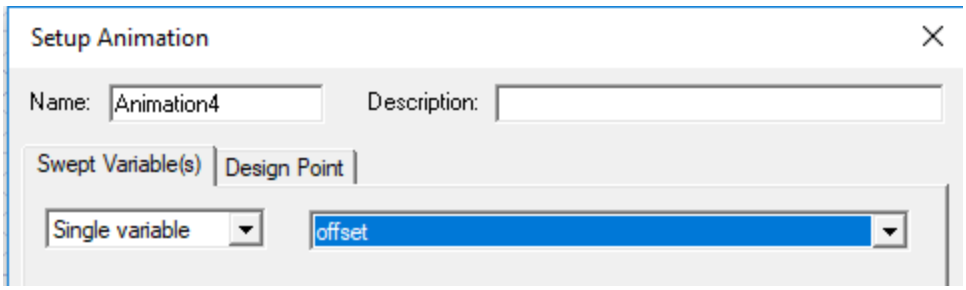
   

   If multiple geometries can be varied in the design, the *Select Drawing* dialog box appears; proceed to step 2. If only one geometry is variable, proceed to step 3.

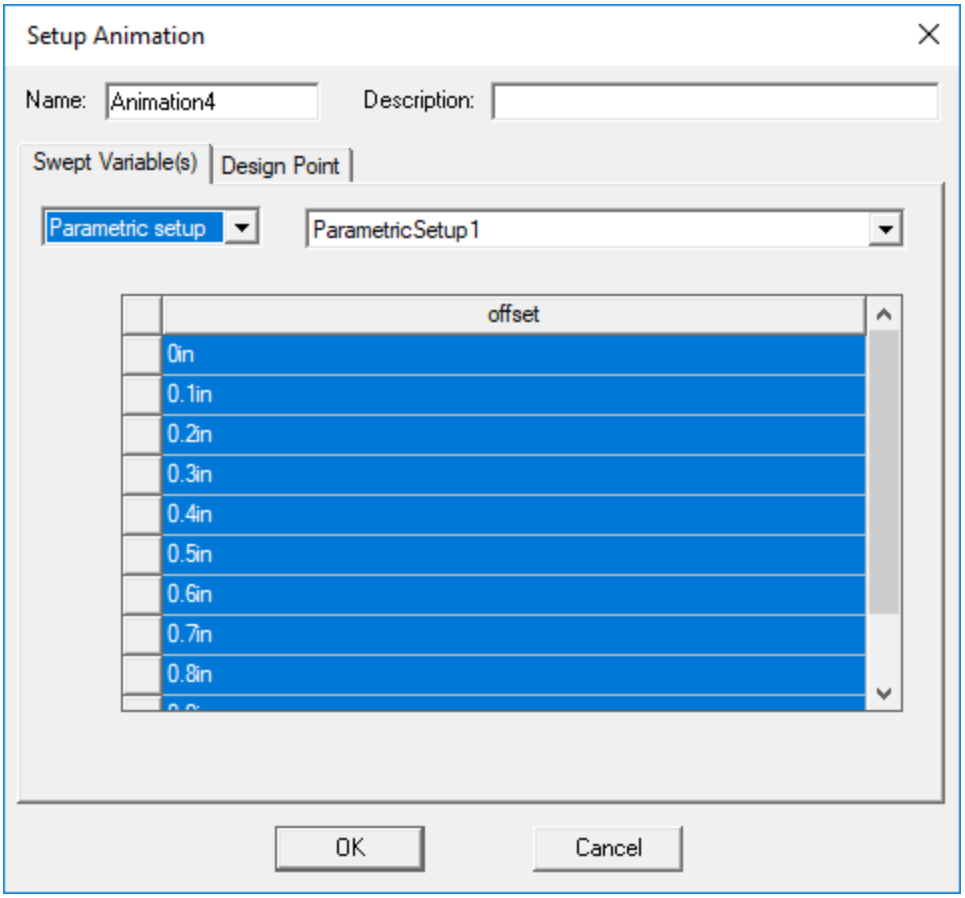
2. In the *Select Drawing* dialog box:
   a. Select the geometry variable to vary in the animation.
   b. Select the object you want to animate.

> **Note:**
>
> If previous animations have been created for this project, the *Select Animation* dialog box 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.

The *Setup Animation* dialog box appears.



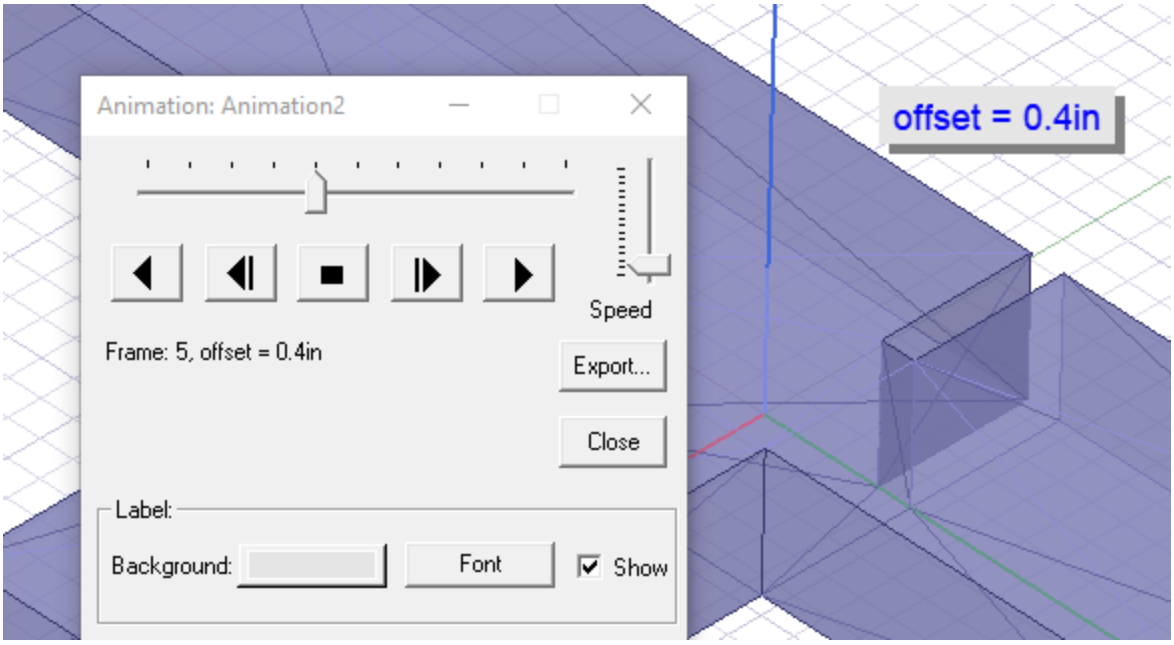
3. If your have created one or more Parametric Sweeps for one or more geometric variables, you can select **Parametric setup**, the setup to use, and the variable values to animate.



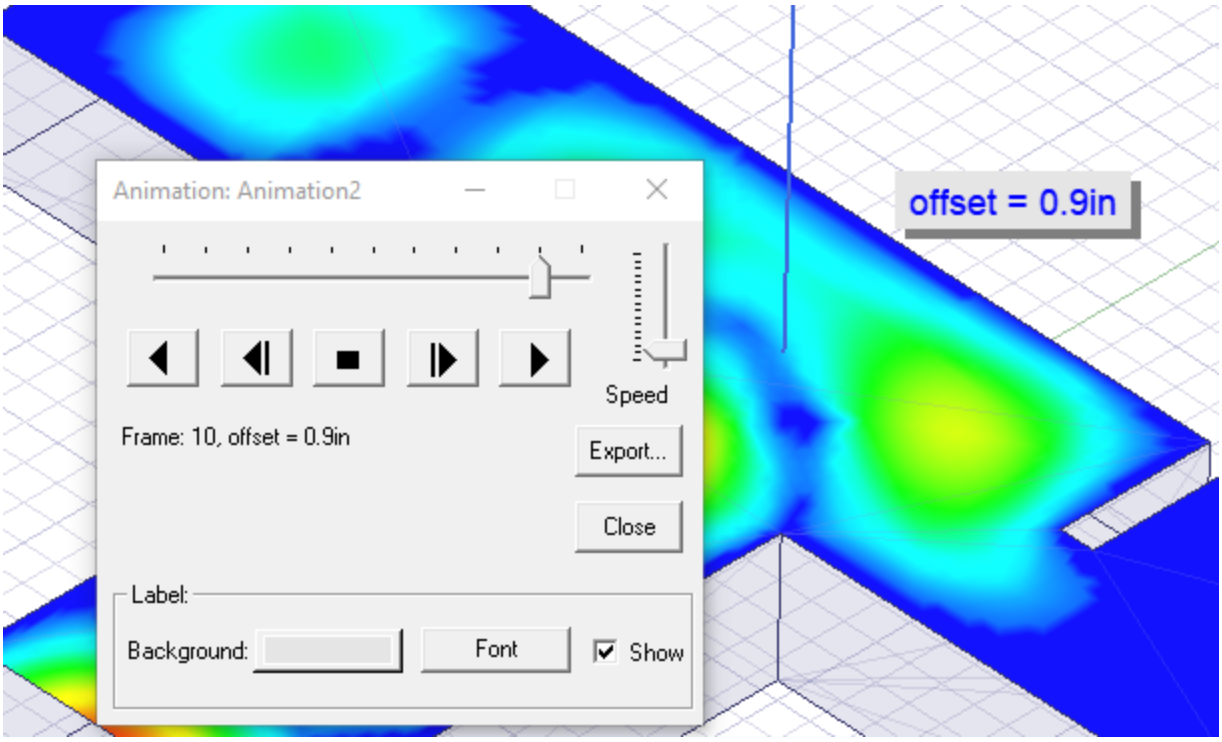
4. Click **OK**.

   The animation begins in the view window. It will display one frame for each variable value.

   The **Animation** control panel lets you to stop, restart, and control the speed and sequence of the frames.



While a parametric animation is running, you can also turn on other field plots.



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 that will be solved for the parametric setup are 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.

Overriding a Variable's Current Value in a Parametric Setup

If you choose not to sweep a variable, the variable's current value set for the nominal design is used 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 **Override** box of the design 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.

Specifying a Solution Setup for a Parametric Setup

To specify the solution setup that the software 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 the software to use when it solves the parametric setup.

   The parametric setup is solved using the solution setup you select. If you select more than one, results are generated for all selected solution setups.

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, Q3D Extractor 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 box. 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 box's **Calculation Expression** field to the Calculations tab of the *Setup Sweep Analysis* dialog box.
4. Click **Done** to close the *Add/Edit Calculation* dialog box.

Setting up Calculations for Optimetrics

The **Setup** dialog boxes for each of the Optimetrics types include a **Setup Calculations** button. Clicking this button displays the **Add/Edit Calculation** dialog box, which 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 box 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.

To set up an Optimetrics calculation:

1. Click **Setup Calculations** to open the **Add/Edit/Calculation** dialog box.
2. In the **Report Type** text field in the **Context** pane, select from the drop-down menu 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 menu of available solutions.
4. If the **Geometry** field is available, select from the drop-down menu.
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:

   i. Click **Output Variables**.

      The **Output Variables** dialog box appears.

   ii. Add the expression you want to evaluate, and then click **Done**.

      The recently created output variable appears in the **Quantity** list.

   iii. Click a new output variable in the **Quantity** drop-down menu.

   > **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 box's **Calculation Expression** field to the Calculations tab of the **Setup Sweep Analysis** dialog box.
8. Click **Done** to close the **Add/Edit Calculation** dialog box.

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. For a Driven Modal or Driven Terminal design, if you selected to extract the solution from the last adaptive solution, Optimetrics uses the adaptive frequency defined in the solution setup. If you selected to extract the solution quantity from a frequency sweep solution, Optimetrics by default will use the starting frequency in the sweep.

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. In the **Value** box, 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.
6. Click **Update**, and then click **Edit**.

Using Distributed Analysis

If you have purchased the appropriate license, the Electronics Desktop supports distributed solve, which involves distributing rows of a parametric table during Optimetrics solve.

If you do a distributed solve, the Electronics Desktop launches solver engines on multiple machines, assuming that you have configured your HPC and Analysis Options correctly. Also see Large Scale DSO for Parametric Analysis.

To run a distributed analysis:

1. Under **Optimetrics** in the project tree, right-click the specific parametric setup.

   A shortcut menu appears.
2. Select **Analyze** 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**.

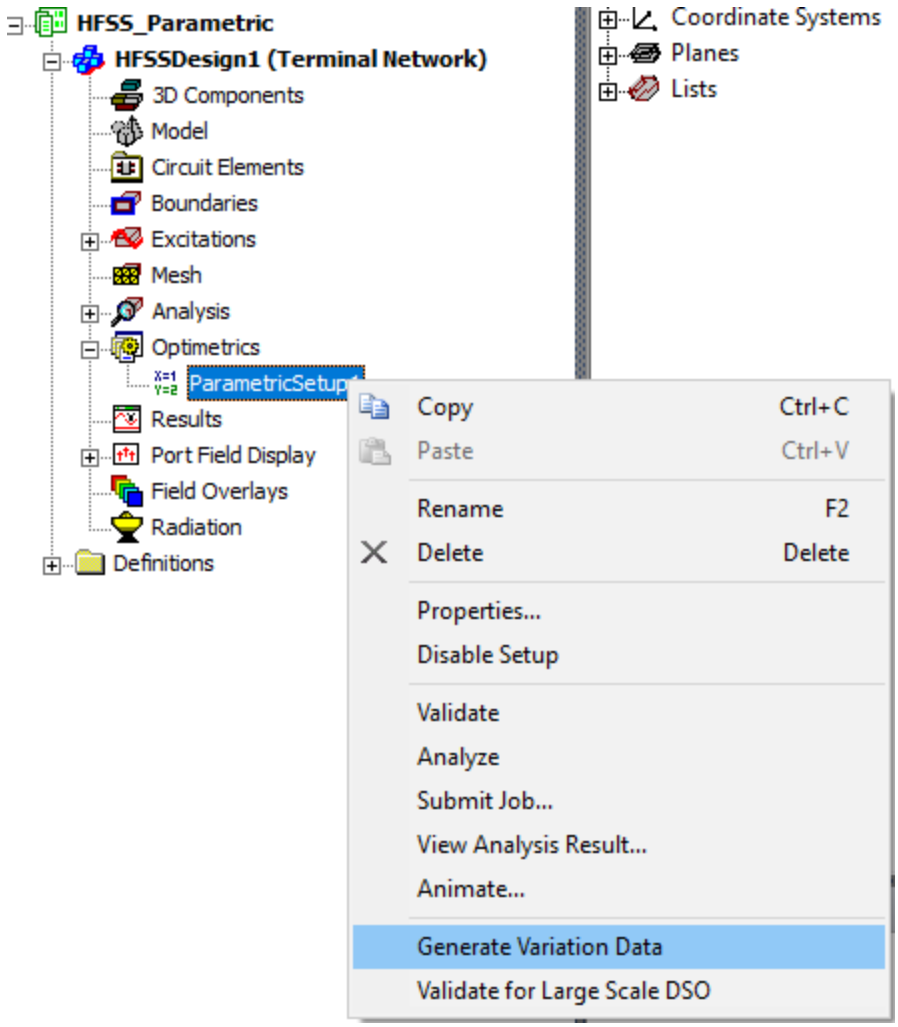
  To hide the child progress bars:

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

Solving a Parametric Analysis on Another Machine

If you would like to run a parametric analysis on another machine, for example, to run a Discovery-linked model on Linux, you can generate the variation data and then export an archive of the project.

To generate the variation data, right-click on the **ParametricSetup** entry in the Project Manager tree, and select **Generate Variation Data**.



When this option is selected, Q3D Extractor will generate geometry for all variations and cache it in the results folder. Users can archive all the data, and then solve on other machines.

> **Note:** If the parametric analysis uses a Discovery-linked model, the machine that the archive is moved to does not have to have Discovery installed.

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 drop-down menu 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).

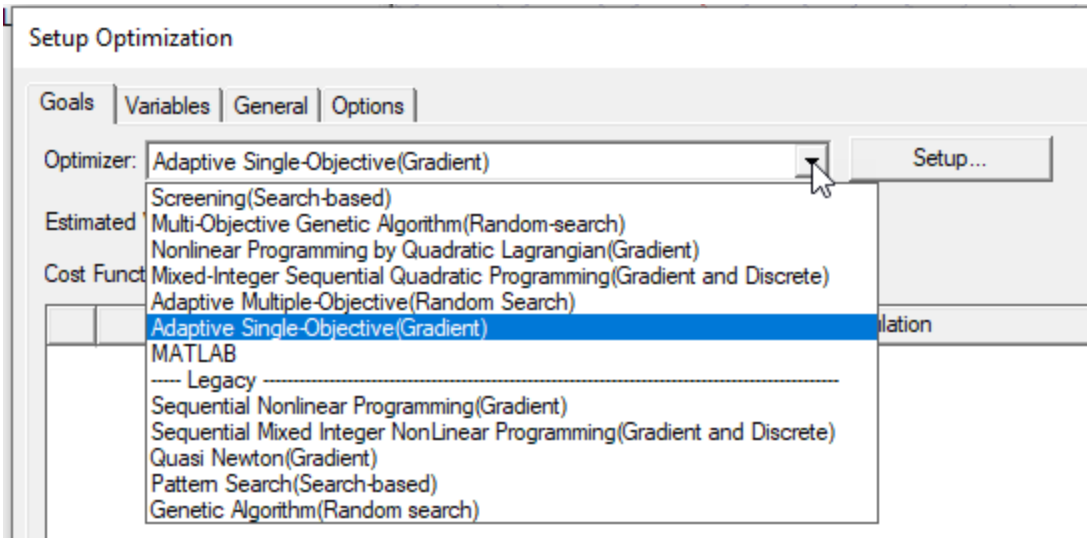
   The design displayed in the **Modeler** window is changed to represent the selected design variation.

Optimization Overview

Optimetrics interfaces with Ansys Electromagnetics products to help you optimize 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.

Choosing an Optimizer

Conducting an optimization analysis allows you to determine an optimum solution for your problem. In optimization analyses, there are many available optimizers.



Goal Driven 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 post-processing action on the Pareto fronts as generated from the results of the various optimization methods.

- Screening (Search based) – 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 – 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.
- Nonlinear Programming by Quadratic Lagrangian (Gradient) – a gradient-based, single-objective optimizer based on quasi-Newton methods. Ideally suited for local optimization.
- Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete) – 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 – 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) – a gradient-based, single-objective optimizer that employs an OSF (Optimal Space-Filling) DOE, a Kriging response surface, and MISQP.
- MATLAB

All 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 SMINLP 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.

Legacy Optimizers

These include

- Sequential Nonlinear Programming (Gradient)
- Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete)
- Quasi-Newton (Gradient)
- Pattern Search (Search-based)
- Genetic Algorithm (Random Search)

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:

$$g_k(\{x\}) \leq 0, \forall k = 1,2,\ldots,K$$

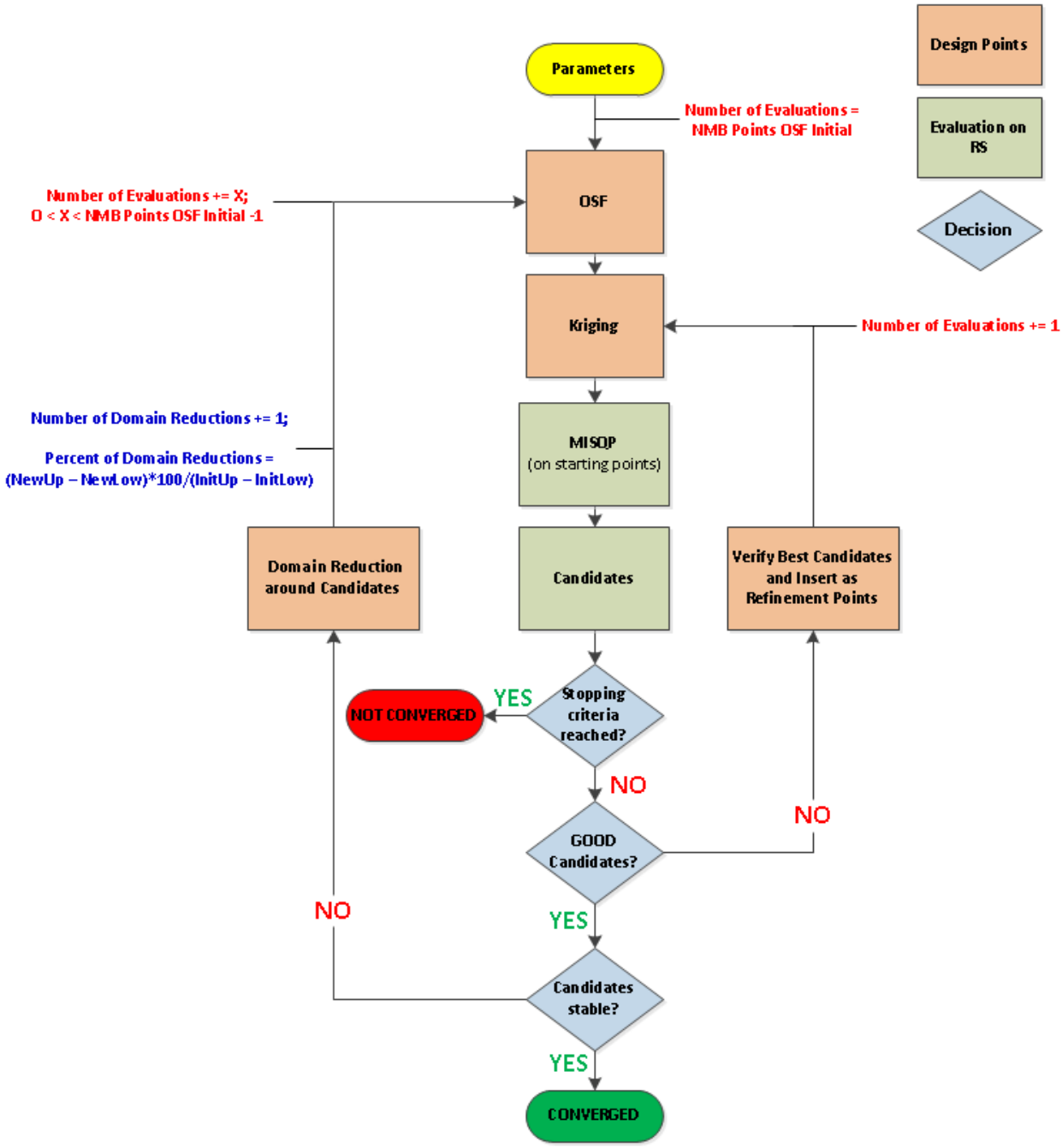
$$h_l(\{x\}) = 0, \forall I = 1,2\ldots,L$$

Where:

$$\{x_L\} \leq \{x\} \leq \{x_U\}$$

ASO Workflow

The workflow of ASO follows:



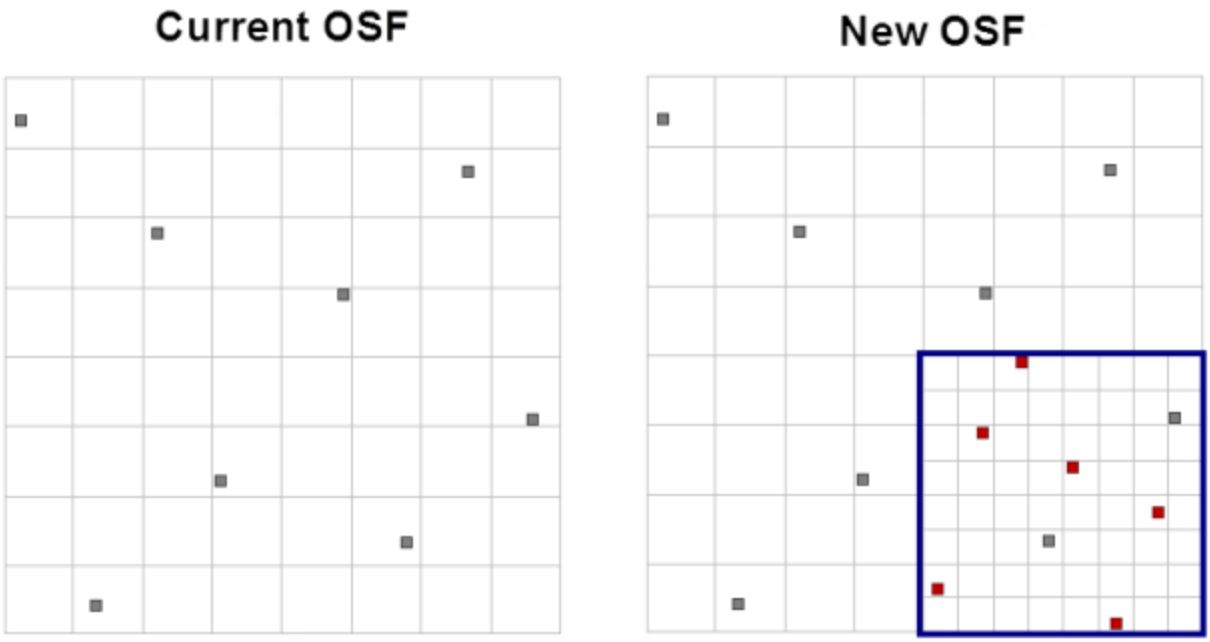
AMO Steps

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.

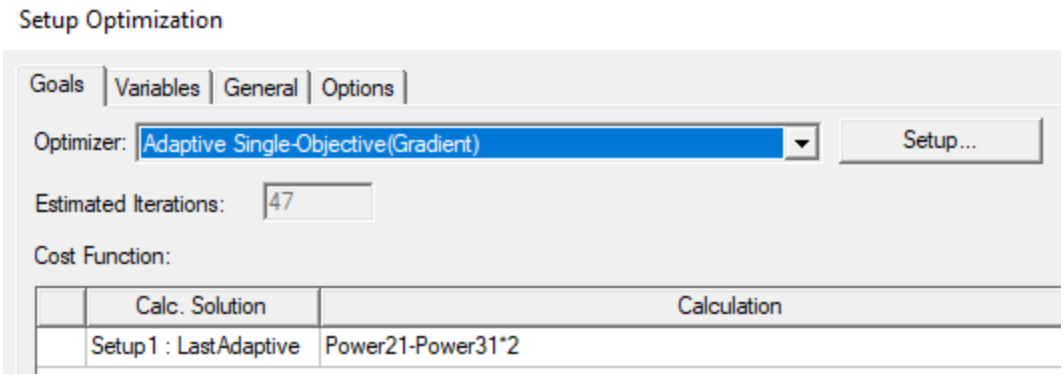
Setting Up Adaptive Single-Objective (Gradient) Optimizer

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.

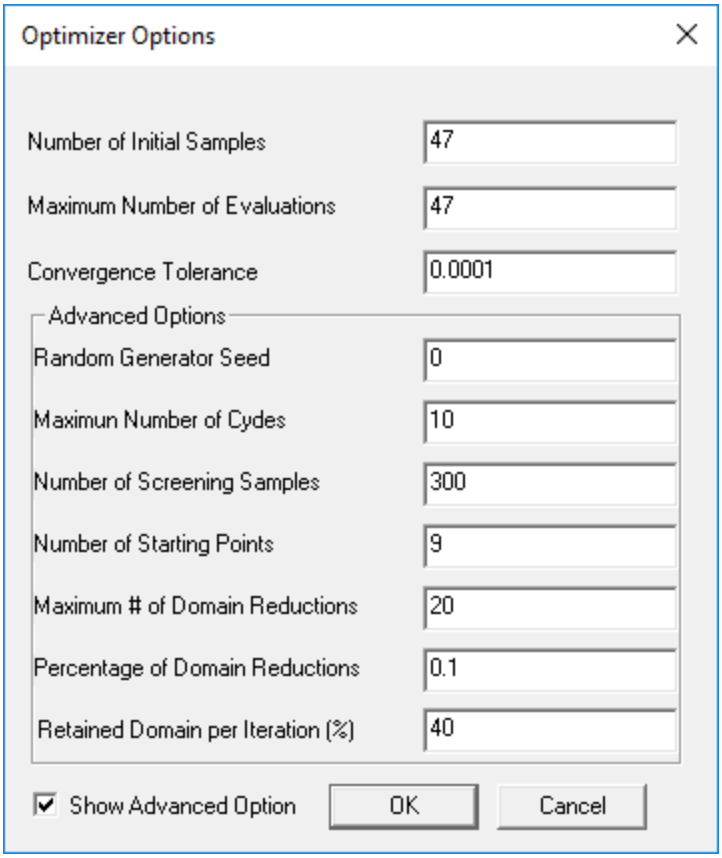
This gradient-based method employs automatic intelligent refinement to provide the global optima. It requires a minimum number of design points to build the Kriging response surface, but, in general, this method reduces the number of design points necessary for the optimization. Failed design points 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 **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & Optimization**. The *Setup Optimization* dialog box appears.
3. Under the **Goals** tab, choose 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.

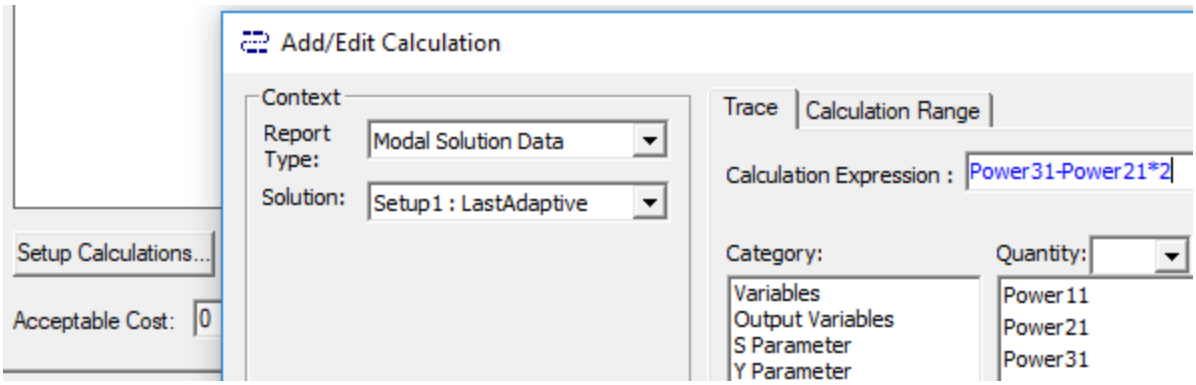
  You can enter a minimum of (NbInp+1)*(NbInp+2)/2 (also the minimum number of OSF samples required for the Kriging construction) or a maximum of 10,000. The default is 100*NbInp for a Direct Optimization system. There is no default for a Response Surface Optimization system.

  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

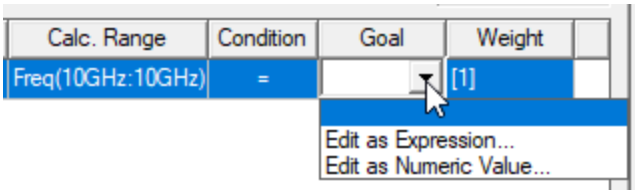
value indicates a faster and more accurate solution, with the risk of it being a local one. The default percentage value is 40.

5. Add a cost function by selecting the **Setup Calculations** button to open the *Add/Edit Calculation* dialog box.

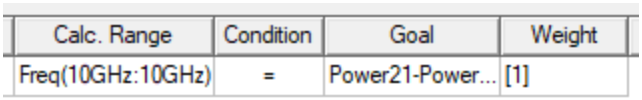


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog box.

6. In the Optimization setup, in the Goal column drop-down menu, select either **Edit as Expression...** or **Edit as Numeric Value...**. This reopens the *Add/Edit Calculation* dialog box.



7. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value into the **Goal** column.



8. 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.

9. Optionally, set the Acceptable Cost and Cost Function Noise.
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.
    - 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.
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** check box 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 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.

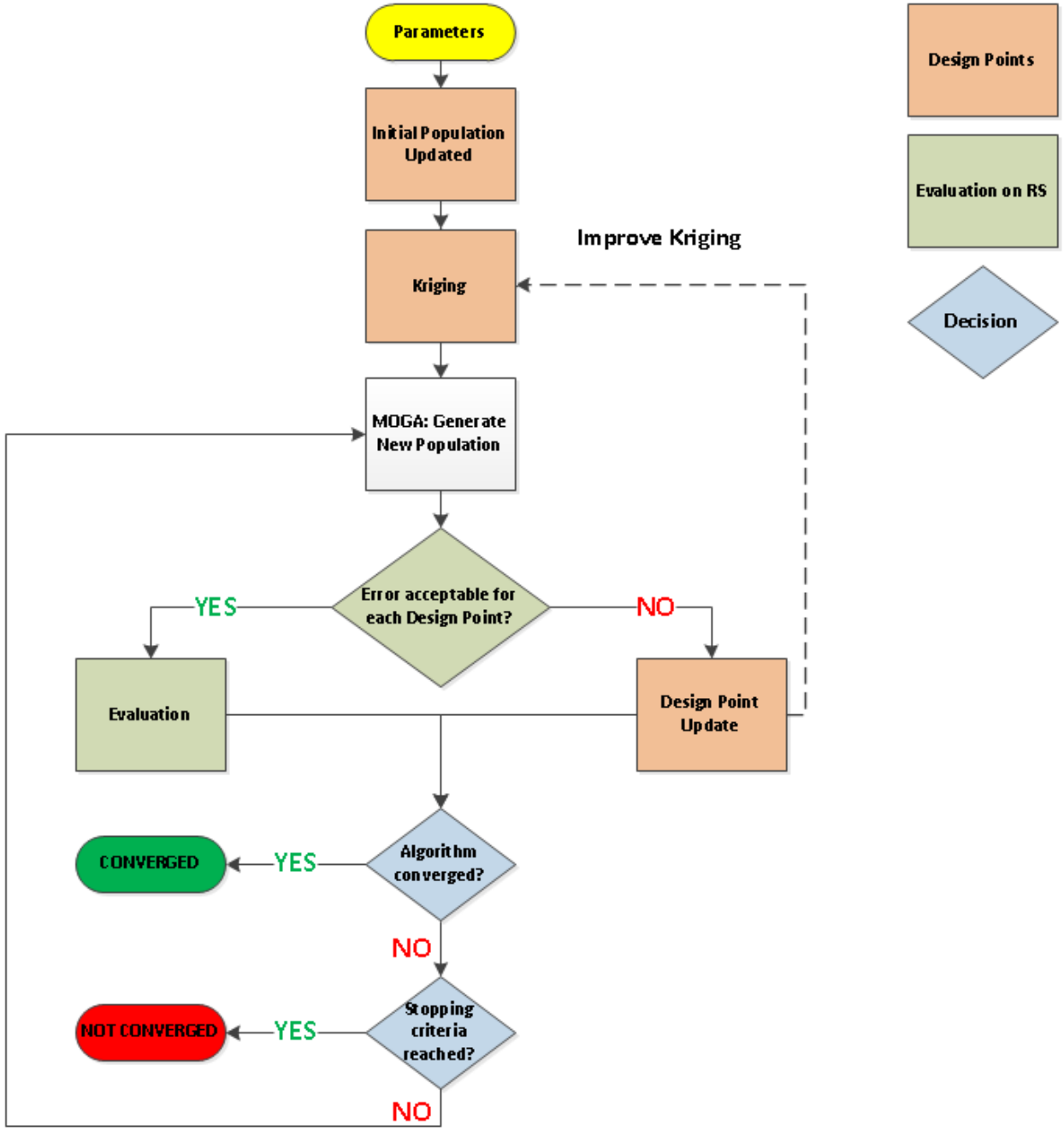
Adaptive Multiple Objective Optimization

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**

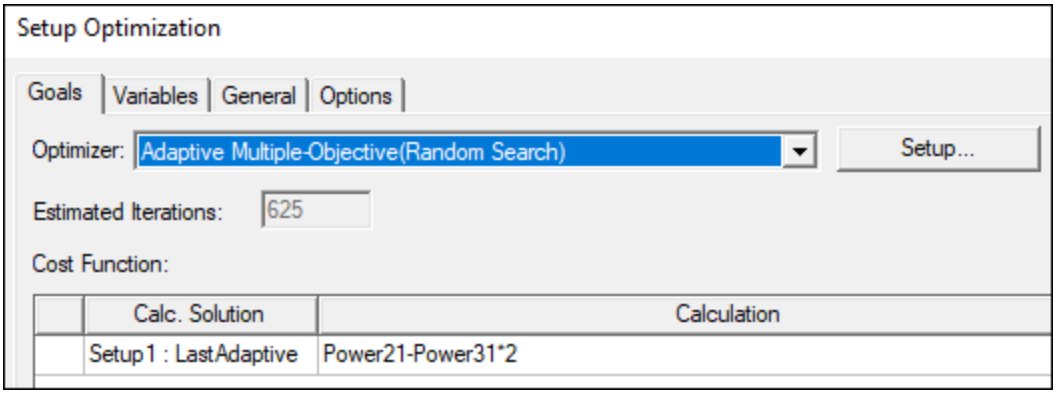
   Steps 2 through 7 are repeated in sequence until the optimization has converged or the stopping criteria have been met. When either of these things occurs, the optimization concludes.

Setting Up Adaptive Multiple-Objective (Random Search) Optimizer

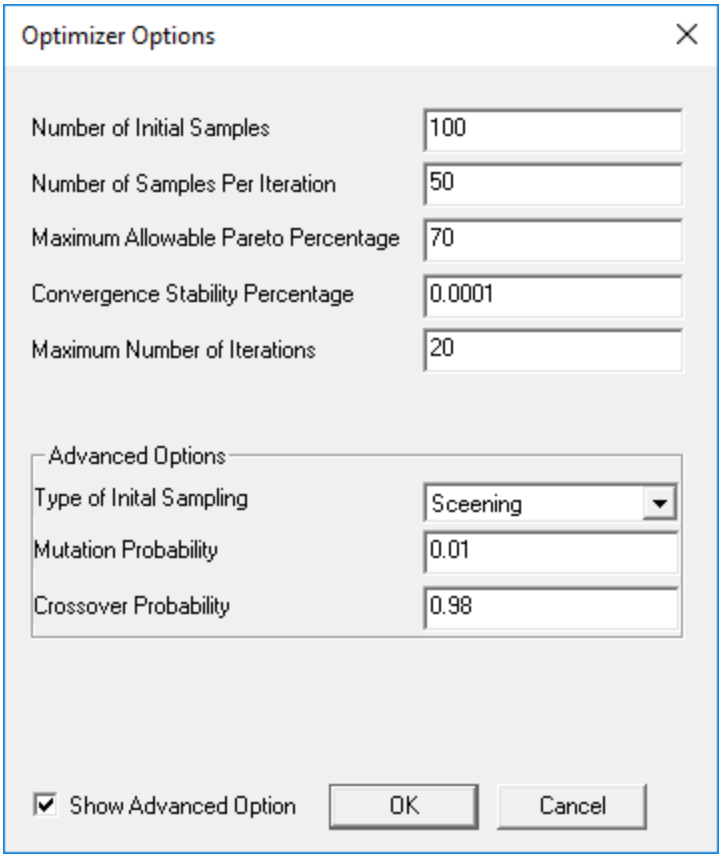
Following is the procedure for setting up an optimization analysis using the Adaptive Multiple Objective (Random Search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

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 **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & 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.

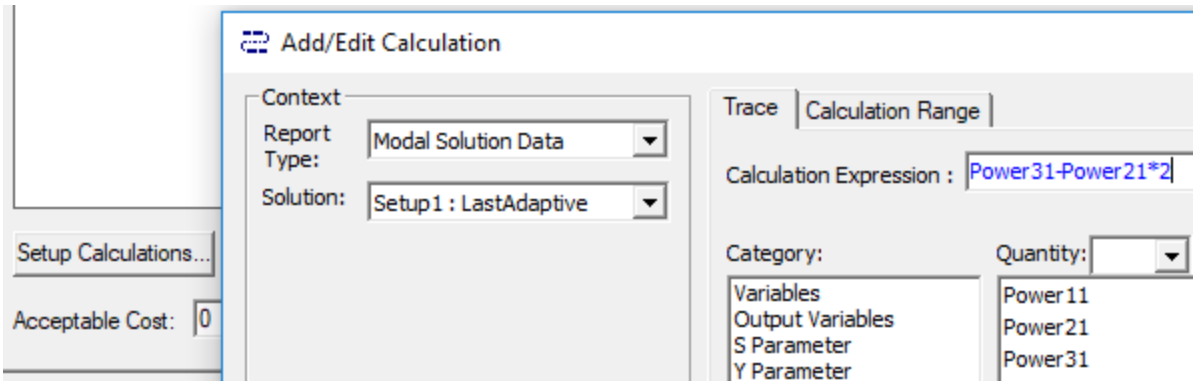
- **Maximum Allowable Pareto Percentage**: Convergence criterion. Percentage value that represents the ratio of the number of desired Pareto points to the number of samples per iteration. When this percentage is reached, the optimization is converged. For example, a value of 70 with Number of Samples 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 0.0001. 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.

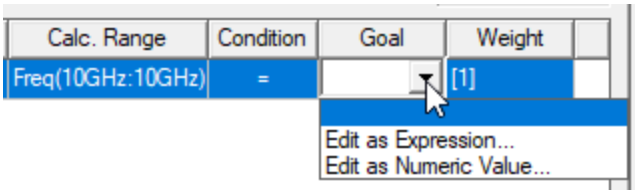
- **Crossover Probability**: Advanced option for specifying the probability with which parent solutions are recombined to generate offspring solutions. The value must be between 0 and 1. A smaller value indicates a more stable population and a faster (but less accurate) solution. If the value is 0, the 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 box.

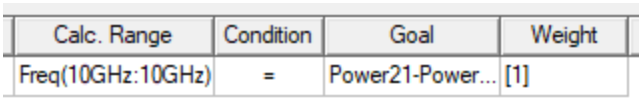


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the *Add/EditCalculation* dialog box.

6. In the Optimization setup, in the Goal column drop-down menu, select either **Edit as Expression...** or **Edit as Numeric Value...**. This reopens the *Add/Edit Calculation* dialog box.



7. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value into the **Goal** column.



8. 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.
9. Optionally, set the Acceptable Cost and Cost Function Noise.
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.
    - 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.
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** check box 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 variations in the optimization analysis, select **Save Fields And Mesh**.

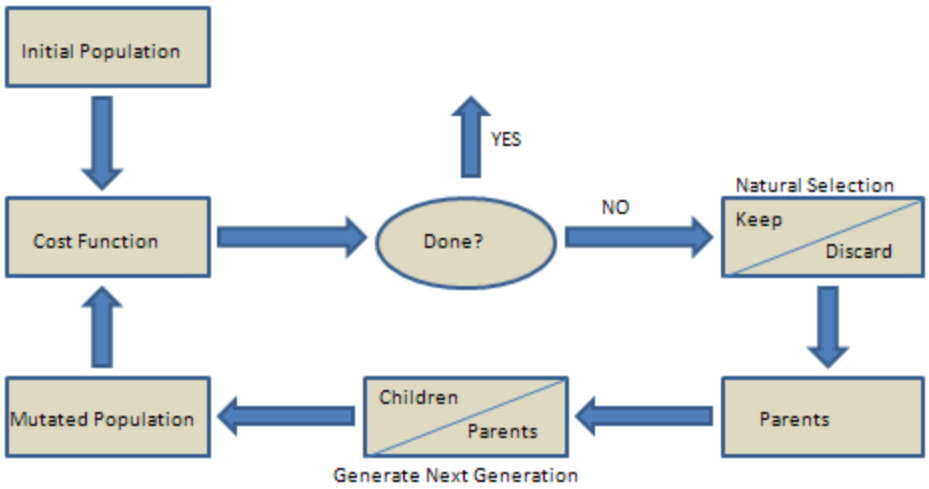
> **Note:**
>
> Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

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.

Genetic Algorithm (Random Search)

Genetic Algorithm (Random Search) 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 fills 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.

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 **Setup** to open the *Advanced Genetic Algorithm Optimizer Options* dialog box.
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 **Reproduction setup** for the dialog box 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 to 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 box, click **OK** to close it and return to the *Advanced Genetic Algorithm Optimizer Options* dialog box.
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 is 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 box.

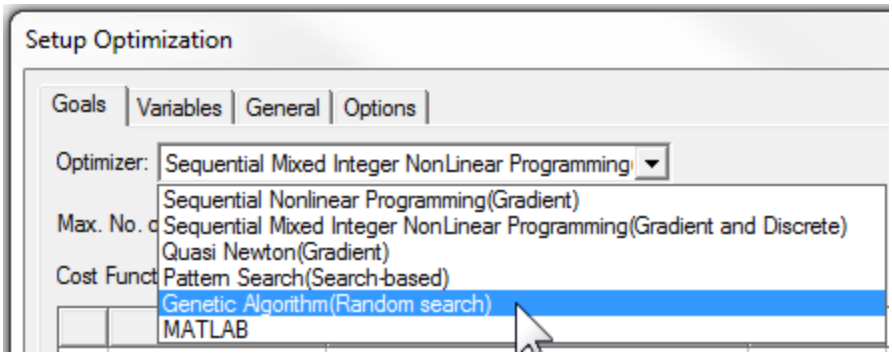
Optimization Setup for the Genetic Algorithm (Random search) Optimizer

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. Click **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & 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 **Setup...** 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 box.
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** 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.
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.

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 (e.g., 32/64 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.
- The MATLAB installation must include the MATLAB Optimization Toolbox. In addition, the MATLAB license must support using the Optimization Toolbox.

To see if the optimization toolbox is installed, users can type the "ver" command at the command prompt of a running MATLAB instance. For example:

```
>> ver
-----------------------------------------------------------------------
--------------------------------
MATLAB Version: 8.1.0.604 (R2023a)
MATLAB License Number: 162684
Operating System: Microsoft Windows 10
```

```
Java Version: Java 1.6.0_17-b04 with Sun Microsystems Inc. Java
HotSpot(TM) 64-Bit Server VM mixed mode
--------------------------------------------------------------------
---------------------------------
MATLAB Version 8.1 (R2023a)
Simulink Version 8.1 (R2023a)
Optimization Toolbox Version 6.3 (R2023a)
```

To see if the optimization toolbox is licensed, you can use the "license('test','optimization_ toolbox')" command at the MATLAB command prompt:

```
>> license('test','optimization_toolbox')
ans =
1
```

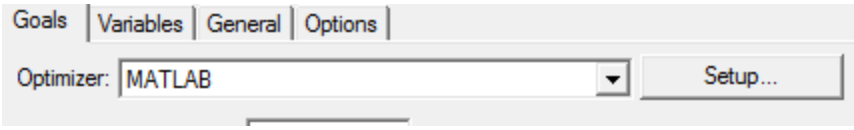
The answer will be 1 if the MATLAB Optimization Toolbox is licensed or 0 otherwise.

Specifying the MATLAB Location

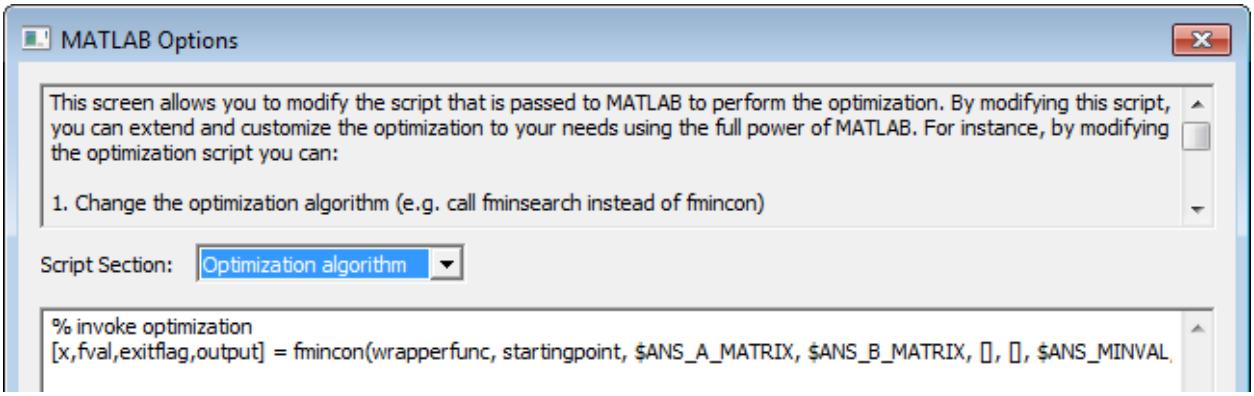
The Tools> General Options: Miscellaneous group contains a setting for the MATLAB location. This setting must to point to the version of MATLAB to be used for performing the optimization. The platform (32/64 bit or Linux) of the specified version of MATLAB must match the platform of this application.

MATLAB Optimization Setup

MATLAB optimization starts by creating an optimization and selecting MATLAB from the optimizer drop-down menu. If you select MATLAB as the optimizer, the *Setup Optimization* dialog box displays a **Setup...** button.



Select **Setup...** to open the *MATLAB Options* dialog box.



The upper text panel is informative. The Script section drop-down menu lets you select a lower panel display for Optimization algorithm, Options, or the Full script template.

This screen 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 our application and perform the optimization, and a lot of that code is unimportant to users. We have addressed this issue by displaying a dropdown to let you view only the portion of code they are interested in 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, users can use a different MATLAB function for optimization. By default we use fmincon() which is a derivative based constrained optimization. By modifying this line, users 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 a multitude of 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
```

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. Click **Q3D Extractor**> **Optimetrics Analysis** > **Add Screening & Optimization**.

   The *Setup Optimization* dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **MATLAB** from the **Optimizer** drop-down menu. Selecting **MATLAB** enables the **Acceptable Cost** and **Noise** fields.
4. Click **Setup...** 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 box.
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 more detail, 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 Focus** 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.

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** check box 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 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.

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.

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.

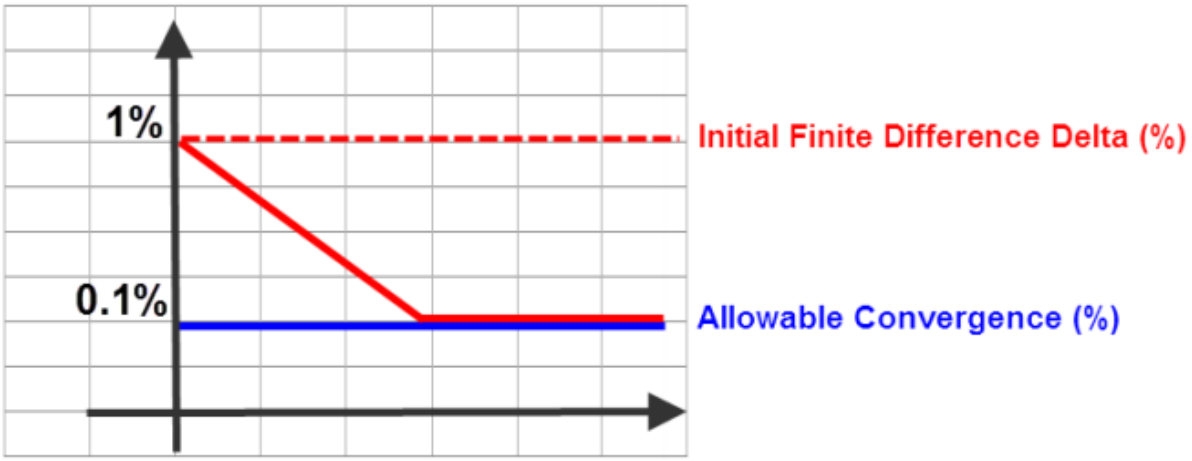
For NLPQL and MISQP, the default convergence rate, which is specified by the Allowable Convergence (%) property, is set to 0.1% for a Direct Optimization system. The maximum value for this property is 100%. This is computed based on the (normalized) Karush-Kuhn-Tucker (KKT) condition. This implies that 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.

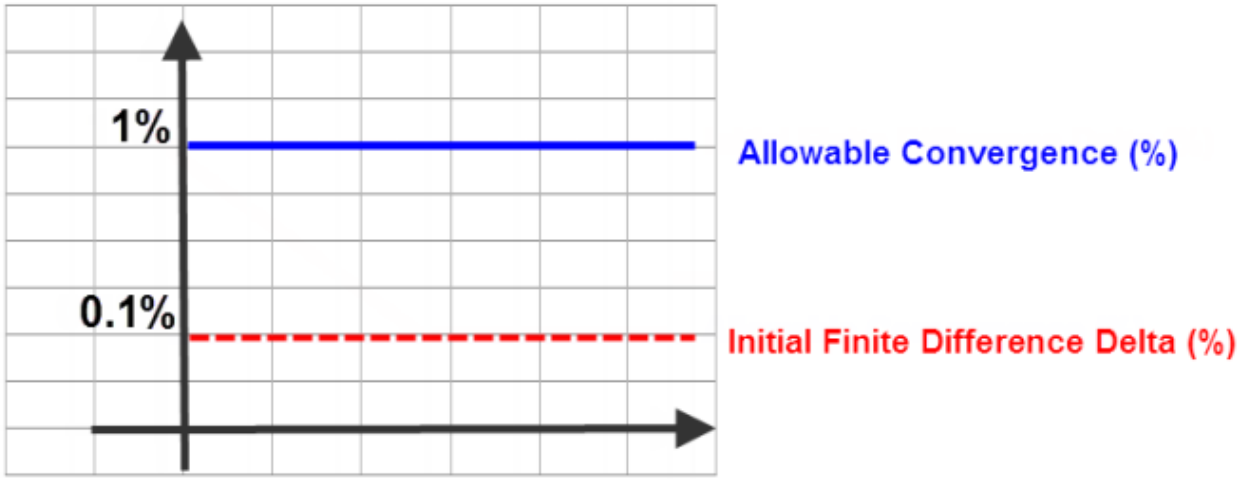
This property defaults to 1% for a Direct Optimization system. You use this property to specify a percentage of the variation between design points to ensure that the Delta use in the calculation of finite differences is large enough to be seen over simulation noise. The specified percentage is defined 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.

Setting Up Merit-based Sequential Quadratic Programming(Gradient) Optimizer

Following is the procedure for setting up an optimization analysis using the Merit-based Sequential Quadratic Programming(Gradient) Optimizer or MBSQ 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. Click **Q3D Extractor**> **Optimetrics Analysis** > **Add Screening & Optimization**.

   The *Setup Optimization* dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Merit-based Sequential Quadratic Programming(Gradient)** from the **Optimizer** drop-down menu.
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 box.
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** 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.
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. Use the **Options tab** if you want to enable use of a fast calculation-update algorithm to speed up Optimetrics and report updates during Optimetrics analyses, and to save the solution data for solved design variations in the analysis.

Setting Up Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete) Optimizer

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 Nonlinear 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

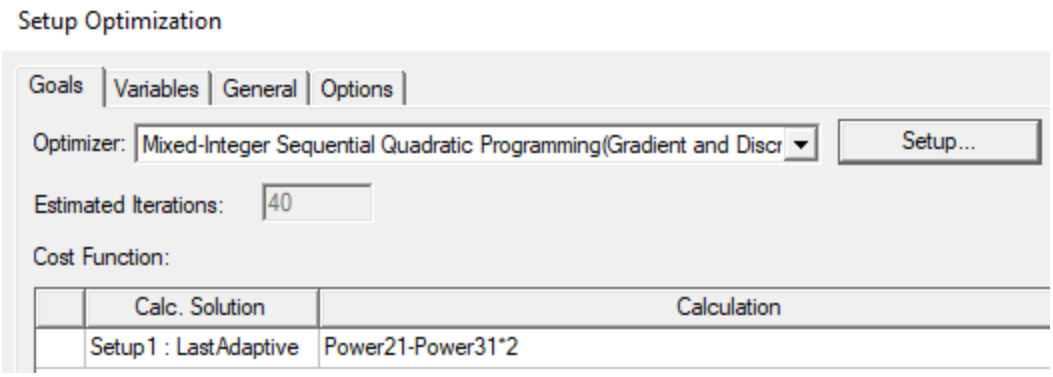
Lagrangian function is approximated by BFGS updates subject to the continuous and integer variables. MISQP is able to solve also non-convex nonlinear mixed-integer programs.

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.

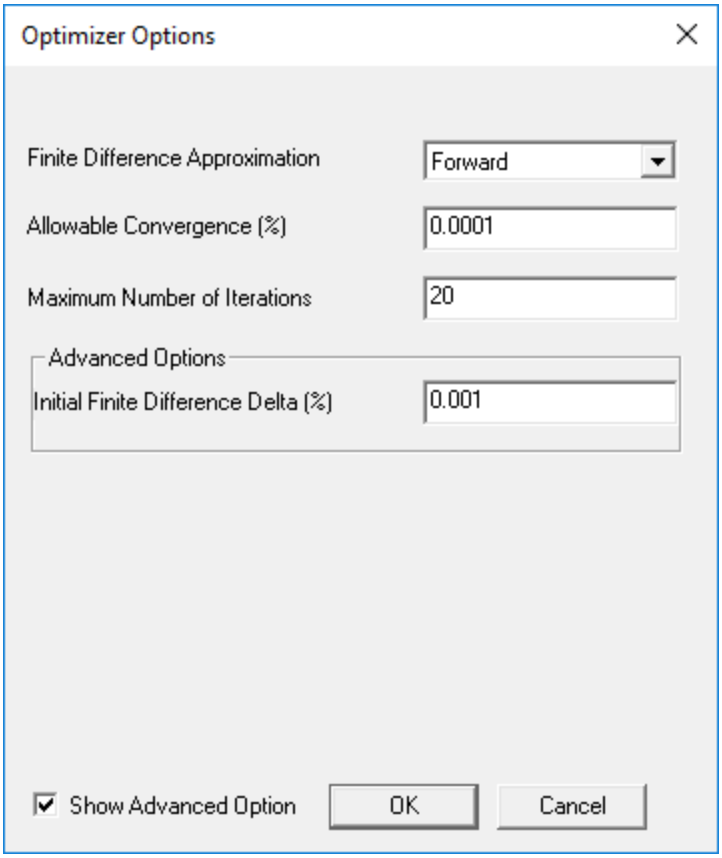
The Mixed-Integer Sequential Quadratic Programming (Nonlinear Programming by Quadratic Lagrangian) method (MISQP)can be used for Direct Optimization systems. It allows you to generate a new sample set to provide a more refined approach than the Screening method. MISQP is available for both continuous 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 **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & 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:

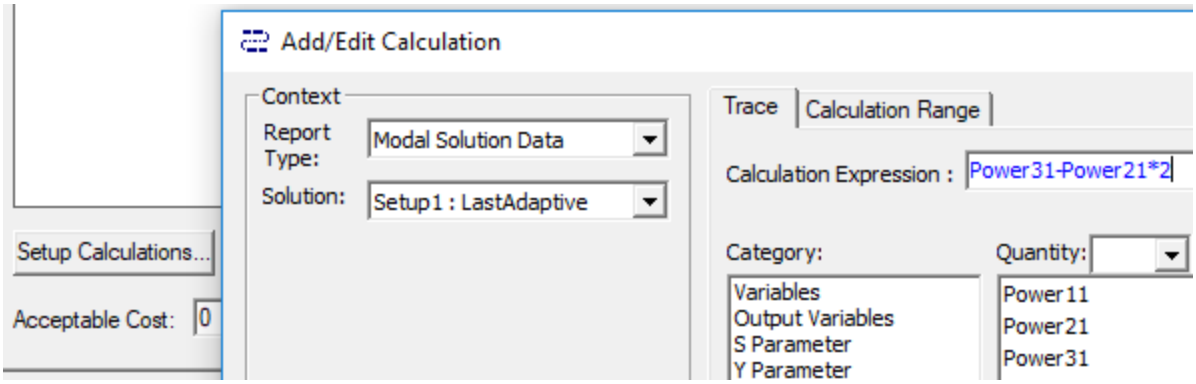
For **Central**: number of iterations * (2*number of inputs +1)

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 0.0001. 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.

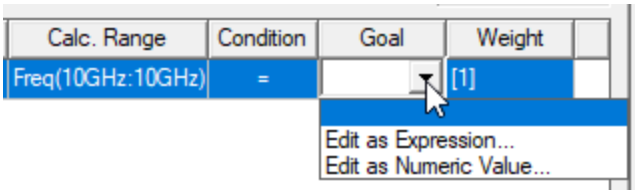
The default percentage value is 0.001. The minimum is 0.0001, and the maximum is 1. For parameters with Allowed Values set to **Manufacturable Values** or Snap to Grid, the value for Initial Finite Difference Delta (%) is ignored. In such cases, the closest allowed value is used to determine the finite difference delta.

5. Add a cost function by selecting the **Setup Calculations** button to open the *Add/Edit Calculation* dialog box.



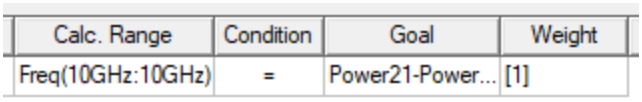
When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the *Add/EditCalculation* dialog box.

6. In the Optimization setup, in the drop-down menu, for the Goal column, select either Edit as Expression or Edit as Numeric Value...



This reopens the *Add/Edit Calculation* dialog box.

7. 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.



8. 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.

9. Optionally, set the Acceptable Cost and Cost Function Noise.
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.

- 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.

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** check box 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 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.

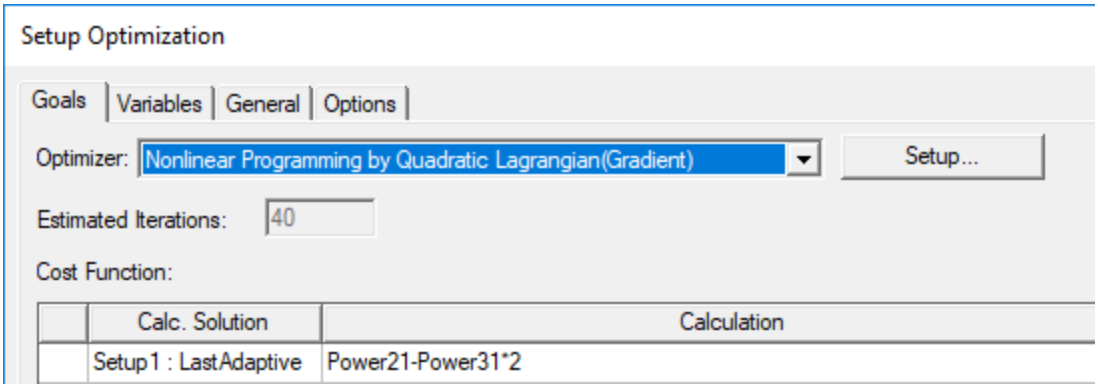
Setting Up Nonlinear Programming by Quadratic Lagrangian (Gradient) Optimizer

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.

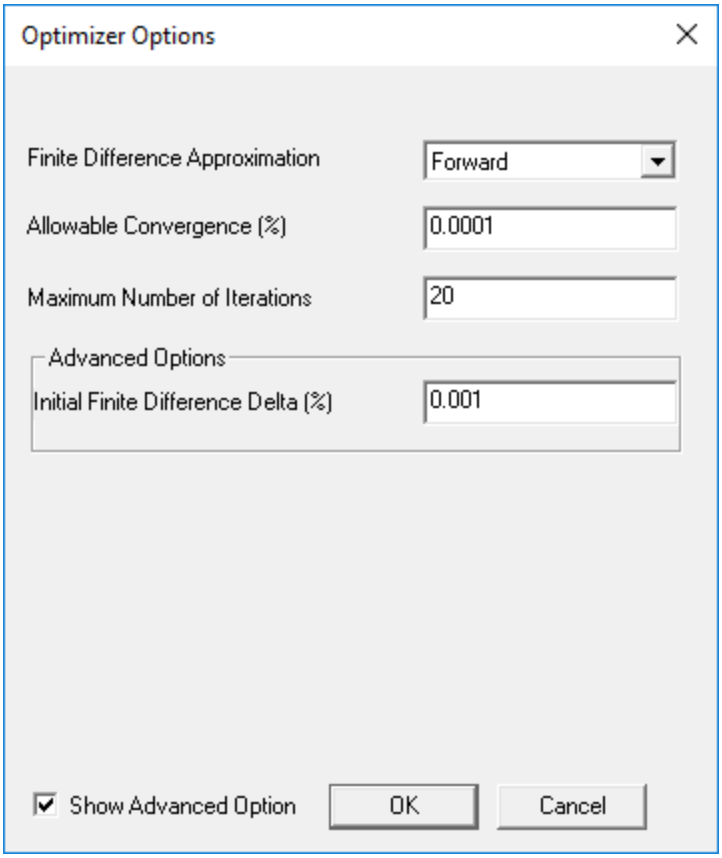
The NLPQL (Nonlinear Programming by Quadratic Lagrangian) method can be used for Direct Optimization systems. It allows you to generate a new sample set to provide a more refined approach than the Screening method. Available for continuous input parameters only, NLPQL can handle only one output parameter 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 **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & 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:

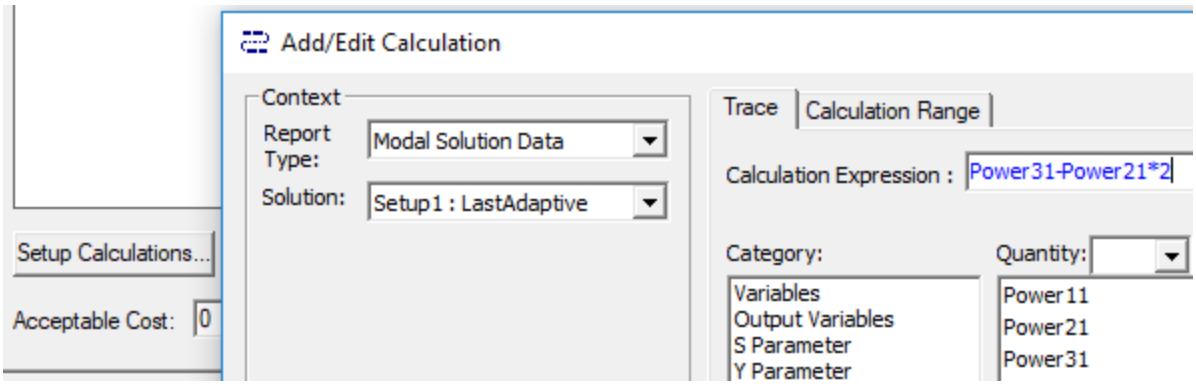
For **Central**: number of iterations * (2*number of inputs +1)

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.

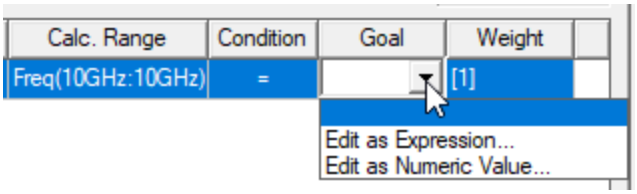
The default percentage value is 0.001. The minimum is 0.0001, and the maximum is 1. For parameters with Allowed Values set to **Manufacturable Values** or Snap to Grid, the value for Initial Finite Difference Delta (%) is ignored. In such cases, the closest allowed value is used to determine the finite difference delta.

5. Add a cost function by selecting the **Setup Calculations** button to open the *Add/Edit Calculation* dialog box.



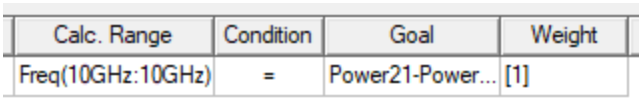
When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the *Add/EditCalculation* dialog box.

6. In the Optimization setup, in the drop-down menu for the Goal column, select either Edit as Expression or Edit as Numeric Value...



   This reopens the *Add/Edit Calculation* dialog box.

7. 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.



8. 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.

9. Optionally, set the Acceptable Cost and Cost Function Noise.
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.
    - 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.

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** check box 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 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.

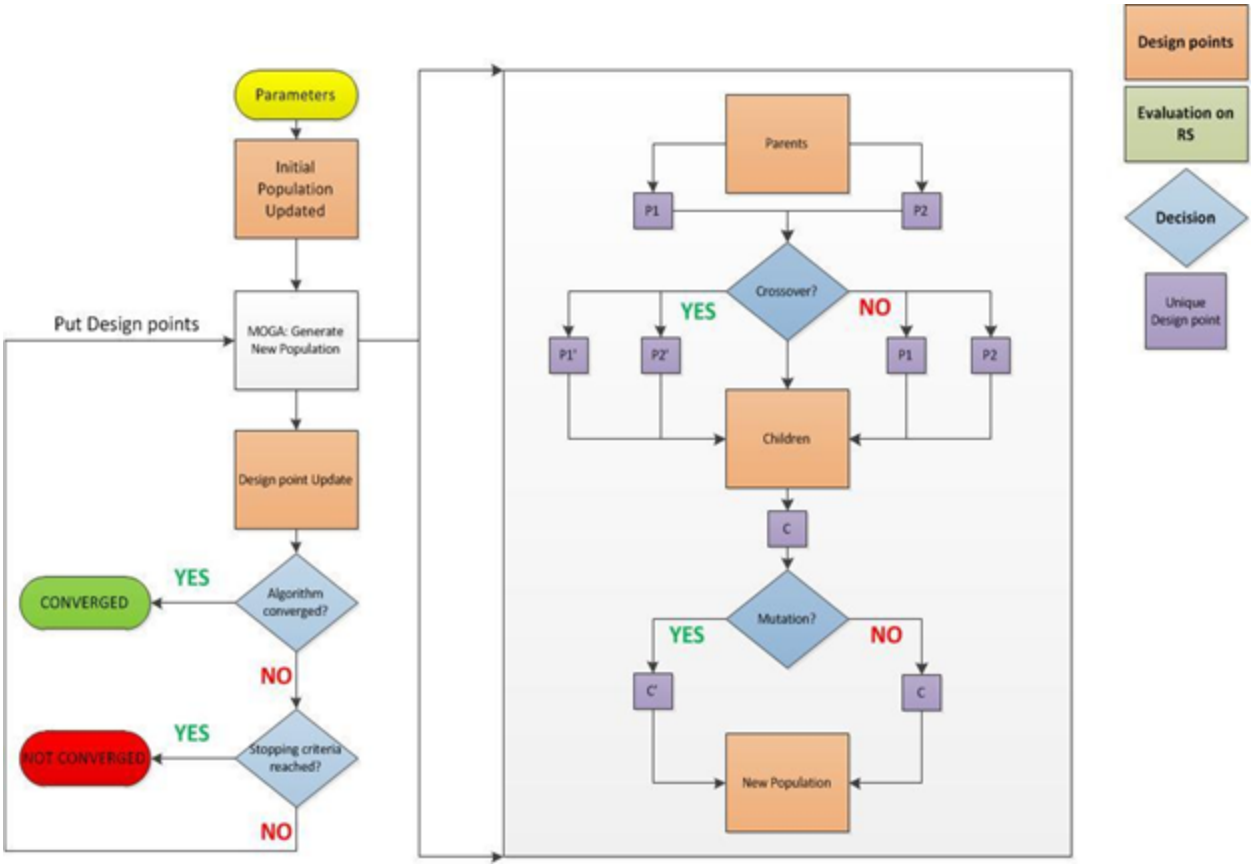
Multi-Objective Genetic Algorithm (MOGA)

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

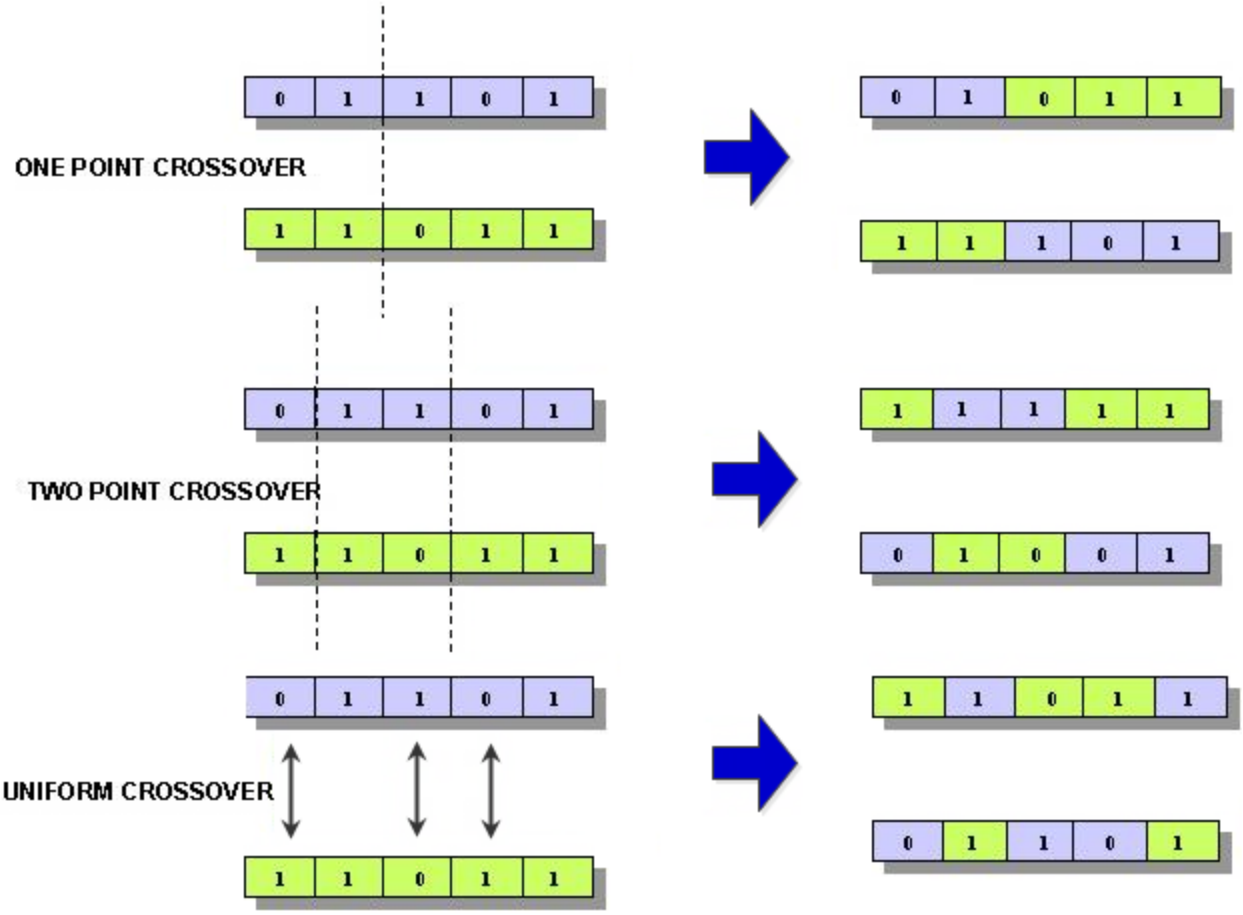
  A one-point cross-over operator that randomly selects a cross-over point within a chromosome then interchanges the two parent chromosomes at this point to produce two new offspring.

- 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.

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

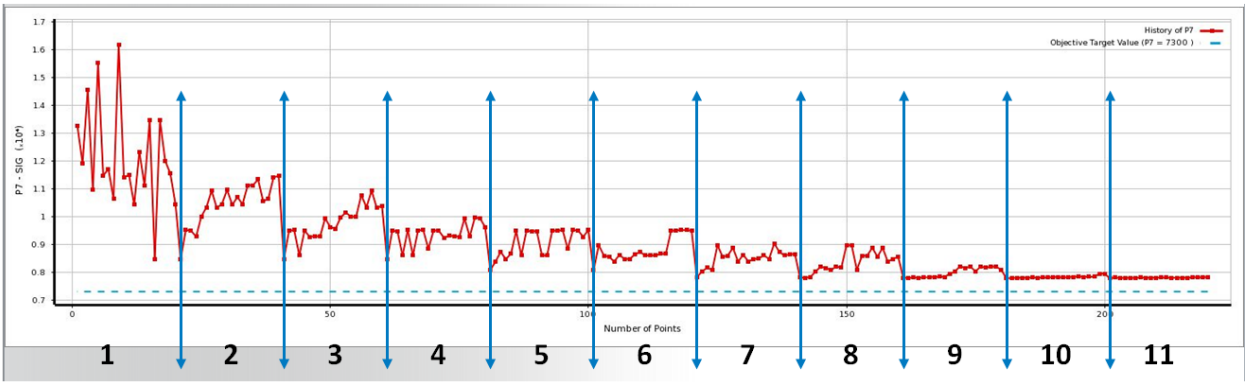
The **Maximum Allowable Pareto Percentage** criterion looks for a percentage that represents a specified ratio of Pareto points per Number of Samples Per Iteration. When this percentage is reached, the optimization is converged.

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

Setting Up Multi-Objective Genetic Algorithm(Random Search) Optimizer

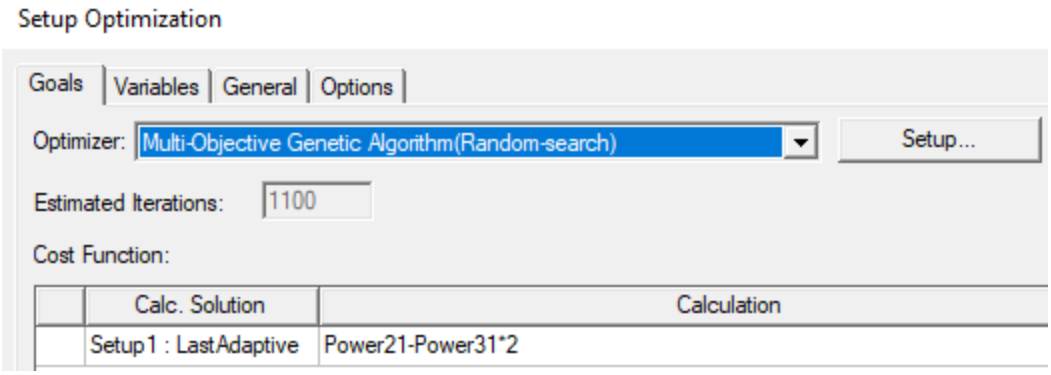
Following is the procedure for setting up an optimization analysis using the Multi-Objective Genetic Algorithm (Random Search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

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

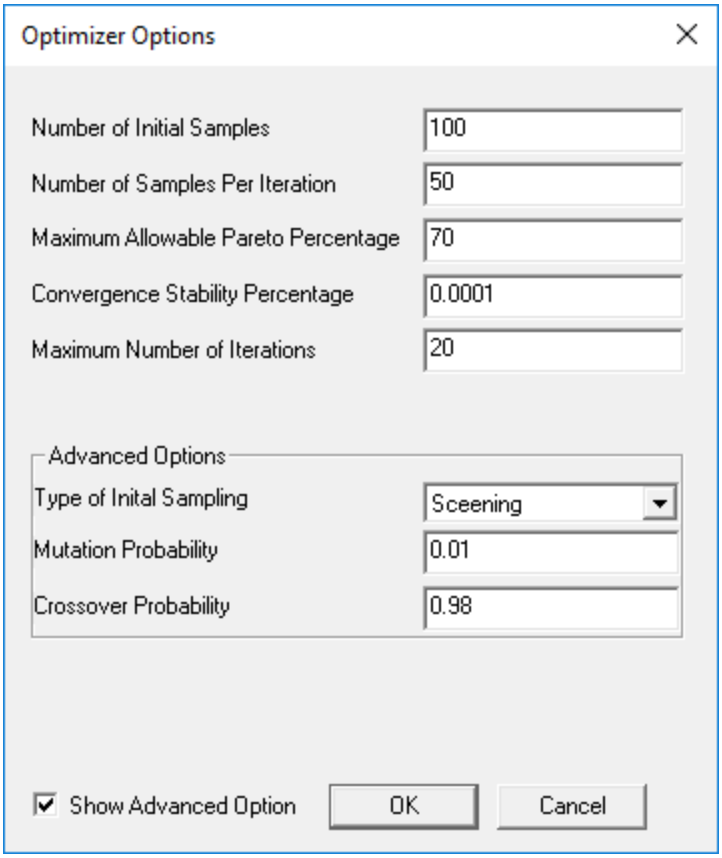
method that is an order of magnitude faster than traditional Pareto ranking methods. The constraint handling uses the same non-dominance principle as the objectives. Therefore, penalty functions and Lagrange multipliers are not needed. This also ensures that the feasible solutions are always ranked higher 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 **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & 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.

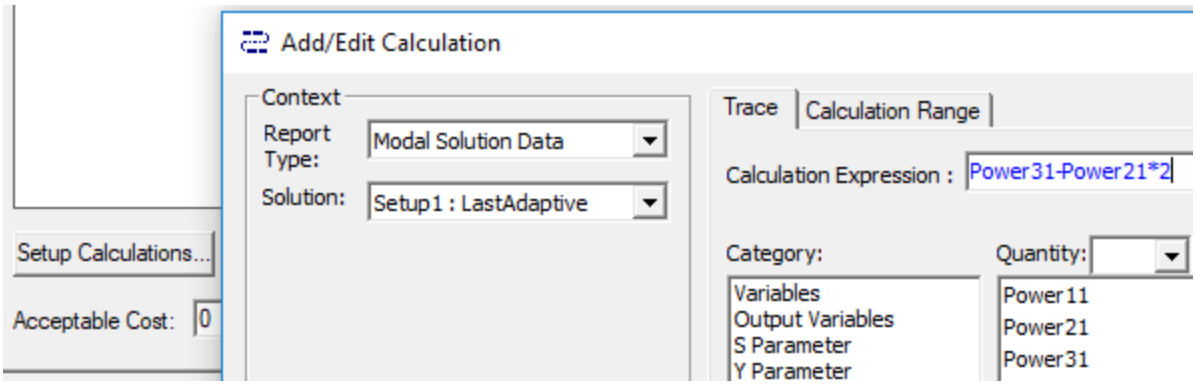
- **Maximum Allowable Pareto Percentage**: Convergence criterion. Percentage value that represents the ratio of the number of desired Pareto points to the number of samples per iteration. When this percentage is reached, the optimization is converged. For example, a value of 70 with Number of Samples 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 0.0001. 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.

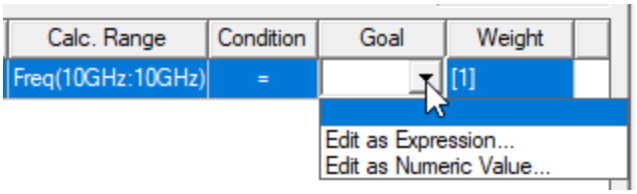
- **Crossover Probability**: Advanced option for specifying the probability with which parent solutions are recombined to generate offspring solutions. The value must be between 0 and 1. A smaller value indicates a more stable population and a faster (but less accurate) solution. If the value is 0, the 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 box.



When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

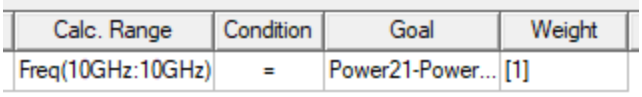
6. In the Optimization setup, in the dropdown for the Goal column, select either **Edit as Expression...** or **Edit as Numeric Value...**.



This reopens the *Add/Edit Calculation* dialog box.

7. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box.

This enters the expression/value into the **Goal** column.



8. 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.
9. Optionally, set the Acceptable Cost and Cost Function Noise.
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.
    - 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.
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** check box 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 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.

Pattern Search (Search-Based)

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.

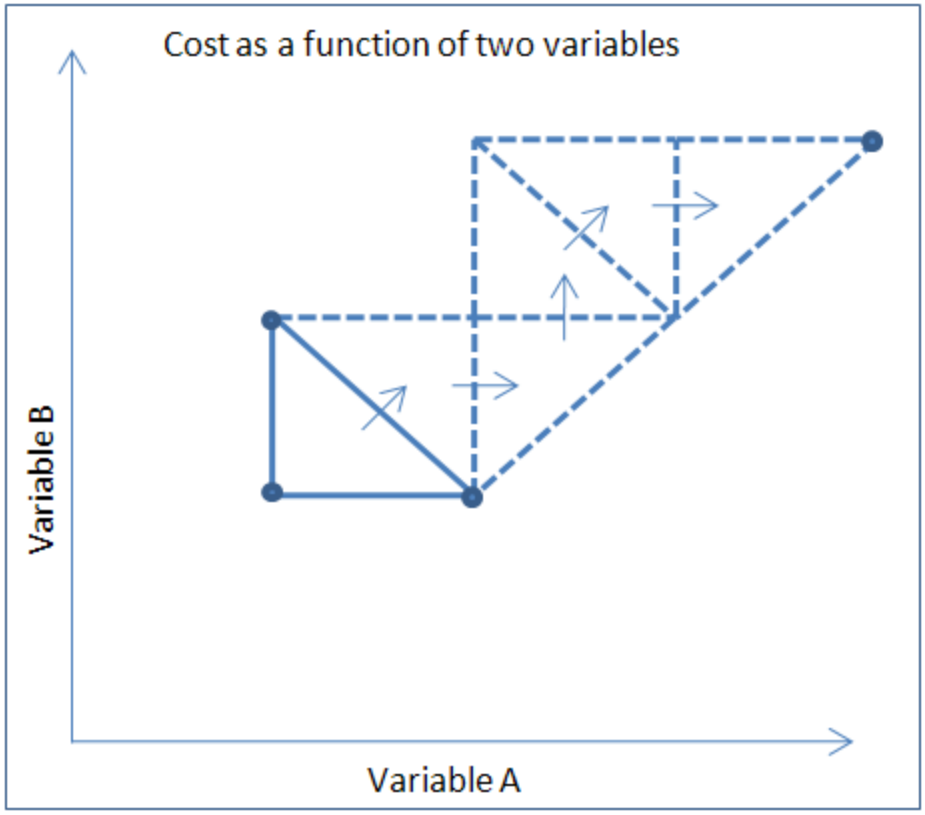
The cost value is calculated at the vertices of the simplex. The optimizer mirrors the simplex across one of its faces based on mathematical guidelines and determines if the new simplex provides better results. If it does not produce a better result, the next face is used for mirroring and the pattern 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 illustrate 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.

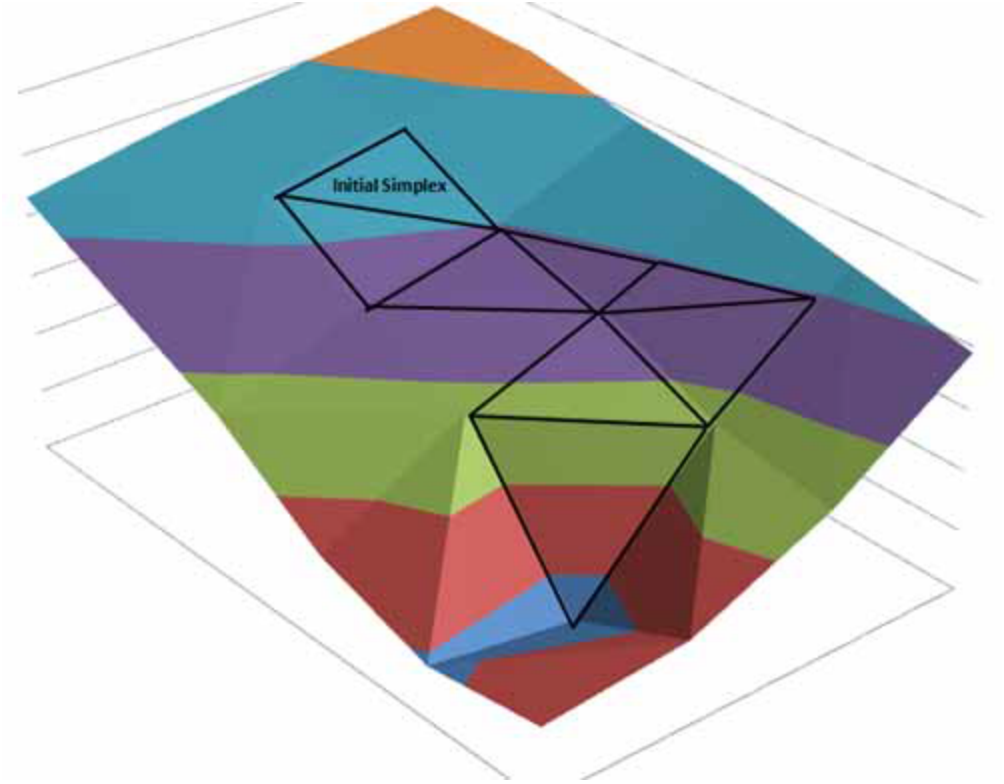
In the case of non-gradient search-based optimization algorithms, such as "pattern search," which are entirely based on function evaluations, one could see drastic changes in the function evaluations depending on how nonlinear the cost function is. This could seem misleading as if the algorithm did not 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 no 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. 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.

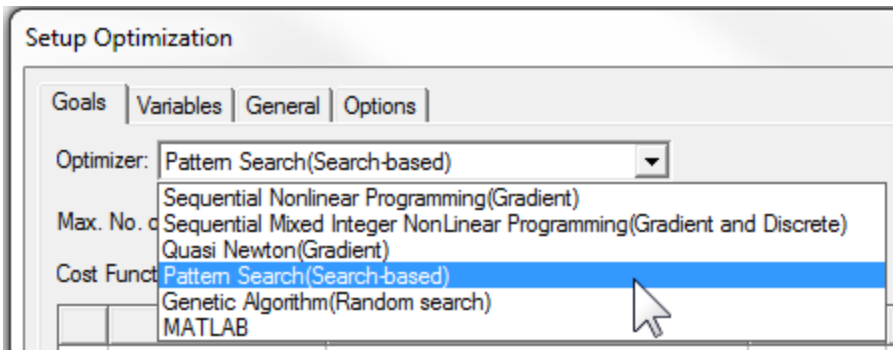
Optimization Setup for the Pattern Search (Search-based) Optimizer

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. Click **Q3D Extractor**> **Optimetrics Analysis** > **Add Screening & Optimization**.

   The *Setup Optimization* dialog box appears.

3. Under the **Goals** tab, select the optimizer by selecting **Pattern Search (Search-based)** from the **Optimizer** drop-down menu.



   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 box.
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.

Quasi Newton (Gradient)

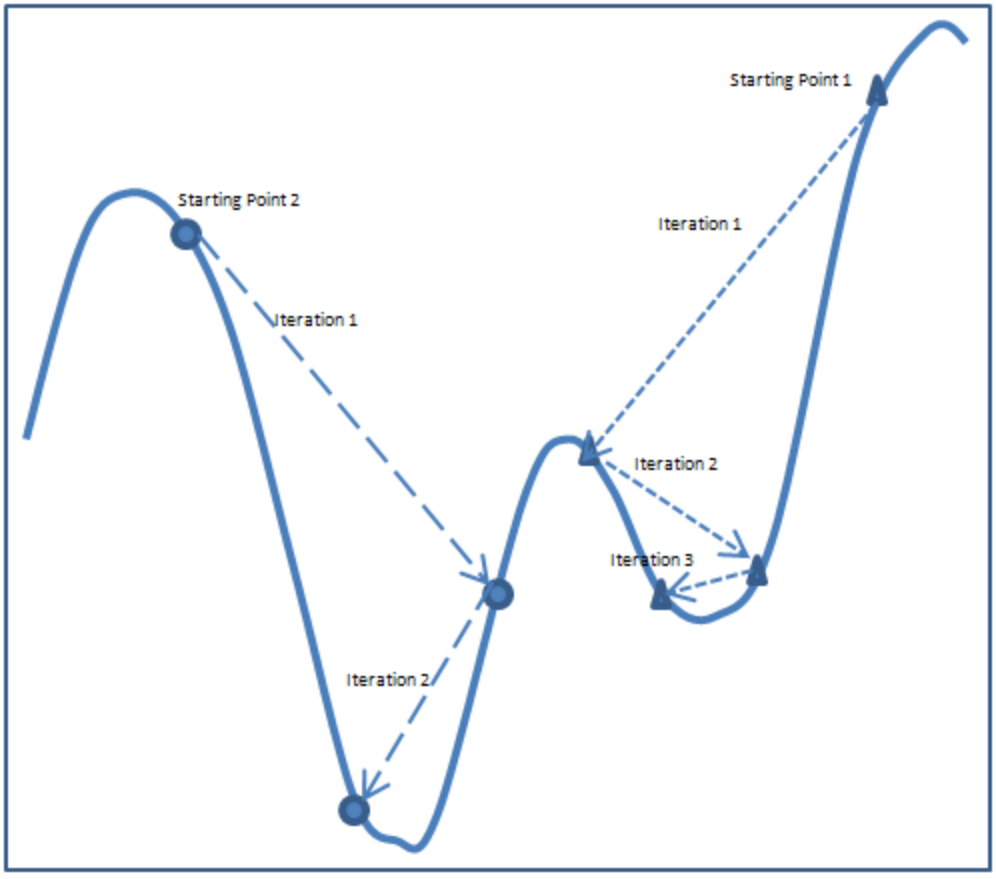
If the Sequential Non Linear Programming Optimizer has difficulty, and if the numerical noise is insignificant during the solution process, use the Quasi Newton optimizer to obtain the results. The Quasi Newton optimizer works on the basis of finding a minimum or maximum of a cost function which relates variables in the model or circuit to overall simulation goals. The user marks one or more variables in the project and defines 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 approximated 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.

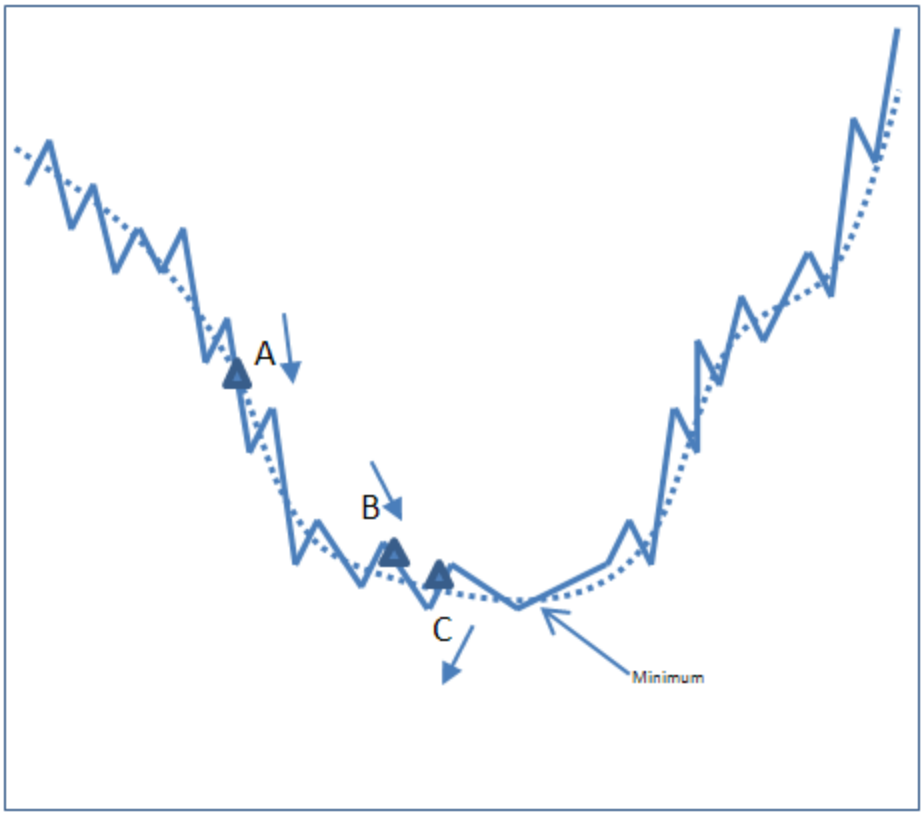
In the Quasi-Newton optimization procedure, the gradients and Hessian are calculated numerically. Essentially, the change in x and the change in the gradient are used to estimate the Hessian for the next iteration. The ratio of the change in cost to the change in the values of x provides the gradient, whereas, 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.

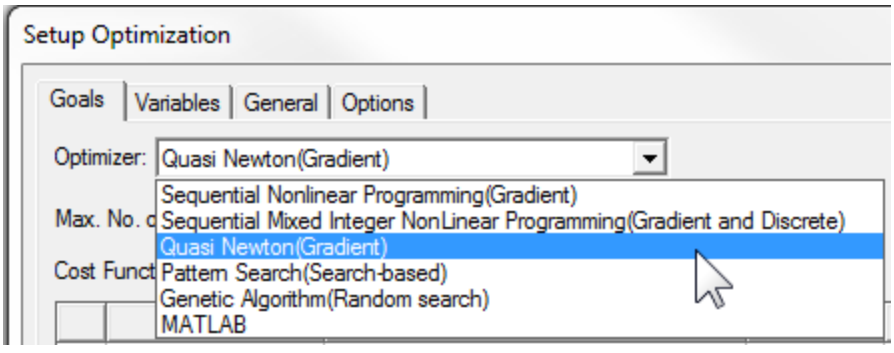
Optimization Setup for the Quasi Newton (Gradient) Optimizer

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. Click **Q3D Extractor**> **Optimetrics Analysis** > **Add Screening & Optimization**.

   The *Setup Optimization* dialog box appears.

3. Under the **Goals** tab, select the optimizer by selecting **Quasi Newton (Gradient)** from the **Optimizer** drop-down menu. Selecting Quasi Newton (Gradient) 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 box.
6. Type the value of the cost function at which the optimization process should stop in the **Acceptable Cost** text box. Note that for Quasi Newton, if you specify 0 as the acceptable cost, the simulation stops after the first analysis.
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.

Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete)

The Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) optimizer is equivalent to the SNLP (Gradient) 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.

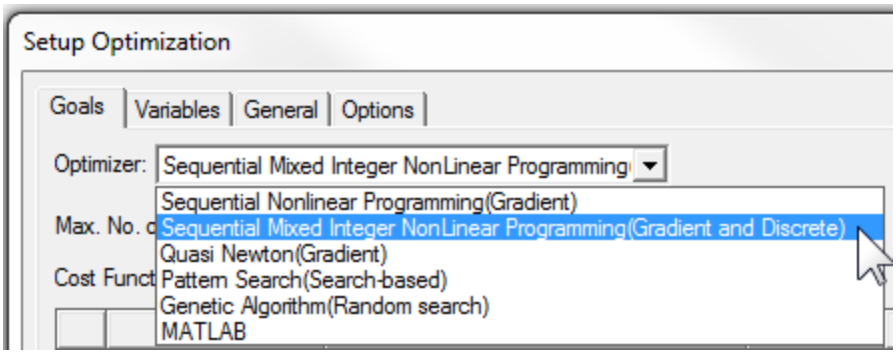
For example, consider N to be an integer optimization variable. By definition it can only assume integer values. You can establish another variable, which further depends on this one: K = 2.345 * N, or K = sin(30 * N). This way K has a discrete value, but is not necessarily integer. Or, one can use N directly as a design parameter.

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. 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. Click **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & 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** drop-down menu.



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 box.
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.
   - Check the Integer box for integer variables.
   - 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.

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.

Sequential Nonlinear Programming (Gradient)

The main advantage of SNLP (Gradient) over Quasi Newton (Gradient) 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 polynomial 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 as 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.

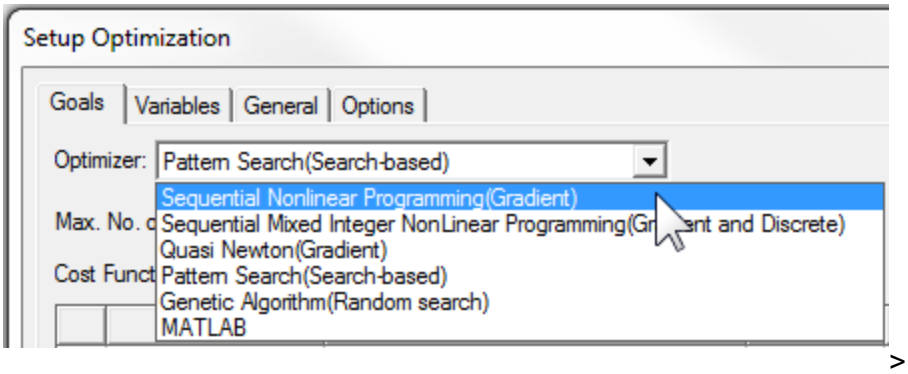
Most importantly, due to the separation of expensive and inexpensive evaluations in the cost calculation, the SNLP optimizer is more tightly integrated with the supporting FEA tools. This tight integration provides more insight into the optimization problem, resulting in a significantly faster optimization 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.

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. 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. Click **Q3D Extractor** > **Optimetrics Analysis** > **Add Screening & Optimization**.

   The *Setup Optimization* dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Sequential Nonlinear Programming (Gradient)** from the **Optimizer** drop-down menu.



>

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 box.
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, 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, 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** 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.

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 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.

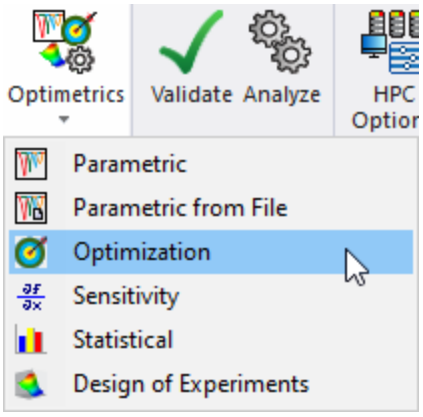
To provide a broad range of capability, Optimetrics incorporates the following types of numerical optimizers:

- Screening (Search based)
- Multi-Objective Genetic Algorithm
- Nonlinear Programming by Quadratic Lagrangian (Gradient)
- Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete)
- Adaptive Multiple Objective (Gradient)
- Adaptive Single Objective
- MATLAB

Legacy Optimizers include

- Sequential Nonlinear Programming (Gradient)
- Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete)
- Quasi Newton (Gradient)
- Pattern Search (Search-based)
- Genetic Algorithm (Random search)

Click on the links above to view the setup procedure for each optimizer. Options for the analysis are listed in the table. Besides setting up an Optimization analysis from the Optimization menu, you can also the **Simulation** tab of the ribbon, and select from the menu under the Optimetrics icon:



The following *optional* optimization solution setup options can also be used:

- Modify the starting variable value.
- Edit the Calc. Range text field or use the Edit Calculation Range dialog box.
- 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 (Gradient) and Pattern Search (Search based optimizers), **Variables** tab).
- Set the minimum and maximum focus size (for the SNLP Gradient and SMINLP Gradient and Discrete 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)
- Open the HPC and Analysis Options window.

> **Note:**
>
> Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

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 **Sequential Nonlinear**

Programming (Gradient), **Sequential Mixed Integer NonLinear Programming**, **Quasi Newton (Gradient)**, or **Pattern Search (Search-based) 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.

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.

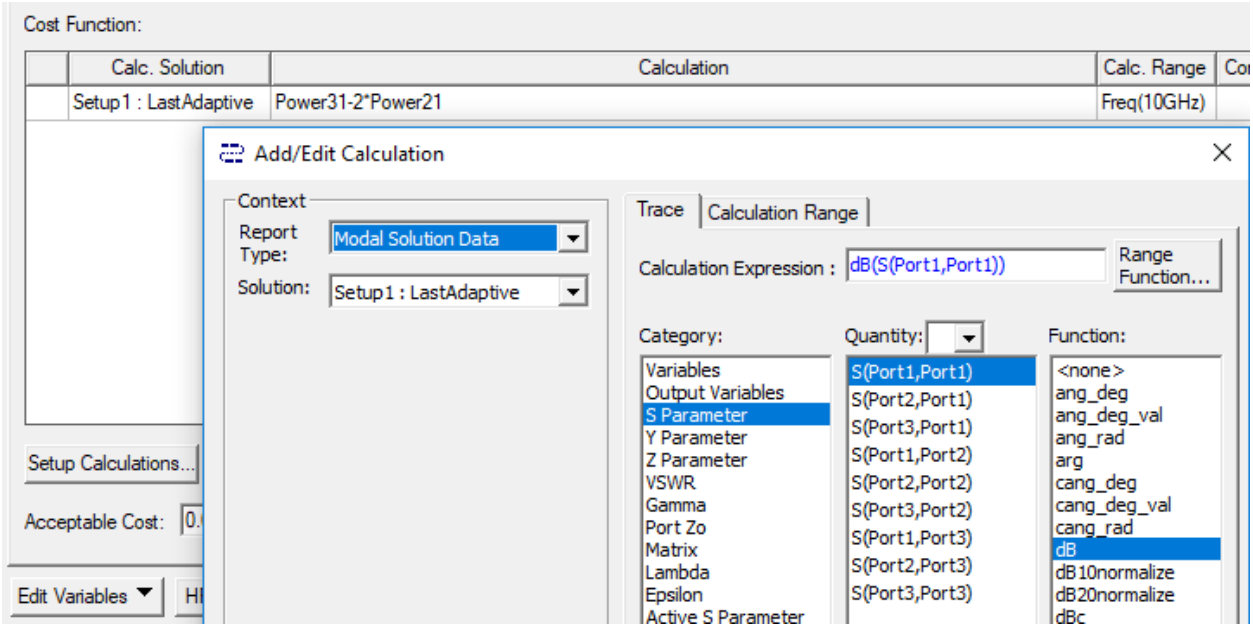
> **Note:**
>
> To revert to the default starting value, clear the **Override** check box.

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. For example, if you vary a design to find the maximum transmission from *Wave Port 1* to *Wave Port 2* (S_{21}=>1), define the cost function to be -mag(S (WavePort2,WavePort1)).

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.



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.

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

$$c = 10000 \cdot |S_{11}|^2$$

where $|S_{11}|$ is the magnitude of the reflection coefficient, at the minimum, $|S_{11}|$ is expected to be very small, $|S_{11}| \approx 0$.

From the solution setup, the error in $|S_{11}|$ is expected to be $E_{S11} \approx 0.01$. The perturbed cost function is therefore

$$c_{perturbed} = 10000 \cdot (|S_{11}|_{min} + E_{S11})^2$$

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.

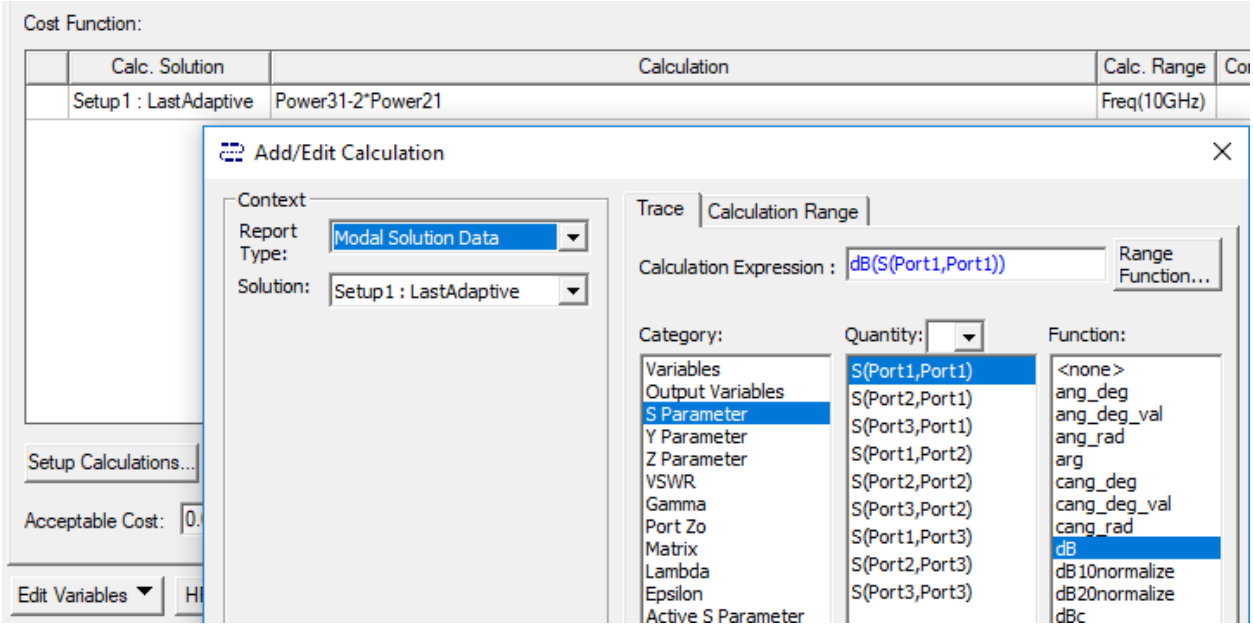
Adding a Cost Function

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* or *Design of Experiments* 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.

> **Note:**
>
> Because Optimetrics works in SI values, you should use ang_deg_val or cang_deg_val functions which return unitless numbers in degrees rather than ang_deg or cang_deg, which return angular values in degree units but evaluate in radians in Optimetrics expressions. For example, you could Optimize successfully by changing the expression from abs(**cang_deg**(S(Port2,Port1))) to abs(**cang_deg_val**(S(Port2,Port1))). See the table of available functions and definitions.

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 drop-down menu.
4. To edit the calculation on which to base the cost function goal, select **Edit** from the drop-down menu.
5. In the **Condition** text box, click one of the following conditions from the drop-down menu:

<=	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.

When Minimize is used as an optimization condition, the value of calculation is used as the cost (there is no target value to compare to). For maximize, the negative of calculation value is used as cost.

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 and set the **Goal Values** to expressions.

8. 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.

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 drop-down menu that displays all of the defined output variables. Click an output variable from the drop-down menu.

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** drop-down menu, 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** drop-down menu. 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** drop-down menu, 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.

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 drop-down menu 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**.

The most recently created output variable is entered 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.**
<=	$e_i = \begin{cases} 0 & s_i \le g_i \\ s_i - g_i & s_i > g_i \end{cases}$
=	$e_i = \lvert s_i - g_i \rvert$
>=	$e_i = \begin{cases} 0 & s_i \ge g_i \\ g_i - s_i & s_i < g_i \end{cases}$

If the total error value is within the acceptable cost, the optimization stops.

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 drop-down menu in the **Cost Function Norm Type** field. The options are **L1**, **L2**, and **Maximum**. **L2** is the default.

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 cost function computes the weighted sum of the squares of individual goal errors.

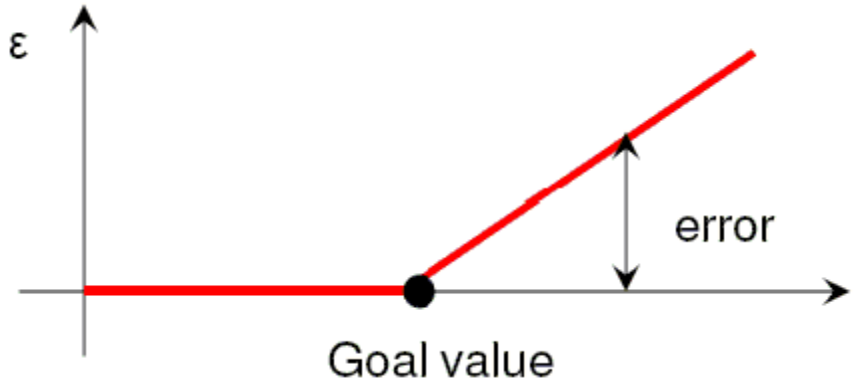
$$Cost = \sum_{1}^{N} w_i \varepsilon_i^2$$

For the **Maximum** norm, the cost function uses the maximum among all the weighted goal errors, which means that cost is always less than zero:

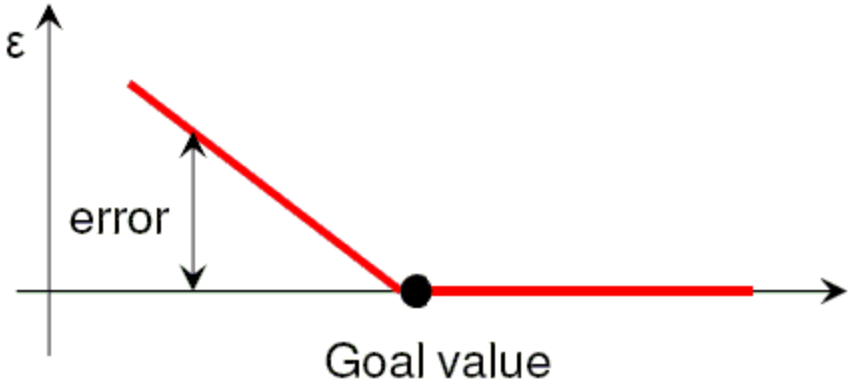
$$Cost = \underset{1}{\overset{N}{Max}} W_i \cdot \varepsilon_i$$

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.

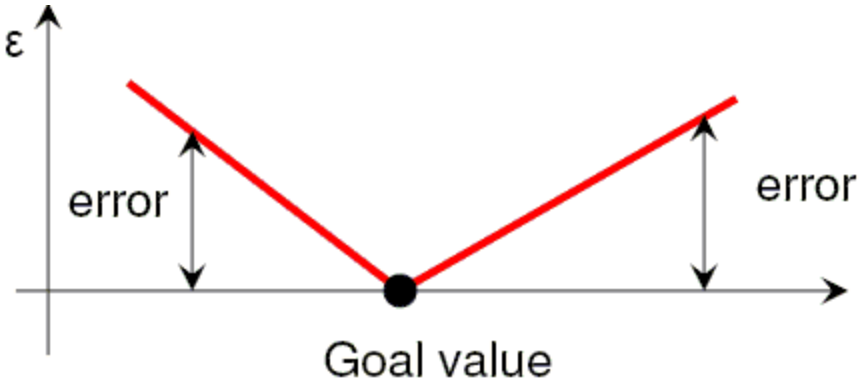
The graphical representation of the error is possible and depends upon the actual condition being used. If a "<" condition is used, the error can be represented as below:



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:

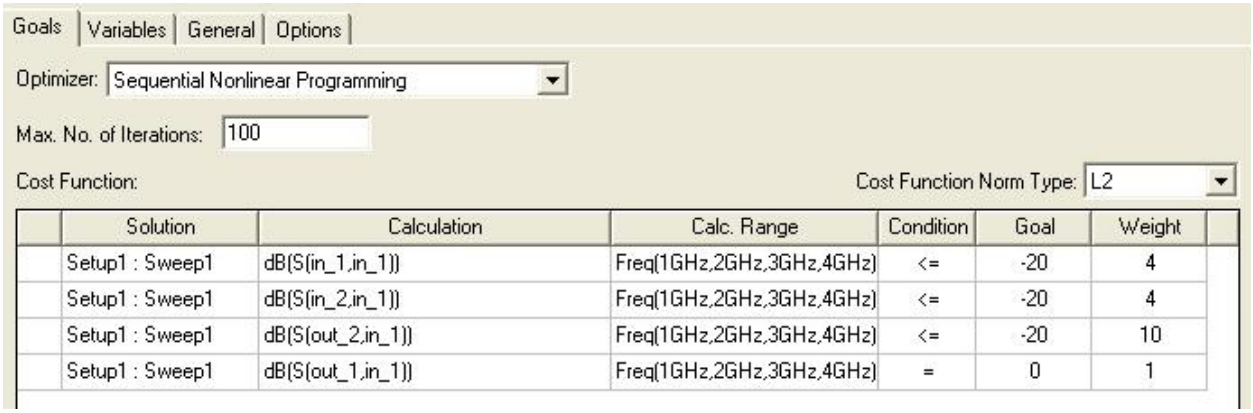


The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios. Note that when using "minimize" or "maximize" settings for the condition there should be a single goal setting which in this case coincides with the cost function.

Example of a More Complex Cost Function

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

As an example of a more sophisticated cost function, consider the figure. It belongs to a connector simulated in HFSS with more than four ports.



Solution	Calculation	Calc. Range	Condition	Goal	Weight
Setup1 : Sweep1	dB(S(in_1,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	<=	-20	4
Setup1 : Sweep1	dB(S(in_2,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	<=	-20	4
Setup1 : Sweep1	dB(S(out_2,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	<=	-20	10
Setup1 : Sweep1	dB(S(out_1,in_1))	Freq(1GHz,2GHz,3GHz,4GHz)	=	0	1

The cost function given here concentrates only on a signal sent into port in_1. Suppose the specifications to be met are: reflection, backward cross talk and forward cross talk all smaller than or equal to -20 dB, of which the forward cross talk is the most important.

The first three entries in the cost function enforce those specifications, with the weight for the forward cross talk being a larger number than the other weights. The actual values for the weights are somewhat arbitrary and serve as examples only. For this cost function, as long as specifications are not met, the optimizer puts the most effort in getting the forward cross talk close to its specification. Once the three specifications have been satisfied, their contributions to the cost function become zero, and only the fourth entry remains. Remember that the connector has more than four ports, so satisfying the given specs does not guarantee maximum transmission.

The fourth line tries to maximize the transmission by asking for S(out_1, in_1) to be 0 dB. That will never be reached, but its presence forces the optimizer to improve the connector a bit beyond the specifications.

The cost function norm type specifies how the four lines are combined into one cost function with one value. With L1 and L2, all four contribute simultaneously, rather than only the largest of the four at any one time.

Setting the Min. and Max. Variable Values for Optimization

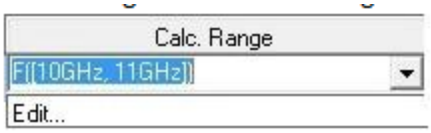
For every optimization setup, Optimetrics automatically sets the minimum and maximum values it will consider for a variable being optimized. Optimetrics sets a variable's minimum value equal to approximately 50% of its starting value. (The starting value is the variable's current value set for the nominal 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 Optimetrics analysis.

Edit Calculation Range

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

In the *Setup Optimization* dialog box, you can enter the Calc. Range Sweep Min/Max by directly editing the Calc. Range field or by accessing an *Edit Calculation Range* dialog box.



The edit field accepts the following forms of text:

- sweep that allows your 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)

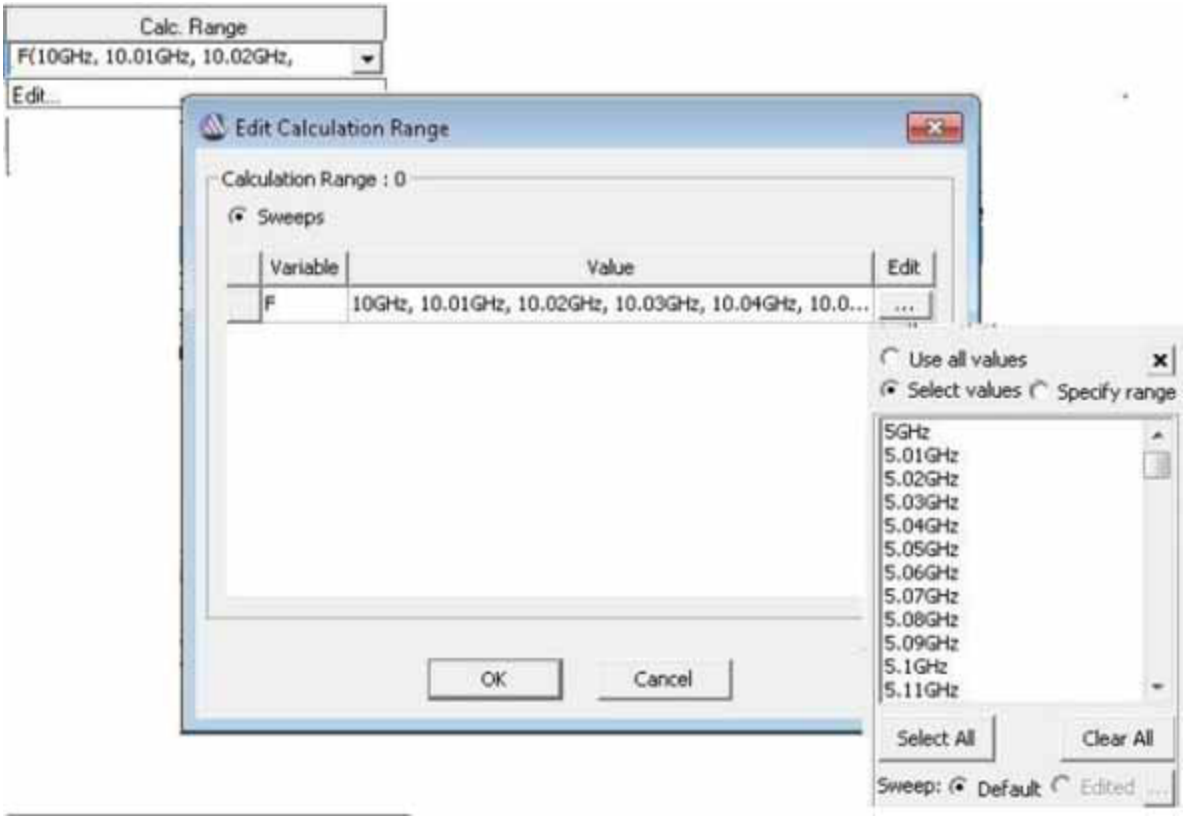
  min/max range, for example, Time([1ms, 2ms]) HFSS/MAXWELL/TWINBUILDER

- 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 comma ","

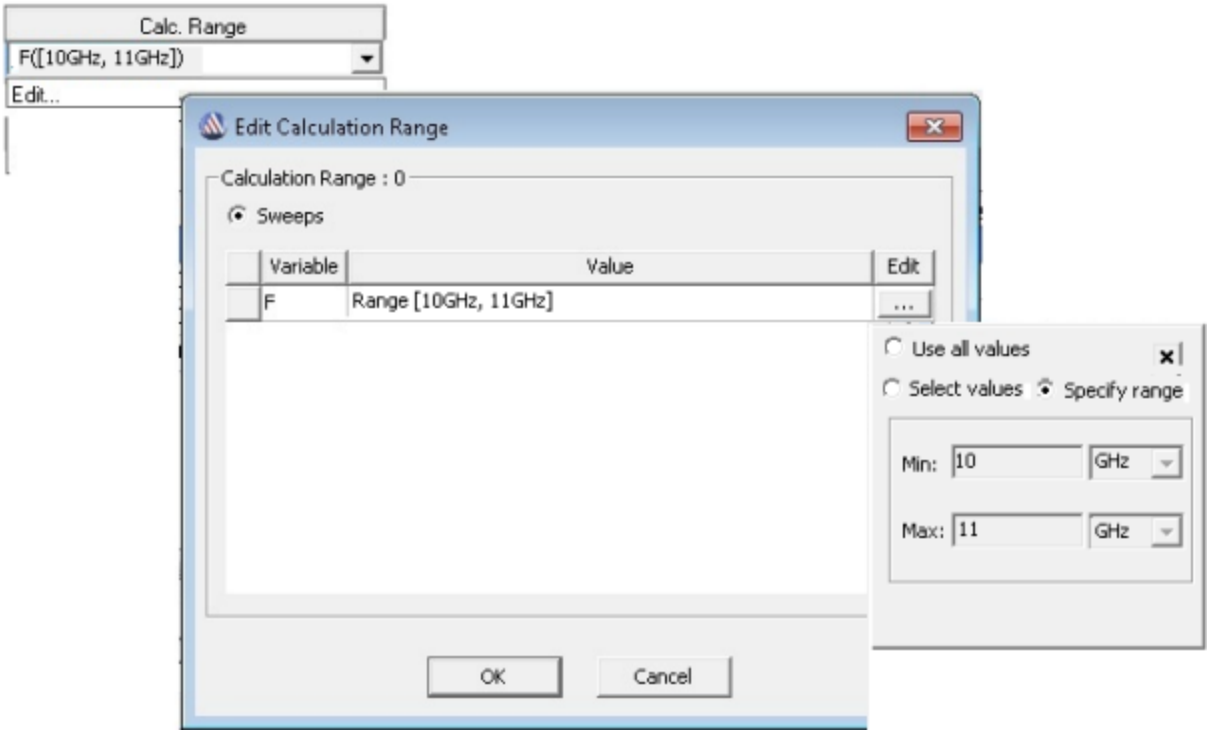
  For example, F(1GHz), cap(1pf, 1.2pf)

  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 radio buttons for Use all values, Select Values or Specify range.



This example shows that when you specify a range, how the range appears in the Calc. Range field.



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, indicating 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 **Q3D Extractor**> **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.

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.

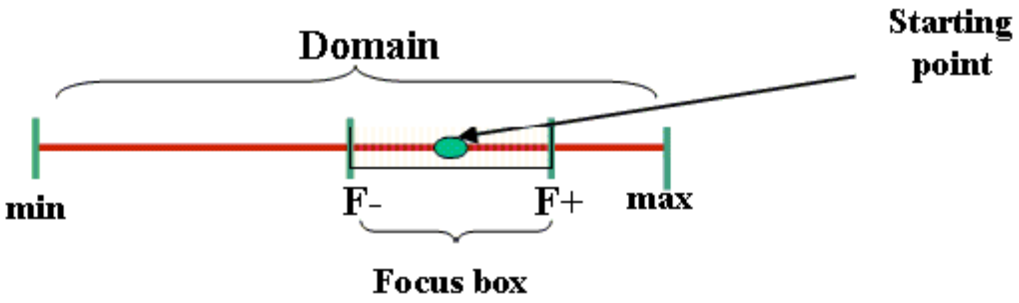
1. In the *Setup Optimization* dialog box, click the **Variables** tab.
2. Optimetrics displays **Min Step** and **Max Step** columns, with default values for each variable to be optimized.
3. In the **Min Step** text box, type the minimum step size value. Optionally, modify the unit system in the **Units** 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**.

> **Tip:**
>
> A value of zero is recommended for the minimum 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.

Setting 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.
3. 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** drop-down menu, 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.

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** drop-down menu, 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 **Edit** 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 box.

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.

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

> **Note:** Sensitivity Analysis has been placed in legacy mode because it is no longer under active development. Go to **Tools** > **Options** > **General Options** > **Miscellaneous** and toggle the **Enable Legacy Optimetrics Tools** check box to show or hide these legacy tools.

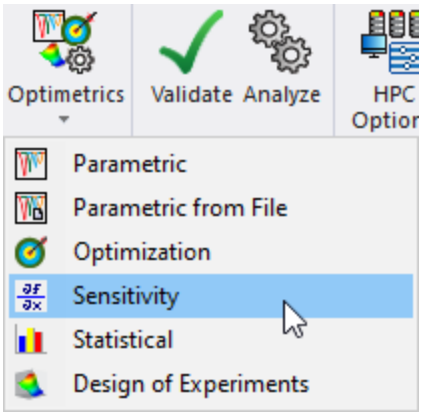
Selecting a Primary Output

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.

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 **Design Properties** dialog box.
2. Click **Q3D Extractor** > **Optimetrics Analysis** > **Add Sensitivity**  . You can also select the **Simulation** tab in the ribbon, and select **Parametric** from the drop-down menu under the Optimetrics icon:



The **Setup Sensitivity Analysis** dialog box appears.

3. Under the **Calculations** tab, type the maximum number of iterations per variable value that you want the software 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.

   Checking the Optional **Worst Case Analysis** option does an extreme value analysis that focuses on the upper and lower boundaries of all the analyzed parameters. Some setup is required before performing Worst Case Analysis.

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.

> **Note:**
>
> Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

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.

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 the output parameter is based on a swept variable, in the **Calculation Range** column, set the value of the variable at which the output parameter is to be computed.
7. 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.

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 box:
   - 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 **Add Calculation** 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.

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. If you specified that the solution be extracted from the last adaptive solution, Optimetrics uses the adaptive frequency defined in the solution setup.

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 **Edit** 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.

   Other examples of available variables include frequency, if you selected an S-parameter solution quantity; and phi or theta, if the solution quantity is a radiated field quantity.

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.

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.

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.

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.

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 **Q3D Extractor**> **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.

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.

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** drop-down menu.

   > **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.

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** drop-down menu.
3. Select **Solve the parametric sweep during analysis**.

Performing Worst Case Analysis

Two popular worst case analysis techniques can be implemented: extreme value analysis and Monte Carlo analysis.

Some setup is required before performing worst case analysis. First, identify uncertainties in design and create a local or project variable for each of them. Second, determine the variation range of each variable (Min and Max) – its statistical distribution is optional. Third, determine a measurement of performance, especially for extreme value analysis.

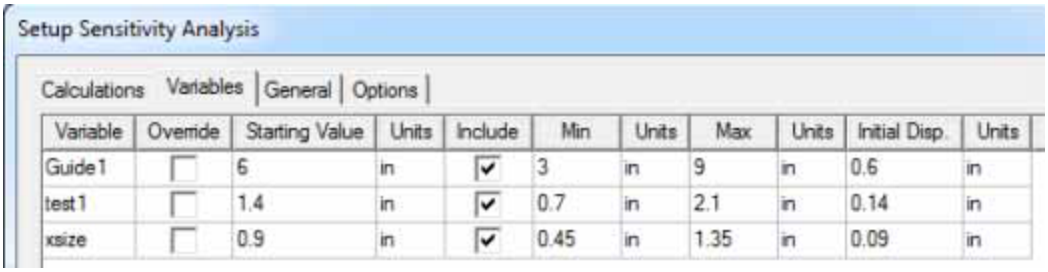
1. **Extreme value analysis:**

   This is one of the most popular methods to estimate worst-case performance. To start, a sensitivity analysis is performed. The results (sensitivities/first derivative) allow us to pick an extreme value (upper or lower bound) for each variable. The corresponding simulation result is used to predict upper and lower bound of performance. The assumption is that extreme performance is reached at boundary value (note that in certain cases, making such an assumption is not valid).

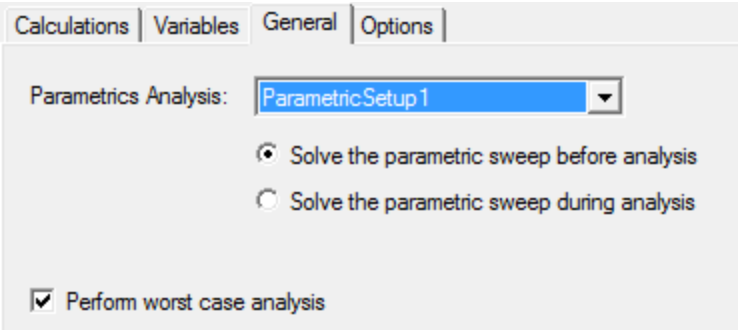
   To perform an extreme value analysis in Ansys Electronics Desktop, create a new analysis under **Optimetrics** > **Sensitivity**.

   - Setup:

     First, include variables in a sensitivity analysis. To do so, go to your list of variables, under Sensitivity, specify Min and Max values of each variable, and check 'Include'. The following figure shows an example with three variables included.



     Second, follow **Setting Up a Sensitivity Analysis** in the help. During this procedure, set your performance measurement in **Calculations** tab; check all variables in **Variables** tab; check **Perform worst case analysis** in **General** tab.



> **Note:**
>
> Checking **Perform worst case analysis** calculates 1st derivatives for each variable. If we have three variables and for Var1 first derivative is negative, Var2 1st derivative is positive, Var3 1st derivative is positive, then for Worst Case Analysis, we request two more variations:

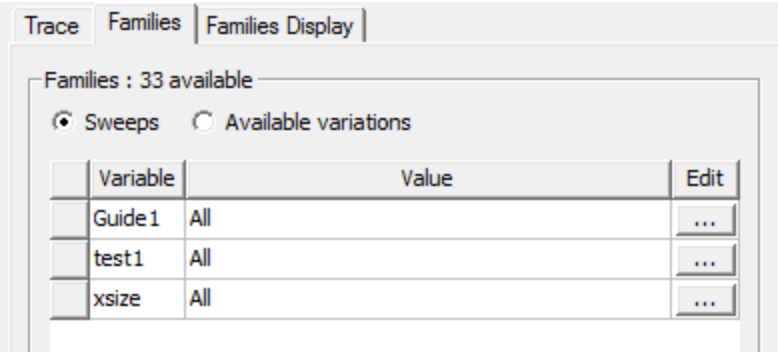
Var1@minimum, Var2@maximum, Var3@maximun

and Var1@maximum, Var2@minimum, Var3@minimum

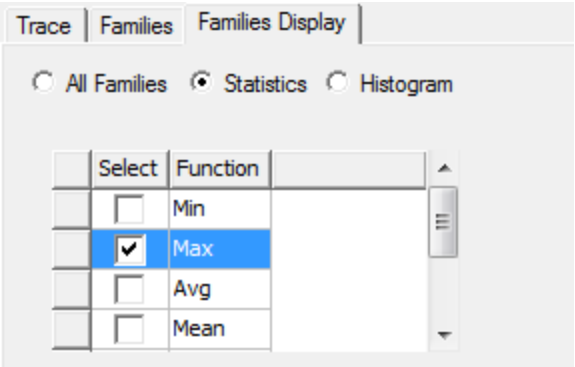
- After setup, analyze your Sensitivity Setup under Optimetrics.
- To view worst-case result (upper bound only):

  Create a Data Table report.

  > **Note:**
  >
  > Under the Context pane in the **Report** dialog box, select matching Solution and Optimetrics setup. Under the **Families** tab, change the setting to 'All' under 'Value' of each variables.



Under the **Families Display** tab, select **Statistics**and then check **Max**.



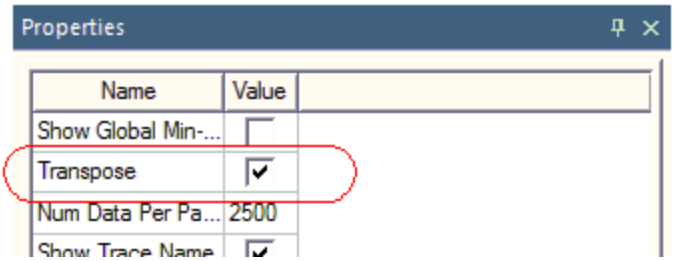
The associated data table shows the Max values.

	Freq [GHz]	dB(S(1:1,1:1)) Setup1 : Sweep Max
1	3.000000	-0.000000
2	4.000000	-0.000000
3	5.000000	-0.105323
4	6.000000	-0.248322
5	7.000000	-0.267220
6	8.000000	-0.202039
7	9.000000	-0.298840

To see the corresponding variable values, select 'All Families' under the **Families Display** tab, and locate the max value to see value of variables.

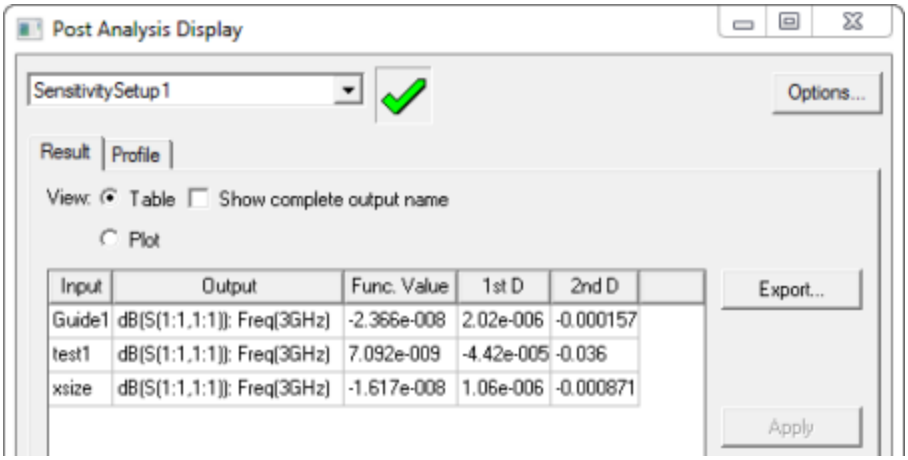
	Freq [GHz]	dB(S(1:1,1:1)) Setup1 : Sweep Guide1='3in' test1='2.1in' xsize='0.45in'
1	3.000000	-0.545922
2	4.000000	-2.300003
3	5.000000	-10.634331
4	6.000000	-4.586505
5	7.000000	-5.950253
6	8.000000	-9.019574
7	9.000000	-3.210097

> **Tip:**
>
> Transpose the table for an alternate view (double-click data table to view the **Properties** window, and on the **Data Table** tab check **Transpose**).



To estimate lower bound of performance:

First, see the sensitivity of performance to the respect of each variable. (Follow the documentation for Viewing Output Parameter Results for a Sensitivity Analysis). Identify the variables that have major influence over performance.



Second, in your project, manually change these variables to the corresponding bounds: choose Min for positive 1st derivative and Max for negative 1st derivative.

2. **Monte Carlo analysis:**

   This method does not assume a circuit is linear - better accuracy is achieved with more iterations. The cost is computing time and resources.

   To perform Monte Carlo Analysis in Ansys Electronics Desktop, create a new analysis under **Optimetrics> Statistical**. Follow the documentation on Statistical Analysis to set

this up. The upper and lower bound of performance can be found on the edge of performance distribution.

> **Tip:**
>
> If distribution of performance is not of interest, set all variables as uniformly distributed.

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 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**.

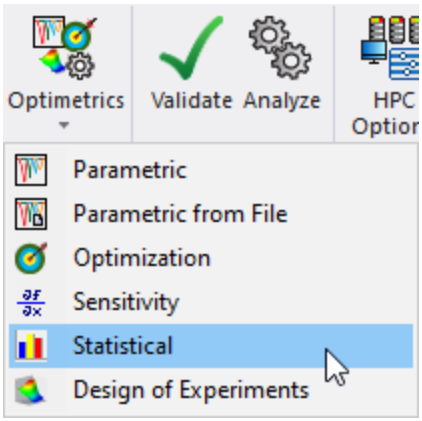
> **Note:**
>
> Statistical Analysis has been placed in legacy mode because it is no longer under active development. Go to **Tools** > **Options** > **General Options** > **Miscellaneous** and toggle the **Enable Legacy Optimetrics Tools** check box to show or hide these legacy tools.

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. Click **Q3D Extractor**> **Optimetrics Analysis** > **Add Statistical**  , or select the **Simulation** tab of the ribbon and, under the Optimetrics icon, select **Statistical** from the drop-down menu:



   The **Setup Statistical Analysis** dialog box appears.

3. Under the **Calculations** tab, type the maximum number of iterations you want Q3D Extractor 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.

9. If you want to save the field solution data for the design variations solved during analysis, select **Save Fields**.

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.

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. You can see the distribution of the solution quantities in the results.

1. In the **Calculations** tab of the *Setup Statistical Analysis* dialog box, click **Setup Calculations**.

   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 box:
   - Select the **Report Type** with a drop-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 RangebFunction** dialog to apply functions to the expression that apply overbthe 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 **Add Calculation** 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.

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. For a Driven Modal or Driven Terminal design, if you specified that the solution be extracted from the last adaptive solution, Optimetrics uses the adaptive frequency defined in the solution setup. 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 **Edit** 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.

   Other examples of available variables include frequency, if you selected an S-parameter solution quantity; and phi or theta, if the solution quantity is a radiated field quantity.

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.

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.

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.

    The design variations are solved using a Gaussian distribution within the specified mean and standard deviation values.
6. If the distribution type is **Uniform**, Eeter a tolerance value in the text box.

    The design variations are solved 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.

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 **Q3D Extractor** > **Design Properties**, select **Statistics**.

    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.

   The design variations are solved using a Gaussian distribution within the specified mean and standard deviation values.

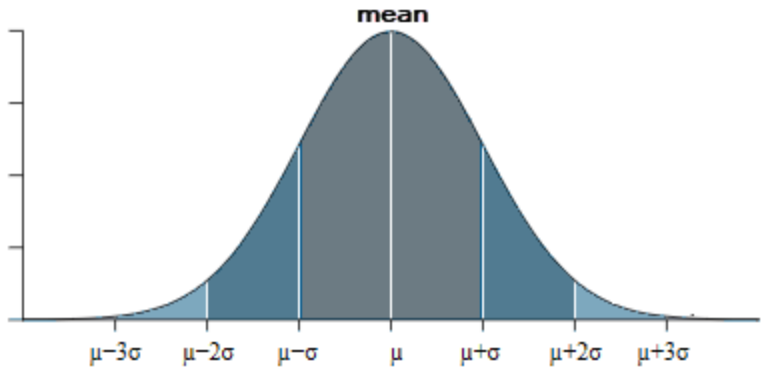
7. If the distribution type is **Uniform**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Type mean and tolerance values in the corresponding text boxes.

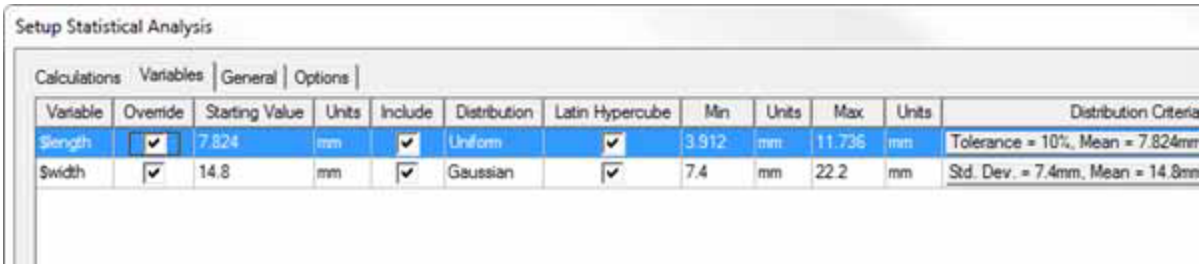
   The design variations are solved within the tolerance range of the starting value, using an even distribution.

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 box.
   c. Either type or import the X and Y data values for the distribution in the *Edit Dataset* dialog box.
10. Click **OK**.

Statistical Cutoff Probability

The cutoff probability values affects the Gaussian distribution criteria. This is a value =>0 and < 0.1. The design variations are solved using a Gaussian distribution using a lower limit cutoff probability and specified mean and standard deviation values.





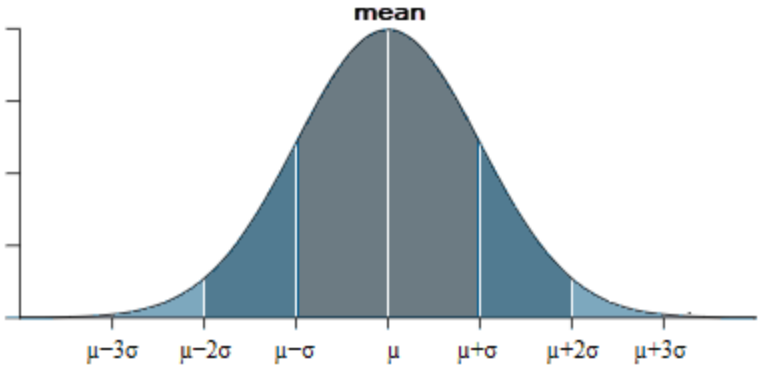
Uniform distributions such as **length** variable above use only the **Tolerance** value and do not have a cutoff probability.

Edit Distribution

When setting the distribution type for a variable, you have the option of changing the distribution parameters from the default values.

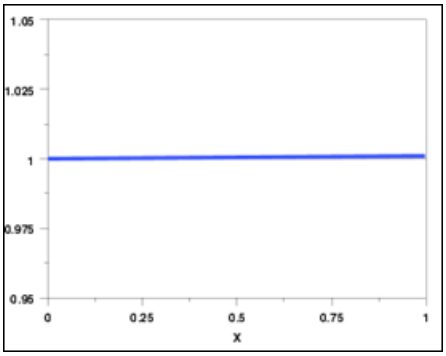
1. If the distribution type is **Gaussian**, do the following:
    a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.
    b. Type the mean value of the distribution in the **Mean** text box.
    c. Type the standard deviation of the distribution in the **Std Dev** text box.

    The design variations are solved using a Gaussian distribution within the specified mean and standard deviation values.



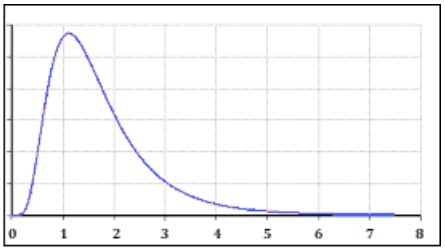
2. If the distribution type is **Uniform**, do the following:
    a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
    b. Type the mean value of the distribution in the **Mean** text box.
    c. Enter the tolerance in the **Tolerance** text box.

    The design variations are solved 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.

    The design variations are solved 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.

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.

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** drop-down menu.
3. Select **Solve the parametric sweep during analysis**.

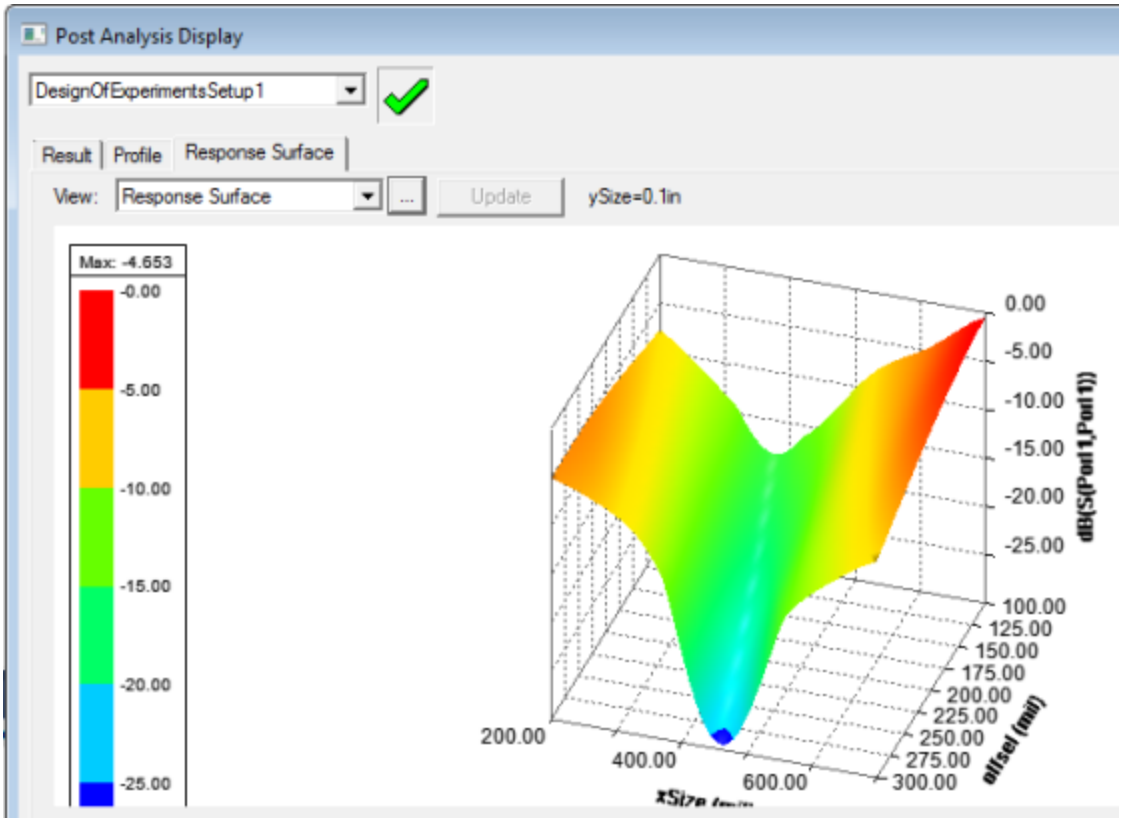
Using Design of Experiments

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

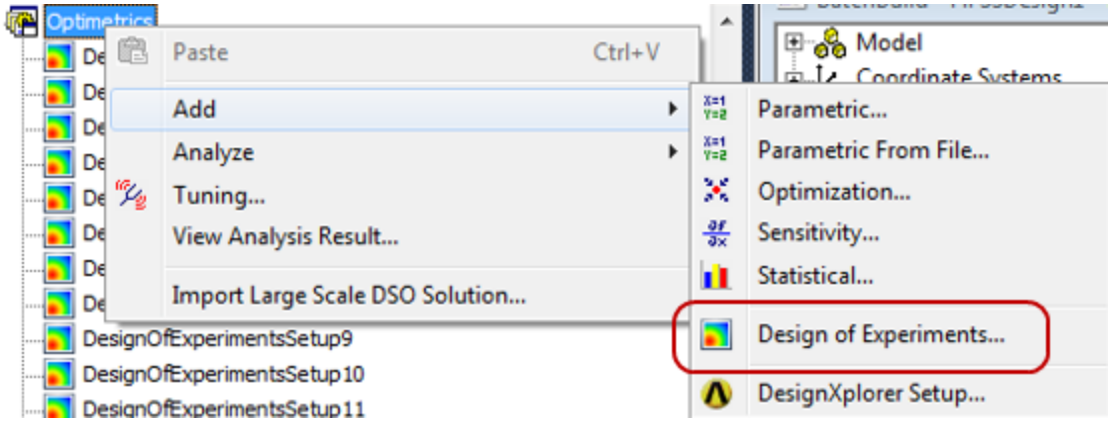
- reduce the number of simulations
- interactively explore the design space before running optimization

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.

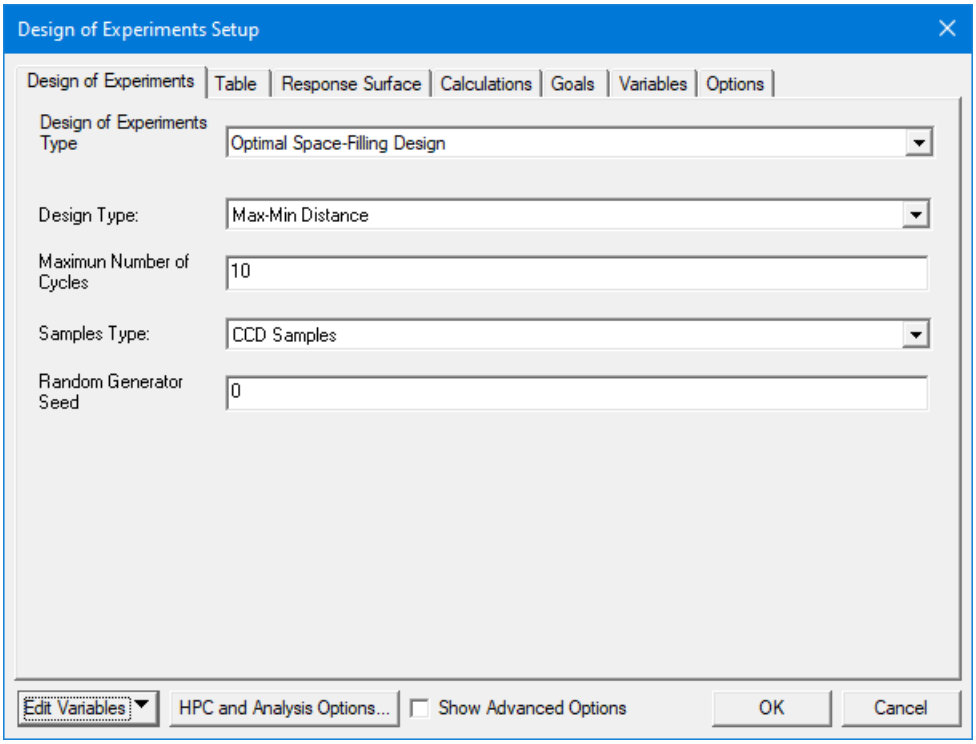
The goal is to create a response surface by interpolating through calculated points (a best curve fit). For each design, you can create a response surface for each output parameter. Once the response surfaces are created, you can share the information can in easily understandable terms: curves, surfaces, 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 **Design of Experiments** under **Optimization** opens a window 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.

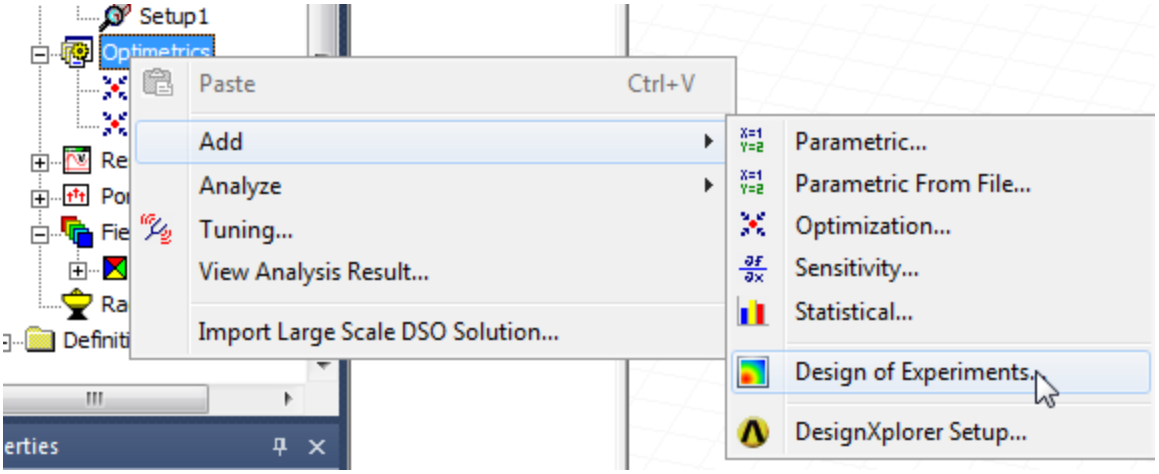
You should set up your DOE Properties before generating your DOE Design Point matrix. The following topics describe setting up and solving your Design of Experiments, and viewing the results.

Setting Up Design of Experiments in Ansys Electronics Desktop

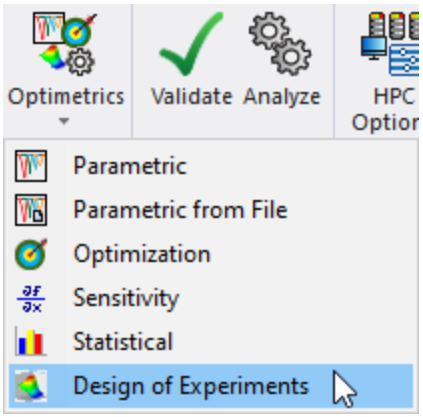
The process for setting up a Design of Experiments analysis in Ansys Electronics Desktop follows:

Add the Design of Experiments Setup

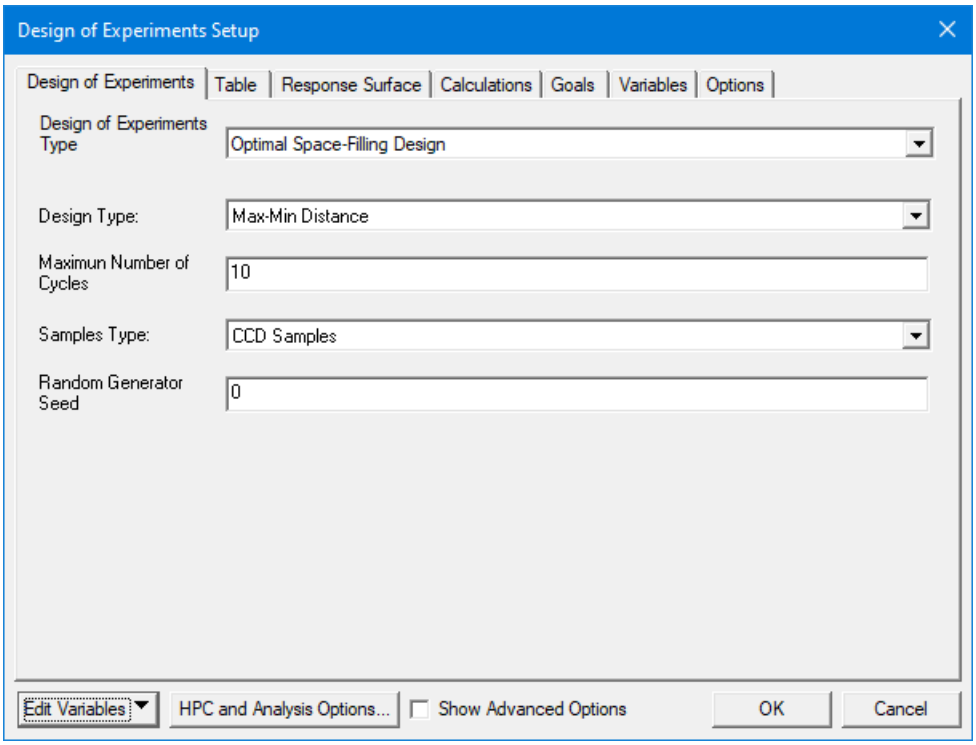
Right-click the Optimetrics icon in the Project tree and select **Add** > **Design of Experiments**. You can also use **Q3D Extractor** > **Optimetrics Analysis** > **Add Design of Experiments**.



You can also select the **Simulation** tab and under the Optimetrics icon, select Design of Experiments from the drop-down menu:



The **Design of Experiments Setup** window appears.



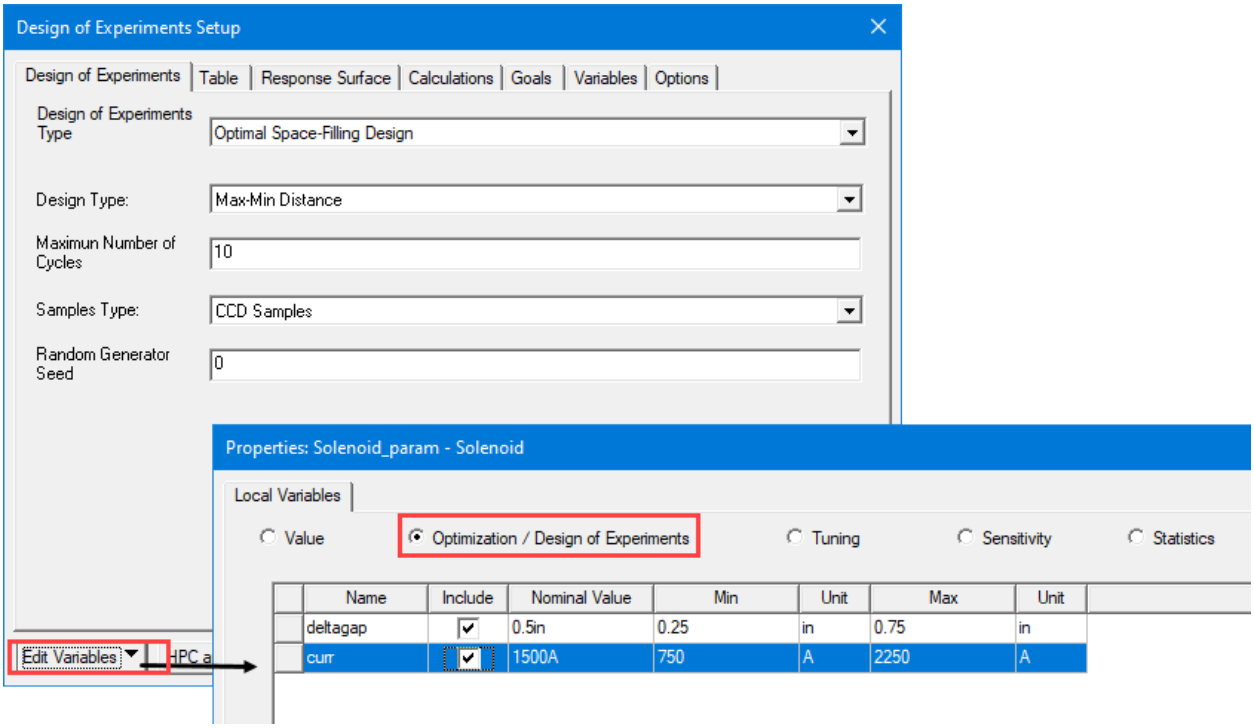
It has tabs for:

- 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

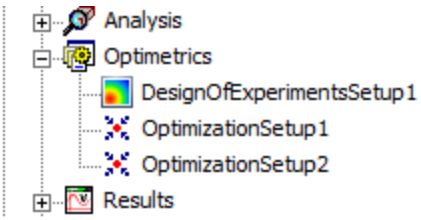
data files as well as add or edit rows. You can also Export files of DOE sampling points you have defined by any method.

- 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.

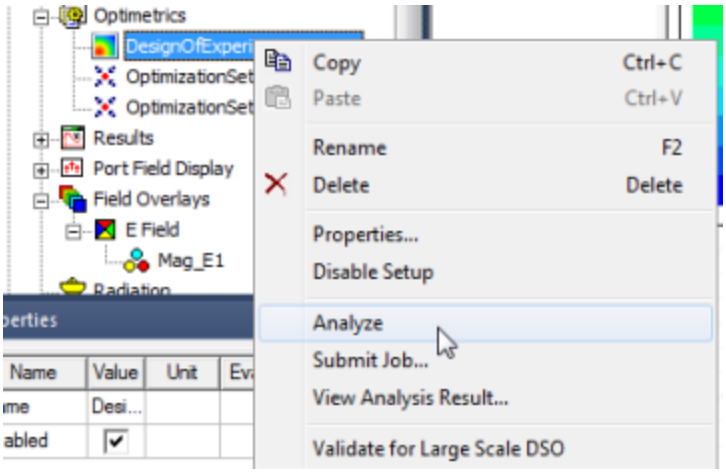
You must define local or project variables as Optimization /Design of Experiments variables for the Design of Experiments setup to include the variable.



Once you have set parameters and click **OK**, the Design of Experiments setup appears under the Optimetrics icon in the Project tree.



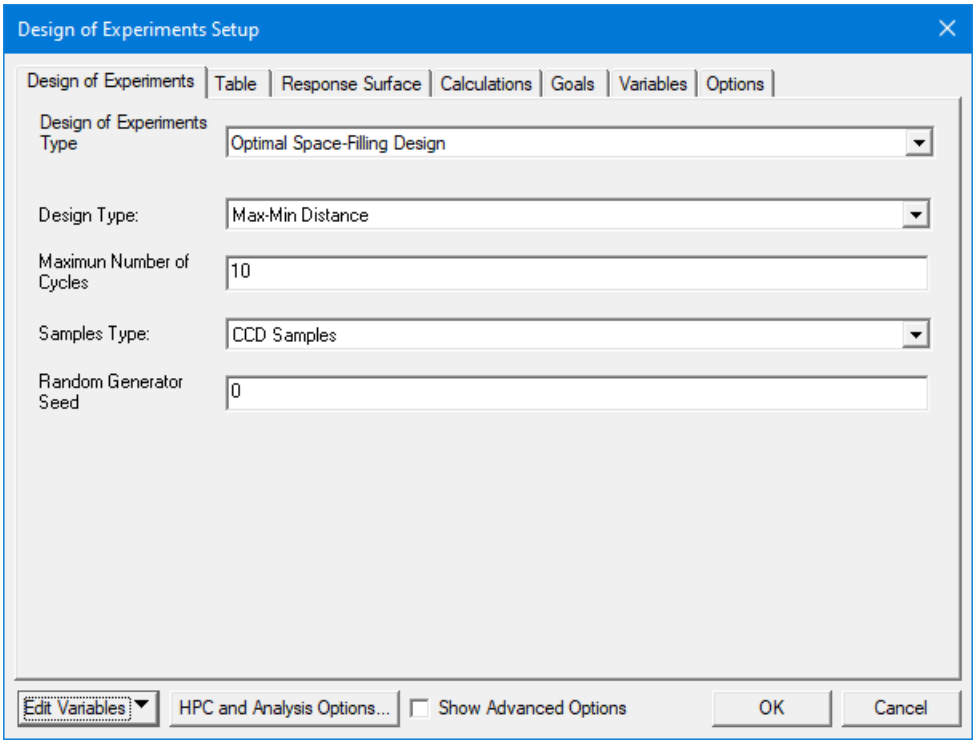
From here you can right-click the DesignOfExperiments setup from the shortcut menu to run **Analyze**, **Submit Job...** or **Validate for Large Scale DSO**.



Stand-alone and distributed simulation will have the same behavior as the parametric setup.

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 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 Design of Experiments Types available in the Desktop include the following:

Design of Experiments Types	Brief Description (see 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 Sampling type determines 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

Design of Experiments Types	Brief Description (see links for more details)
	• 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 • 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.

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.
- **Random Generator Seed**: 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.

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

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). Because 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 twelve input parameters is supported.

No additional properties are available for the Box-Behnken Design DOE type.

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.

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.

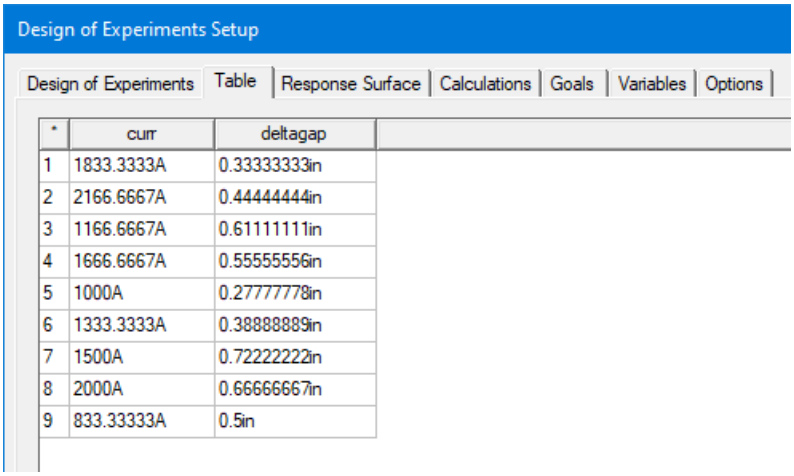
> **Note:**
>
> The Optimal Space-Filling Design DOE type is an LHS design that is extended with post-processing.

The following properties are available for the LHS DOE type:

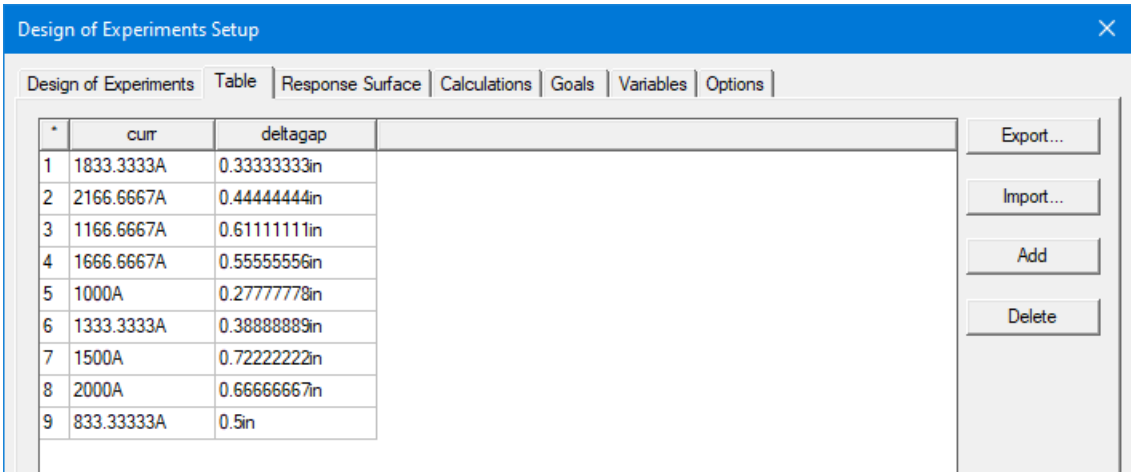
- **Samples Type**: Determines the number of DOE points the algorithm should generate. This option is suggested if you have some advanced knowledge about the nature of the metamodel. The following choices are available:
  - **CCD Samples** (default): Supports a maximum of 20 inputs. Generates the same 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.
- **Random Generator Seed**: 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.

Table Tab for Design of Experiments

The **Table** tab for the **Design of Experiments Setup** window 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.



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. All 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.



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.

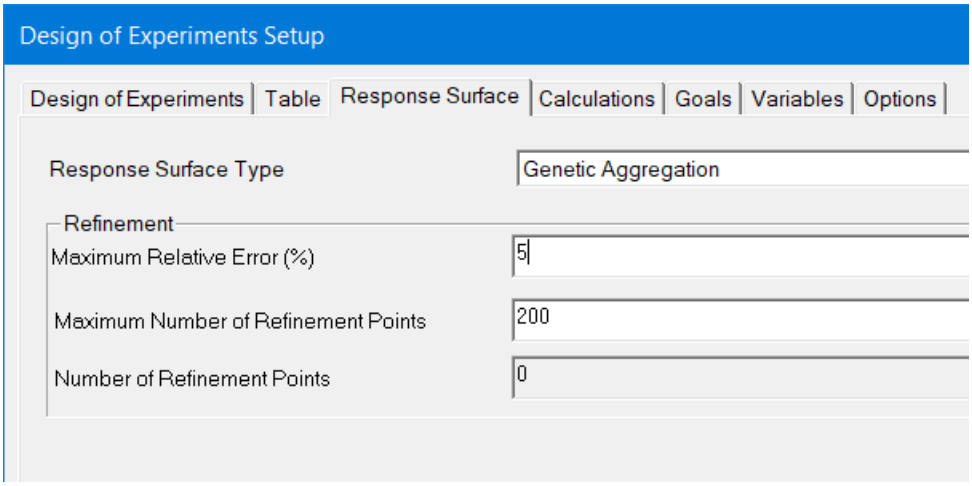
The Custom Table view also includes an Import button. Import and Export files can be

- 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.

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.



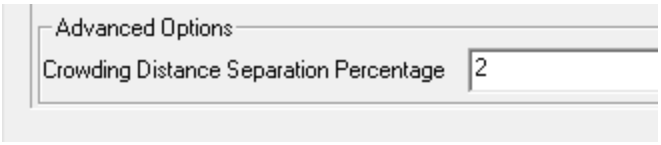
The Response Surface type selection specifies the refinement applied to the initial Design of Experiments. The Genetic Aggregation Response Surface finds the best possible response surface for each output automatically by combining

- 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:

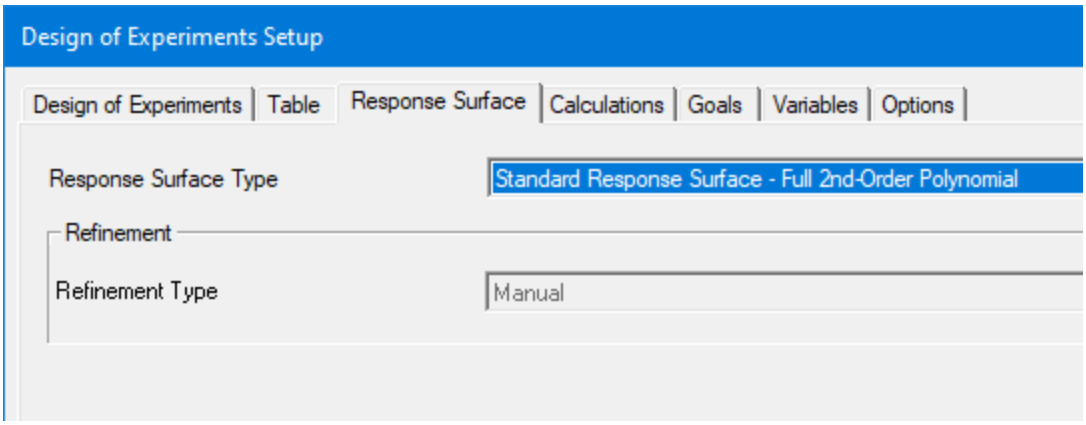
- **Maximum Relative Error %**: This apply to all output calculations. Empty this field to turn off Auto refinement. (Note: Workbench does not have this field.)
- **Maximum Number or Refinement Points**: Determines the maximum number of refinement points that can be generated for use with the Genetic Aggregation algorithm.
- **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** window, you can also specify Crowding Distance Separation Percentage:



Crowding Distance Separation Percentage: determining the minimum allowable distance between new refinement points, implemented as a constraint in the search for refinement points.

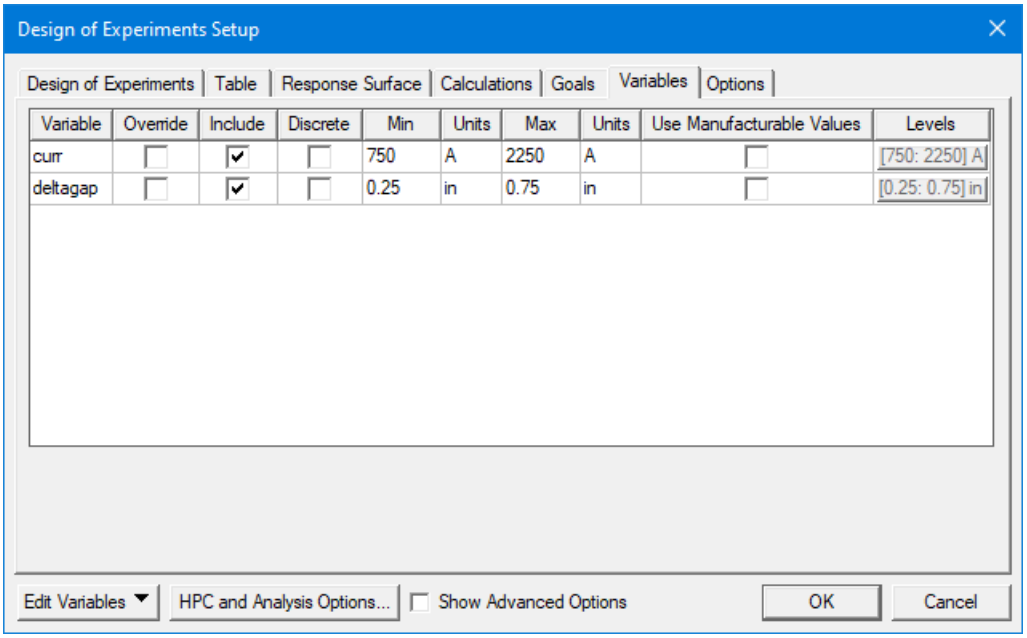
You do have the option of selecting Standard Response Surface- Full 2nd Order Polynomial.



After you have completed an analysis you can view the generated plot.

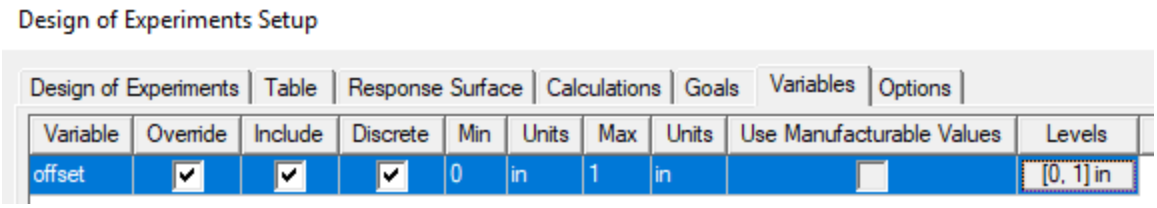
Variables Tab for Design of Experiments

The Variables tab for the **Design of Experiments Setup** window displays list of the variables defined for the Design and Project as Optimetrics/Design of Experiments variables.



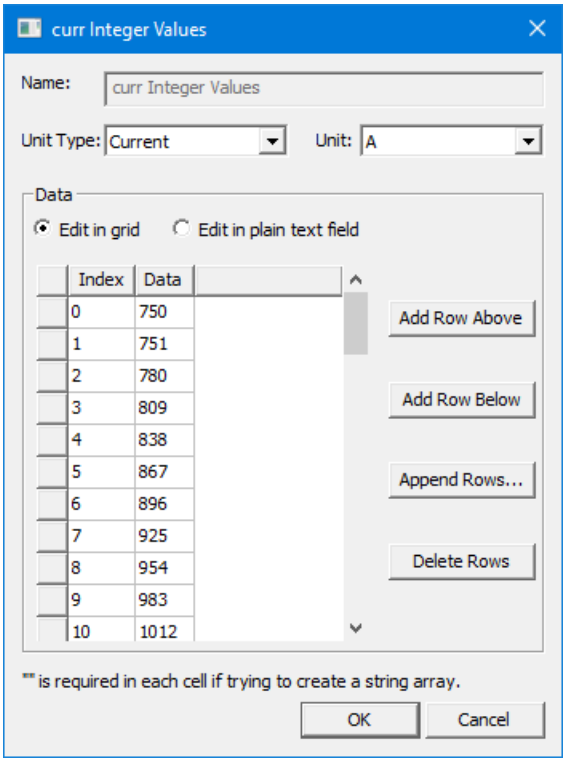
The columns list the Variable names, the current Min and Max Values and Units, and provides options for the following:

- **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 check box in the Discrete column is checked, the button in the “Levels” column will be enabled and Use Manufacturable Values is disabled.



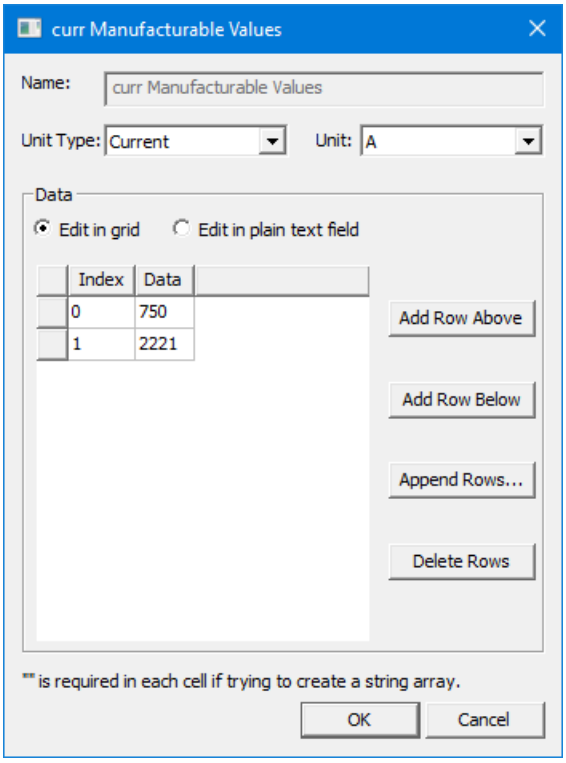
The Levels show the variable values. For continuous variable, Levels is determined by the min/max columns. For Integer variables Levels can be a subset of the min/max range. The discrete values can be bounded by a min/man range and/or manufacturable values.

Click the enabled **Levels** button for the row to edit the Discrete values. An edit dialog box for the variable appears:



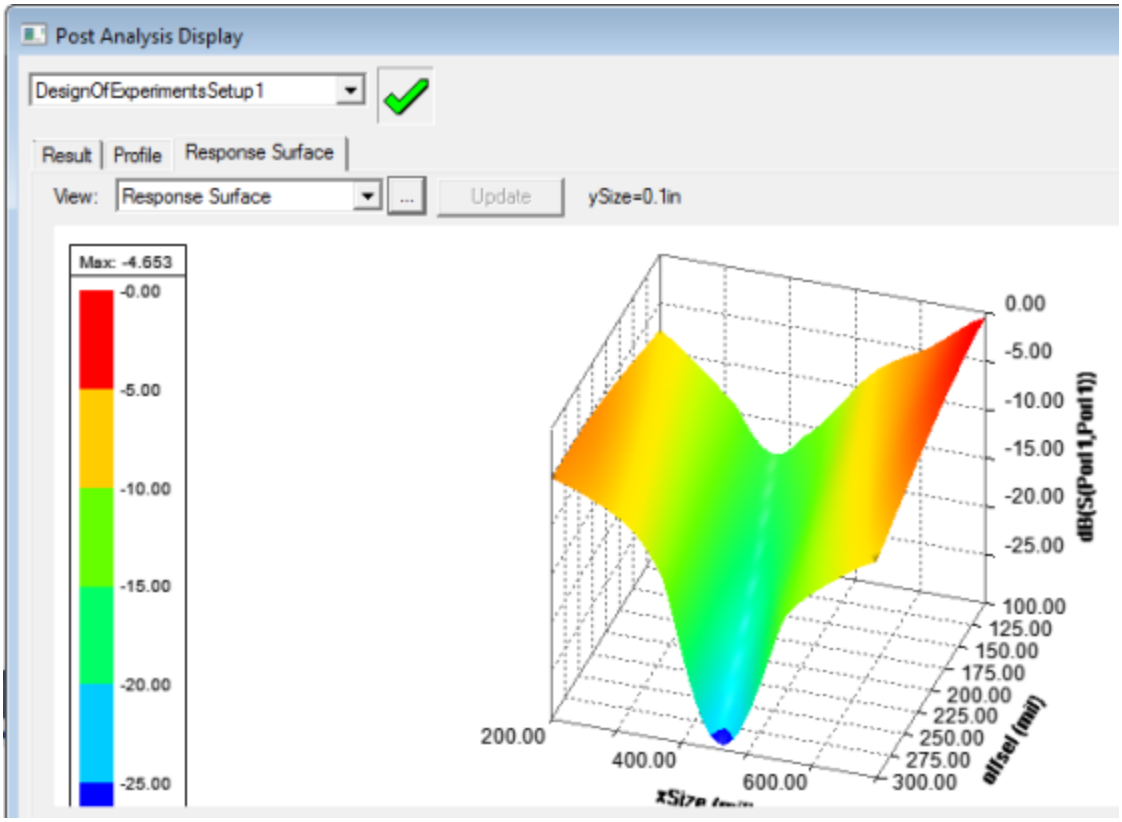
For continuous variables (with **Discrete** unchecked), the values physically vary in a continuous manner between a lower and an upper bound (min/max) defined by the user. With continuous variables, you can also check **Use Manufacturable Values**.

- When you enable the check box in the "**Use Manufacturable Values**" column the button in the "**Levels**" column is enabled. Manufacturable Values represent real-world manufacturing or production constraints. The min max values for the variable will be upper and lower constraints on the available manufacturable values bounded by the min/max range. If you check **Use Manufacturable values**, you edit the Levels to define the Manufacturable values. Click the Levels button to edit the values. An edit dialog for the variable appears.



Viewing Analysis Result for Design of Experiments

After the Analysis has competed, you can right-click the Design of Experiments setup in the **Project tree** and click **View Analysis Result** on the shortcut menu. 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 ellipsis [...] button next to the view list box.

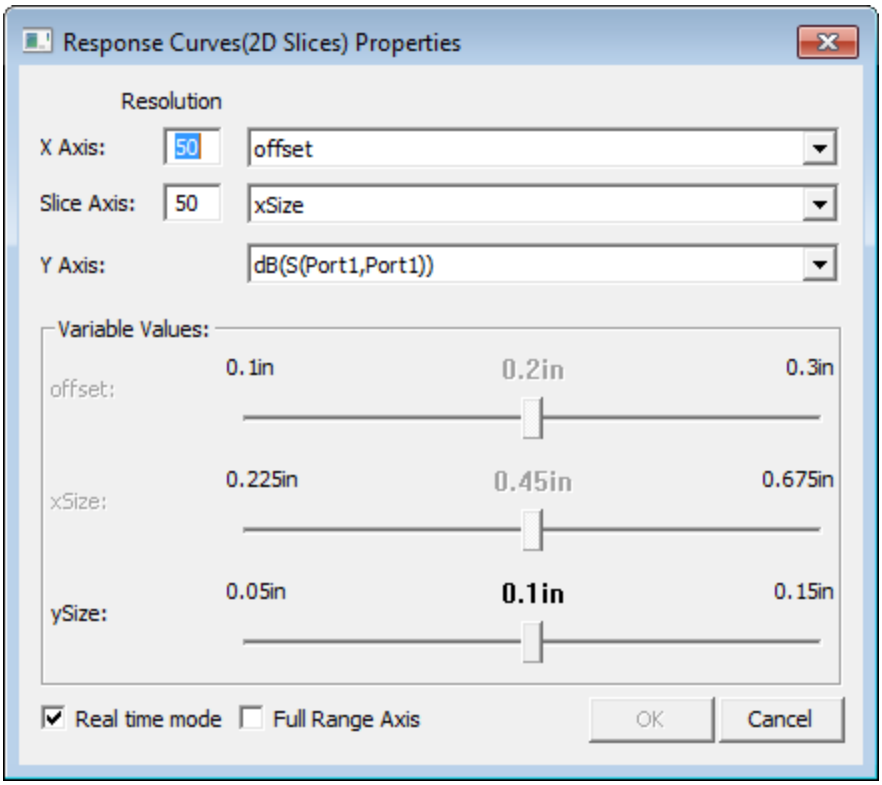


From the **Response Surface** tab, the “View” list box lets you select all available views of the selected response-surface-setup.

- Min Max Search
- Refinement points table
- Response points table
- Verification points table
- Goodness of Fit
- Response Curve
- Response Curve (2D Slices)
- Response Surface
- Local Sensitivity Charts
- Local Sensitivity Curves

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 regenerate 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.

Clicking the ellipsis button [...] by the View for Response Surface opens a dialog box 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)
- Ansys Plot data files, (*.dat)

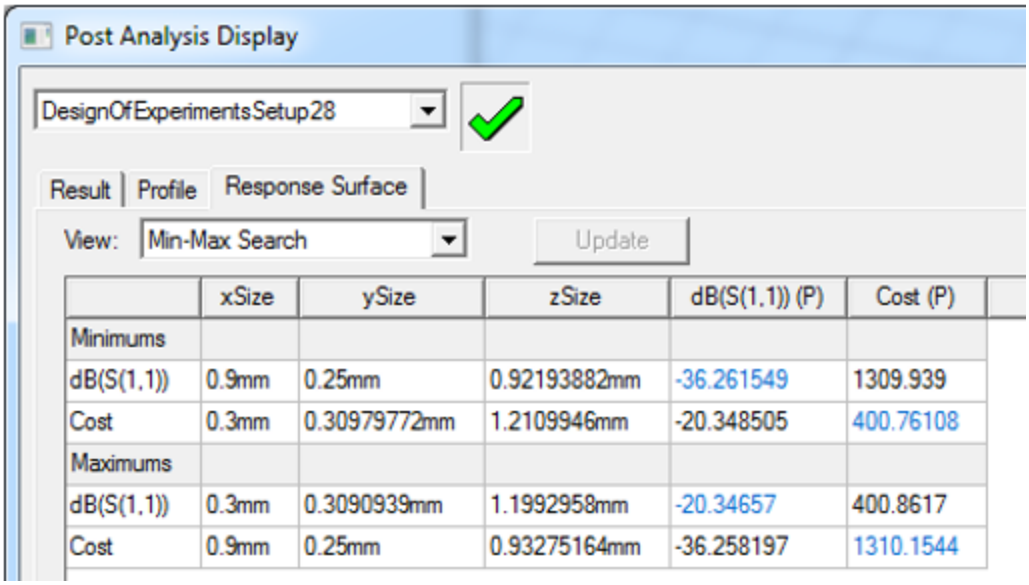
- Post Processor format data files, (*.txt)
- Ansys Report Data *.rdat files

Exported files can be imported into a report.

Min-Max Search View for Design of Experiments Result

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.

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 **Apply** is enabled, and you can then apply the selected variation variable values to the variables' nominal values.



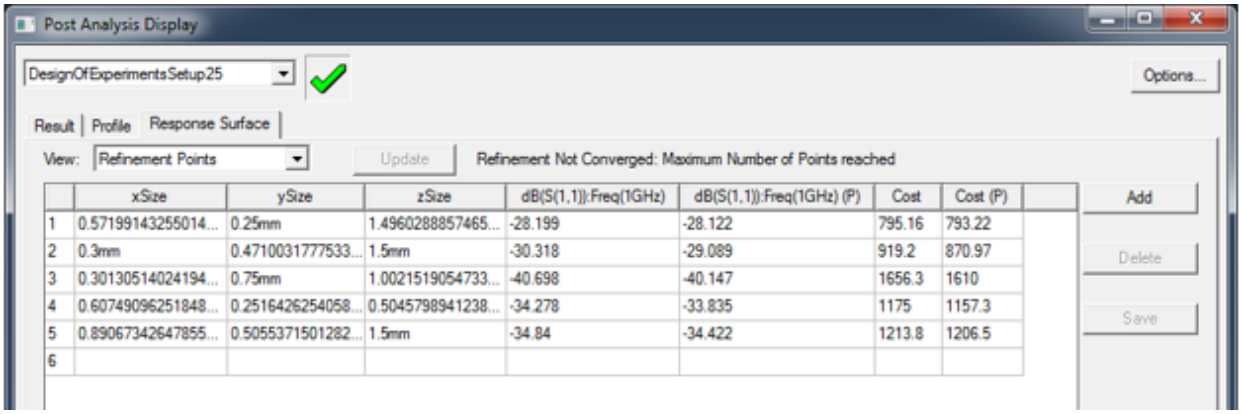
	xSize	ySize	zSize	dB(S(1,1)) (P)	Cost (P)
Minimums					
dB(S(1,1))	0.9mm	0.25mm	0.92193882mm	-36.261549	1309.939
Cost	0.3mm	0.30979772mm	1.2109946mm	-20.348505	400.76108
Maximums					
dB(S(1,1))	0.3mm	0.3090939mm	1.1992958mm	-20.34657	400.8617
Cost	0.9mm	0.25mm	0.93275164mm	-36.258197	1310.1544

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)

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 you do not need to solve for an initial response surface (without the refinement point) before updating it with your manual refinement. Manual refinement is available for all response surface types except for Sparse Grid.

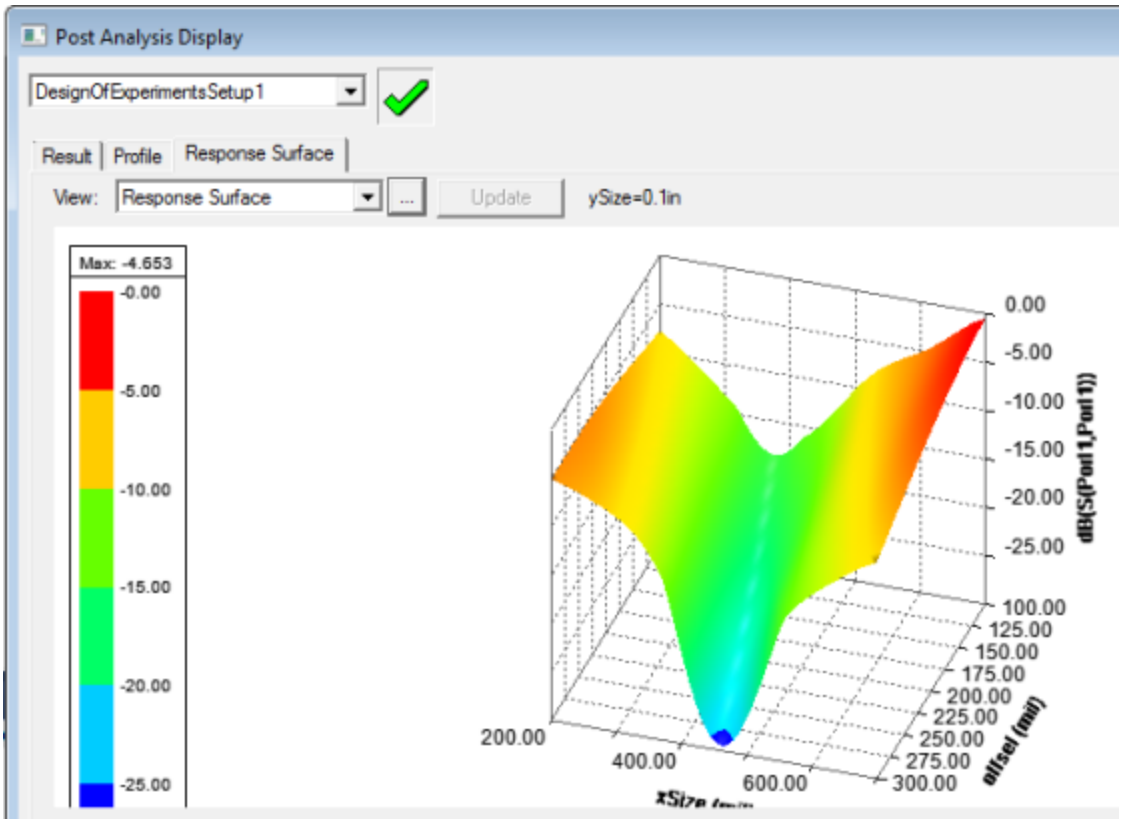
You can add, delete, or modify refinement points by clicking the **Add** or **Delete** buttons, or by modifying point values directly in the grid.

Manual Refinement Point can be inserted from the Response point table and Verification Points Table.

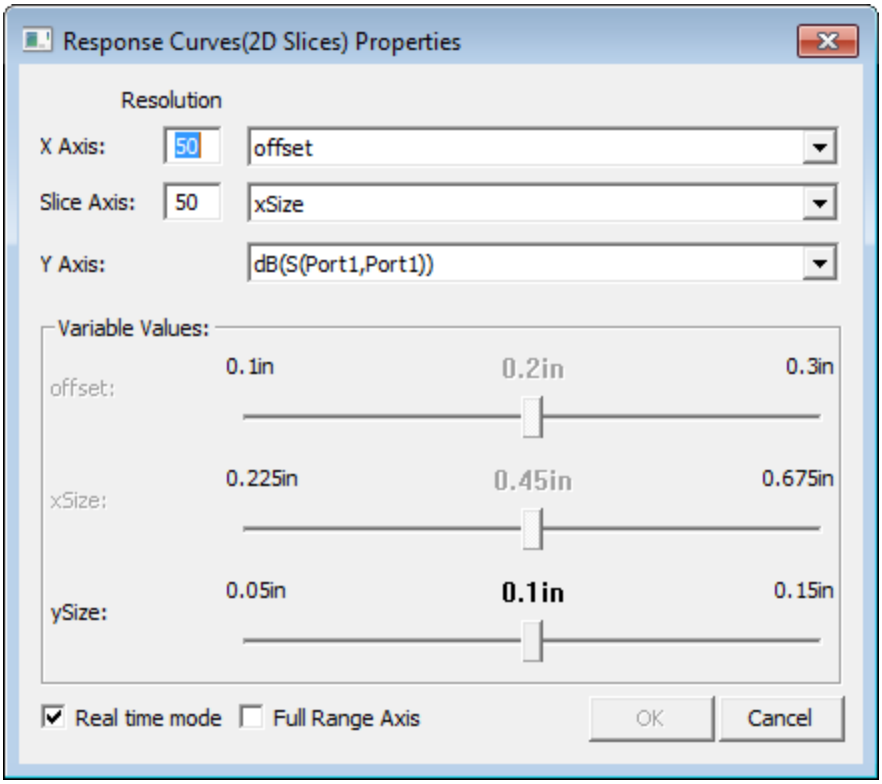
Response Surface Results

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.

Exporting Response Surface Data

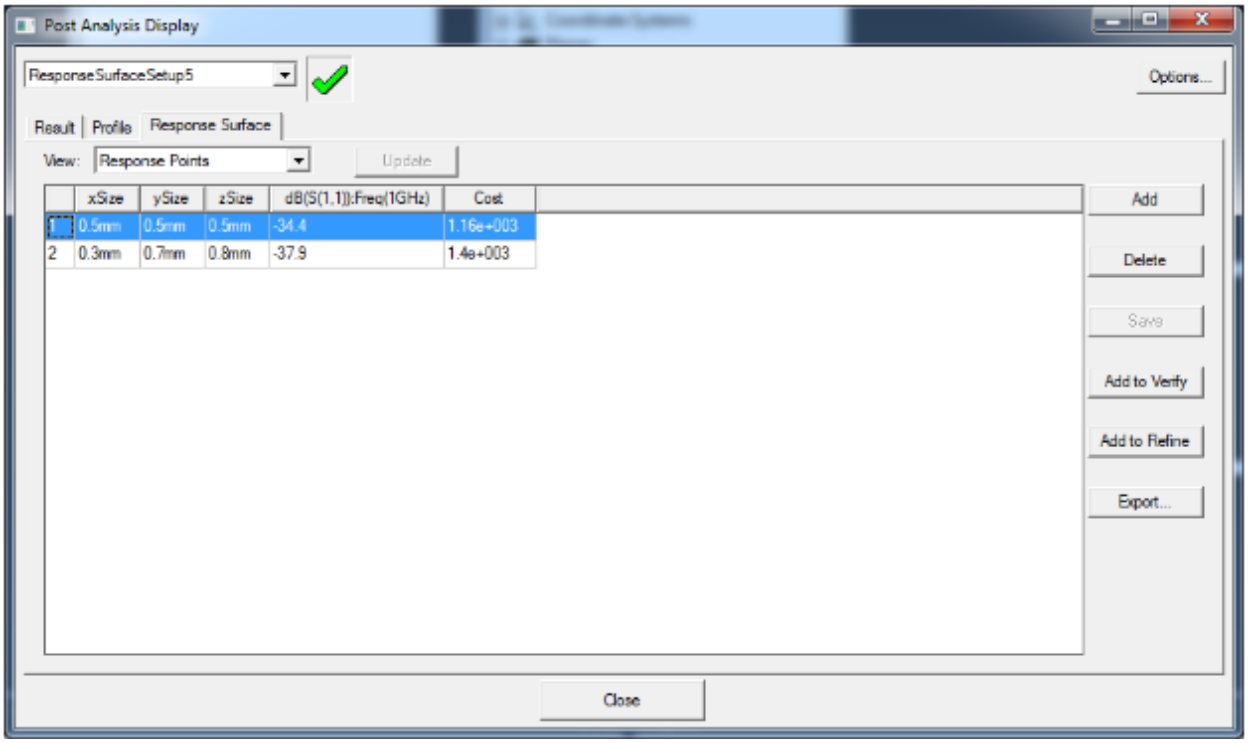
You can Export the Response surface data as a table in the following formats:

- 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.

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:



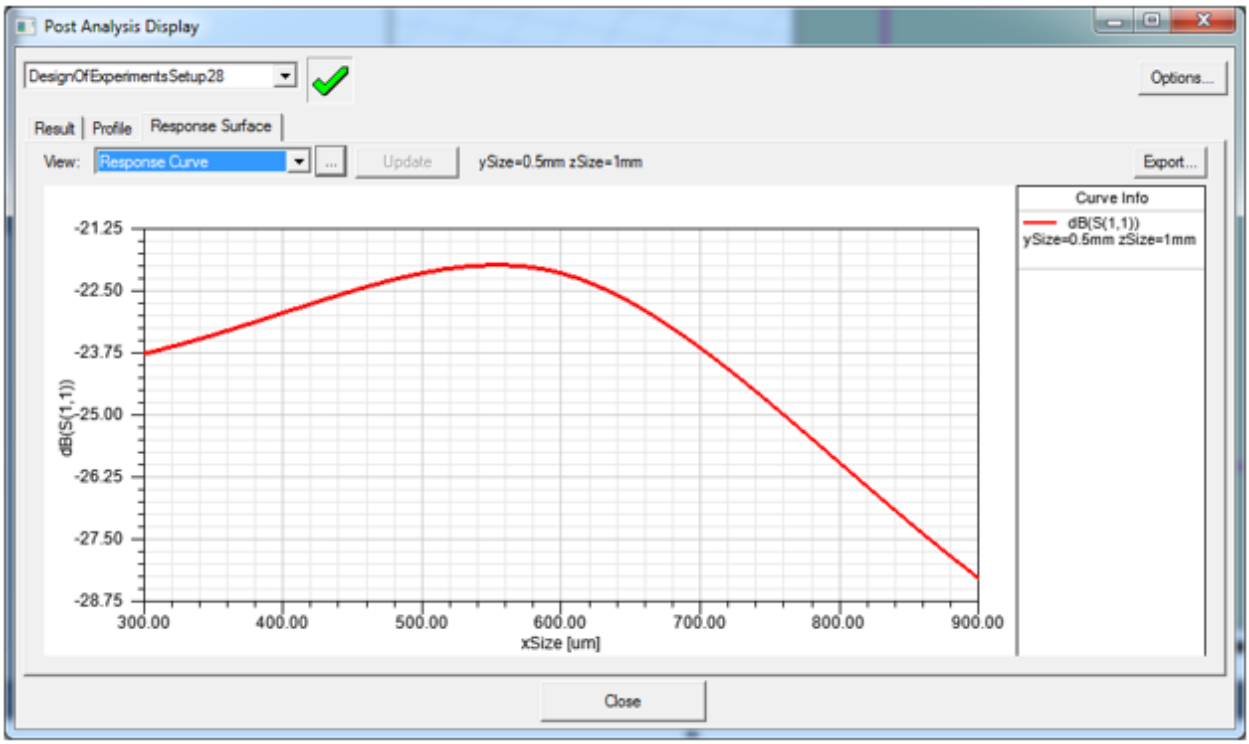
A response point is defined by a snapshot of variable values where output calculation values were calculated in Ansys DesignXplorer from a response surface. As such, the output calculation(or cost) values are approximate and calculated from response surfaces.

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.

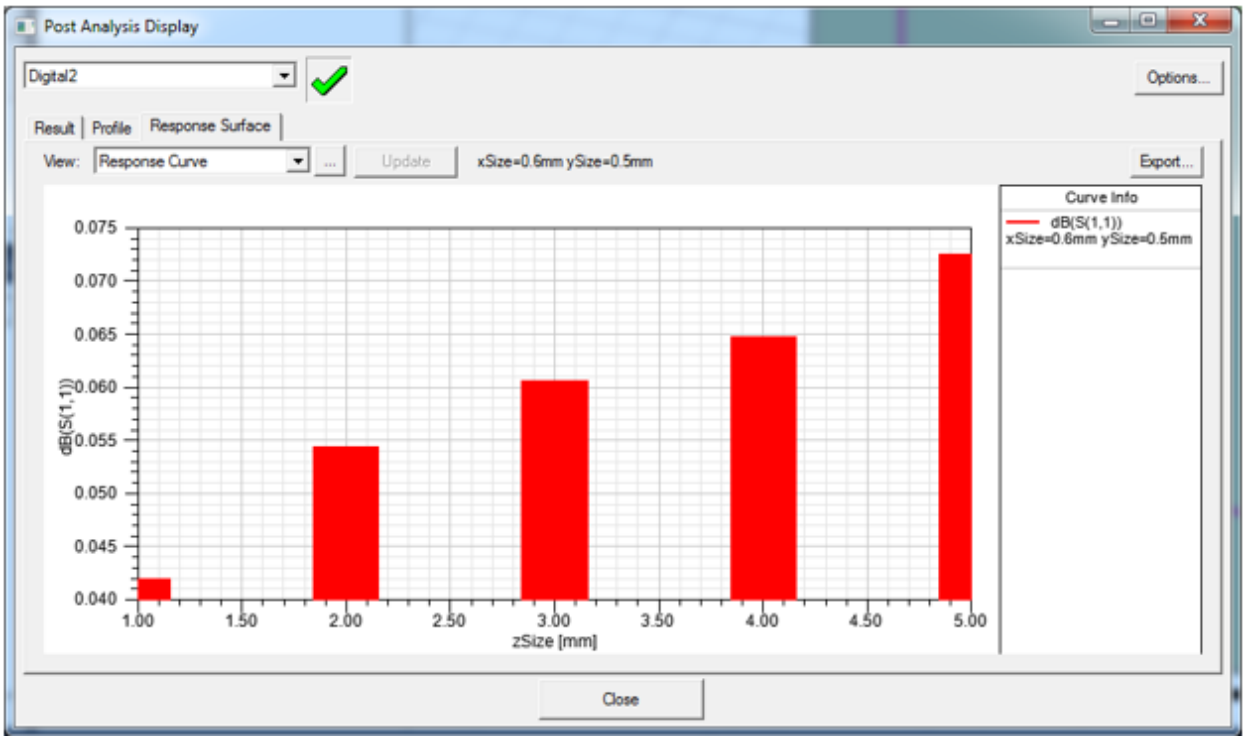
Response Curve

When Continuous Variable is chosen as the X-axis, and Continuous XY plot will be shown.



Discrete Variable

If you choose Discrete Variable as the X-axis, a bar chart plot is shown:



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.

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 check box won't clear the accumulated curves, just stop to accumulate new curve. When the Axis variable is changed, all accumulated curves will be cleared.

Exporting Response Curve Data

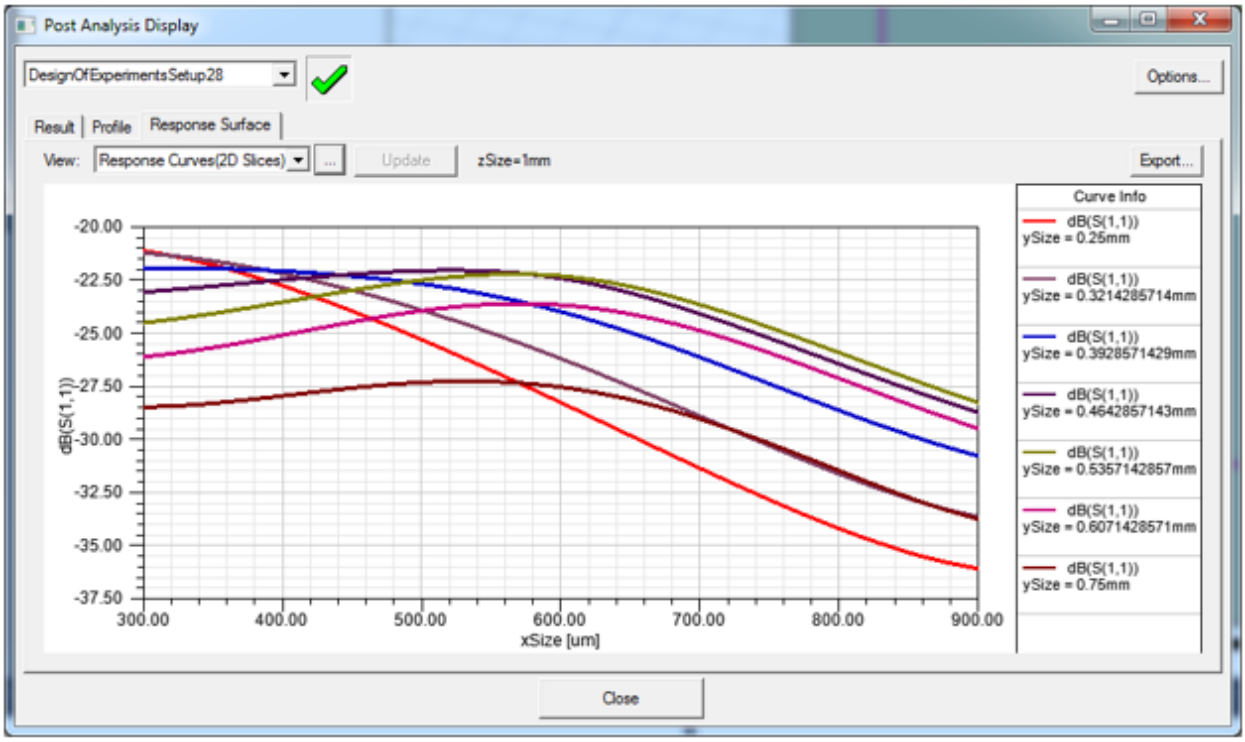
You can Export the response curve data as a table in the following formats:

- Comma delimited data files, (*.csv)
- Tab delimited data files, (*.tab)
- Ansys Plot data files, (*.dat)
- Post Processor format data files, (*.txt)
- Ansys Report Data *.rdat files

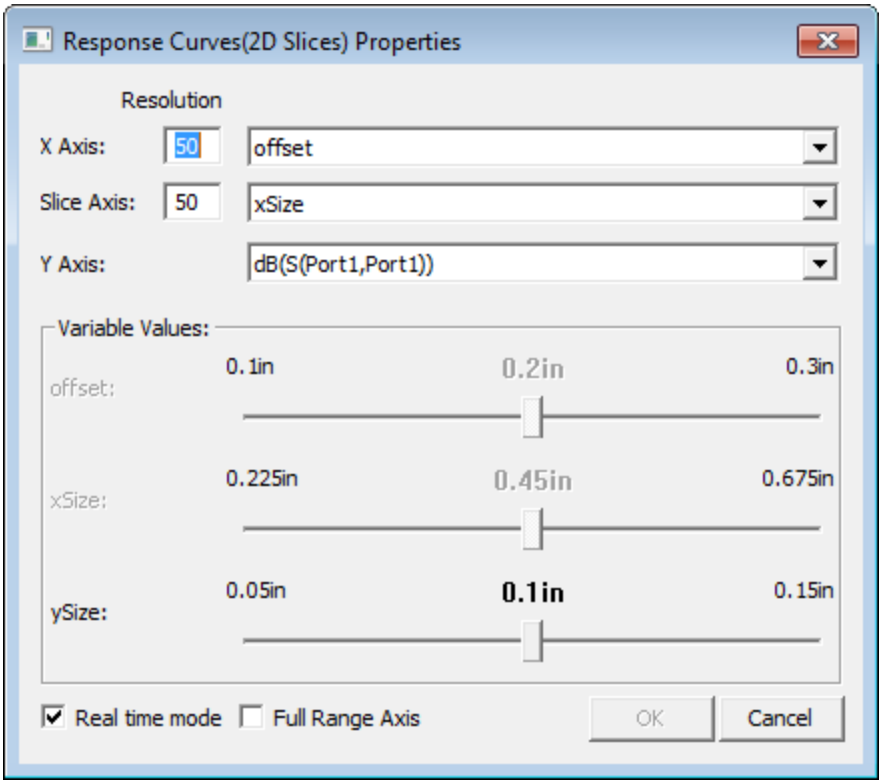
Exported files can be imported into a report.

Response Curve (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

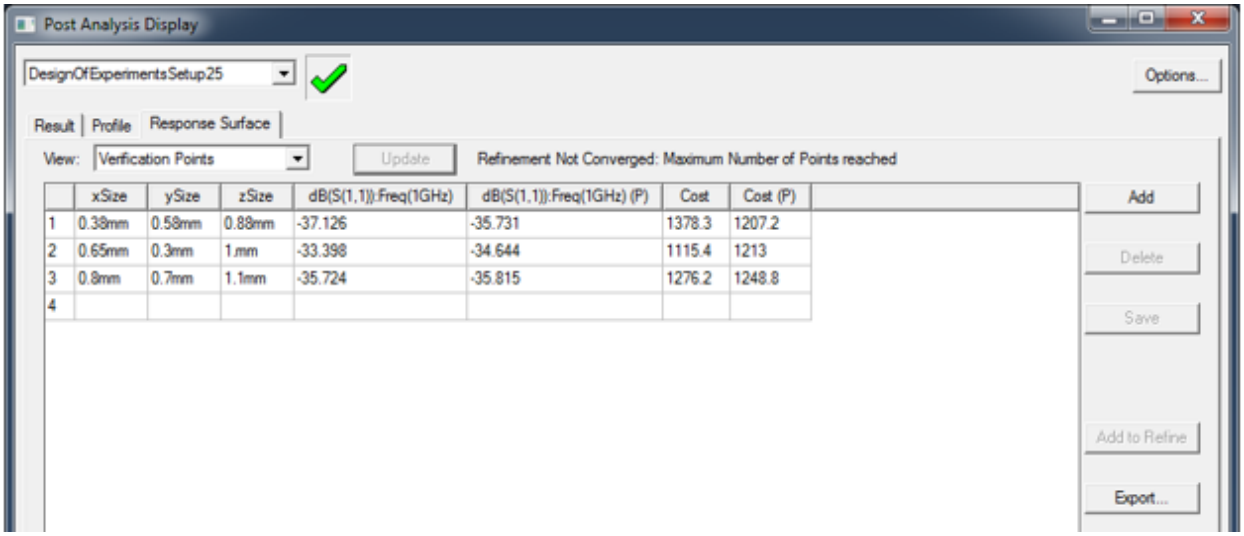
You can Export the response curve data as a table in the following formats:

- Comma delimited data files, (*.csv)
- Tab delimited data files, (*.tab)
- Ansys Plot data files, (*.dat)
- Post Processor format data files, (*.txt)
- Ansys Report Data *.rdat files

Exported files can be imported into a report.

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 **Add to Refine** button to insert the selected response point to the refinement table.

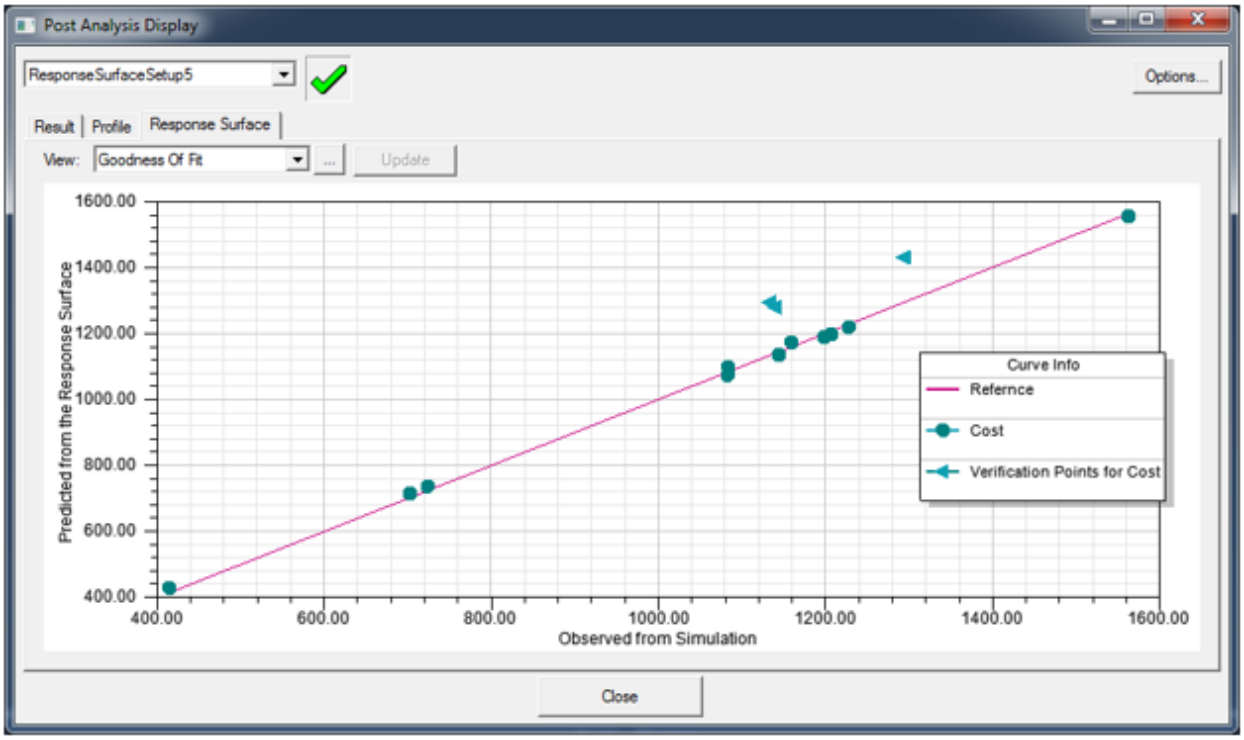
A design point update (that is, a "real solve") calculates each verification point. These verification point results are then compared with the response surface predictions and the difference is calculated.

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.

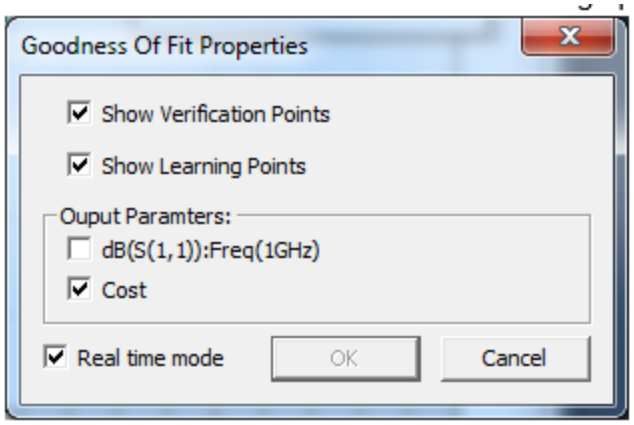
Goodness of Fit (Predicted vs Observed Chart)

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.

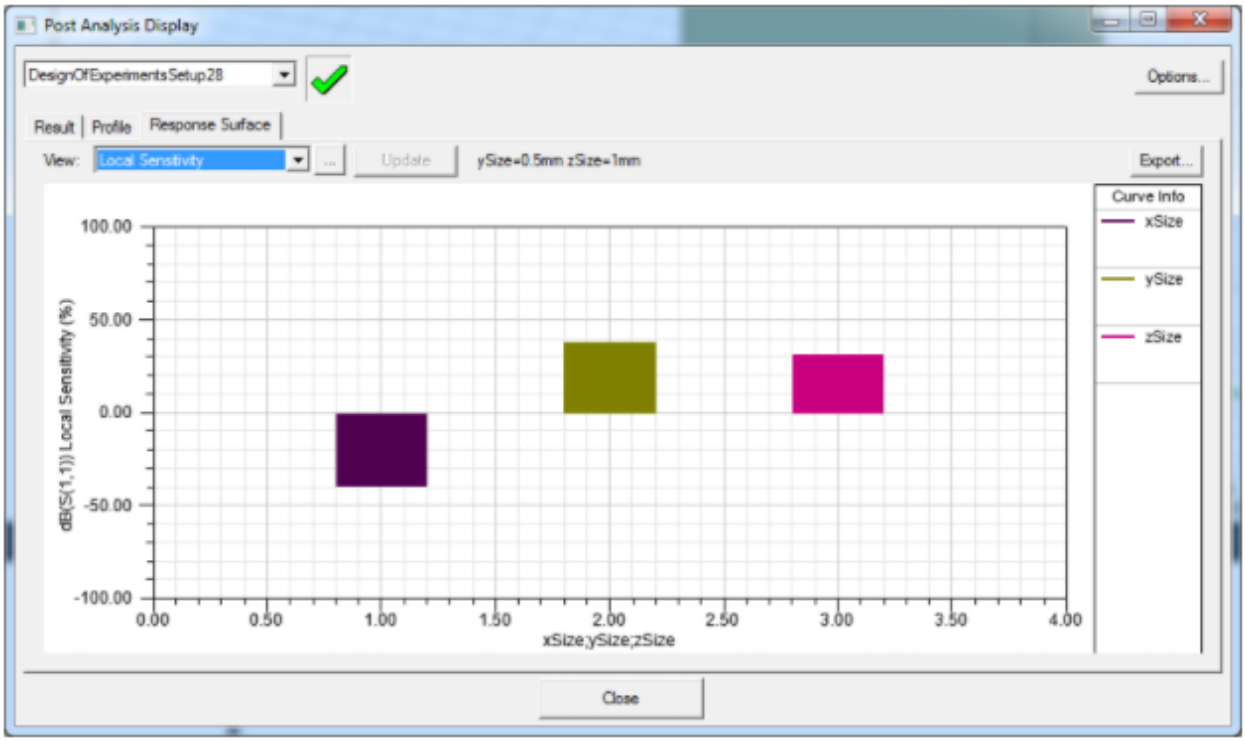


You can view Goodness of Fit information for any of the output parameters in a response surface. To do so click the ellipsis button [...] button to bring up this dialog box:

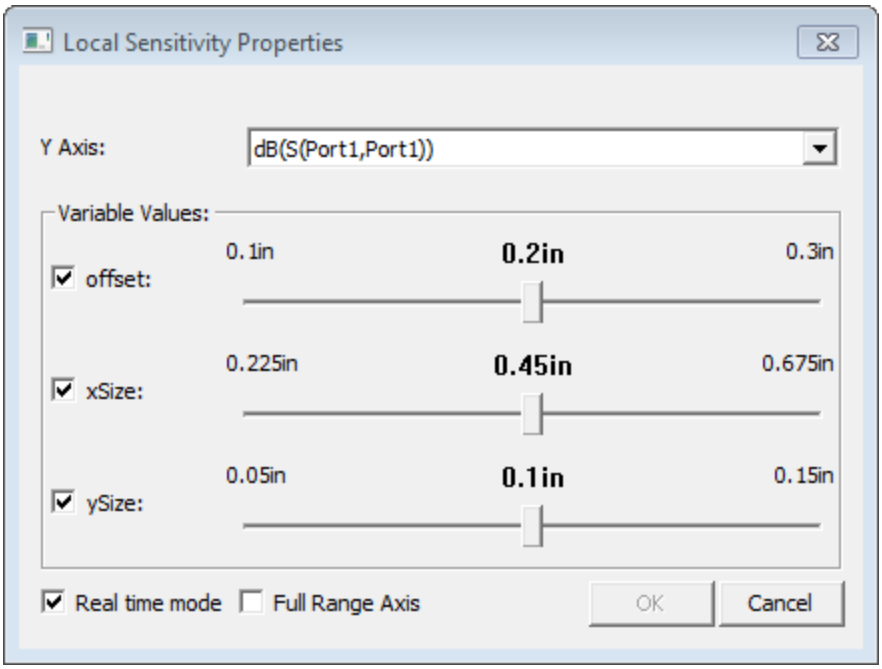


Local Sensitivity Charts

From the **Response Surface** tab, the **View** list box lets you select all available views of the selected response-surface-setup. Local Sensitivity Charts allow you to graphically view the impact that changing each input parameter has on the output parameters. The Local Sensitivity chart can be a powerful exploration tool. For each output, it allows you to see the weight of the different input. This chart calculates the change of the output based on the change of each input independently, at the current value of each input parameter in the project.

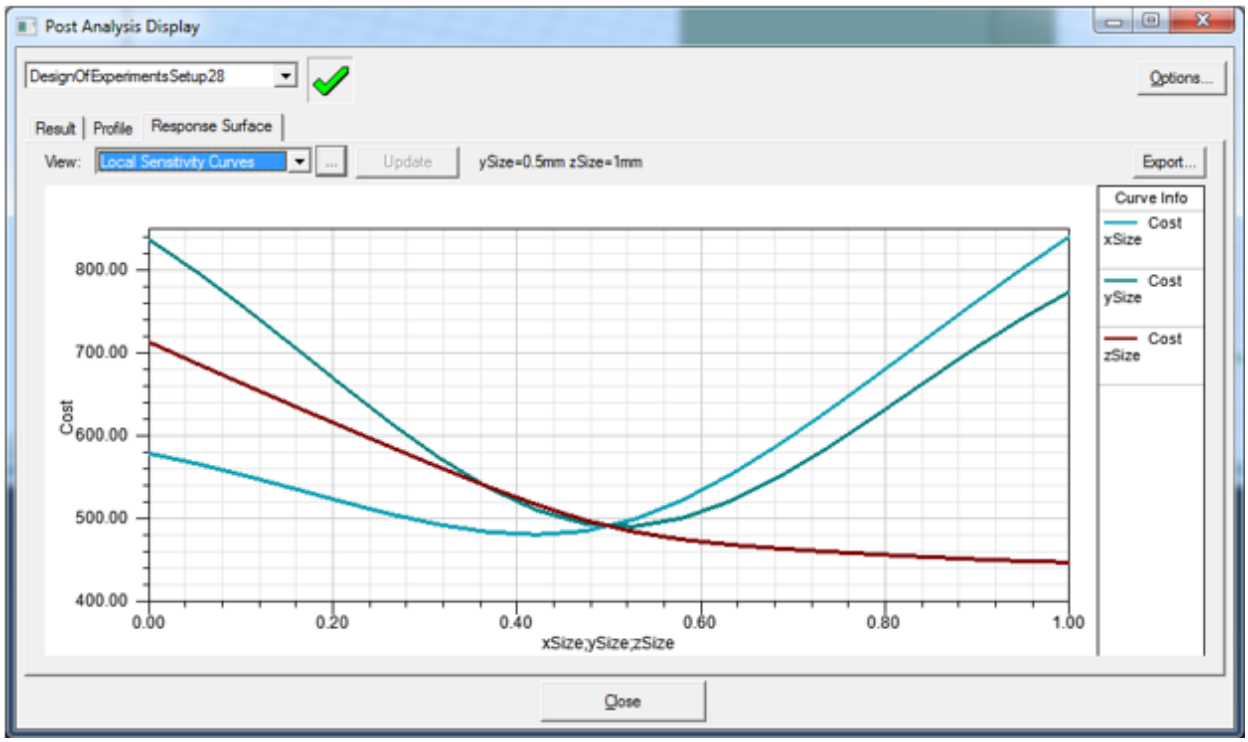


Clicking the ellipsis button [...] opens the **Local Sensitivity Properties** dialog that lets you adjust the variables selected and the values applied.

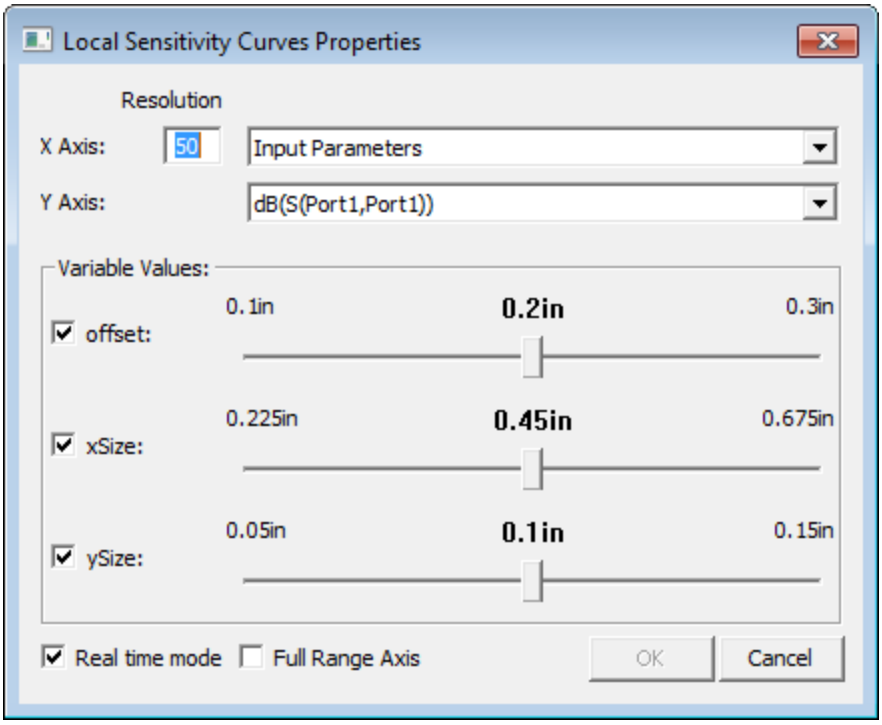


Local Sensitivity Curves

From the **Response Surface** tab, the **View** list box lets you select all available views of the selected response-surface-setup. Local Sensitivity Curves allow you to graphically view the impact that changing each input parameter has on the output parameters. The Local Sensitivity Curves chart helps you to focus your analysis by allowing you to view independent parameter variations within the standard Local Sensitivity chart. This multi-curve chart provides a means of viewing the impact of each input parameter on specific outputs, given the current values of the other parameters. The Local Sensitivities Curves chart shows individual local sensitivities, with a separate curve to represent the impact of each input on one or two outputs.



Clicking the ellipsis button [...] opens the **Local Sensitivity Properties** dialog box that lets you adjust the variables selected and the values applied.



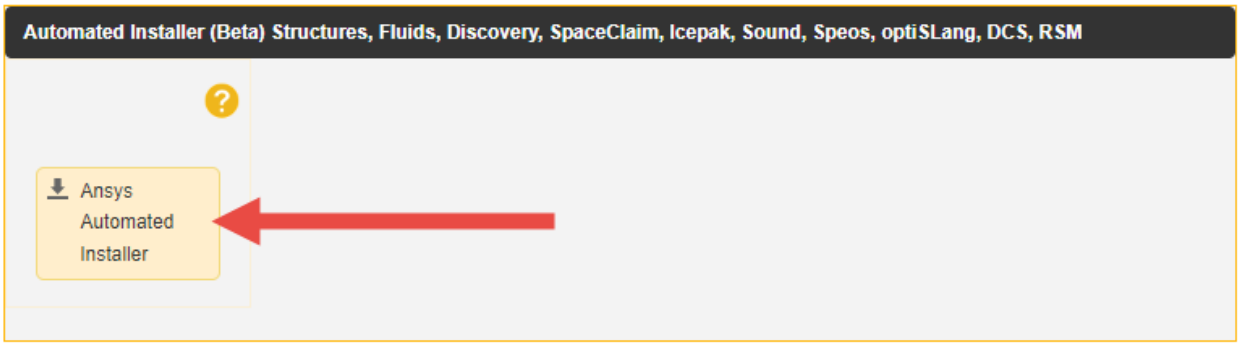
Running an Optimetrics Analysis

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.

optiSLang Integration with Electronics Desktop

optiSLang is a tool for graphical programming, process integration, and automation that can be integrated with Ansys Electronics Desktop (AEDT) for Optimetrics analysis. It is available as part of the Ansys Automated Installer package on the Ansys Customer Portal.



Comprehensive online help for optiSLang is available from the optiSLang Help menu, or at:

https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v251/en/opti_ug/opti_ug_intro.html

This section covers only its integration with Ansys Electronics Desktop.

optiSLang in AEDT allows AEDT and optiSLang to stay in continuous interaction by exchanging API commands between each other. optiSLang algorithms can be executed without exiting AEDT because optiSLang setups follow Optimetrics norms and have an explicit association with a specific analysis setup. AEDT stays continuously open while optiSLang algorithms demand set after set of designs.

Each optiSLang setup resides in one specific design of an AEDT project and is linked to one specific analysis setup. When an optiSLang setup is executed within AEDT, optiSLang runs in the background in batch mode and executes the algorithm that the optiSLang setup represents. optiSLang administers an optiSLang project file (*.opf) and a normal optiSLang project directory (*.opd), stored relative to the AEDT project.

The **standalone optiSLang GUI** allows for further editing of the optiSLang project structure, with each analysis setup represented as a single Ansys EDT node:

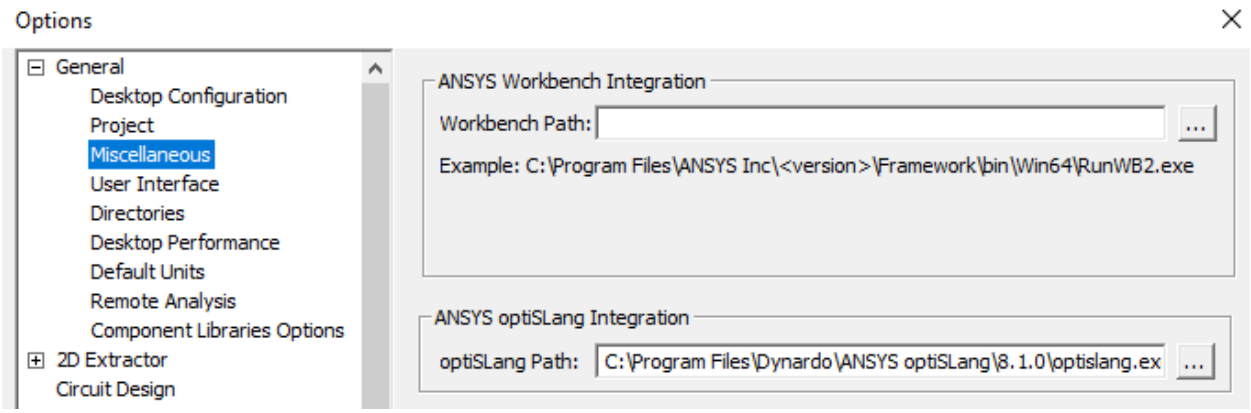
If the original Ansys EDT node and its copies are all kept in the optiSLang Setup run mode, they will remain linked to the specific optiSLang setup existing in the referenced AEDT project. The optiSLang project can contain and execute multiple algorithm systems, which means that the linked optiSLang setup in the AEDT project represents a connector, not an algorithm, in that case.

The topics in this section cover:

- optiSLang User Workflow
- Creating an optiSLang Setup in AEDT
- Solving an optiSLang Setup in AEDT
- optiSLang Menu Options
- Viewing optiSLang Postprocessing Results

Prerequisites

Before working with optiSLang in AEDT, specify the path to the optiSLang installation using **Tools** > **General Options** > **General** > **Miscellaneous**.



Additionally, ensure that the following are true:

- The project is solved
- Parameters exist (See: Parametrization for optiSLang Integration)
- Results reports exist (See: Results and Reports for optiSLang Integration)

optiSLang User Workflow

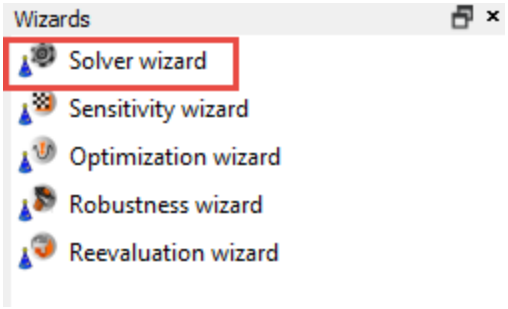
This topic explores the general workflow for creating an optiSLang project from Ansys Electronics Desktop (AEDT).

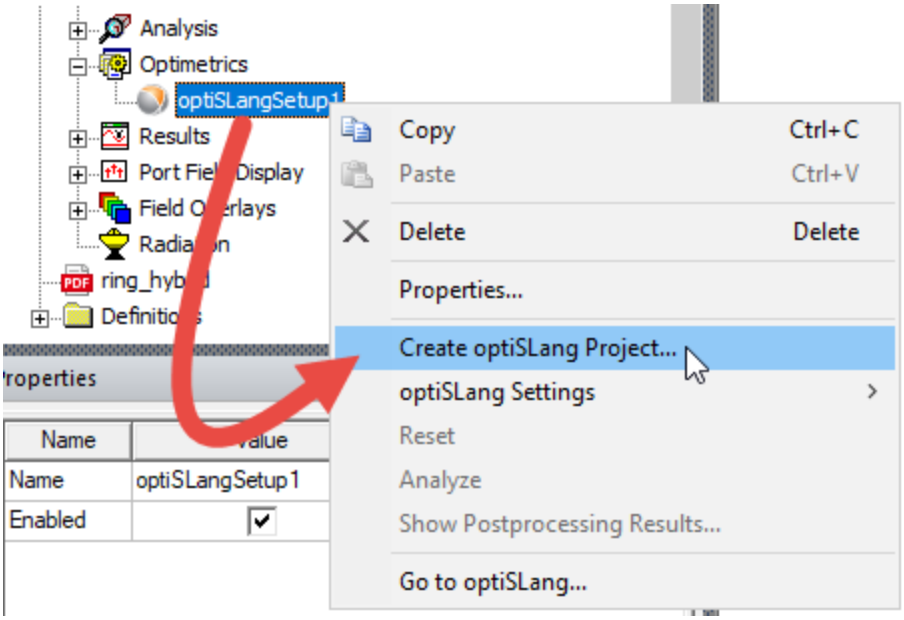
1. Prepare the AEDT project by ensuring:
   - Parameters exist.

     > **Note:**
     >
     > optiSLang scans parameters at both the project and design level. Hidden parameters are also scanned. See: Parametrization for optiSLang Integration.

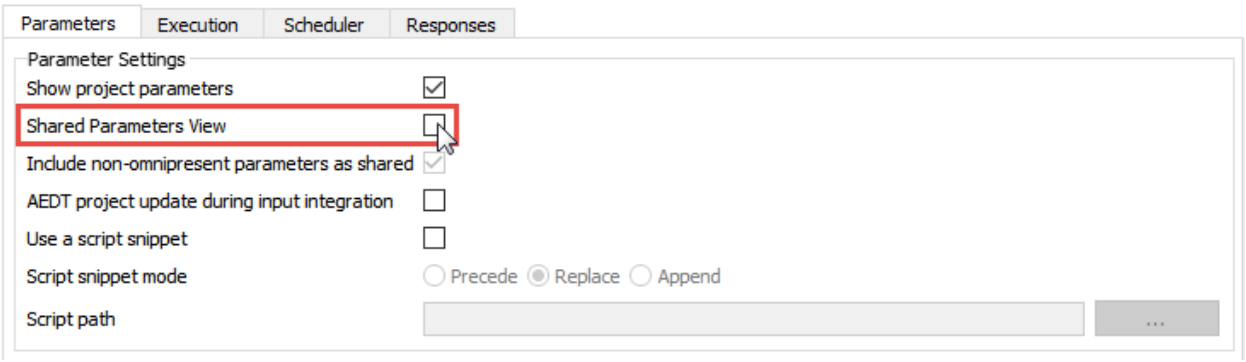
   - Project is solved.
   - Results reports exist and are prepared for the specific use case
     - For a standard use case, set all variables to nominal.
     - For a use case with a user-defined sweep, display all variations.

     See: Results and Reports for optiSLang Integration.
2. Invoke the Solver Wizard in optiSLang or create an optiSLang project directly from AEDT.





3. Create a one-node setup or a two-node setup.
   - A one-node setup registers both parameters and responses. This is the type of setup created by the wizard.
   - A two-node setup registers parameters in the first node and responses in the second. This is a more complicated setup that requires cloning the integration node, setting function switches, and setting absolute or relative paths. Consult the optiSLang help for more information.
4. View results in either tabular or tree view.
5. Enable **Shared Parameters View**. By default, parameters from different models are treated separately and a prefix based on the model name ensure uniqueness of names. In shared mode, sub-models are kept in sync and prefixes are unnecessary.

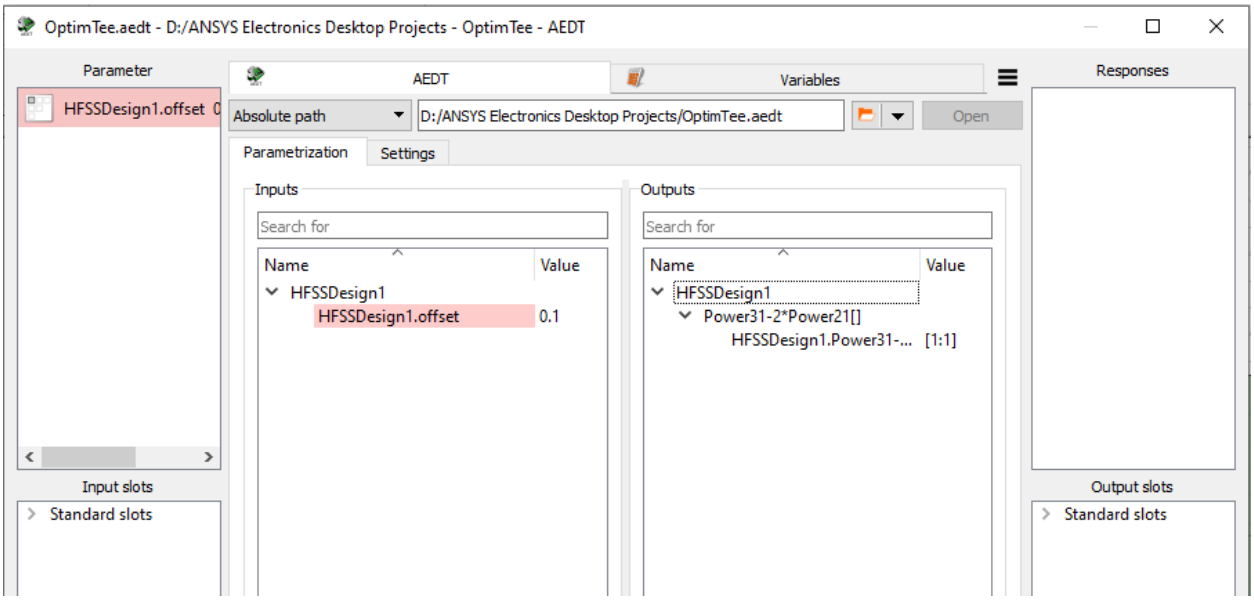


6. Reload the AEDT project in optiSLang to see the effects of changed settings.

Parametrization for optiSLang Integration

Certain AEDT data must exist before optiSLang can use it.

Variables act as **Inputs** in optiSLang, while simulation results (reports) act as **Outputs**.

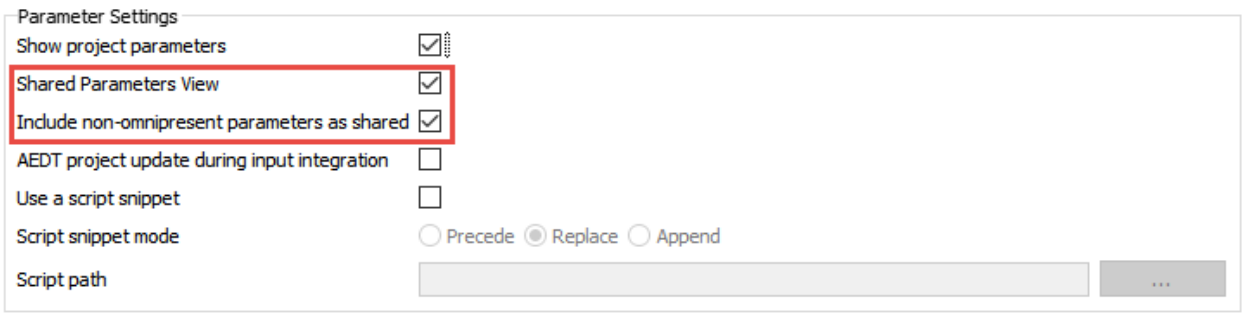


Inputs

When integrated with AEDT, optiSLang automatically detects project variables and design variables for every design in the project. Note that:

- Dependent parameters are skipped.
- Any parameter marked as "hidden" is always included.

Users decide whether to treat parameters from different models in the project as independent or communal (kept synchronized). In optiSLang, the AEDT integration has a **Shared Parameters View** with an additional **Include non-omnipresent parameters as shared** option.



In AEDT, project variables can be used to steer model parameters.

It is best to leverage features on both sides of the optiSLang-AEDT integration for managing overlapping (that is, partially synced) parameter sets. optiSLang's standard parametrization management tools offer the parameter type Dependent for aligning or otherwise controlling parameters.

In the context of bundling optiSLang-demanded designs for simultaneous execution as DSO jobs, remember that DSO design variation tables are hosted in the Optimetrics section of each design, and that there is no Optimetrics branch at the project level. So, even varying only project variables will still yield separate DSO tables and thus separate jobs for each model in the project.

A project walker mechanism inquires a given AEDT project and stores the generated project structure info in a JSON file for optiSLang. If the same reference project is linked for use in a different AEDT node even from a different optiSLang project, the stored project inquiry results can be recalled and the parametrization can be instantly displayed. This mechanism spares users wait time for the project inquiry call.

Outputs

The framework for result data transfer from AEDT to optiSLang is very simple: each report trace shows up in optiSLang response listings as a signal object. You can register the signals directly or apply math functions offered by optiSLang's calculator with each integration node. The transfer vessels are CSV files.

Even such plots as Smith charts or 3D polar plots can be collected as groups of signal objects using the uniform CSV file export function offered in AEDT.

Parameterization with optiSLang Analysis Goals in Mind

The parametrization scheme should connect to the goals of the optiSLang analysis.

Bad parametrization might hide the design optimum from the best optimizer or destroy the connection of a robustness sampling with reality. Thoughtfully improved parametrization can achieve simulation goals with a fraction of the designs that would be needed for a task within the most simple ad hoc parametrization scheme.

The algorithms being applied in a specific scenario should also guide your parametrization scheme. For example, it makes sense to give a robustness sampling algorithm the chance to examine a parameter space where the simulation can represent the relevant physical effects happening in the real world. An optimizer algorithm, meanwhile, needs the chance to hit feasible designs frequently.

Some tips:

- Think ahead to devise a stable and meaningful parametrization
- Be attentive to the kinds of errors encountered by failing designs

- Be quick to eliminate the origins of errors that appear most frequently in error logs
- Learn to use optiSLang post-processing as a diagnostic toolbox
- Apply feedback and lessons learned to your parametrization schemes

Results and Reports for optiSLang Integration

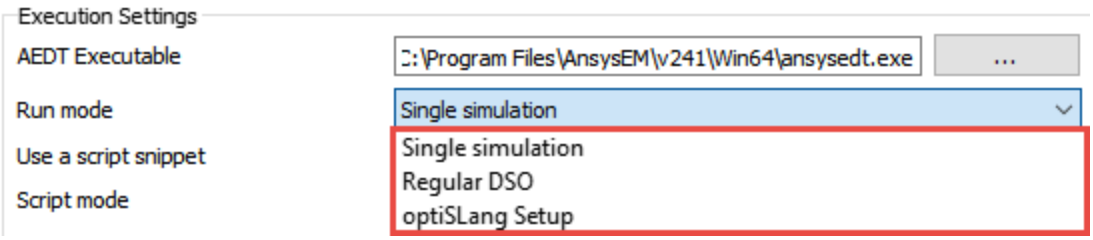
AEDT can export simulation reports as CSV files that can be read in optiSLang.

Additionally, all Optimetrics setups in AEDT (including optiSLang setups) have a **Calculations** tab, allowing for additional results. See: Setting up Calculations for Optimetrics.

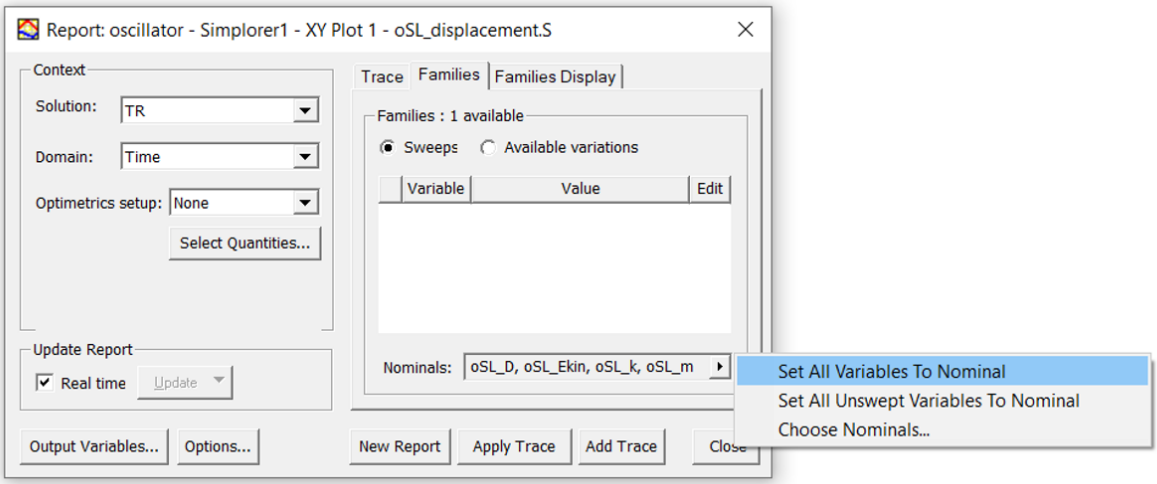
When the scripted AEDT integration machinery within optiSLang exports reports, it is just as if a user had exported a CSV file from within the AEDT UI. Any reports that can be exported as CSV (rectangular plots, data tables, polar plots, etc.) are sent in CSV format, and the reports' traces can be registered as responses in optiSLang. As a general rule, each data column of a CSV file is exposed as a signal trace.

> **Note:**
>
> For 3D Polar Plots, this can cause a small disadvantage by yielding, for example, 180 signal objects instead of a matrix. However, the simplicity of the routine achieves good performance and robustness, and allows the AEDT integration node in optiSLang to cover a large array of use cases.

Below, find considerations for setting up reports for use in optiSLang's different run modes.

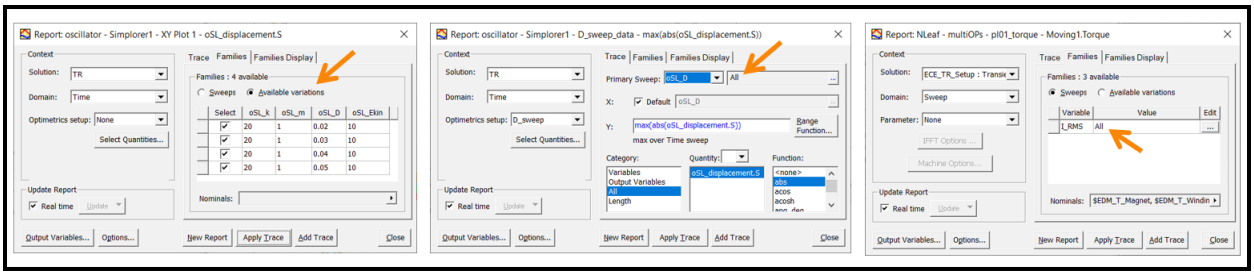


In optiSLang's **Single Simulation** run mode, optiSLang modifies the parameters of the nominal design and triggers the solution process. Therefore, reports used in this way should be configured to display the nominal design if they are to contain traces after modification and solution update. To ensure all report parameters are set to nominal, open the report settings and select the **Families** tab. From the **Nominals** field, select **Set All Variables to Nominal**.



In optiSLang's **Regular DSO** mode, the result export script cycles over all design variations of the Optimetrics setup created by previous script commands for holding the set of design variations demanded by optiSLang. In the loop, it applies the next design to be displayed as the nominal design, forces a refresh of all reports, and exports all reports. This loop is repeated for all design variations and the CSV files are stored each time in the corresponding design folder. Therefore, reports used in this mode should also be configured to display the nominal design. To ensure all report parameters are set to nominal, open the report settings and select the **Families** tab. From the **Nominals** field, select **Set All Variables to Nominal**.

In optiSLang's **optiSLang Setup** mode, the software solves a pre-existing Optimetrics setup (either serially or as a DSO job). This is commonly used to represent a parametric sweep or a prescribed set of operating points. In these use cases, reports should contain all data from all design variations to support a single export report call. To ensure the reports are adequate, open the report settings and enable all available variations:

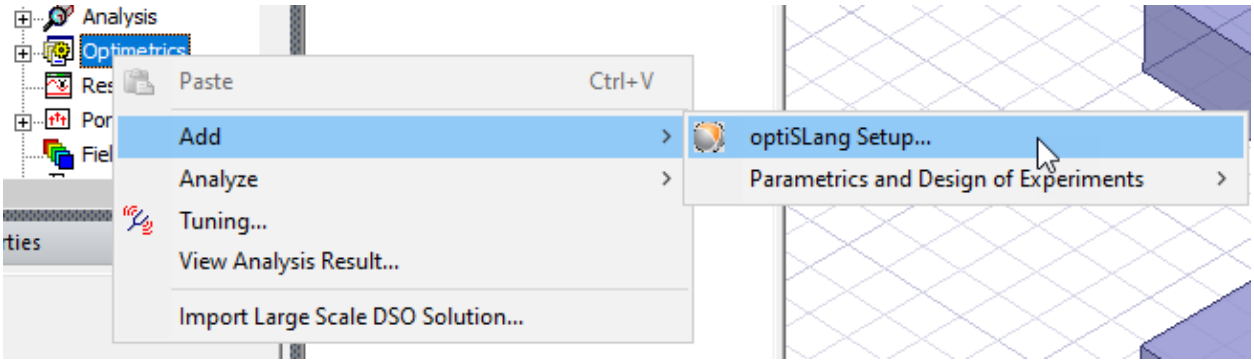


Creating an optiSLang Setup in AEDT

optiSLang setups are defined similarly to Optimetrics setups.

Create an optiSLang setup one of two ways:

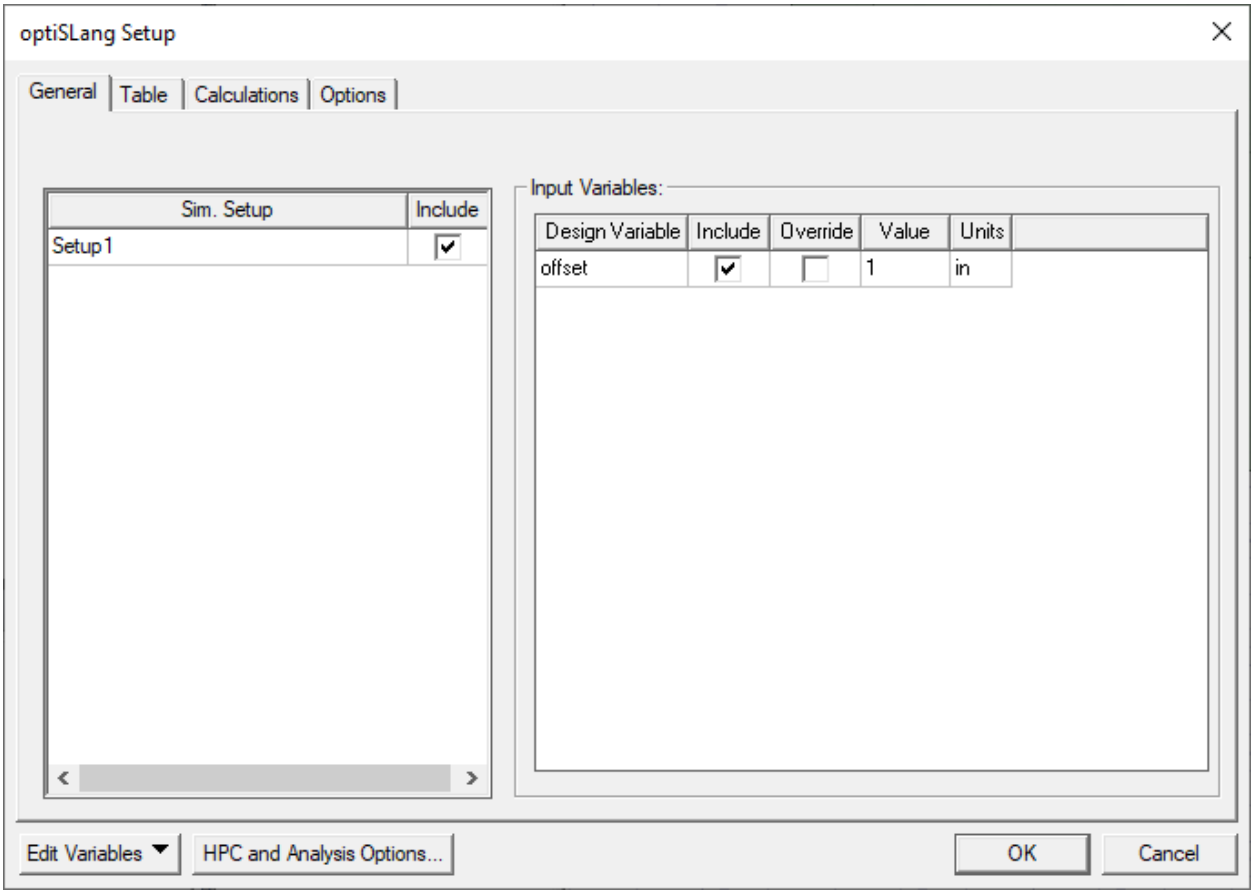
- From the **Project Manager**, right-click **Optimetrics** and select **Add** > **optiSLang Setup**.
- Click **Q3D Extractor** > **Optimetrics Analysis** > **Add optiSLang Setup**.



The **optiSLang Setup** window appears, with content based on the current design.

This window contains several tabs:

- **General** – lists solution setups and design variables associated with the current design, and allows you to specify which setups and variables to include, as well as to override a variable's value.



- **Table** – lists the values of the any variable(s) selected for inclusion.
- **Calculations** – allows for the setup of optiSLang calculations, similar to Optimetrics or Parametric setup. See: Setting up Calculations below.
- **Options** – lists mesh options. See: Setting Options below.

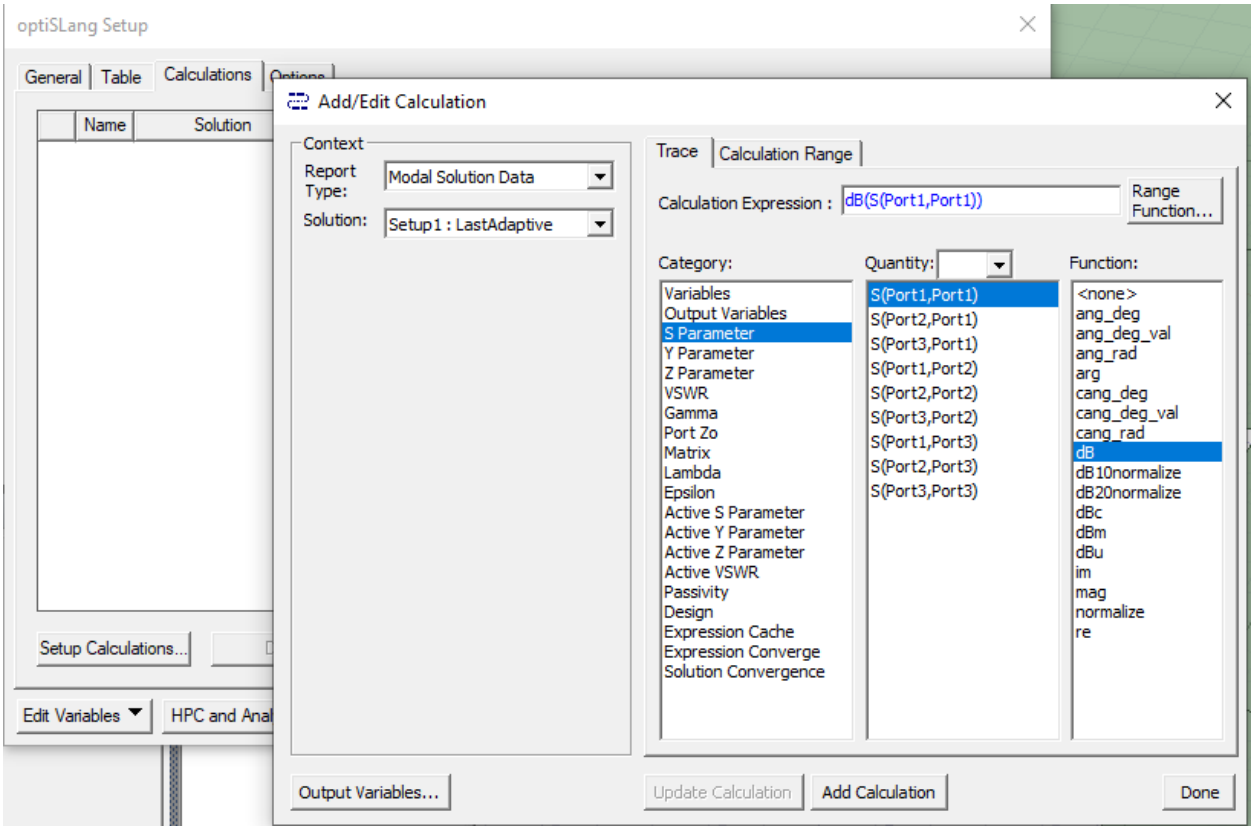
Each tab of the window also provides access to Edit Variables and HPC and Analysis Options.

Setting up Calculations

Initially, the **Calculations** tab is empty.

To set up a calculation, click **Setup Calculations**.

The **Add/Edit Calculation** window appears.



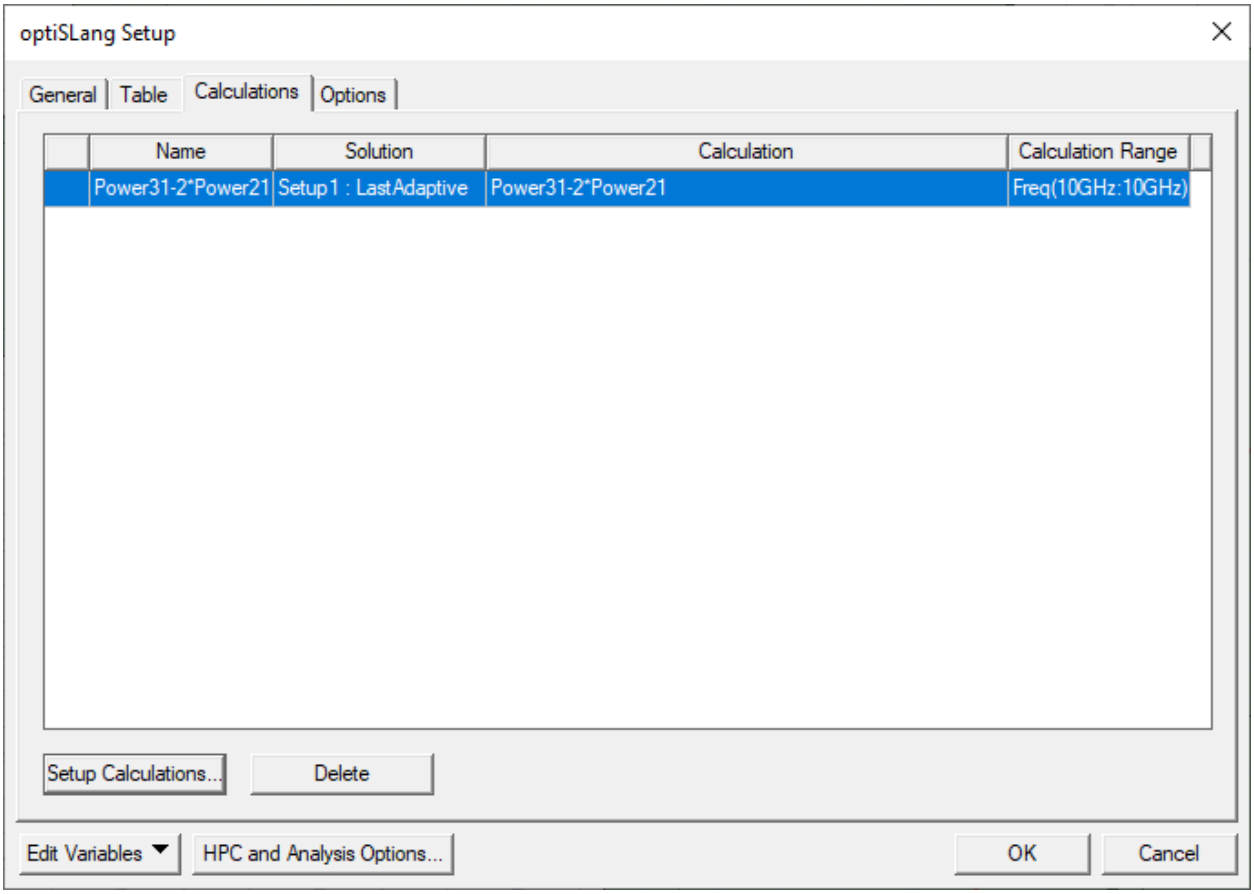
This window contains **Context**, **Category**, **Quantity**, and **Function** fields that can be used to build an optimization calculation expression, just as for other optimization or parametric calculations.

See: Setting Up Calculations for Optimetrics.

Once you have created an expression, click **Add Calculation** to add it to the **optiSLang** setup.

Click **Done** to close the **Add/Edit** window and return to the **Calculations** tab.

The new calculation is listed:

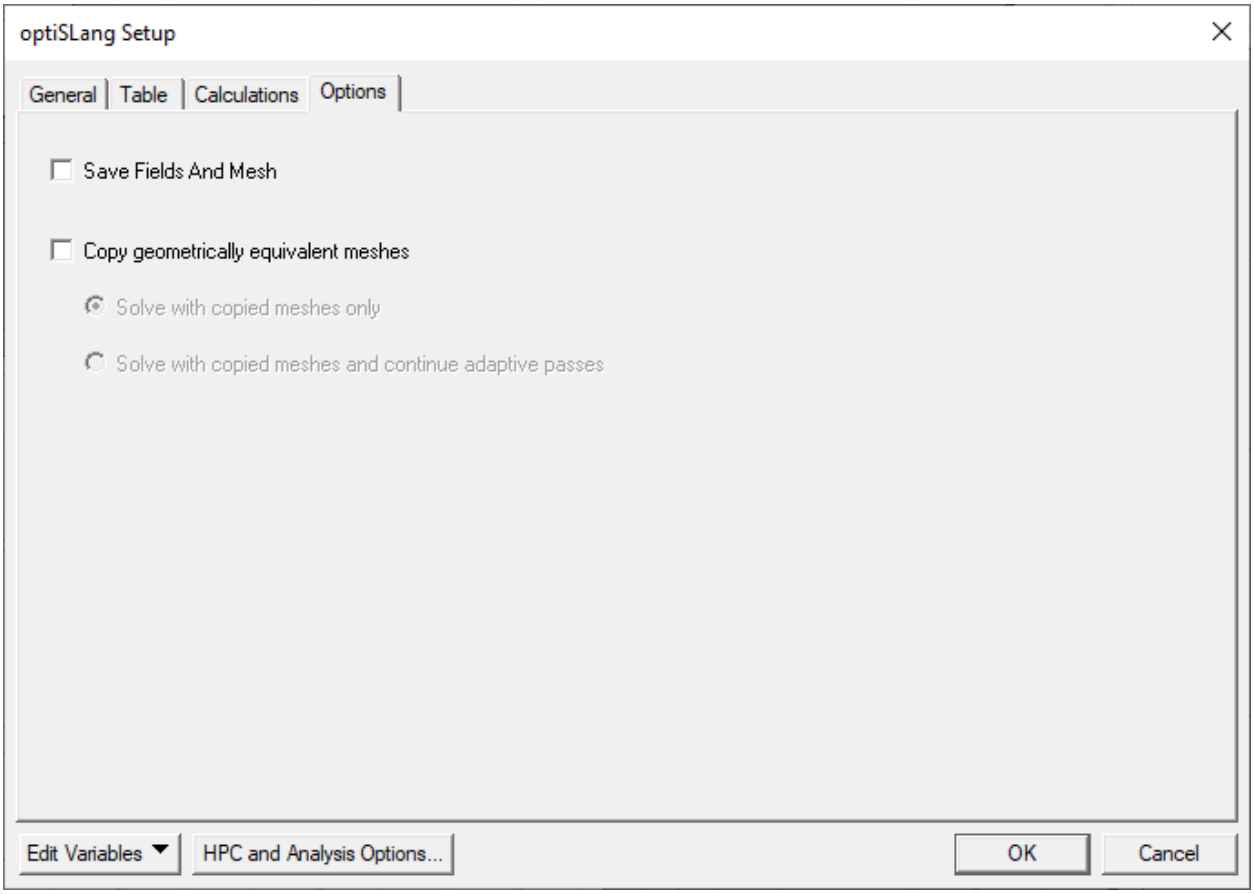


Setting Options

The **Options** tab contains options determining whether to **Save Fields and Mesh** and whether to **Copy geometrically equivalent meshes**.

In order to preserve disk space, by default AEDT does not save field solution data for every solved design variation. **Save Fields and Mesh** overrides this behavior.

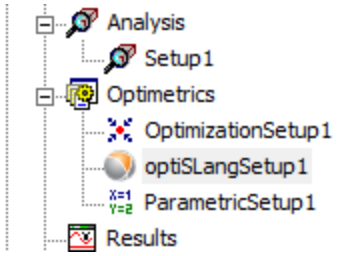
Copy geometrically equivalent meshes directs AEDT to copy a mesh that was calculated for one sweep variation for reuse on a geometrically equivalent sweep variation. See: Copying Meshes in Optimetrics Sweeps.



Finishing Setup

Click **OK** when you have completed the setup.

The optiSLang setup appears in the **Project Manager**, under **Optimetrics**:



Example optiSLang Setups

The following are example scenarios illustrating different considerations that might be taken when preparing an optiSLang setup. See: Solving an optiSLang Setup.

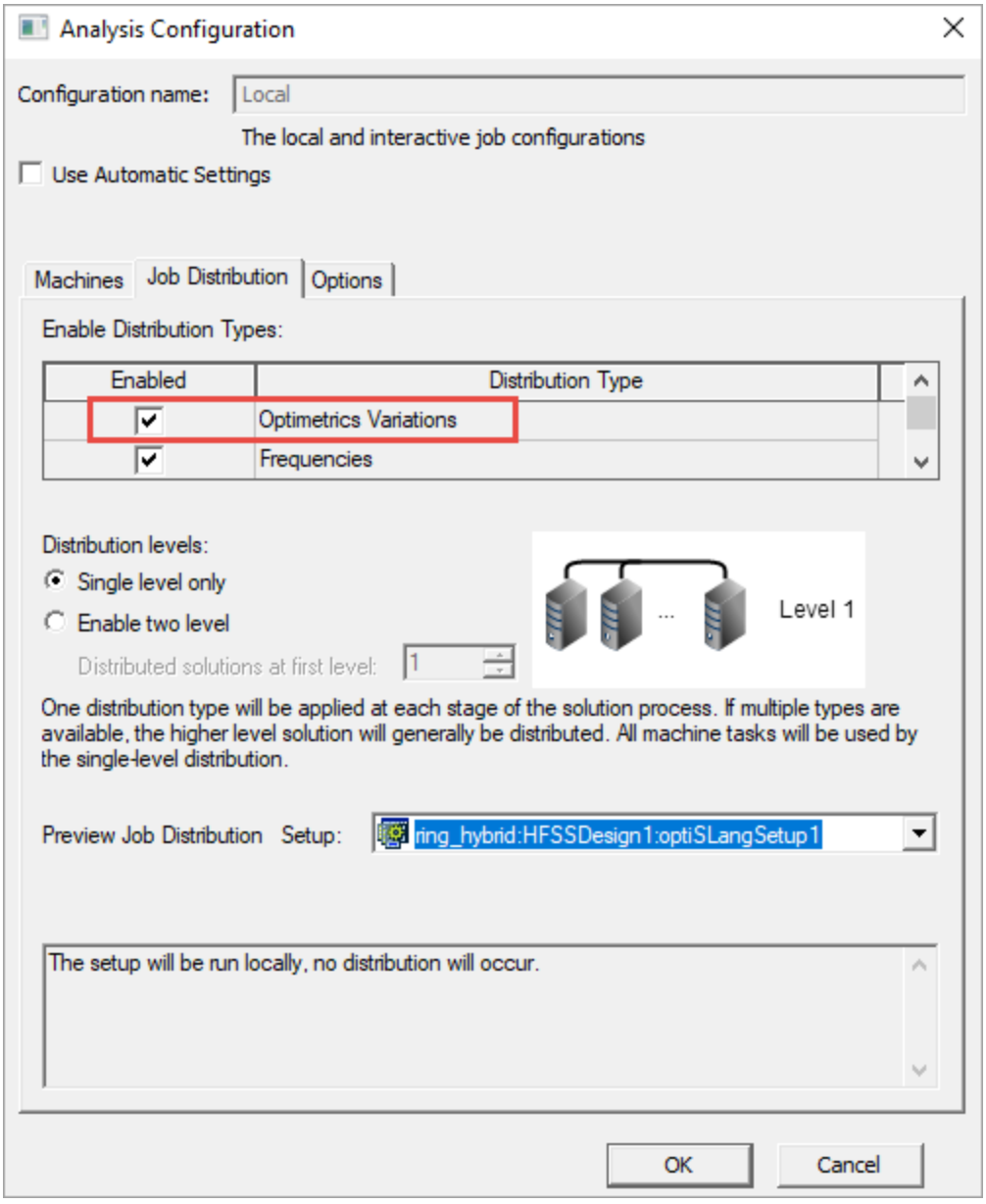
Settings in these examples are handled both in AEDT and in optiSLang.

Example 1: Solving a User-created, Pre-existing optiSLang Setup

Recall that optiSLang setups are handled as Optimetrics setups in AEDT.

By preparing an optiSLang setup in an AEDT project, you can create regular DSO jobs even though the AEDT integration node is left in the single simulation run mode. In every design folder, the clone of the reference project contains the optiSLang setup and its solution is triggered under the **Analyze All** umbrella. Depending on the HPC settings active in AEDT for the model type in question, this yields optiSLang DSO jobs.

A key setting is the check box for enabling **Optimetrics Variations** in the **Job Distribution** tab of the **Analysis Configuration** window:



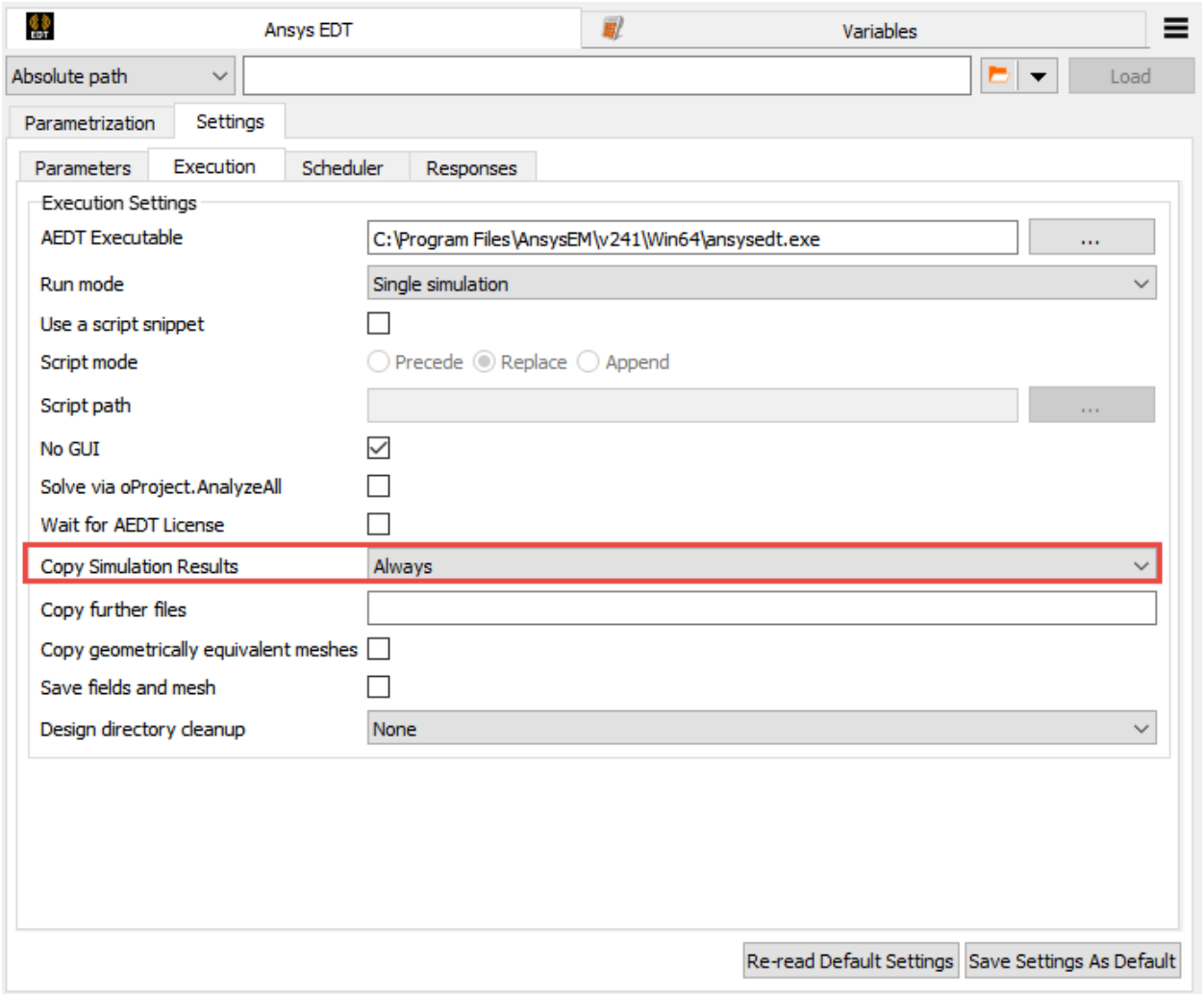
Note:

The **Use Automatic Settings** check box must be disabled to view this tab.

The working points being solved completely independently (simultanization) enables nice speed-up degrees for all designs within the optiSLang setup.

Example 2: Increasing Performance by Avoiding Wasteful Solution of the Nominal Design

In Example 1, performance drag can exist if the analysis setup of the nominal design is also solved in every optiSLang design folder in series with a desired Optimetrics setup. Keeping the optiSLang setting **Copy Simulation Results** on the value **Always** can avoid redundant solution of the nominal design at the cost of the time for copying the *.aedtresults folder of the reference design repeatedly into every design folder.



Example 3: Avoiding Redundant Computations while Exploiting AEDT Features around Postprocessing Variables

In optiSLang, an AEDT node can be used to copy a reference project along with its associated *.aedtresults directory of solution data. When this node is used for evaluating different combinations of postprocessing variables, AEDT does not call any solver and that the evaluation time per design is accordingly fast. The only drag consists in loading a fresh instance of the AEDT program for every new design.

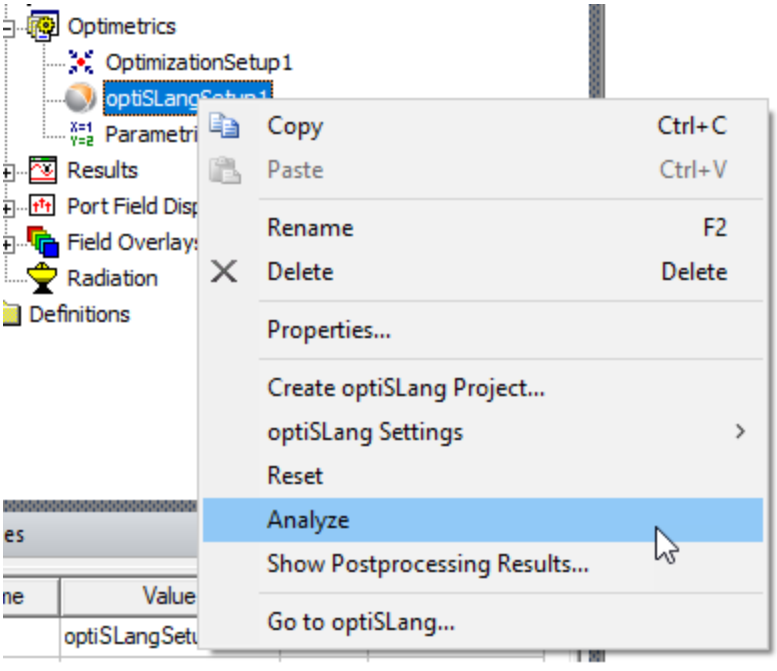
Example 4: Parametrization of an AEDT Design for optiSLang Integration

Consider a 2D Maxwell model of a permanent magnet synchronous motor. For design optimization, the magnet positions and angles can be the crucial parameters and due to symmetry a quarter model can represent the entire machine. For questions of robustness, reliability, tolerances, and achieving specs, it might be necessary to parametrize eccentricity and ellipticity. If magnet quality scatter is important, this might be another aspect requiring a full 360 degree model because it would be drastically unrealistic to assume that 16 worst-case magnets can find their way into one single machine. See: Parametrization for optiSLang Integration.

Solving an optiSLang Setup

To analyze an optiSLang setup within Electronics Desktop:

- From the **Project Manager**, right-click a setup and select **Analyze**.



Considerations for Using the Analyze All Command

The **Q3D Extractor** > **Analyze All** command solves all projects in a design, including Optimetrics setups. See: Running More Than One Simulation.

> **Warning:** Before using the **Analyze All** command, remove any unnecessary analysis setups from the project.

Some reasons for doing so:

- During regular DSO run mode, optiSLang creates Optimetrics DSO design tables. By leaving just one setup for analysis, there is no ambiguity about which analysis setup is associated with any new table.
- Numerical convergence settings steer the performance trade-off between speed and accuracy. If a small number of designs take exceptionally long to converge, it can make a substantial performance difference whether designs are solved independently or in groups (DSO jobs). In group jobs, one slowly converging simulation can delay the entire group. This increases the importance of upper bounds on iterations for mesh refining, interpolation point addition, and so on.
- The unintended presence of Optimetrics setups (including optiSLang setups) can lead to long execution times. When used intentionally, however, Optimetrics setups can be used to define parametric sweeps. These sweeps are easy to set up and represent the sweep per design as a nested system in optiSLang. Note that reports must be adjusted accordingly, so that data is drawn from the entire sweep and not a single point, in order to reach the optiSLang database.
- Analysis setups contain settings that drastically impact the resulting file and folder sizes (for example, the **Save Fields** check box). If saved fields consume disk space in the range of gigabytes for a single project, then the creation of hundreds of design folders with solution data by optiSLang can quickly fill a hard drive.

Considerations for Distributed Analysis

There are additional considerations for using Distributed Analysis and Large-Scale DSO to analyze optiSLang setups.

- Using the AEDT node's run mode "regular DSO" is not the only way to create DSO jobs. Also in run mode "single simulation" DSO jobs can be spawned. For example, due to the **Analyze All** command, a user-created pre-existing Optimetrics setup could be solved for every one of optiSLang's design evaluations.
- The integration node's **Designs Per Execution** setting can work with HPC and DSO settings for AEDT solvers in many different ways. If set to 4, optiSLang can send 4 designs per AEDT call when AEDT is geared at solving 4 design variations simultaneously. If **Designs Per Execution** is set to 9999, however, each optiSLang

algorithm will always send all designs waiting in the pipeline at once (a robustness sampling of 1000 designs can be cast into one single DSO job). Using HPC settings, AEDT can still be made to break it down with ntasks=ncores=4. The drawback is that the success/failure info becomes visible in optiSLang only at the very end of the job. A large value for **Designs Per Execution** can make sense for optimization algorithms demanding design sets of fluctuating size.

- With population-based optimizers, it can be useful to match the population size with compute resources to avoid job load fluctuations.
- Several optiSLang nodes offer the feature to list files to be copied from the reference location into each design folder. However, this may not work for directories. A simple solution is to use a Python node and the shutil.copytree command.
- Avoid redundant solve actions. It may be helpful to copy not only the bare *.aedt file but also the *.aedtresults directory into each design folder. When the AEDT node issues the **Analyze All** command on the project level, pre-existing solution data can reduce the scope of analysis setups that are triggered to be solved. Compare the time for copying the data of solved projects against the time gain due to less computation. Avoid disc space concerns by activating design directory purging rules to be applied to every design folder after finished evaluation.
- AEDT offers a special type of postprocessing variables. When they are varied while other parameters stay constant, a new design variation can be evaluated with very little computational burden as compared to solving from scratch. The AEDT integration node provides copy functionalities for *.aedtresults folders allowing to benefit from AEDT features around postprocessing variables in certain use cases.

optiSLang Menu Options in AEDT

Once an optiSLang setup has been created in AEDT, additional menu options become available.

Right-click options include:

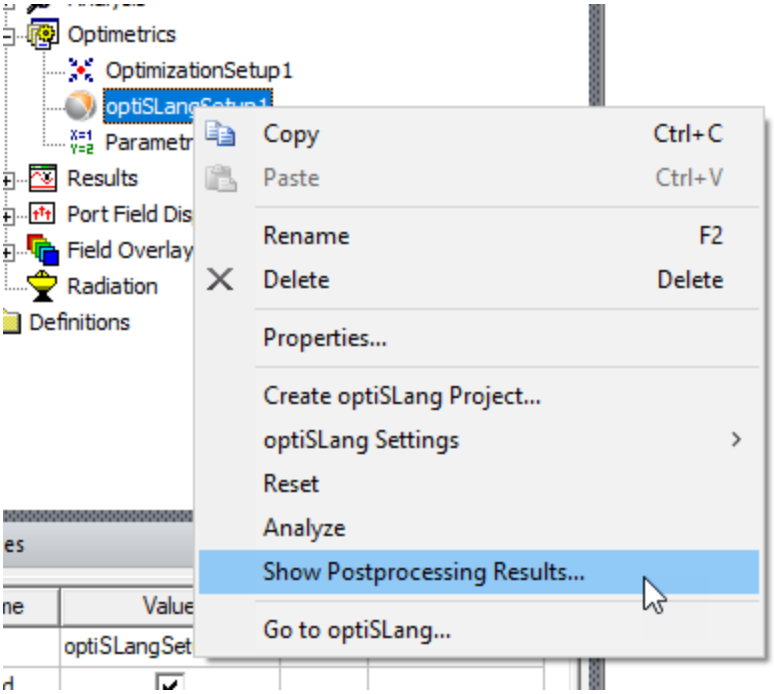
- **Create optiSLang Project** – creates an optiSLang project from the current design.
- **Edit AEDT Node Settings** – launches optiSLang with the AEDT project connected, on the Parametrization tab. For more information on node settings, consult the optiSLang help.
- **Edit Algorithm settings** – launches optiSLang and opens the Analysis Wizard, with options to analyze for Sensitivity, Optimization, or Robustness/Reliability. For more information on the Analysis Wizard, consult the optiSLang help.
- **Reset** – resets the optiSLang project.
- **Analyze** – launches the analysis of the optiSLang Optimetrics setup. When it has finished, post-processing results are available.
- **Show Postprocessing Results** – launches an optiSLang postprocessing results

window. This menu option only available after analysis has been performed.

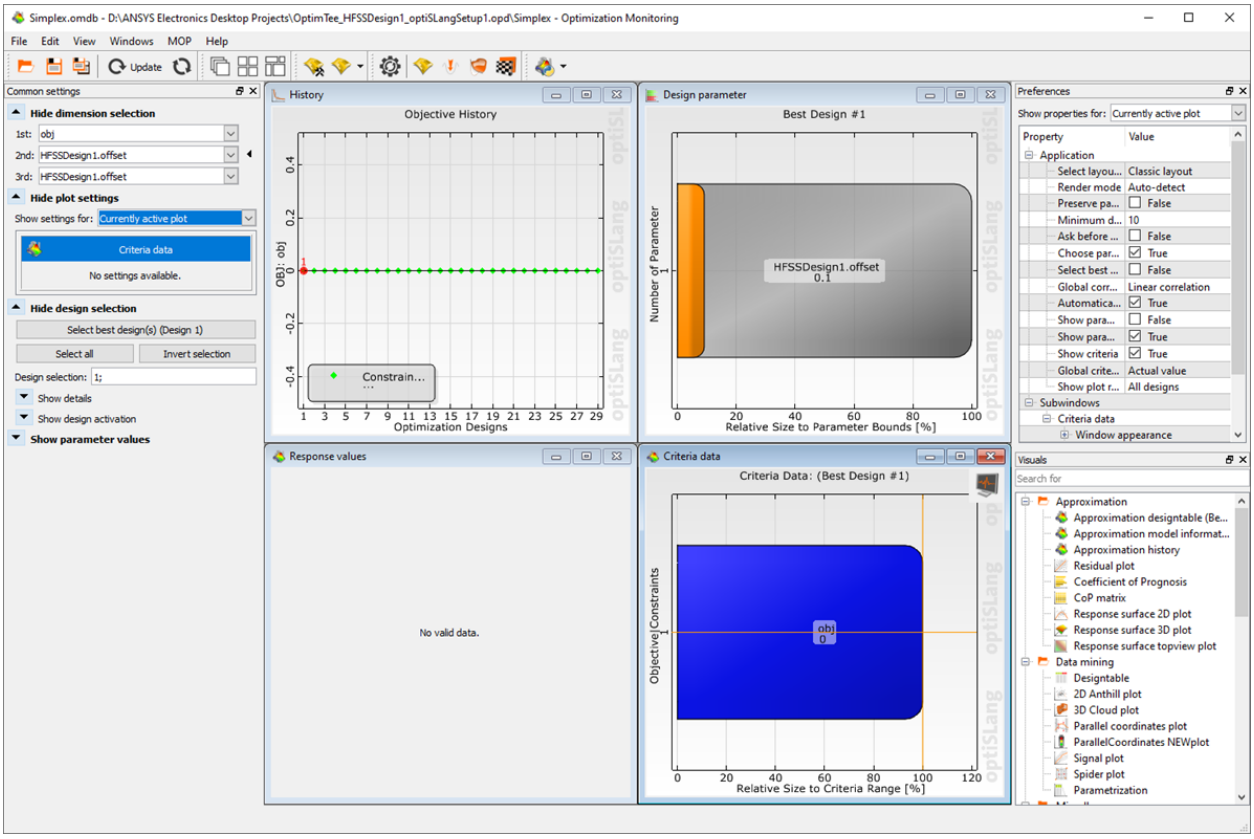
- **Go to optiSLang** – launches optiSLang.

Viewing optiSLang Postprocessing Results

Once you have created and run an optiSLang setup, right-click it and select **Show Postprocessing Results**.



An optiSLang window opens, showing postprocessing features.



For more information, use optiSLang's **Help** menu to access the optiSLang online help.

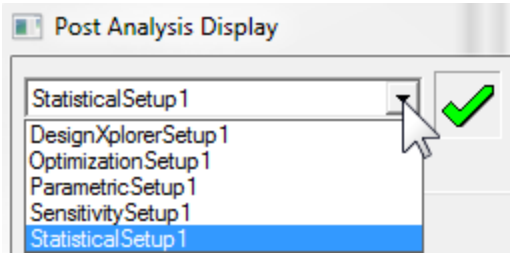
Viewing Analysis Results for Optimetrics Solutions

To view data specific to an Optimetrics solution:

1. In the project tree, right-click the Optimetrics setup for which you want to view the results, and select **View Analysis Result**.

   The **Post Analysis Display** dialog box appears.

2. Use the drop-down menu to select from available setups.



3. Click the **Results** tab to view results in plot or table form.

   When you view results in Table format, you can sort the results based on each column. Click a column's header to sort. Click again to invert the current sort.

4. Click **Options...** to open a dialog box that allows you to specify the Maximum number of significant digits to display when showing the analysis result. The default is 4.
5. Select the **Profile** tab to view start, stop, and elapsed times for each variable, and the analysis machine for each variation. Click a column heading to sort the table by variation number, variable value, start, stop, elapsed time, or machine.

See the help topics in this section for more details about viewing Optimetrics analysis results.

Viewing Solution Data for an Optimetrics Design Variation

To view the convergence information, or 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** window and via the **Results** > **Apply Solved Variation** command.

1. Click the **Q3D Extractor** menu 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.

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 Q3D Extractor as it solved each design variation. Optimetrics writes the variation information to the profile table before the solve. It then updates the entry with end data (end time, elapsed time, etc.) once the solve variation is completed.

1. In the project tree, right-click the Optimetrics solution setup of interest, and select **View Analysis Result**.

   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 drop-down menu 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)

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 drop-down menu 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 **Modeler** window is changed to represent the selected design variation.

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 select **View Analysis Result**.

   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** drop-down menu.
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** drop-down menu.

   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 select **View Analysis Result**.

   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**.

   The software 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.

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 select **View Analysis Result**.

   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**.

   The software 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.

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 select **View Analysis Result**.

   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** drop-down menu.
4. Select the output parameter results you want to plot on the y-axis from the **Y** drop-down menu.

   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 select **View Analysis Result**.

   The **Post Analysis Display** dialog box appears.

2. Select the statistical setup with the results you want to view from the drop-down menu 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 **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** drop-down menu.

   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).

   The software now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window changes to represent the selected design variation.

Click **Revert** to return the design in the view window to the original value.

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 select **View Analysis Result**.

   The **Post Analysis Display** dialog box appears.

2. Select the statistical setup with the results you want to view from the drop-down menu 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** drop-down menu.

   A histogram plot appears in the **Post Analysis Display** dialog box. It displays the distribution of the selected solution quantity.

Saving Field Solutions for Optimetrics Analyses

In order to preserve disk space, by default Ansys Electronics Desktop 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 sweep. If the nominal design is not included in the optimization analysis, all field solutions are deleted.

You can change the general options for a particular design type (HFSS, Q3D, and so on) so that any new Optimetrics setup you define will default to saving the fields. The instructions to do so are included on this page.

Alternatively, you can choose to save the fields on an individual setup basis, without enabling the option to save fields by default. See the subtopics listed in the Related Topics section at the bottom of this page for instructions to do so for each type of optimetrics setup.

To save the fields for all design variations, change the default setting for all projects:

1. Using the menu bar, select **Tools** > **Options** > **General Options**, or from the **Desktop** ribbon tab, click  **General Options**.

   The **Options** dialog box appears.

2. In the tree on the left side of the dialog box, expand the branch for the product/design type of interest (HFSS, Q3D, Mechanical, and so on).

3. Select **General**.

   The appropriate group of options appears on the right side of the dialog box.

4. Select **Save Optimetrics field solutions**.
5. Repeat steps 2 through 4 for all products in which you want to save field results by default for all new Optimetrics setups.

   The **Save Fields** option will now be selected by default when you create a new Optimetrics setup in each product for which you selected the **Save Optimetrics field solutions** option.

Saving Field Solutions for a Design of Experiments Setup

In order to preserve disk space, by default Ansys Electronics Desktop 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 Design of Experiments 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. Right-click **Optimetrics** in the Project Manager and choose **Add** > **Design of Experiments** from the shortcut menu to create a new Design of Experiments setup.

   Alternatively, to change the settings of an existing setup, right-click the desired Design of Experiments setup under **Optimetrics** in the Project Manager and choose **Properties**.

   The **Design of Experiments Setup** dialog box appears.

2. Select the **Options** tab.
3. Select **Save Fields And Mesh**. 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**.

   Ansys Electronics Desktop will save the field solution data for every solved design variation in the design of experiments setup.

> **Note:**
>
> Alternatively, you can modify the Ansys Electronics Desktop General Options to save fields by default for any new Optimetrics setup. See: Saving Field Solutions for Optimetrics Analyses for instructions.
>
> Changing the default setting in the General Options has no effect on previously defined Optimetrics setups.

Saving Field Solutions for a Parametric Setup

In order to preserve disk space, by default Ansys Electronics Desktop 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. Right-click **Optimetrics** in the Project Manager and choose **Add** > **Parametric** from the shortcut menu to create a new parametric setup.

   Alternatively, to change the settings of an existing setup, right-click the desired Parametric setup under **Optimetrics** in the Project Manager and choose **Properties**.

   The **Setup Sweep Analysis** dialog box appears.

2. Select the **Options** tab.
3. Select **Save Fields And Mesh**. 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**.

   Ansys Electronics Desktop will save the field solution data for every solved design variation in the parametric setup.

> **Note:**
>
> Alternatively, you can modify the Ansys Electronics Desktop General Options to save fields by default for any new Optimetrics setup. See: Saving Field Solutions for Optimetrics Analyses for instructions.
>
> Changing the default setting in the General Options has no effect on previously defined Optimetrics setups.

Saving Field Solutions for an Optimization Setup

In order to preserve disk space, by default Ansys Electronics Desktop 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. Right-click **Optimetrics** in the Project Manager and choose **Add** > **Optimization** from the shortcut menu to create a new optimization setup.

Alternatively, to change the settings of an existing setup, right-click the desired Optimization setup under **Optimetrics** in the Project Manager and choose **Properties**.

The **Setup Optimization** dialog box appears.

2. Select the **Options** tab.
3. Select **Save Fields And Mesh**. 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**.

   Ansys Electronics Desktop will save the field solution data for every solved design variation in the optimization setup.

> **Note:**
>
> Alternatively, you can modify the Ansys Electronics Desktop General Options to save fields by default for any new Optimetrics setup. See: Saving Field Solutions for Optimetrics Analyses for instructions.
>
> Changing the default setting in the General Options has no effect on previously defined Optimetrics setups.

Saving Field Solutions for a Sensitivity Setup

In order to preserve disk space, by default Ansys Electronics Desktop 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. Right-click **Optimetrics** in the Project Manager and choose **Add** > **Sensitivity** from the shortcut menu to create a new sensitivity analysis setup.

   Alternatively, to change the settings of an existing setup, right-click the desired Sensitivity analysis setup under **Optimetrics** in the Project Manager and choose **Properties**.

   The **Setup Sensitivity Analysis** dialog box appears.

2. Select the **Options** tab.
3. Select **Save Fields And Mesh**. 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**.

Ansys Electronics Desktop will save the field solution data for every solved design variation in the sensitivity analysis.

> **Note:**
>
> Alternatively, you can modify the Ansys Electronics Desktop General Options to save fields by default for any new Optimetrics setup. See: Saving Field Solutions for Optimetrics Analyses for instructions.
>
> Changing the default setting in the General Options has no effect on previously defined Optimetrics setups.

Saving Field Solutions for a Statistical Setup

In order to preserve disk space, by default Ansys Electronics Desktop 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. Right-click **Optimetrics** in the Project Manager and choose **Add** > **Statistical** from the shortcut menu to create a new statistical analysis setup.

   Alternatively, to change the settings of an existing setup, right-click the desired Statistical analysis setup under **Optimetrics** in the Project Manager and choose **Properties**.

   The **Setup Statistical Analysis** dialog box appears.

2. Select the **Options** tab.
3. Select **Save Fields And Mesh**. 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**.

   Ansys Electronics Desktop will save the field solution data for every solved design variation in the statistical analysis.

Note:

Alternatively, you can modify the Ansys Electronics Desktop General Options to save fields by default for any new Optimetrics setup. See: Saving Field Solutions for Optimetrics Analyses for instructions.

Changing the default setting in the General Options has no effect on previously defined Optimetrics setups.

Adding an Expression in the Output Variables Window

When you are in the **Output Variables** window (after clicking **Edit Calculation** from 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 drop-down menus.
    b. Select a **Category**, **Quantity**, and **Function** from the lists, and click **Insert Quantity Into Expression**.
    c. If you want to insert a specific predefined function, select one from the **Function** drop-down menu, and click **Insert Function**.
3. You can also type numbers or expression by hand directly into the **Expression** area.

Copying Meshes in Optimetrics Sweeps

An option in the Optimetrics Analysis setups allows you to request Ansys Electronics Desktop 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 intend to set up.
2. Select Ansys Electronics Desktop and then select the appropriate **Optimetrics** > **Add** command to display the associated **Setup** dialog box.
3. Click the **Options** tab in the setup dialog box.

4. Select **Copy geometrically equivalent meshes**.
    - When this option is enabled, you can additionally select **Solve with copied meshed only** or **Solve with copied meshes and continue adaptive passes**. The **Solve with copied meshes only** option is not available for Maxwell 3D/2D magnetic and electric transient designs. It is available for HFSS Transient but does not apply for a Transient solve setup with a mesh link.

   Ansys Electronics Desktop copies the mesh 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.

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**.

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

$$cost(x): R^n \rightarrow R$$

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 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} parameter.

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 drop-down menu.
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.

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.

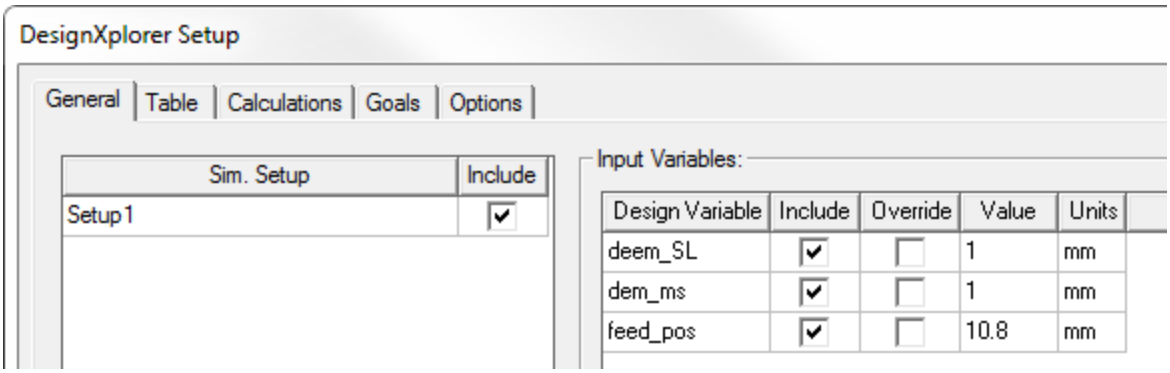
Link to DesignXplorer

You can export a .xml file containing information on a Q3D Extractor setup, optimization variables, and output variables that enables Ansys Design Xplorer to manage the simulations (for example, for design of experiments and optimization). Design Xplorer will launch Ansys Electronics Desktop simulations of design variations and evaluate the outputs.

To do so:

1. Click your product on the menu bar and then **Optimetrics Analysis** > **Add Design Xplorer Setup** or right-click on **Optimetrics** in the **Project Manager** window, and select **Add Design Xplorer Setup** from the short-cut menu.

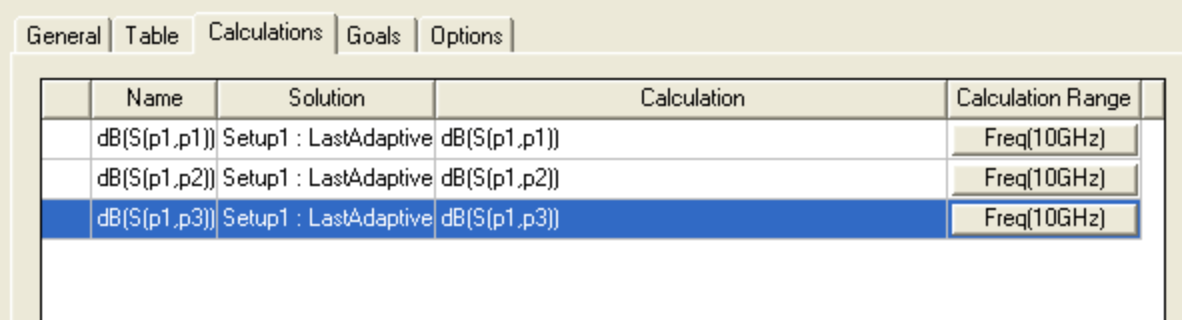
   This opens the **Design Xplorer** dialog box with the **General** tab selected. it lists the setups available in the current project, and the input variables it contains.



2. Select **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 set up any output calculations, click the **Calculation** tab and click **Setup Calculations**.

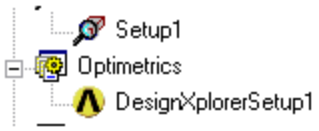
   This opens the **Add/Edit Calculation** dialog box. Here you can define the simulation results of interest. The dialog box contains distinct panes and tabs to set the **Context**, the **Calculation Expression**, and the **Calculation Range**. See: Setup Calculations for Optimetrics for details.

   Use the **Add Calculation** button to add expressions to the Calculations table.

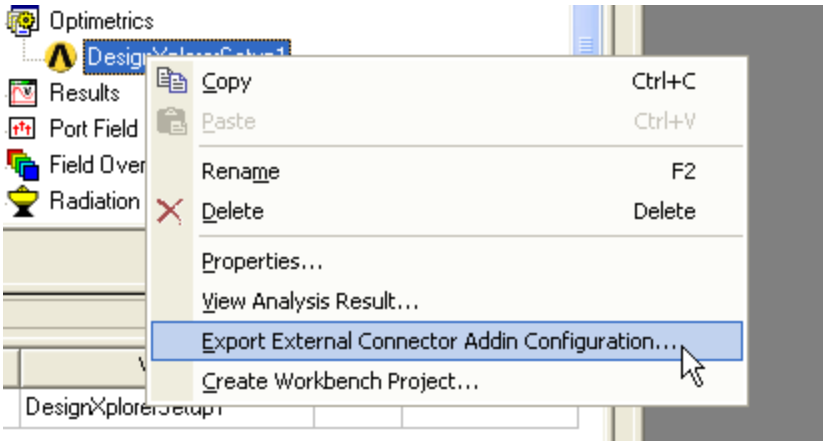


5. When you have added the calculations of interest, click **OK** to save the setup.

   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 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. This file contains information regarding the 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. You should have provided a path to the Workbench installation in the **Tools** > **General Options** dialog box Miscellaneous tab, to provide a path.

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 project established. To this connection, you can add a Design Xplorer Setup. See the documentation of Ansys Workbench for details on Design Xplorer.

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.

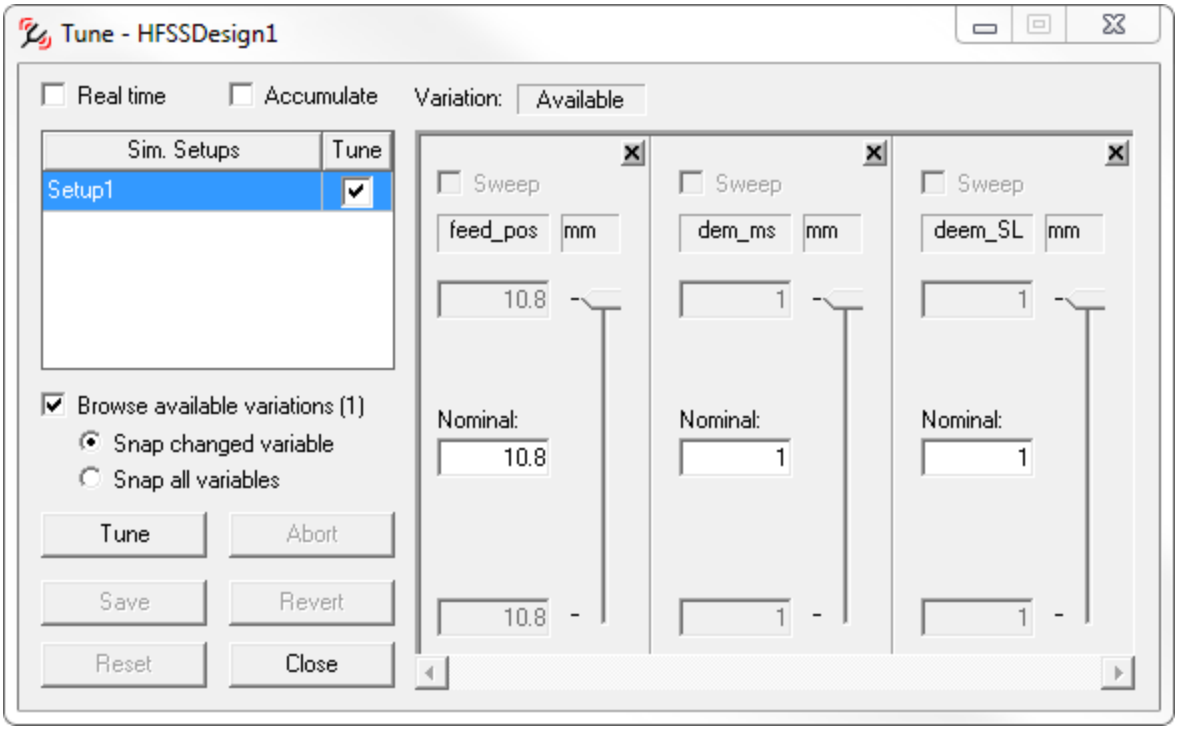
Tuning a Variable

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. After running the simulation, click the product in the menu bar and then select **Tune**.

   Alternatively, right-click **Optimetrics** in the Project Manager and choose **Tuning** from the shortcut menu.

   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 to use when it solves the specified design variation.

   The analysis is solved 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 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 to solve a range of values, specify a linear range of values with a constant step size:

   a. Select the **Sweep** check box. (You must have cleared the **Browse available variations** 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. The model is solved 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 box 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 box 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 box and go back to the **Tune** dialog box.

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 associated setup dialog box.

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
- When saving a tuned state
- When reverting to a tuned state

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**.

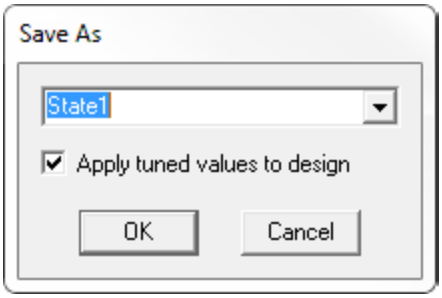
   The variable values from the solved design variation become the current variable values for the nominal design. If you have applied variant values, you should see the new values listed in the relevant Design or Project Properties lists of variables and values, and if the changes affect plots or physical features of a model, those changes should also be apparent.

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.

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 drop-down menu.
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.

Resetting Variable Values after Tuning

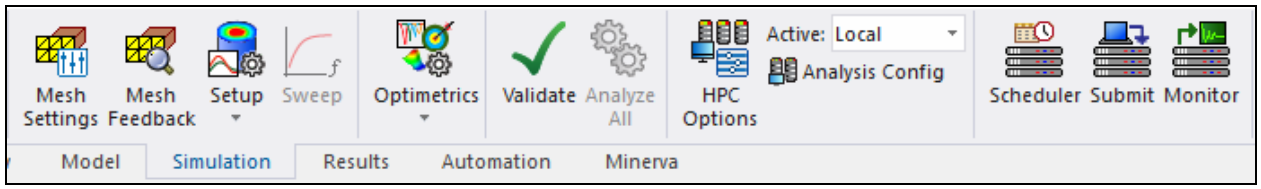
If you want to reset variable values to the values they were set to when you started the current session of the **Tune** dialog box:

- After tuning a variable, click **Reset** in the **Tune** dialog box.

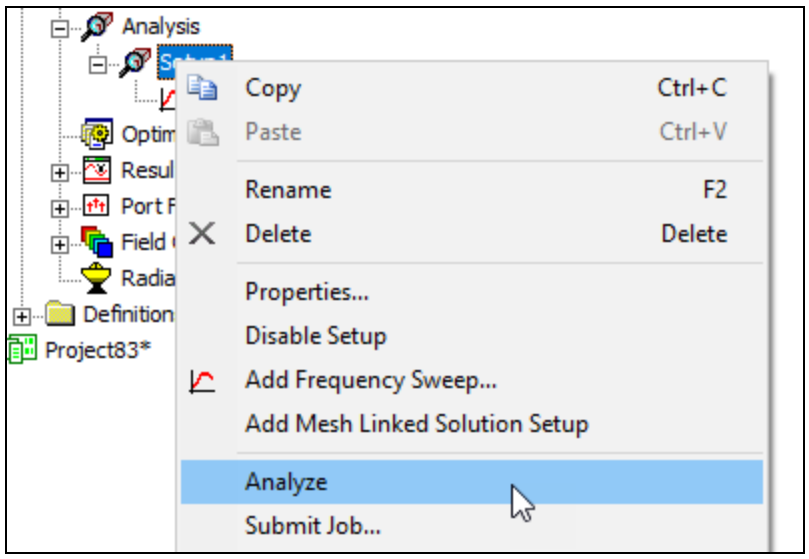
  Solutions for the design variations solved during tuning analyses remain available for post processing.

16 - Running Simulations

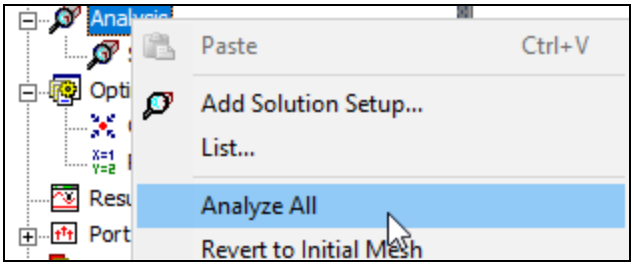
After you specify how Ansys Electronics Desktop is to compute the solution, you need to begin the solution process. You can access the **Analyze** commands for a specific setup by right-clicking on a solution or sweep in the **Project** tree and selecting from the shortcut menu, you by selecting the **Analyze All** from the solver menu, or the **Analyze All** icon on the **Simulation** tab of the ribbon.



In general, the **Analyze** command on the shortcut menu applies to the selected setup and associated sweeps, if any, or to a selected sweep. To use this command, right-click on a setup or sweep in the Project Manager and click the command on the context menu.



The **Analyze All** command applies to all, and sweeps at or below the level invoked in the Project Manager. To use this command, either click **Q3D Extractor**> **Analyze All** or right-click the **Analysis** icon in the *Project Manager* and select **Analyze All**.



Important:

If the UI shows an error with a code, it means that solver exited unexpectedly.

What do you want to do?

- Local Analysis
- Solve a single setup with or without sweeps (when applicable)
- 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.
- Submit Job to RSM or HPC Scheduler
- Monitor queued simulations
- 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

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. If running a distributed analysis, consult Distributed Analysis.

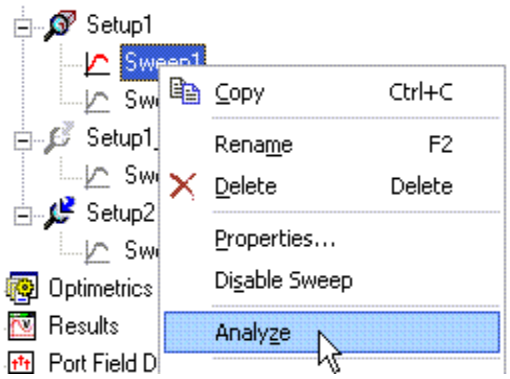
> **Note:**
>
> MPI distribution may be affected by starting or stopping a VPN, even if the MPI distribution is limited to the local host.

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.

Solving a Single Setup or Sweep

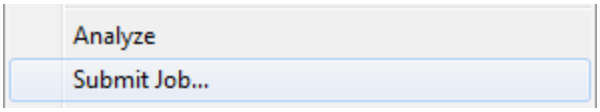
To run a single setup or sweep:

1. Select a solution setup or sweep in the Project Manager.
2. Right-click and select **Analyze** from the shortcut menu. The graphic shows a single sweep selected for analysis.



The 3D field solution is computed inside the structure for a solution.

When you right-click a Setup, rather than a Sweep, the shortcut menu also includes the **Submit Job...** command.



For more information on the **Submit Job...** command see, Distributed Analysis and High Performance Computing (HPC) Integration.

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), Ansys Electronics Desktop will not allow another dependent simulation to start until the first use of the source has completed.

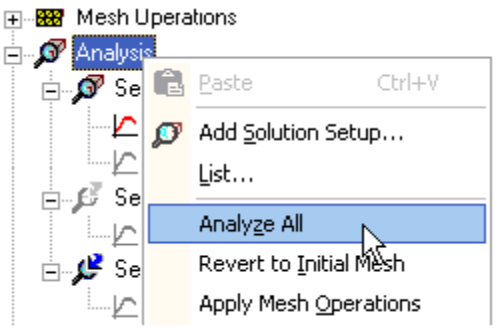
Running More than One Simulation

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 **Q3D Extractor**> **Analyze All**.

Each enabled solution setup is solved in the order it appears in the Project Manager.

The example here shows an analysis invoked from the Project Manager shortcut menu with three setups: one disabled, two enabled. The first setup has one sweep enabled, and one disabled (shaded icon). The second setup is disabled, and the third is enabled, with a disabled sweep.



Note:

The **General** tab for the Setup includes an **Enabled** check box. By default, this is checked. Clearing the **Enabled** check box excludes a setup from running

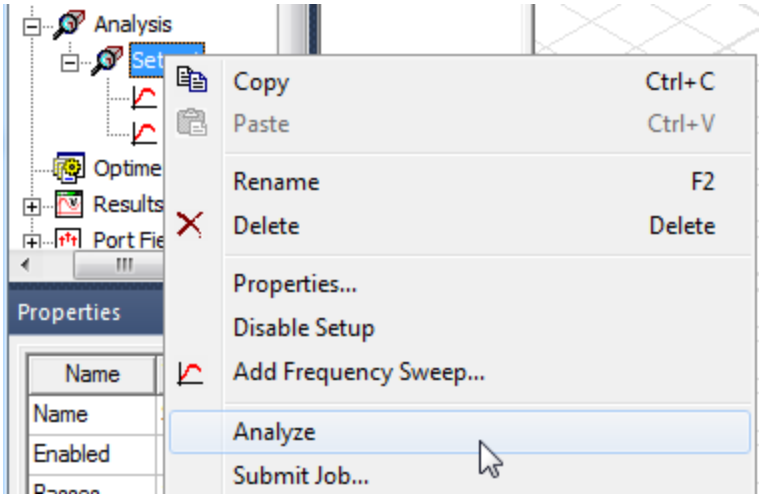
Note:

You can 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 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 Manager, using the available machines. The following example shows a setup with two enabled sweeps.



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 Manager. You can prioritize setups by changing their order in the queue.

1. To view the solution queue, click **Tools> Show Queued Simulations**.

   This displays a dialog box showing all the simulations and their current status. You can 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. The progress bar shows the relative progress of the simulation.

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 showing the status of the analysis process, including the progress and number of valid passes completed for adaptive solutions, elapsed time, memory usage, and more
- Mesh statistics (numbers of elements, lengths, volumes, and volume standard deviation; all on a per-object basis).

To view the *Solutions* window:

1. Right-click the solution **Setup** in the Project Manager.

   A shortcut menu appears.

2. Select **Convergence**, **Solutions**, **Mesh Statistics**, or **Profile** 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 solution status:

- Click **Q3D Extractor** > **Results** > **Browse Solutions**.

  The *Solutions* dialog box appears with the **Browse** tab selected. It displays data about the number of valid passes completed (when adaptive solutions are applicable). 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 (when applicable), and the state of the solution.

- You can use the **Properties** button to display a dialog box that lets you change the way the Setup, Solution, and Variation are listed in the tree structure of the *Solutions* dialog box.
- 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 a license is lost, the software waits for the license to be regained, checking every 2 minutes or until you abort.

- The **Statistics** tab of the *Solutions* dialog box displays path information as well as format, number of files, and size.

Changing a Solution Priority for System Resources

You can modify the priority of Ansys Electronics Desktop simulations so that system resources are allocated to other computer processes before the solver. If you reduce the priority of Ansys Electronics Desktop simulations, your other software tools will respond as they normally would, but Ansys Electronics Desktop simulations may take longer.

> **Note:**
>
> The Windows Task Manager does not indicate a reduced priority for the Ansys Electronics Desktop solvers. 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 **Tools> Options> HPC and Analysis** and click the **Options** tab of the resulting dialog box.
2. From the **Change Priority** menu (or the **Default Process Priority** drop-down menu), select one of the following priorities:

Highest	
Above Normal	
Normal	The default.
Below Normal	
Lowest Priority	

3. Click **OK**.

Related Topics

Monitoring Queued Simulations

Aborting an Analysis

To end the solution before the currently running process is complete:

- Right-click in the **Progress** window and click **Abort**.

  The solver ends the analysis immediately. The data for the currently running adaptive pass, frequency point, time step, or variation is lost. Previously completed solutions may or may not be retained (depending on what process was interrupted by the Abort command). If you want to ensure that the results of all completed solutions are saved, use *Clean Stop* instead of *Abort*.

To abort the solution after the currently running process is complete:

- Right-click the **Progress** window and click **Clean Stop** on the shortcut menu.

  The analysis ends after the currently running adaptive pass, frequency point, time step, or variation has been solved. Solutions completed before the stop request are retained.

If you request a clean stop during the third adaptive pass, the solution for the third pass will be available once the third pass has finished solving, but the fourth pass will not run.

Ansys EM Application as an LSF Job

If you have an Ansys EM application running as an LSF job, you can use the command "bkill -s SIGTERM *jobid*" to terminate that 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*.

Linux

For Linux, you can use TERM commands. Sigterm handling is done in Desktop library. You can abort a running batchsolve by sending a TERM signal to hfss.exe.

Re-solving after Modifying a Design

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."

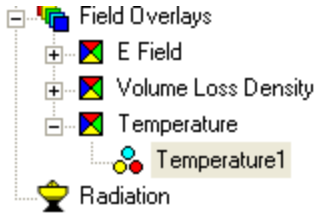
To generate a new solution after modifying a design, follow the procedure for running a simulation.

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 box is checked, you can manage analysis with feedback in Ansys Workbench. After solving an HFSS or Q3D Extractor or 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 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 HFSS/Maxwell/Q3D. 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 Electronics Desktop to Ansys Workbench and back can continue until you decide that Temperature delta reported in the Solution Report low and stable for the designs.

17 - Post Processing and Generating Reports

As Ansys Electronics Desktop completes a solution, you can display and analyze the results:

- Post Processing and Generating Reports in Q3D Extractor
- Post Processing and Generating Reports in 2D Extractor

Post Processing and Generating Reports in Q3D Extractor

After a solution is complete, you can display and analyze the results in the following ways:

- View solution data, including convergence information, computing resources used during the solution process, and matrices 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).
- Create 2D or 3D reports of RLGC 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.
- Change an excitation 's magnitude.

Viewing Solution Data in Q3D Extractor

While a solution is being generated, or when it is complete, you can use the **Solutions** dialog box to view the following:

- Solution Profile
- Convergence Data
- Matrix Data
- Mesh Statistics

To access the **Solutions** dialog box:

- Click **Q3D Extractor** > **Results** > **Solution Data**.
- Right-click **Results** in the project tree, and select **Solution Data** on the shortcut menu.

Viewing a Solution Profile in Q3D

At any time during or after the solution process, you can examine the computing resources or profile data that were used during the analysis. The profile data is essentially a log of the tasks performed by during the solution. The log indicates the length of time each task took and how much physical/disk memory was required.

From the **Project Manager**, right-click the solution setup and select **Profile** from the shortcut menu.

The **Solutions** dialog box appears, on the **Profile** tab. The displayed data depends on the type of problem and solution setup. If one or more dependent setups exist, the profile information for these can be selected from drop-down menu in the **Simulation** text field at the top of the dialog box.

In general, you can view the following information:

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, and totals.

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 takes the format "Solver *pdsn*" where: • Precision Type *p* = *M* (Mixed, for Direct Solver) or *D* (Double, for Iterative Solver) • Matrix Data Type *d* = *R* (Real) or *C* (Complex) • Symmetry Type *s* = *S* (Symmetric), *A* (Asymmetric), or *H* (Hermitian) • *n* = Number of processors used. Specify the number available on the local machine. If a solve does not require all available processors, the number reported may be less than the number available. Examples: Solver MRS2 or Solver DCS4-L2. 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, the lower the memory. You will never see L1, as this would be equivalent to Direct Solver. 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.

Information	The matrix solver information line includes: • **Disk** – the amount of hard disk space used during the calculation of the matrix solution (for example, 0 KBytes). 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 (for example, 11137). • **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 (for example, 20.3). 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. • **Number of Iterations** – used with Iterative Solver.

To export the profile data:

1. Open the **Solutions** dialog box. The **Profile** tab should already be selected.
2. Click **Export Profile**.

   This opens a file save dialog that lets you provide a file name and location.

3. Click **Save**.

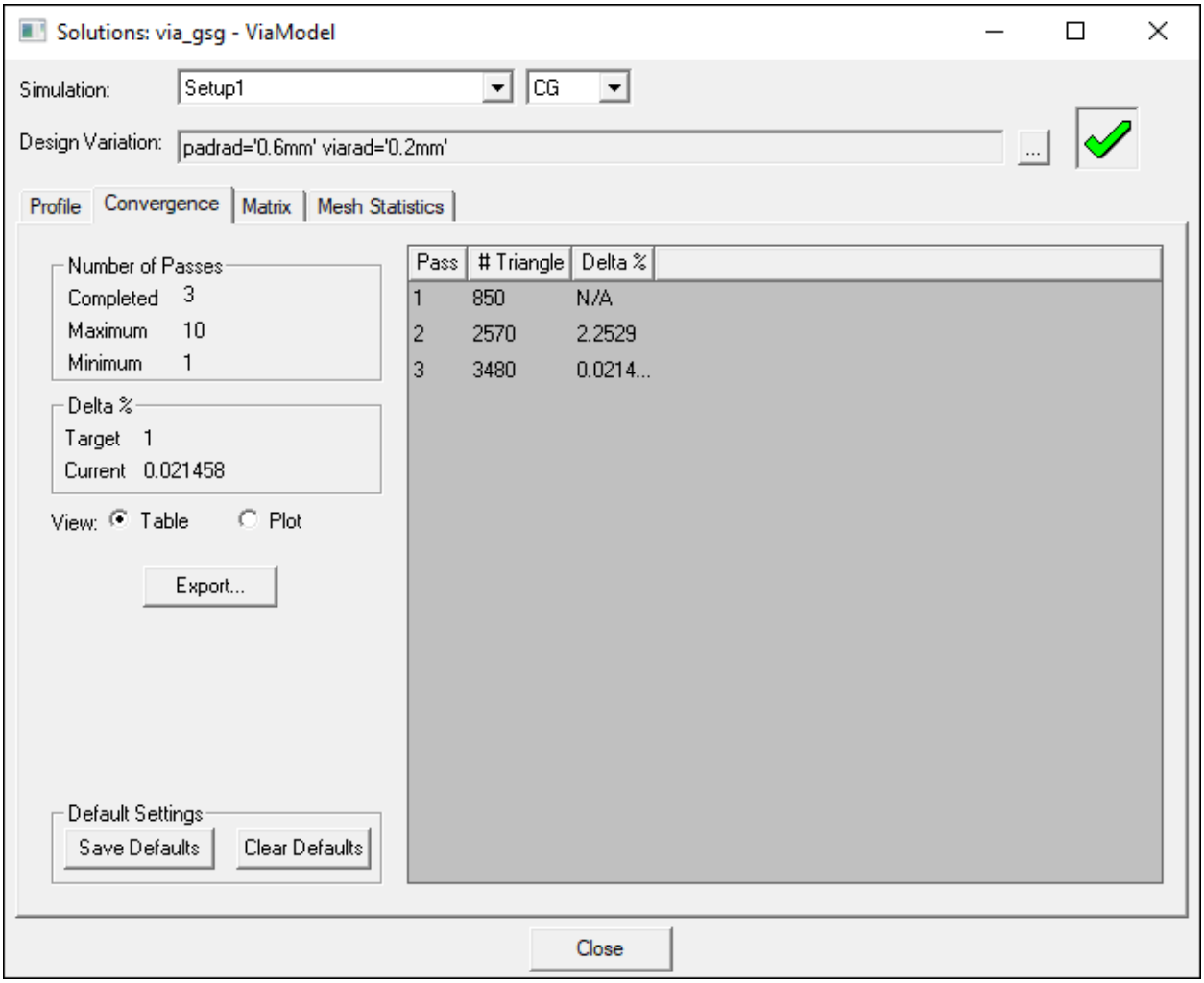
   The data is saved in a text file with a .**prof** extension.

Viewing Convergence Data in Q3D Extractor

To view convergence data:

- From the **Project Manager**, right-click a solution setup and select **Convergence**.

The **Solutions** dialog box appears, with the **Convergence** tab selected.



Convergence Data includes:

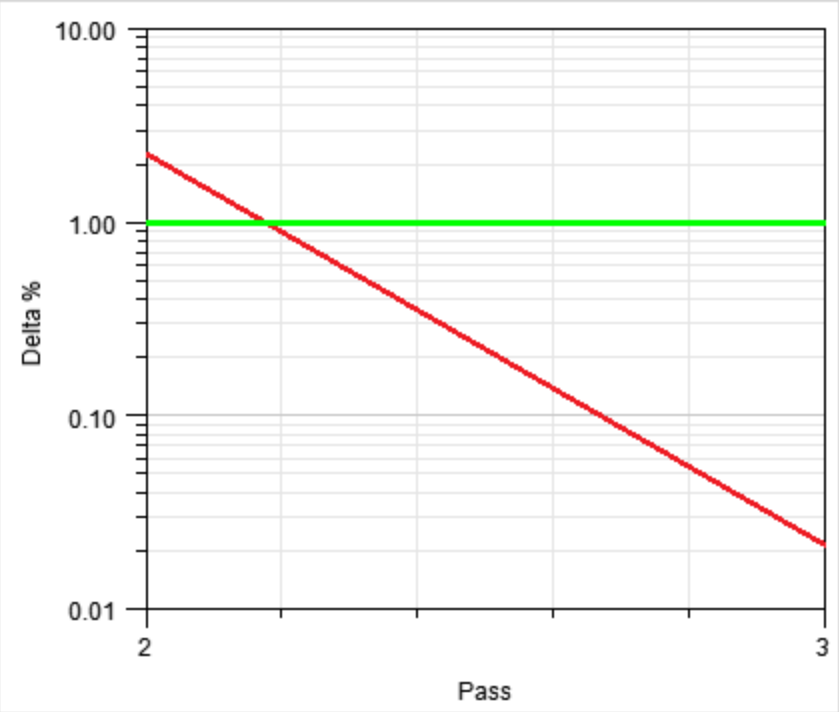
- **Number of Adaptive Passes** – during the solution process, this information includes the number of solve → error analysis → refine cycles completed / yet to be completed. After analysis has completed, only completed cycles are listed.

  > **Note:**
  >
  > If the solution converged within the specified stopping criteria, fewer passes than requested may have been performed.

- **Number of Triangles** – the number of tetrahedra or triangles created at each adaptive pass. For magnetic materials, convergence data shows information for surface mesh triangles.

- **Delta C, R, L** – the change in the C, R or L matrices between two consecutive passes, depending on the requested parameters. This information is available after two or more passes are completed.

Use the radio buttons to switch between **Table** view (shown above) and **Plot** view:



Convergence Criteria in Q3D Extractor

During adaptive refinement, the convergence of the solution is monitored by reviewing the changes in the entries of the RLGC parameter matrices.

The change **Delta_%** is expressed as:

$$\mathrm{delta\_\%} = 100 \times \frac{\max|M_{ij}[k] - M_{ij}[k-1]}{\max|M_{ij}[k]}$$

Where:

- $M_{ij}\,[k]$ is the entry of matrix M at row i and column j, at the k^{th} adaptive pass.

The maximum is taken across all rows and columns. A small Delta_% is usually a good indicator that the result is accurate.

By default, Q3D monitors only the convergence of the C and L parameter matrices. This convergence criteria emphasizes matrix entries with larger relative values. Small off-diagonal

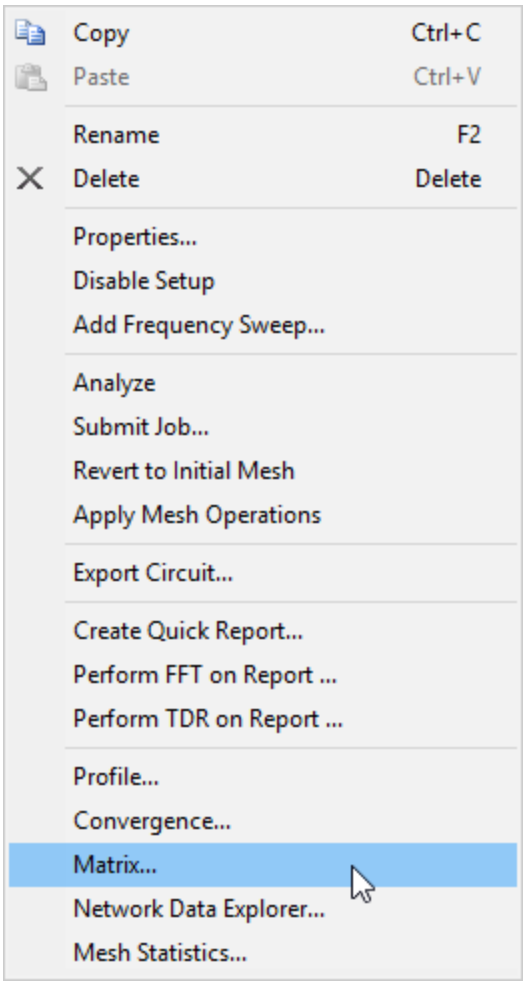
terms are not weighted as heavily. You can monitor the convergence of off-diagonal terms of the R and G matrices using the expression cache.

Viewing Matrix Data in Q3D Extractor

You can view matrices computed for resistance, inductance, and capacitance during each adaptive or non-adaptive solution.

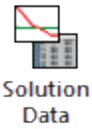
1. Access the **Solutions** window one of two ways:
   - In the project tree, under **Analysis**, right-click the solution setup.

     The right-click menu appears.

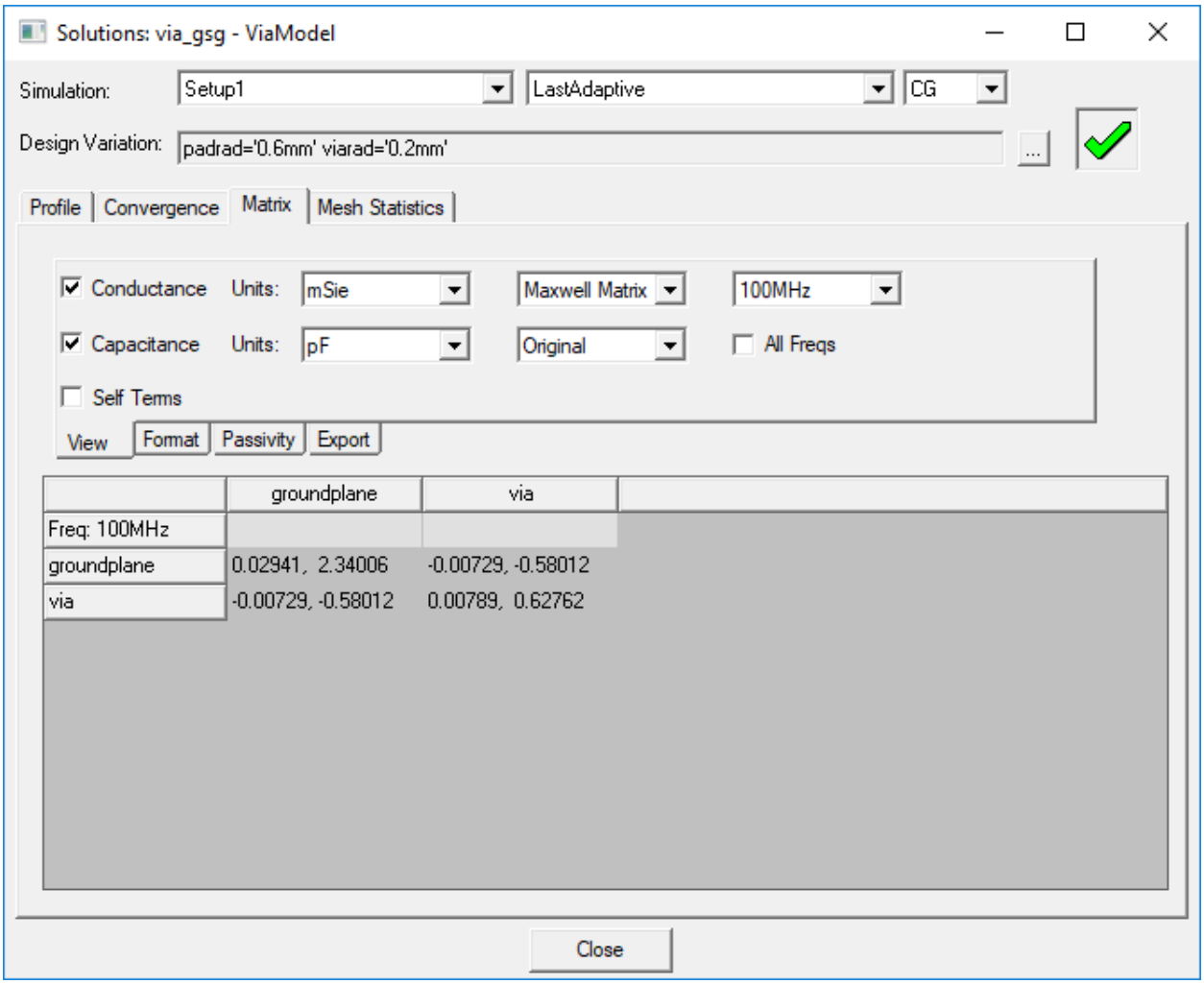


     Select **Matrix...**

- Or, from the ribbon, click **Results > Solution Data**.

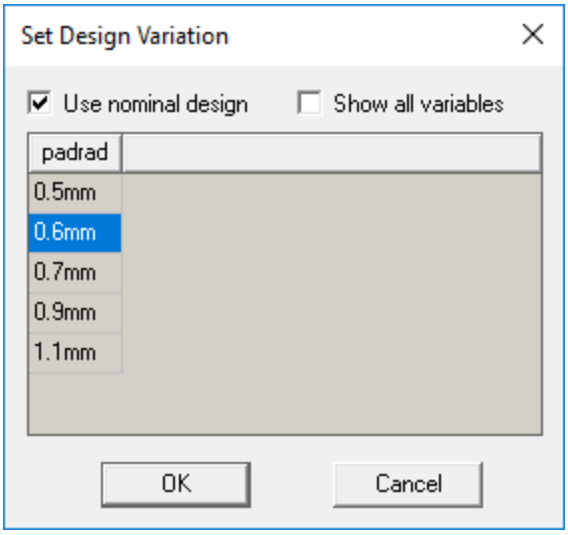


The **Solutions** window appears, with the **Matrix** tab selected.



2. In the **Design Variation** text box, set the design variation by clicking the ellipses (...) icon.

   The **Set Design Variation** window appears, listing all solved variations in the design.



> **Note:**
>
> You can also access this window from the project tree by right-clicking **Results** and selecting **Apply Solved Variation**.

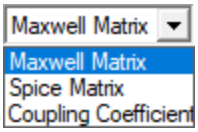
3. From the **Simulation** drop-down menus, you can select setups and either adaptive or sweep solutions. Select the solution and type of matrix you want to view (CG, DC RL, or AC RL). Available types differ based on the solution type.
4. The **Matrix** tab contains four sub-tabs: **View**, **Format**, **Passivity**, and **Export**. The **View** tab displays by default.

   From the **View** tab:

   - Select the **Units** in which to display matrix information. Available units depend on the matrix type.

     The units are saved in the matrix panel. Units are stored only for the current session and current project. Changing the units in a project does not affect the units in another project open at the same time. Closing the project resets the units, and changes are lost.

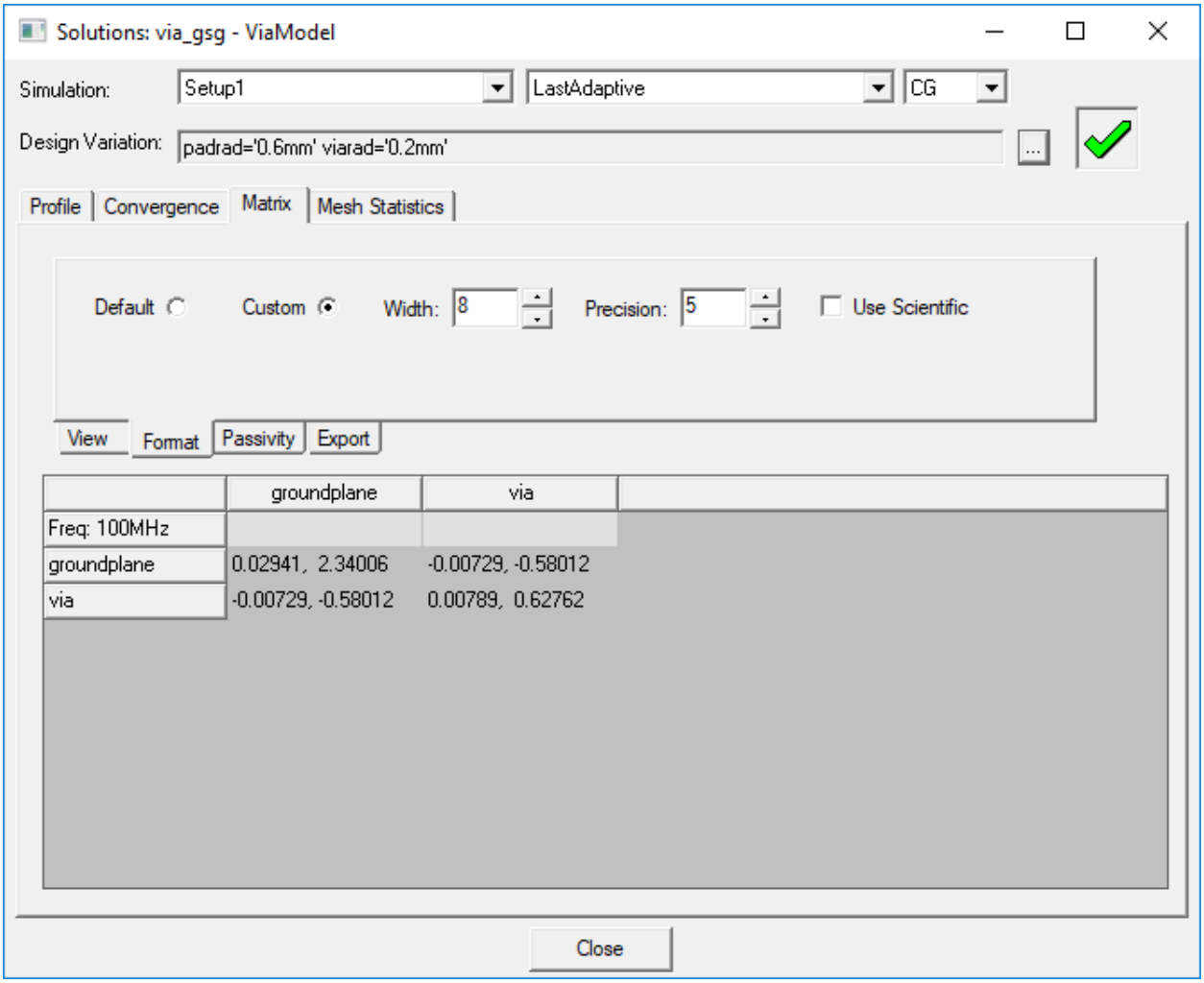
   - Use the drop-down menus to select reduced matrices to display. Choices include Coupling Coefficient, Maxwell Matrix, and Spice Matrix. Maxwell Matrix and Spice Matrix are only available in the Capacitance matrix. By default, the **Original** solved matrix displays.



- To show matrix entities for reduced operations, click the **Original** menu and select **Reduced** matrices.
- Select the solved frequencies to display using the provided options:
  - **Frequency Drop-Down Menu** – allows you to select the frequency for which you want to view matrix entries.
  - **All Freqs** – allows you to display matrix entries for all solved frequencies.
  - **Pass** – for an Adaptive Pass, allows you to select the pass for which you want to view matrix entries.
  - **Edit Freqs** – for an Interpolating Sweep, opens the **Edit Sweep** window, allowing you to insert or delete displayed frequencies.
- Use the **Self Terms** check box to display only the diagonal of the selected matrix.

5. Click the **Format** tab.

   The **Format** tab displays, with additional options.

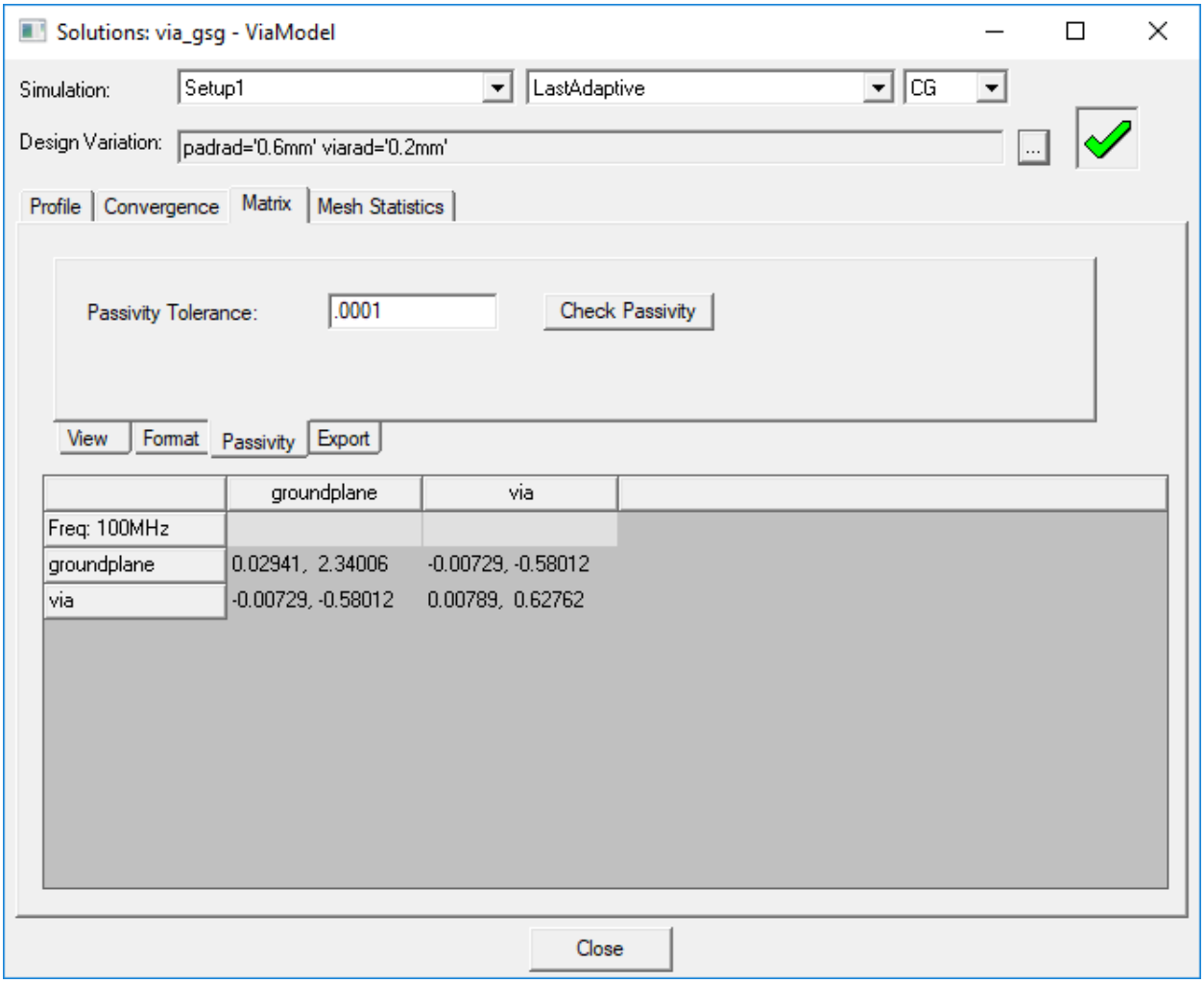


From the **Format** tab:

- Select either the **Default** or **Custom** display.
- If you select **Custom**, specify the matrix field **Width** and **Precision** in digits.
- Select the **Use Scientific** check box to display matrix data using scientific notation.

6. Click the **Passivity** tab.

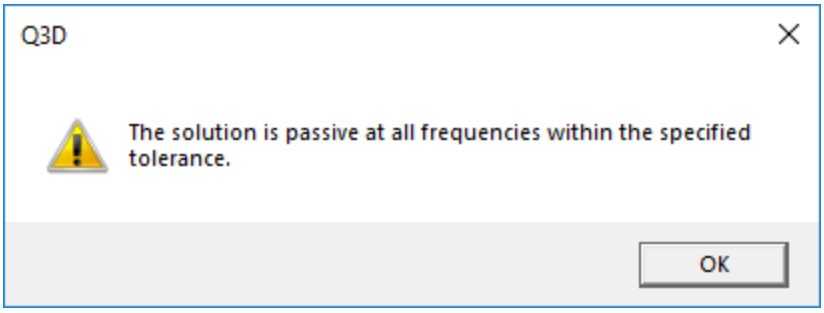
   The **Passivity** tab displays.



From the **Passivity** tab:

- Set the **Passivity Tolerance**.
- Click **Check Passivity** to check passivity under the specified Passivity Tolerance.

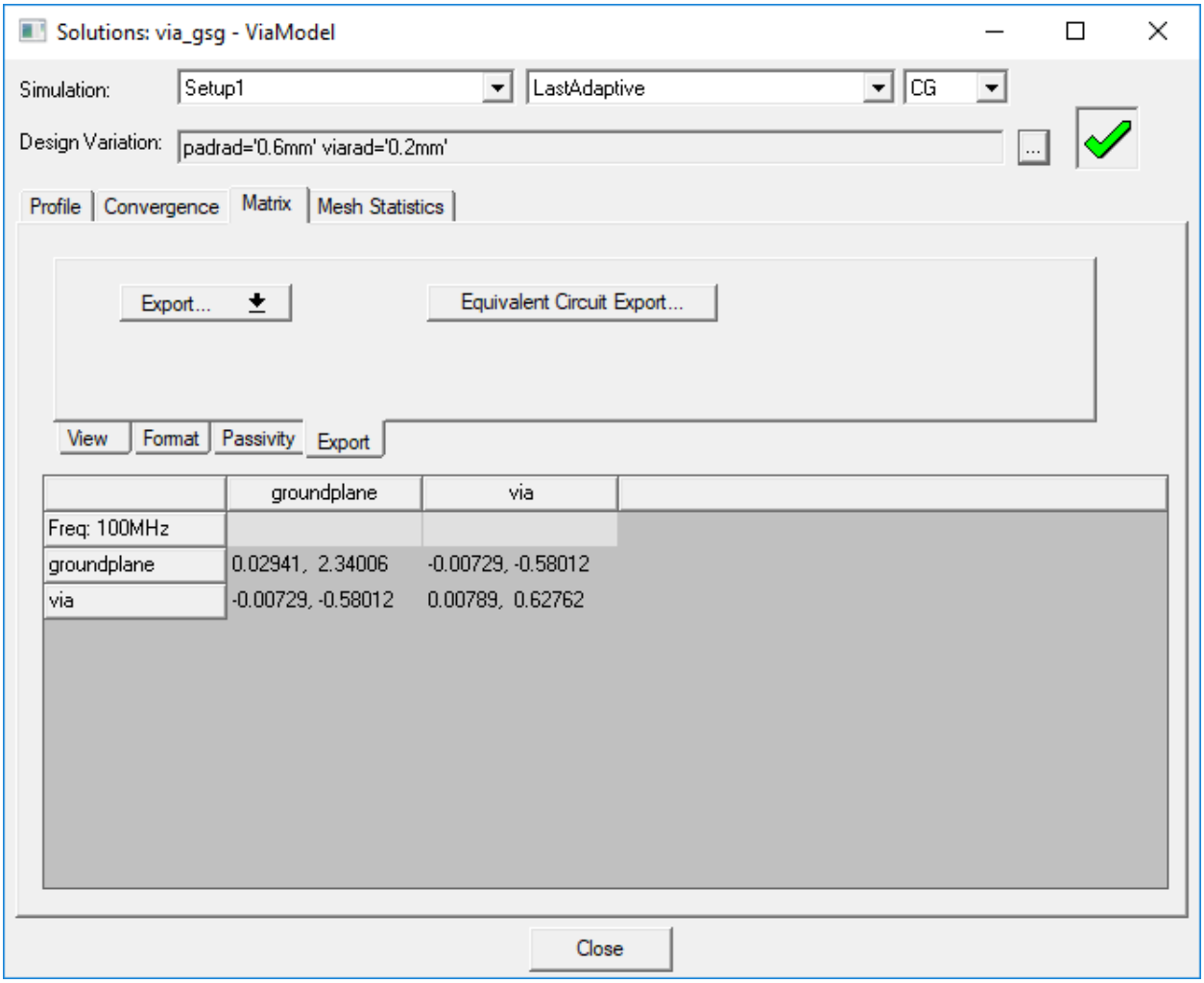
  An alert appears, showing the result.



- Click **OK** to close the alert.

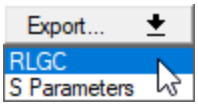
7. Click the **Export** tab.

   The **Export** tab displays.

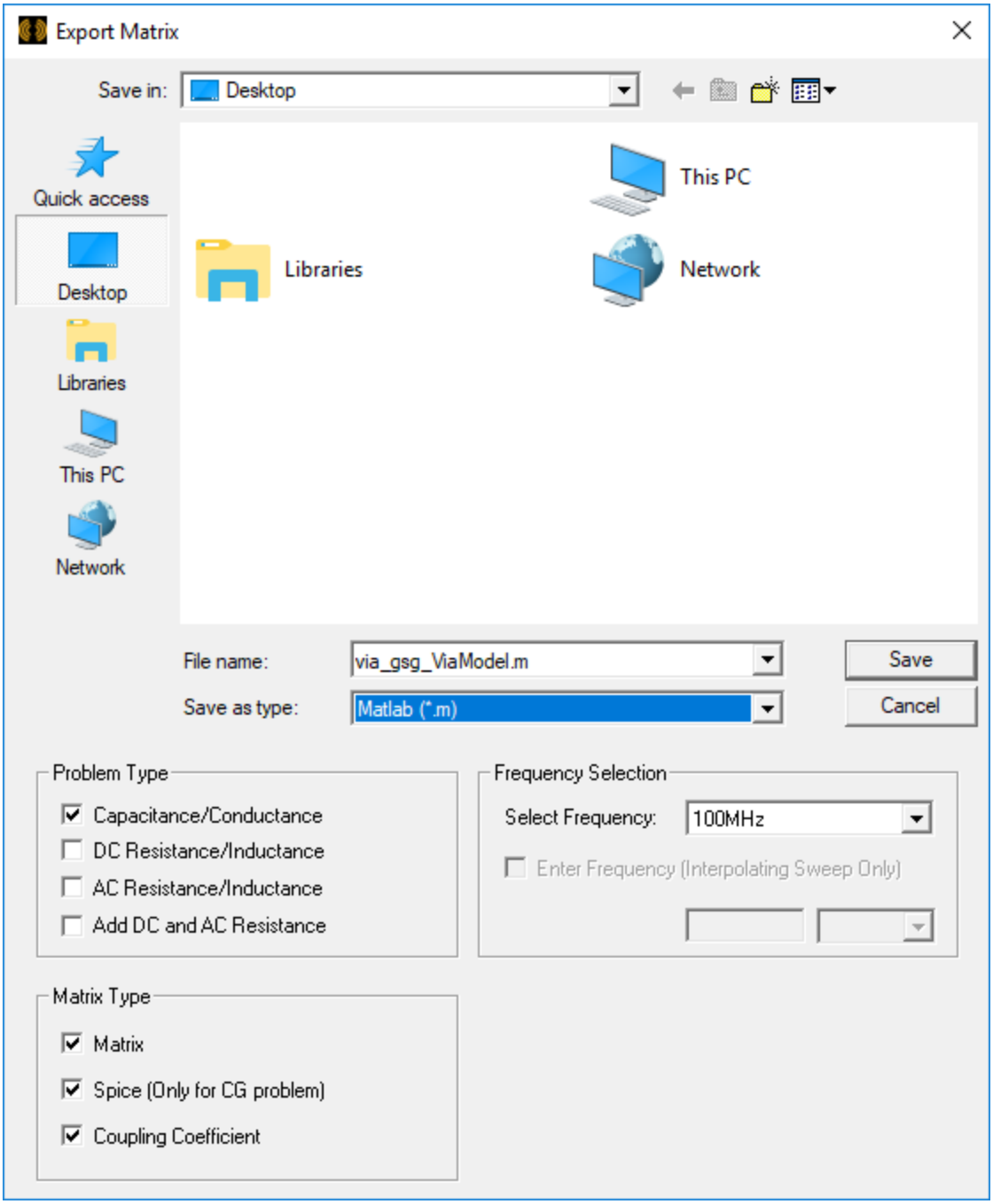


From the **Export** tab:

- To export matrix data, click **Export** and select either **RLGC** or **S Parameters** from the drop-down menu that appears.



The **Export Matrix** window appears.



Select your desired options and click **Save**.

- To export the equivalent circuit, click **Equivalent Circuit Export**.

> **Note:**
>
> You can only export the current problem type, and will receive an error message if you try to export both CG and RL.

8. Click **Close** to exit the **Solutions** window.

> **Note:**
>
> Selections in the **Solutions** window persist during execution, but are reset to their defaults on exit.

Coupling Coefficient

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:

$$abs\,(M_{ij})/\left(\sqrt{(M_{ii} \times M_{jj})}\right)$$

Maxwell Matrix

The entries of the Maxwell capacitance matrix represent the total charge on a conductor due to the voltages on the various conductors.

If Q_i is the total charge on conductor i, and V_1, V_2, ..., V_N are the conductor voltages, then:

$$Q_i = C^M{}_{i,1} * V_1 + C^M{}_{i,2} * V_2 + \ldots + C^M{}_{i,N} * V_N$$

where the superscript *M* is used for the Maxwell capacitance matrix.

Spice Matrix

The entries of the Spice capacitance matrix represent the values of a network of 2-terminal capacitors. Imagine that between every pair of conductors (say, i and j, i different from j), you have a capacitor of value $\mathbf{C^S_{i,j}}$.

In addition, let every conductor (say the i^{th}) have a capacitor $\mathbf{C^S_{i,0}}$ connected to "ground" (the values of these grounded capacitors are the entries on the main diagonal of the Spice capacitance matrix).

The total charge on a given conductor will be found by summing up the charges on the individual capacitor plates connected to it.

$$Q_i = C^S{}_{i,0} * Vi + C^S{}_{i,1} * (V_i - V_1) + C^S{}_{i,2} * (V_i - V_2) + \ldots\ldots + C^S{}_{i,N} * (V_i - V_N)$$

Relationship between Maxwell Matrix and Spice Matrix

Because there are two different expressions for the same thing, the entries of the Maxwell matrix and Spice capacitance matrices can relate to each another.

Matching the coefficients of the voltages results in:

$(1) C^{M}_{i,i} = C^{S}_{i,0} + C^{S}_{i,1} + C^{S}_{i,2} + \ldots + C^{S}_{i,(N-1)}$

$(2) C^{M}_{i,j} = - C^{S}_{i,j}$ *(for i not equal to j)*

So the off-diagonal entries of the Maxwell capacitance matrix (which are generally negative) are the opposite of the off-diagonal entries of the Spice capacitance matrix (which are positive). The main-diagonal entries of the Maxwell matrix are the sum across an entire row of the Spice capacitance matrix.

Alternatively, you could express the grounded capacitor in the Spice matrix as the sum across a whole row of the Maxwell capacitance matrix:

$C^{S}_{i,0} = C^{M}_{i,1} + C^{M}_{i,2} + \ldots + C^{M}_{i,N}$

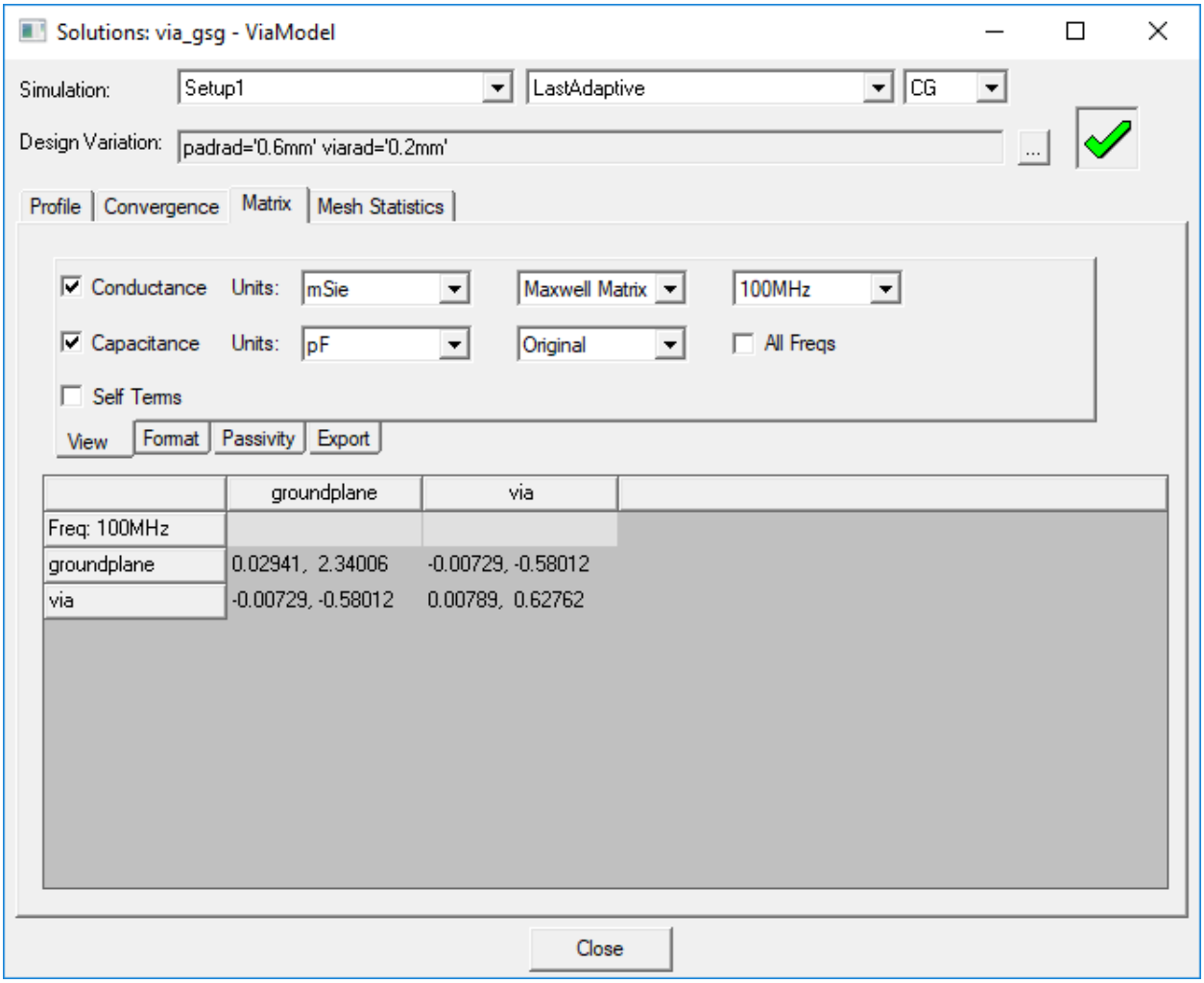
> **Note:**
>
> Since the off-diagonal entries of the Maxwell matrix are negative, the diagonal entry of the Spice matrix is generally smaller than the diagonal entry of the Maxwell matrix. It can even go to zero if the conductor is well shielded from the ground at infinity.

Exporting RLGC Matrix Data

To export RLGC matrix data:

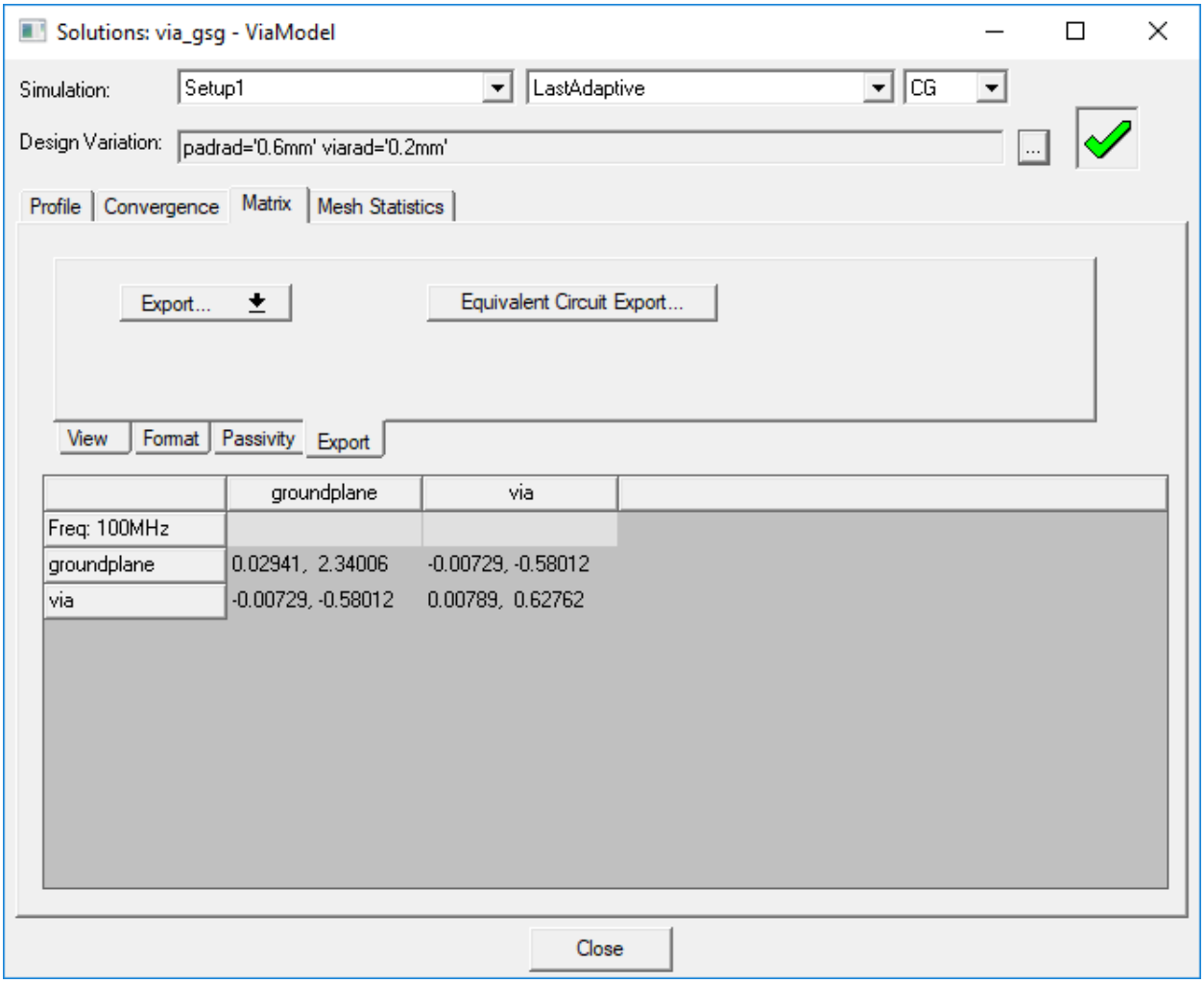
1. In the project tree, right-click the solution setup and select **Matrix** from the shortcut menu. Alternatively, click **Results** > **Solution Data** and select the **Matrix** tab.

   The **Solutions** window appears with the **Matrix** tab selected.

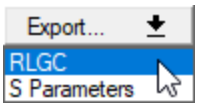


2. Click the **Export** tab.

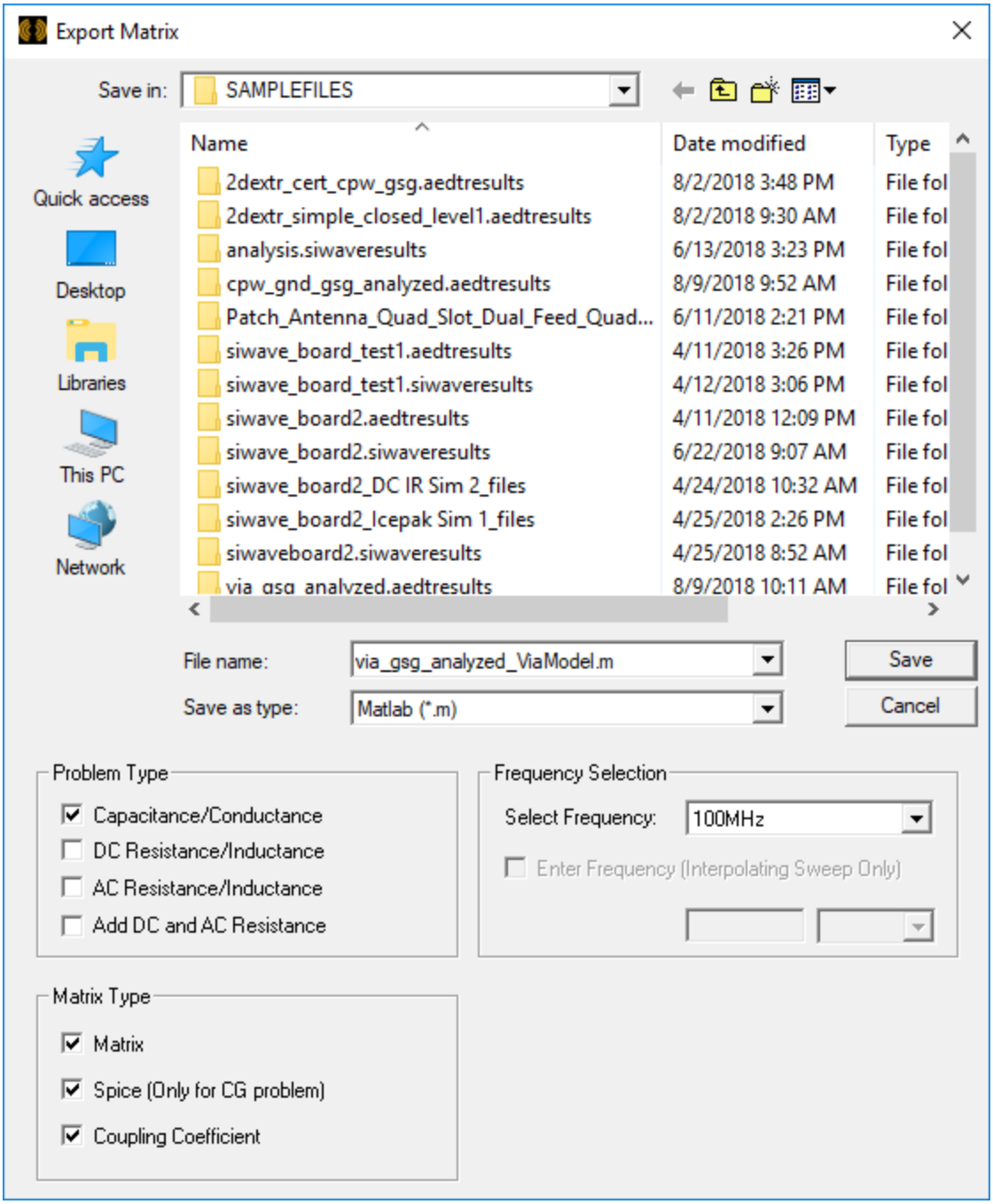
   The **Export** tab appears.



3. Click Export, and select **RLGC** from the drop-down menu.



The **Export Matrix** window appears.



4. Enter the name of the file you are exporting to in the **File name** field.
5. Select one the following file formats from the **Save as type** drop-down menu:
    - Ansys EM Legacy Format Files (*.lvl)
    - Data Table(spreadsheet) (*.txt)

- Matlab (*.m)

If you select the data table format, a text file will be created. The elements of the RLGC-matrix are arranged in a series of columns that are tab-separated and include a first row of headings.

If you select the Matlab format, the elements of the R-,L-, G-, or C- matrix will be arranged in a series of rows.

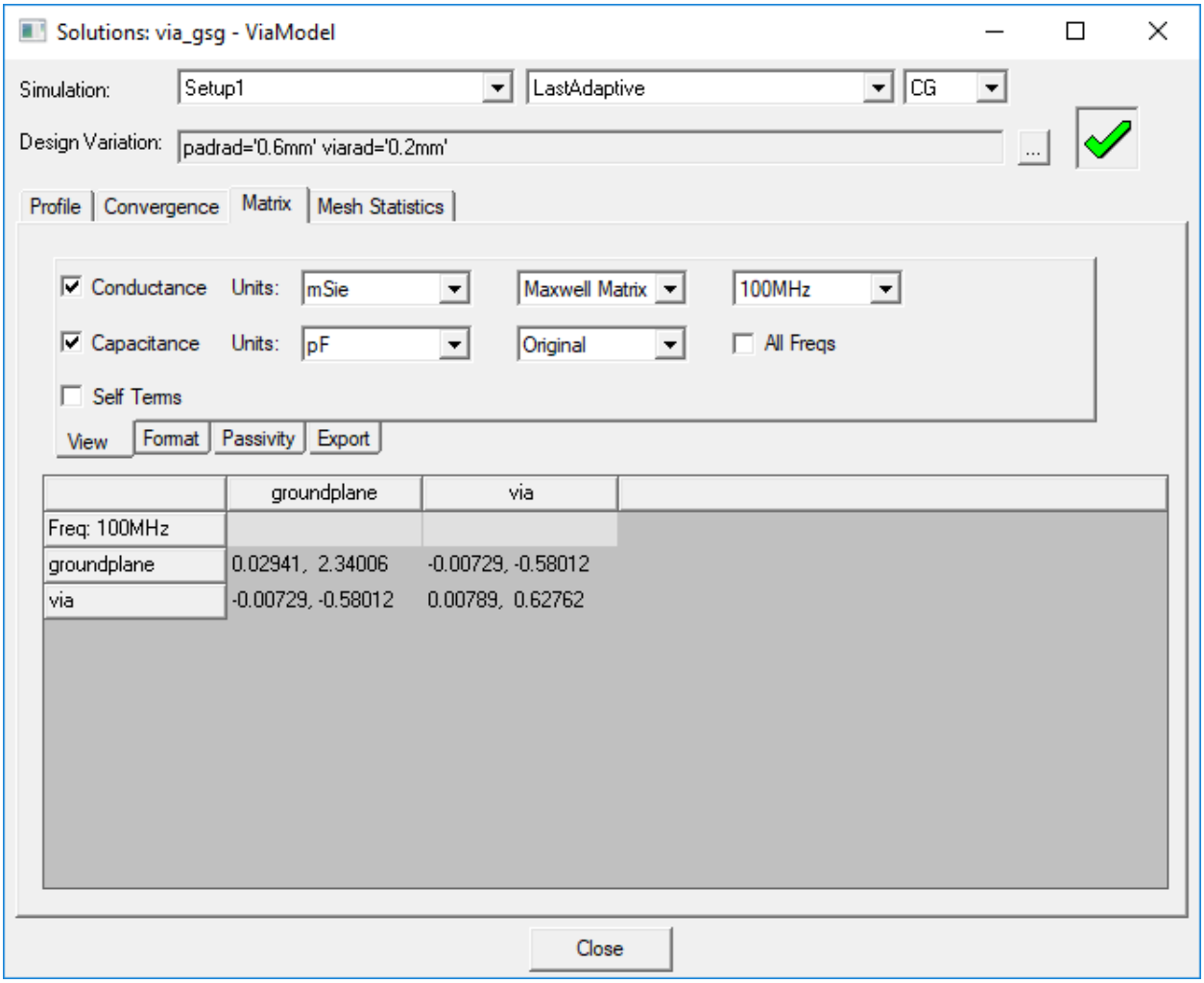
6. Select the RLGC data to be exported. You can select **Capacitance/Conductance**, **DC Resistance/Inductance**, **AC Resistance/Inductance** or **Add DC and AC Resistance**.
7. Select the **Matrix Type**.
8. Select the **Frequency**.
9. Click **Save**.

Exporting S Parameter Data

To export S parameter data:

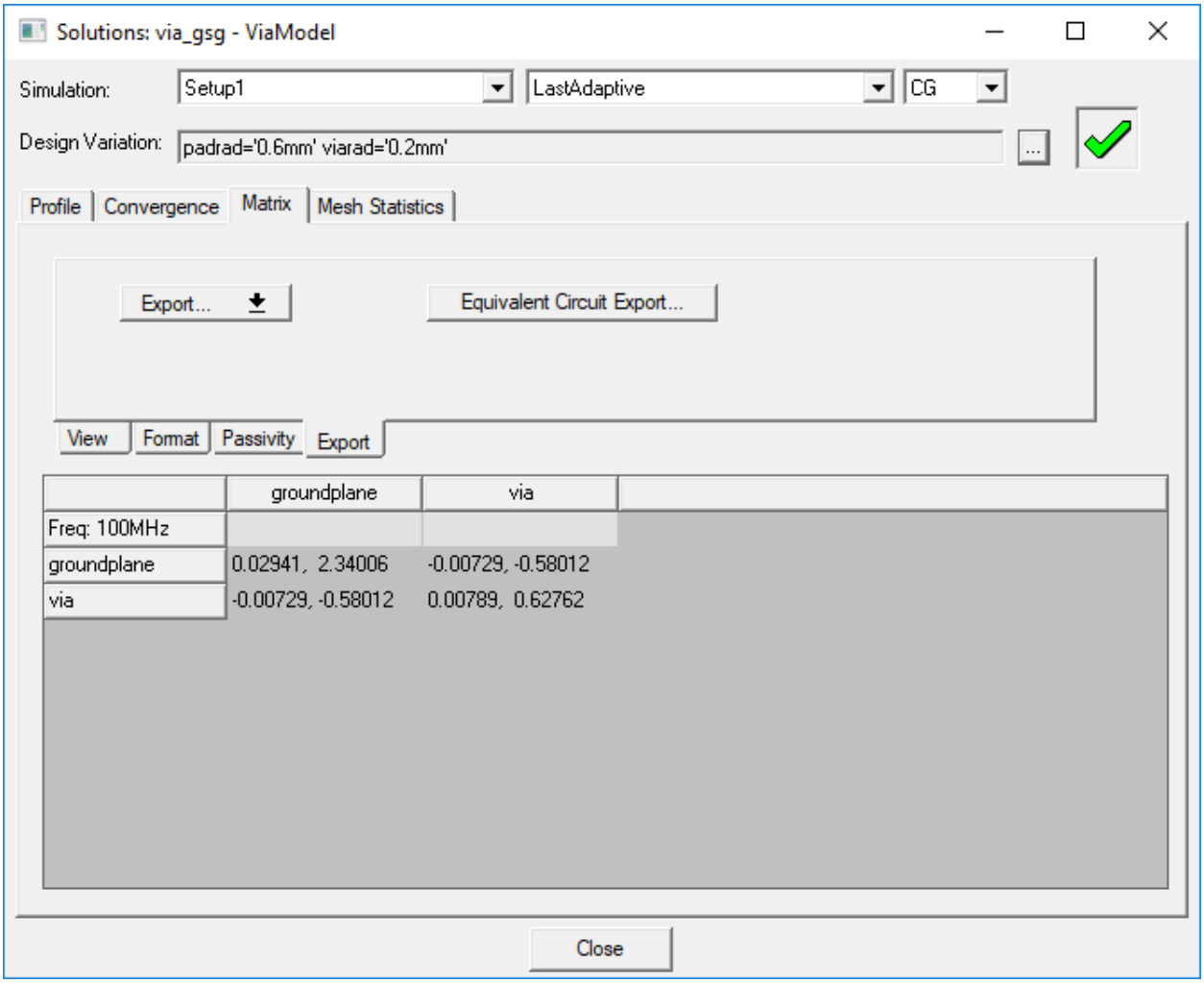
1. In the project tree, right-click the solution setup and select **Matrix** from the shortcut menu. Alternatively, click **Results** > **Solution Data** and select the **Matrix** tab.

   The **Solutions** window appears with the **Matrix** tab selected.

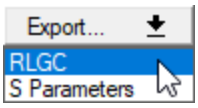


2. Click the **Export** tab.

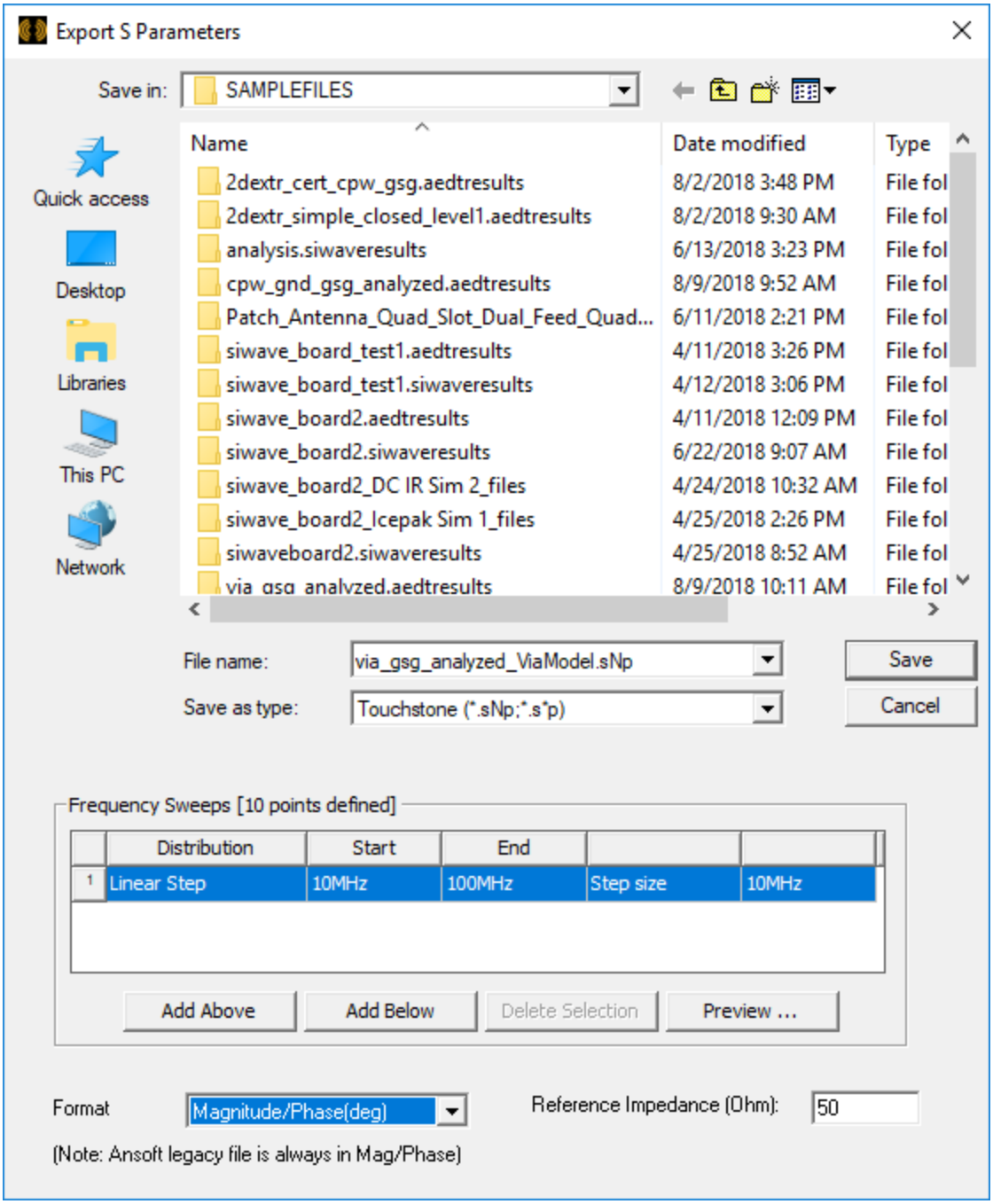
   The **Export** tab appears.



3. Click **Export** and select **S Parameter** from the drop-down menu.



The **Export S Parameters** window appears.



4. Enter the name of the file you are exporting to in the **File name** field.
5. Select one the following file formats from the **Save as type** drop-down menu:
    - Ansys EM Legacy Format Files (*.szg)
    - Citi Files (*.cit)

- Data Table(spreadsheet) (*.txt)
- Matlab (*.m)
- Neutral Model Format (*.nmf)
- Touchstone (*.sNp, *.s*p)

If you select the data table format, a text file will be created. The elements of the S-matrix are arranged in a series of columns that are tab-separated and include a first row of headings.

If you select the Matlab format, the elements of the S-,Y-, or Z- matrix will be arranged in a series of rows.

If you select the Touchstone format, a Touchstone S parameter file will be created. The number of sources and sinks is indicated by *N*. For example, a Touchstone file with one source and sink would have the extension *.**s2p**.

6. Specify the **Frequency Setup**:
    a. Specify the **Start** frequency, **Stop** frequency, and **Step** size.
    b. Select the units.
    c. Click **Display** to see the selected values.
7. Select the format in which the data is to be exported. Available formats depend on the matrix type being displayed.
8. Enter a value for the **Reference Impedance**.
9. Click **Save**.

Viewing Mesh Statistics in Q3D Extractor

To view an adaptive solution's mesh information, either during or after the solution process:

- From the **Project Manager**, right-click a solution setup and select **Mesh Statistics**.

The **Solutions** dialog box appears, with the **Mesh Statistics** tab selected.

The table lists the design elements and depending on the solution type may include: Num Elements, Min edge length, Max edge length, RMS edge length, min elem area, max elem area, mean elem area, and standard deviation.

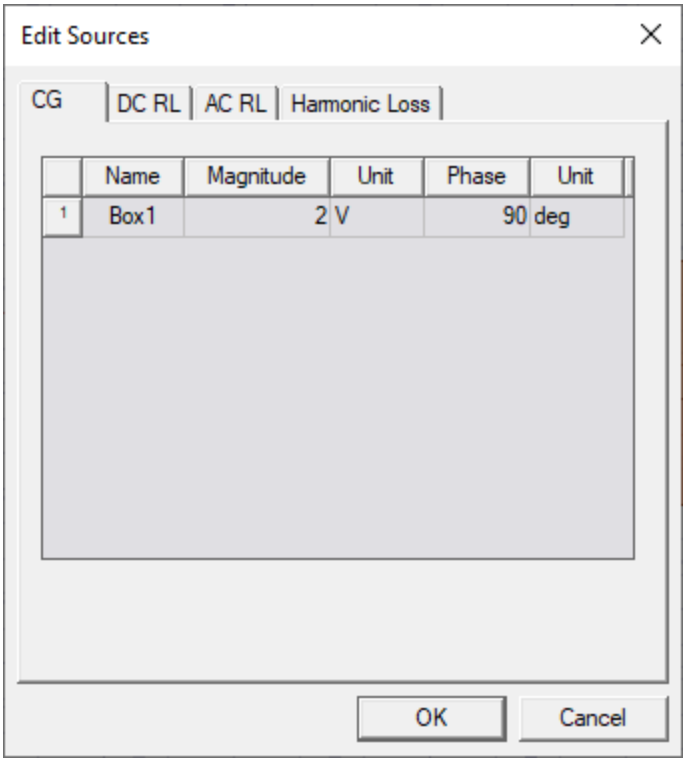
If mesh repairs have been performed, two additional columns appear in the table: **Recovered %**, and **Repaired %**. These columns indicate the fraction of an object that was successfully recovered and the fraction that needed some repair.

Editing Sources in Q3D Extractor

Change source values in the **Edit Sources** dialog box.

1. Launch the Edit Sources window one of two ways:
   - Click **Q3D Extractor** > **Fields** > **Edit Sources**.
   - In the **Project Manager**, right-click **Field Overlays** and select **Edit Sources**.

   The **Edit Sources** window appears, containing four tabs:



   - **CG** – allows you to set magnitude and phase values.
   - **DC RL** – allows you to select either **Voltage** or **Current**, and specify those values. Also provides an **Assume true DC in loss calculation** check box.

> **Note:**
>
> When **Assume true DC in loss calculation** is enabled, DC volume loss calculations will ignore phases other than 0 or 180 degrees. The loss is calculated as *P* = *IRI*, and loss calculated with this option matches legacy results.
>
> When this option is deselected, the calculation assumes that DC excitation includes sinusoidals at very low frequencies with given phase shift, and the loss is calculated as RMS loss:
>
> $$P = \frac{1}{2} I_{re} R I_{re} + \frac{1}{2} I_{im} R I_{im}$$

- **AC RL** – allows you to select either **Voltage** or **Current**, and specify those values. AC RL voltage and current sources are assumed to be peak quantities.
- **Harmonic Loss** – allows you to specify real and imaginary currents for each source, in order to compute harmonic loss.

2. Select the tab whose values you wish to edit.
3. Click **OK**.

Post Processing and Generating Reports in 2D Extractor

When 2D Extractor 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 used during the solution process, and matrices 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.
- Create 2D or 3D reports of RLGC matrices and basic and derived field quantities.
- Plot the finite element mesh on surfaces or within 3D objects.

- Change an excitation's magnitude.

> **Note:**
>
> If a particular pass is not solved for CG or RL, last adaptive solution is used. However if last adaptive solution is not valid, then you can see the data for only those passes for which both CG and RL are available.

Viewing Solution Data in 2D Extractor

While a solution is being generated, or when it is complete, you can use the **Solutions** dialog box to view the following:

- Solution Profile
- Convergence Data
- Matrix Data
- Mesh Statistics
- Mode Data
- Tline Data

To access the **Solutions** dialog box:

- Click **2D Extractor** > **Results** > **Solution Data**.
- Right-click **Results** in the project tree, and select **Solution Data** on the shortcut menu.

Viewing a Solution Profile for 2D Extractor

At any time during or after the solution process, you can examine the computing resources or profile data used for the analysis. The profile data is essentially a log of the tasks performed by during the solution. The log indicates the length of time each task took and how much physical/disk memory was required.

From the **Project Manager**, right-click the solution setup and select **Profile** from the shortcut menu.

The **Solutions** dialog box appears, on the **Profile** tab. The displayed data depends on the type of problem and solution setup. If one or more dependent setups exist, the profile information for these can be selected from drop-down menu in the **Simulation** text field at the top of the dialog box.

In general, you can view the following information:

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, and totals.

To export the profile data:

1. Open the **Solutions** dialog box. The **Profile** tab should already be selected.
2. Click **Export Profile**.

   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.

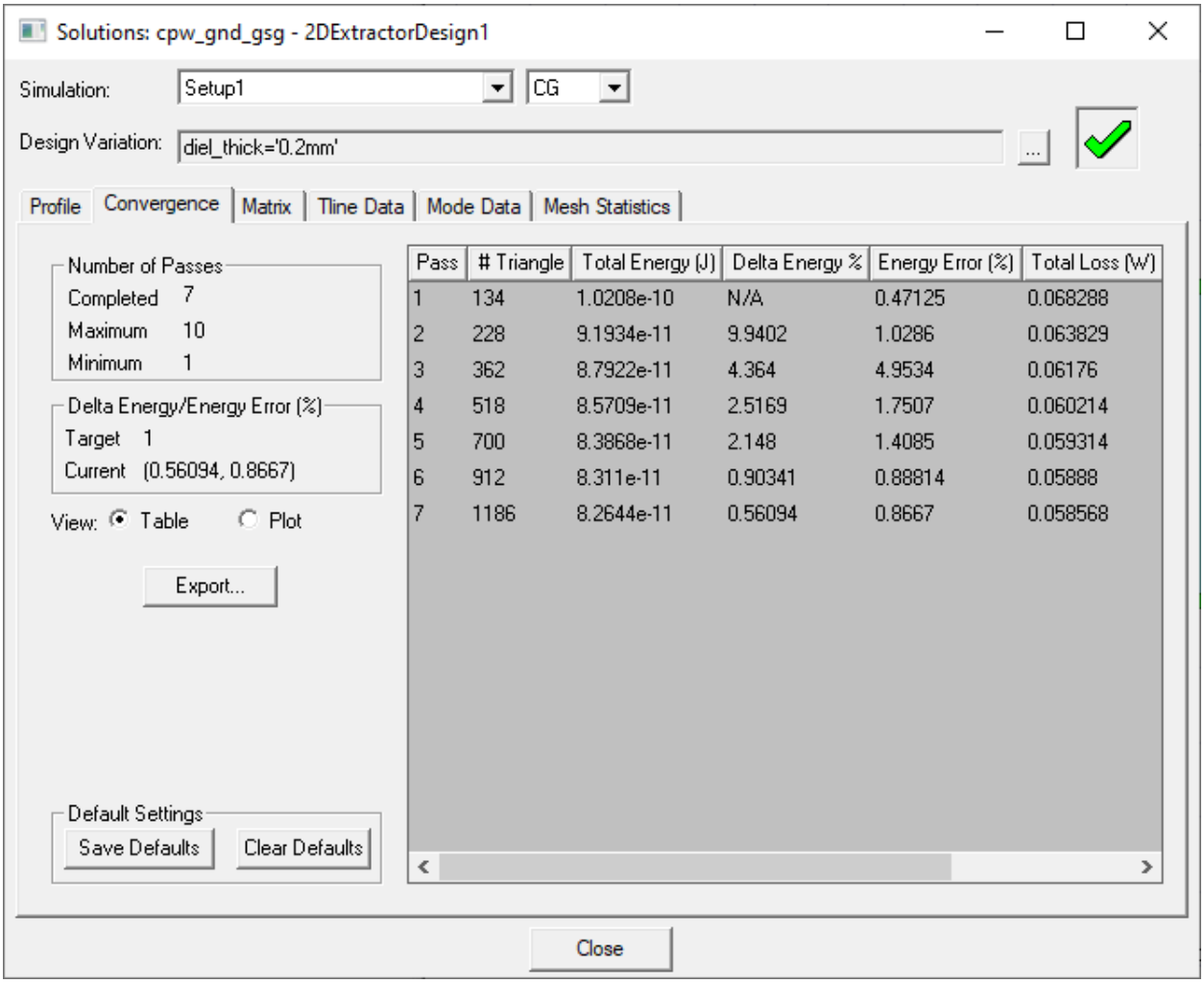
Viewing Convergence Data for 2D Extractor

2D Extractor performs adaptive analysis for accurate simulations. It uses “h-adaptive” mesh refinement, iteratively refining the initial mesh by reducing the size of individual elements in areas of high approximation error—thus improving the accuracy of the solution. Different convergence control mechanisms such as Energy Error, Loss Error and Parameter Error exist are used during the analysis.

To view convergence data:

- From the **Project Manager**, right-click a solution setup and select **Convergence**.

The **Solutions** dialog box appears, with the **Convergence** tab selected.

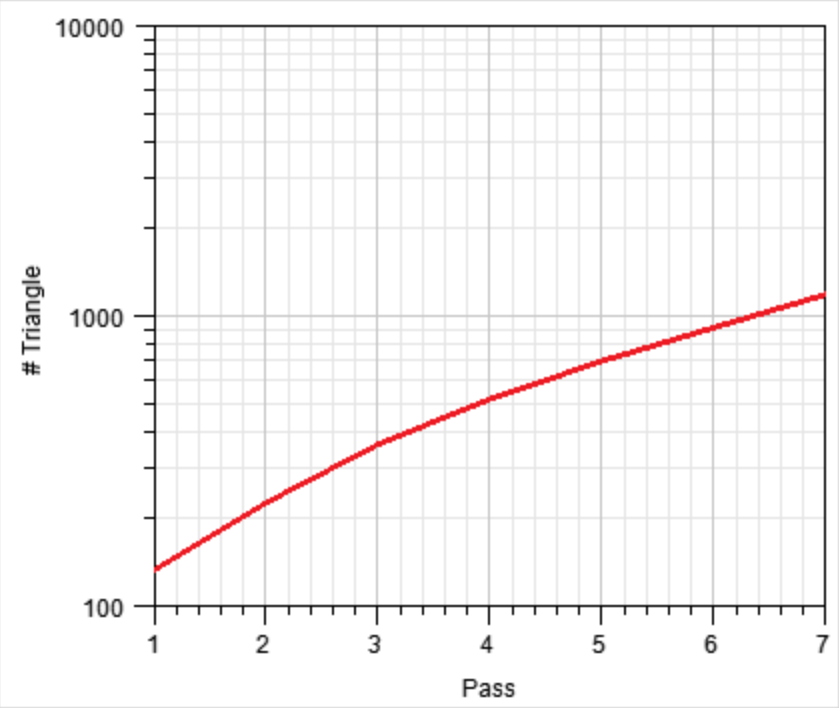


Convergence Data includes:

- **Number of Adaptive Passes** – during the solution process, this information includes the number of solve → error analysis → refine cycles completed / yet to be completed. After analysis has completed, only completed cycles are listed.
- **Number of Triangles** – the number of tetrahedra or triangles created at each adaptive pass. For magnetic materials, convergence data shows information for surface mesh triangles.
- **Total Energy** – the energy stored in the electric and magnetic fields over the entire computational domain, in Joules.
- **Delta Energy** – the percentage change in the total energy from the previous adaptive pass *k-1* to the current pass *k*, expressed as a percentage of the energy in the current pass.
- **Energy Error** – a metric for estimating the error in the finite element solution.

- **Total Loss** – power loss measured over the entire computational domain.
- **Delta Loss** – the percentage change in the total loss from one pass to the next.
- **Loss Error** – the error in the losses computed by the finite element solution.
- See Convergence Criteria below for additional details.

Use the radio buttons to switch between **Table** view (shown above) and **Plot** view:



Convergence Criteria in 2D Extractor

Parameters	Description

Total Energy	Total Energy is the energy stored in the electric and magnetic fields over the entire computational domain, in Joules. For CG solutions, Total Energy can be written as: $\mathbf{energy} = \frac{1}{2}\mathrm{Re}\int_{\Omega} \vec{E}^{*} \cdot \varepsilon \vec{E} d\Omega$ Where: • ε is the effective complex permittivity • $\vec{E}$ is the electric field vector • Ω denotes the problem domain For RL solutions, Total Energy can be written as: $\mathbf{energy} = \frac{1}{2}\mathrm{Re}\int_{\Omega} \vec{H}^{*} \cdot \mu \vec{H} d\Omega$ Where: • μ is the effective complex permeability • $\vec{H}$ is the magnetic field vector The capacitance (C) and inductance (L) values are closely related to the Total Energy. For CG or RL matrices, these values can be derived from the Total Energy stored in the system under certain excitation configuration (for example, an excitation configuration of 1A for one signal line and 0A for all others is used in eddy current solution, which results in L to be directly proportional to the total magnetic energy as L=2*Total Energy). A good indication of accurate C and L values is when the total energy stabilizes during the adaptive refinement process.
Delta Energy	Delta Energy is the percentage change in the Total Energy (as defined above) from the previous adaptive pass ***k-1*** to the current pass ***k***, expressed as a percentage of the energy in the current pass: $\mathbf{Delta\ Energy}\ \% = 100 \times \left\lvert \frac{\mathbf{energy}[k] - \mathbf{energy}[k-1]}{\mathbf{energy}[k]} \right\rvert$ Delta Energy is calculated for $k > 1$.

Energy Error	A finite element solution approximates the electric and magnetic fields in the problem domain using low-order polynomial basis functions. Energy Error is a metric for estimating the error in the finite element solution. It should decrease as the mesh is refined. Energy Error is displayed as a percentage of the Total Energy. In a well-converged solution, the Energy Error should be much less than the Total Energy. By default, Delta Energy and Energy Error are used for convergence of the solution during the adaptive process, unless you select one of the **CG/RL Advanced** options: **Use Loss Convergence** or **Use Parameter Convergence**.
Total Loss	Total loss is the power loss due to mechanisms such as Joule I-R heating (Ohmic loss), dielectric dissipation (loss in dielectric materials with loss tangent under alternating electric fields) and non-ideal magnetic materials (Hysteresis loss in magnetic materials under alternating magnetic fields), measured over the entire computational domain. The total loss is displayed for AC conduction or Eddy Current solutions, in Watts. The conductance (G) and resistance (R) values are closely related to the Total Loss. For CG or RL matrices, the values can be derived from the Total Loss in the system under certain excitation configuration (for instance, an excitation configuration of 1A for one signal line and 0A for all others is used in Eddy Current solution, which results in ***R*** to be directly proportional to the Total Loss as: $\mathbf{R} = \mathbf{2} \times \mathbf{Total\ Loss}$ When Total Loss stabilizes during adaptive refinement , this is a good indicator that the G and R values will be accurate.
Delta Loss	Delta loss is analogous to delta energy. It is the percentage change in the total loss from one pass to the next one as formulated below: $\mathbf{Delta\ Loss\ \%} = 100 \times \left\lvert \frac{\mathrm{total\_loss}[k] - \mathrm{total\_loss}[k-1]}{\mathrm{total\_loss}[k]} \right\rvert$ Delta Loss is calculated for $k > 1$, where ***k*** is the current adaptive pass number.

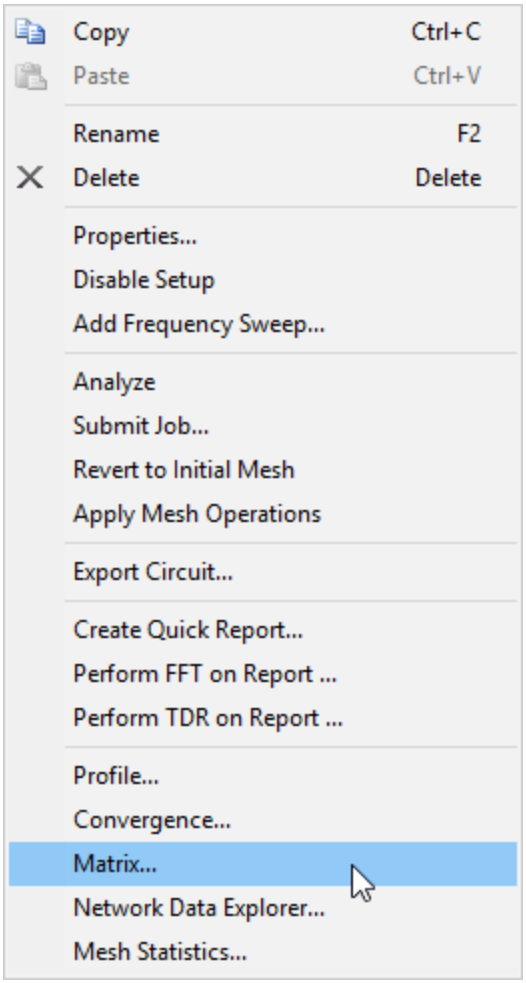
Loss Error	Loss Error is a metric used to estimate the error in the losses computed by the finite element solution. It should decrease as the mesh is refined. Loss Error is displayed as a percentage of the total loss. In a well converged solution, the Loss Error should be much less than the Total Loss. Delta Loss and Loss Error criteria are used during the adaptive process only if the **CG/RL Advanced** option **Use Loss Convergence** is selected.

Viewing Matrix Data in 2D Extractor

You can view matrices computed for resistance, inductance, and capacitance during each adaptive or non-adaptive solution:

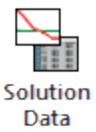
1. Access the **Solutions** window one of two ways:
    - In the project tree, under **Analysis**, right-click the solution setup.

      The right-click menu appears.

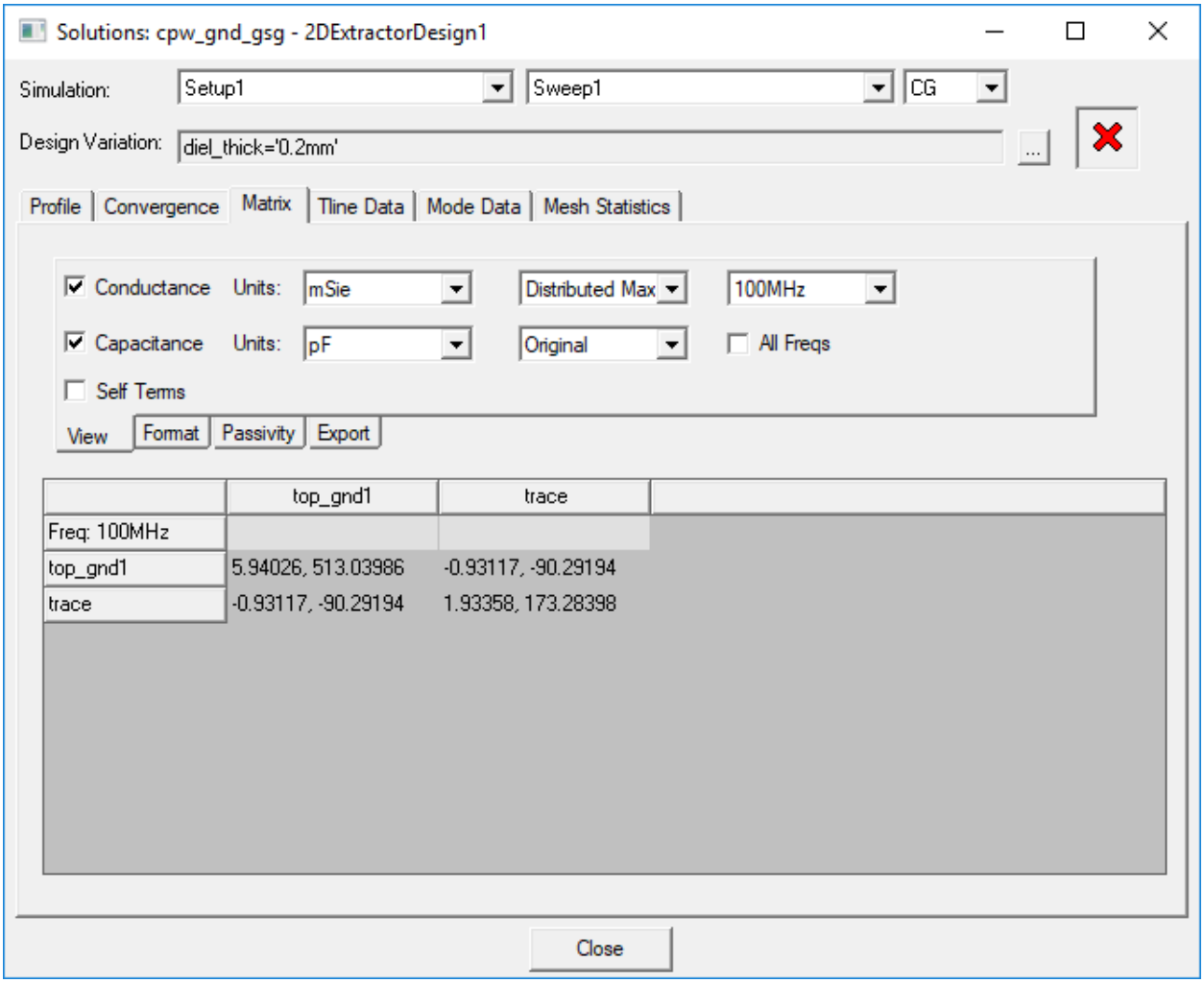


Select **Matrix...**

- Or, from the ribbon, click **Results > Solution Data**.

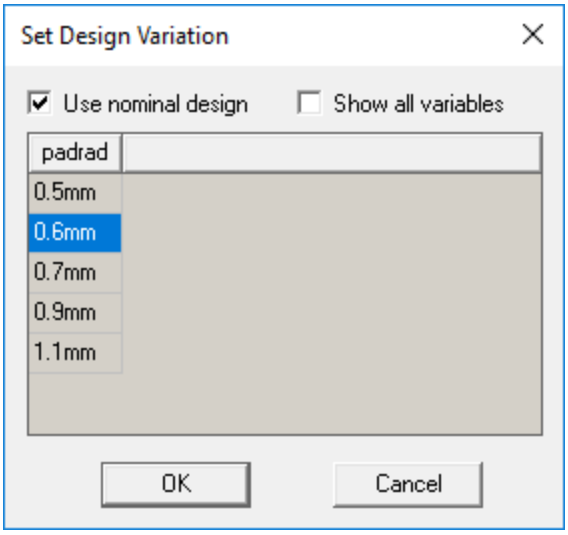


The **Solutions** window appears, with the **Matrix** tab selected.



2. In the **Design Variation** text box, set the design variation by clicking the ellipses (...) icon.

   The **Set Design Variation** window appears, listing all solved variations in the design.



> **Note:**
>
> You can also access this window from the project tree by right-clicking **Results** and selecting **Apply Solved Variation**.

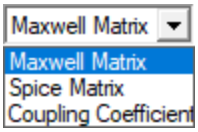
3. From the **Simulation** drop-down menus, you can select setups and either adaptive or sweep solutions. Select the solution and type of matrix you want to view (CG, DC RL, or AC RL). Available types differ based on the solution type.
4. The **Matrix** tab contains four sub-tabs: **View**, **Format**, **Passivity**, and **Export**. The **View** tab displays by default.

   From the **View** tab:

   - Select the **Units** in which to display matrix information. Available units depend on the matrix type.

     The units are saved in the matrix panel. Units are stored only for the current session and current project. Changing the units in a project does not affect the units in another project open at the same time. Closing the project resets the units, and changes are lost.

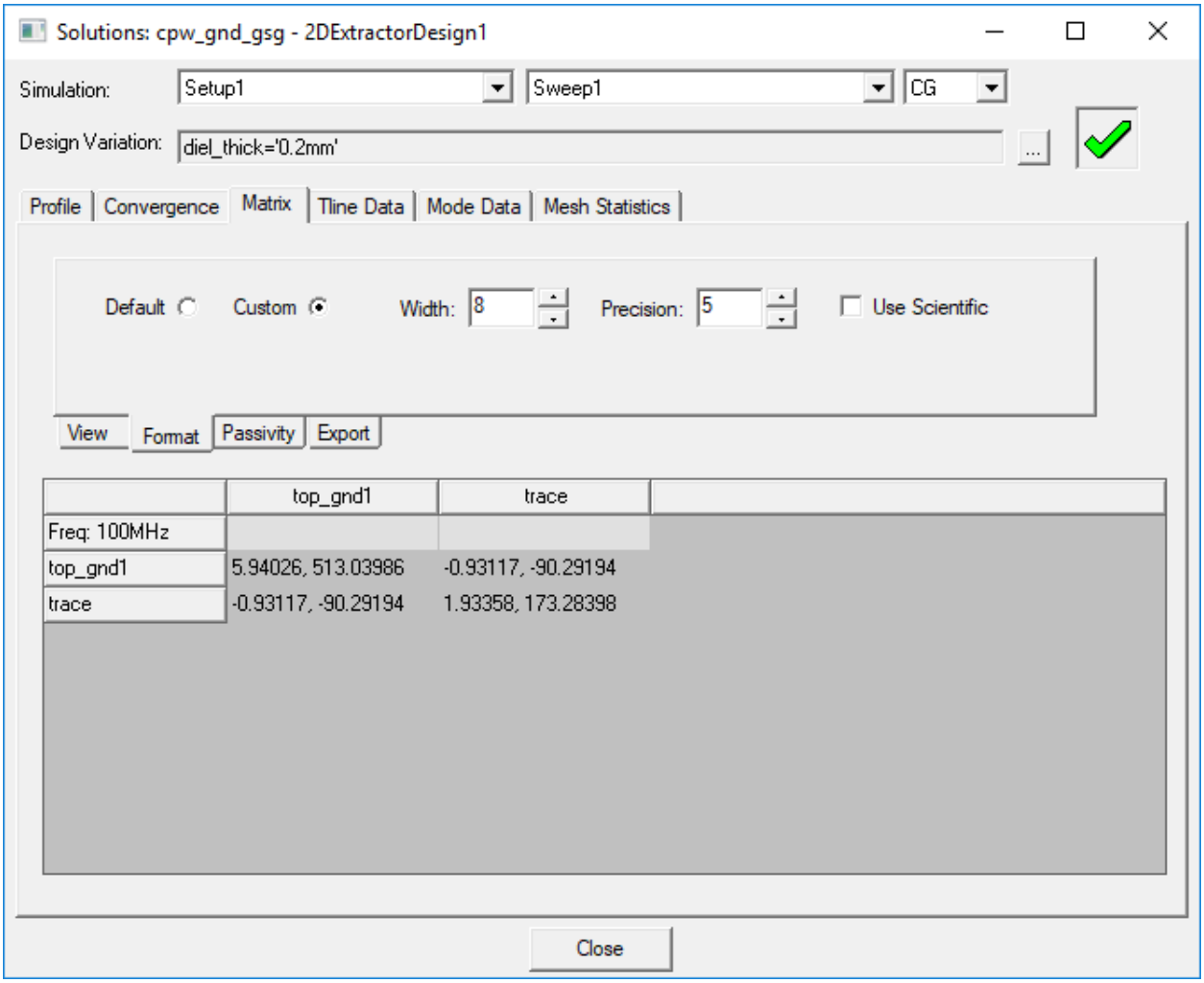
   - Use the drop-down menus to select reduced matrices to display. Choices include Coupling Coefficient, Maxwell Matrix, and Spice Matrix. Maxwell Matrix and Spice Matrix are only available in the Capacitance matrix. By default, the **Original** solved matrix displays.



- To show matrix entities for reduced operations, click the **Original** menu and select **Reduced** matrices.
- Select the solved frequencies to display using the provided options:
  - **Frequency Drop-Down Menu** – allows you to select the frequency for which you want to view matrix entries.
  - **All Freqs** – allows you to display matrix entries for all solved frequencies.
  - **Pass** – for an Adaptive Pass, allows you to select the pass for which you want to view matrix entries.
  - **Edit Freqs** – for an Interpolating Sweep, opens the **Edit Sweep** window, allowing you to insert or delete displayed frequencies.
- Use the **Self Terms** check box to display only the diagonal of the selected matrix.

5. Click the **Format** tab.

   The **Format** tab displays, with additional options.

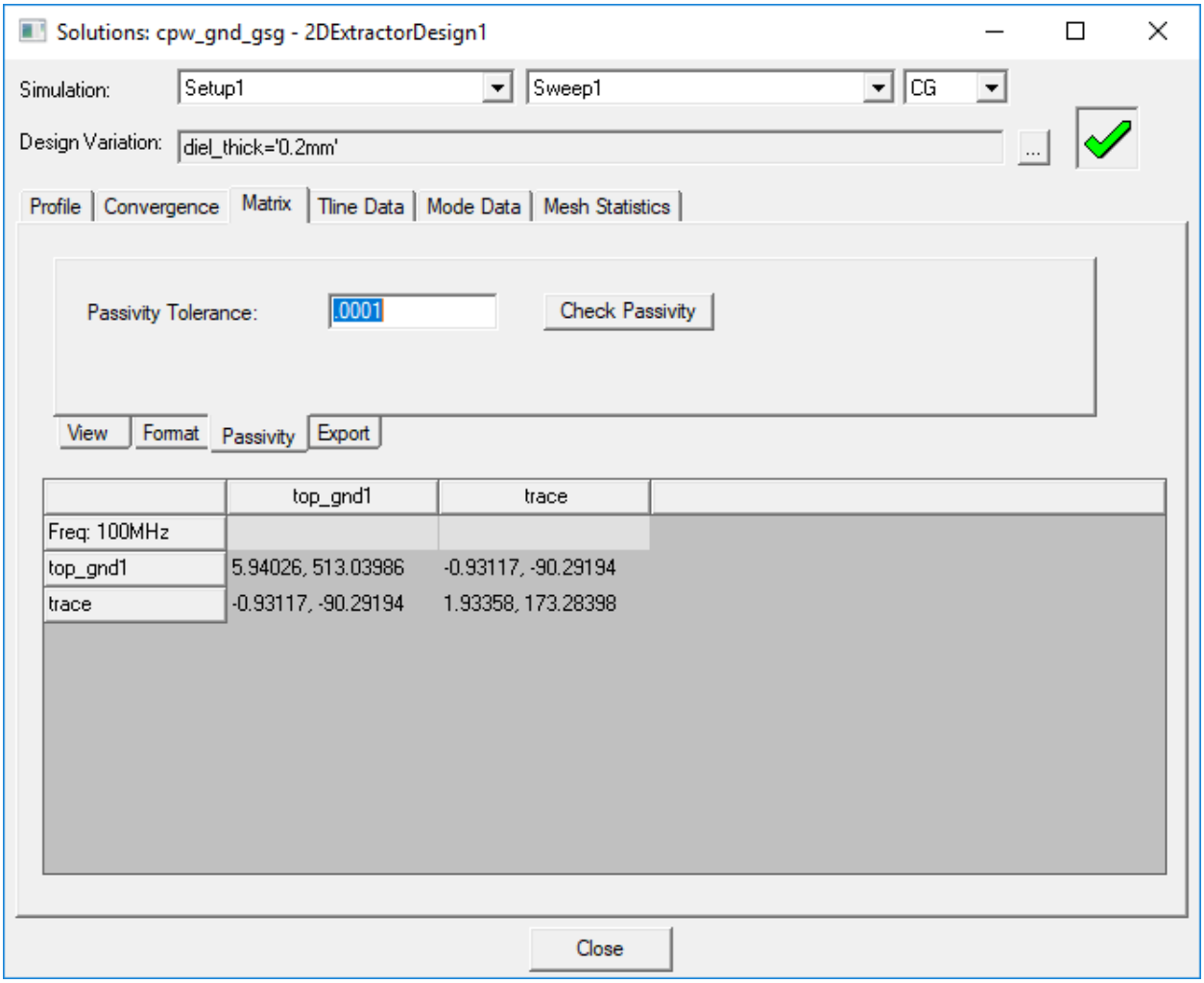


From the **Format** tab:

- Select either the **Default** or **Custom** display.
- If you select **Custom**, specify the matrix field **Width** and **Precision** in digits.
- Select the **Use Scientific** check box to display matrix data using scientific notation.

6. Click the **Passivity** tab.

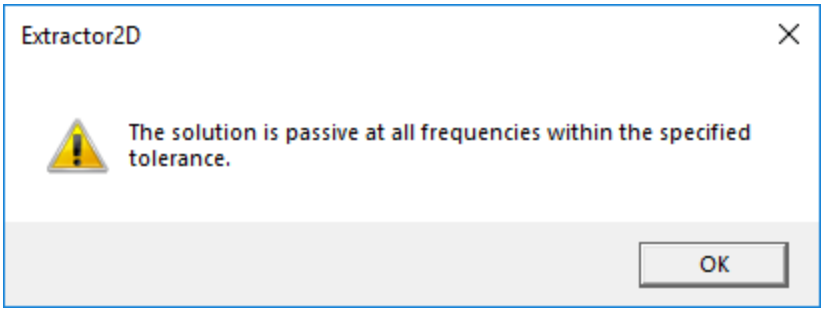
   The **Passivity** tab displays.



From the **Passivity** tab:

- Set the **Passivity Tolerance**.
- Click **Check Passivity** to check passivity under the specified Passivity Tolerance.

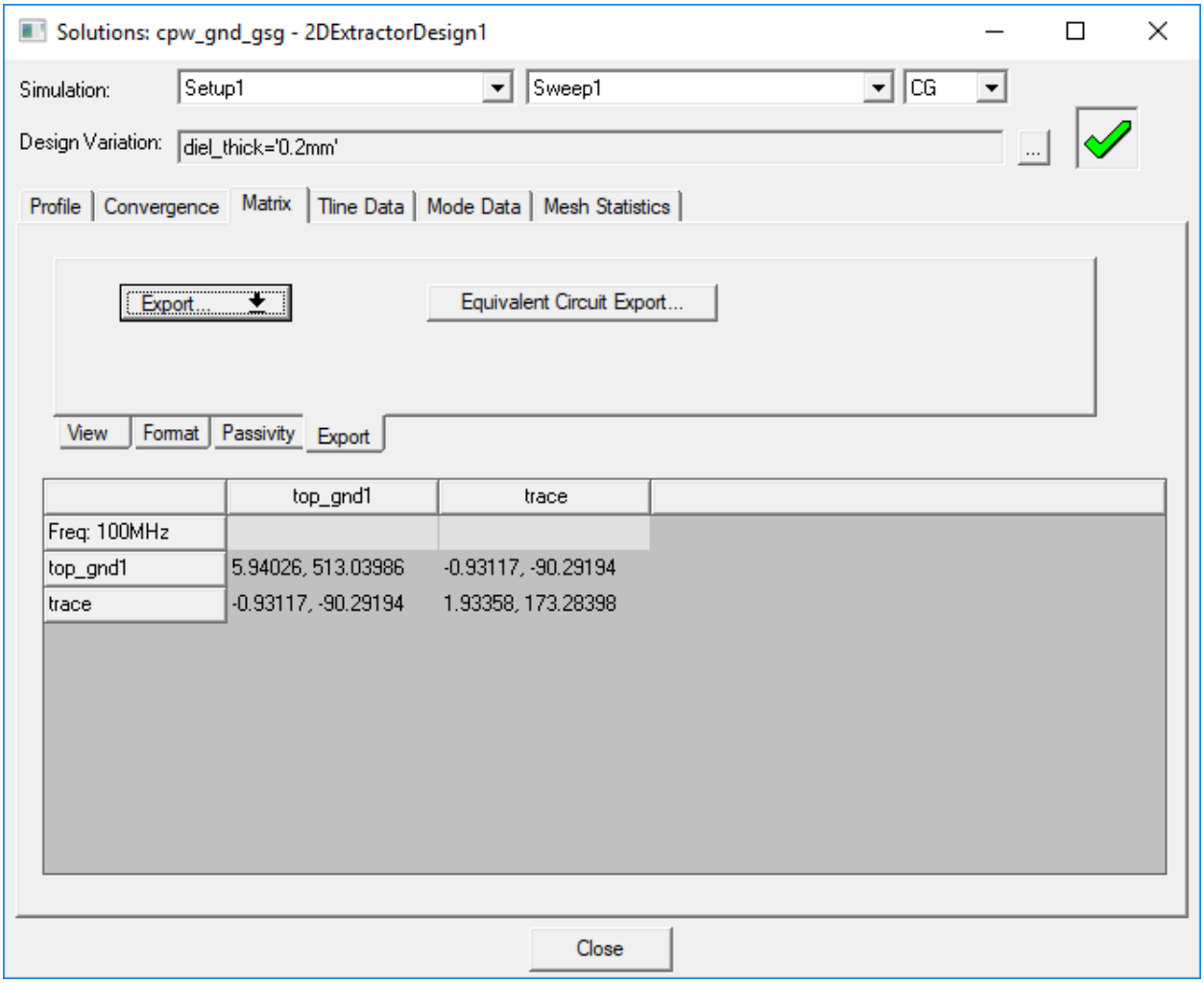
  An alert appears, showing the result.



- Click **OK** to close the alert.

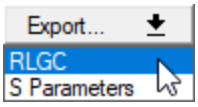
7. Click the **Export** tab.

   The **Export** tab displays.

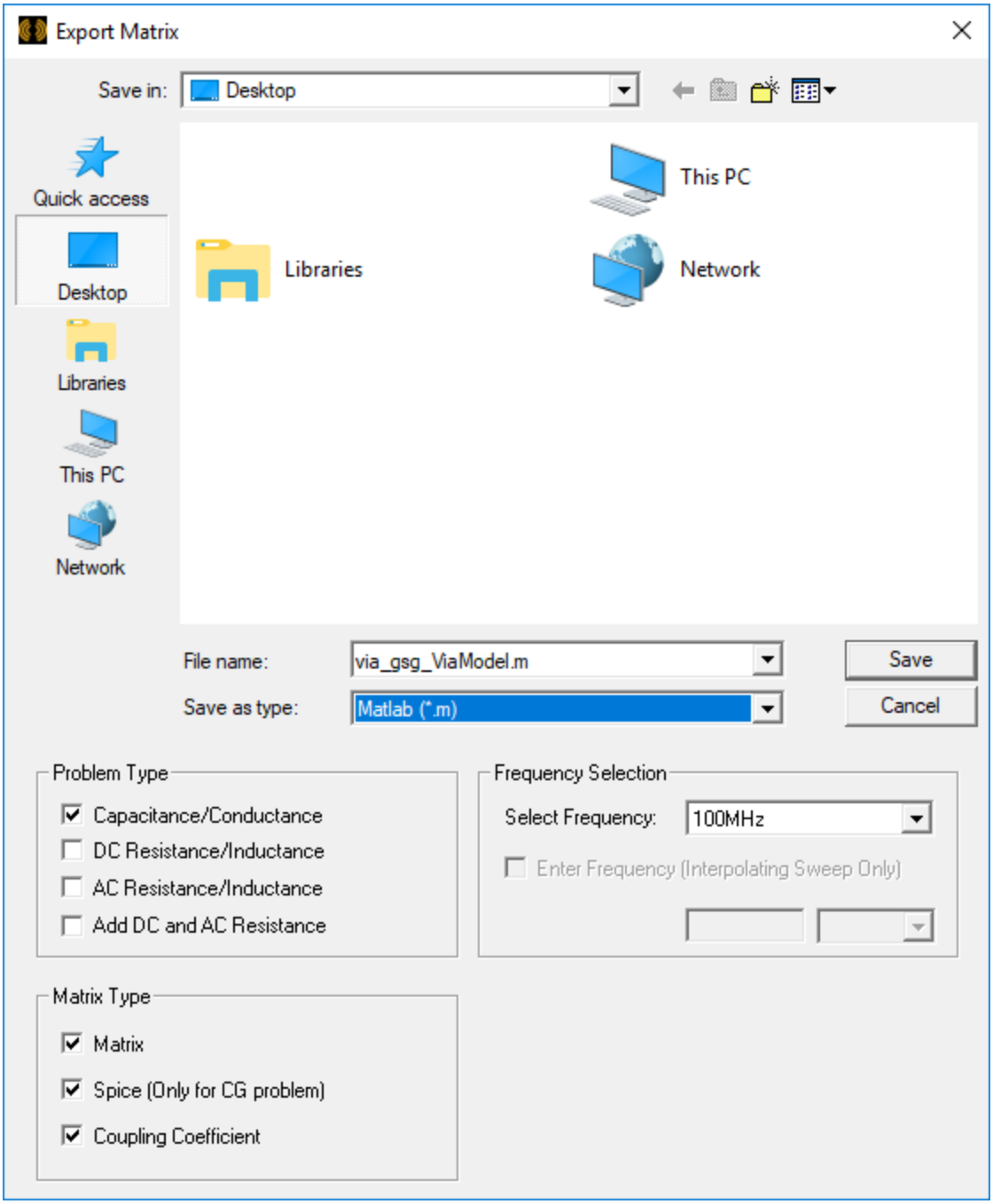


From the **Export** tab:

- To export matrix data, click **Export** and select either **RLGC** or **S Parameters** from the drop-down menu that appears.



The **Export Matrix** window appears.



Select your desired options and click **Save**.

- To export the equivalent circuit, click **Equivalent Circuit Export**.

> **Note:**
>
> You can only export the current problem type, and will receive an error message if you try to export both CG and RL.

8. Click **Close** to exit the **Solutions** window.

> **Note:**
>
> Selections in the **Solutions** window persist during execution, but are reset to their defaults on exit.

Coupling Coefficient for 2D Extractor

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:

$$abs\left(M_{ij}\right) / \left(\sqrt{\left(M_{ii} \times M_{jj}\right)}\right)$$

Maxwell Matrix for 2D Extractor

The entries of the Maxwell capacitance matrix represent the total charge on a conductor due to the voltages on the various conductors.

If Q_i is the total charge on conductor i, and V_1, V_2, ..., V_N are the conductor voltages, then:

$$Q_i = C^M{}_{i,1} * V_1 + C^M{}_{i,2} * V_2 + \ldots + C^M{}_{i,N} * V_N$$

where the superscript *M* is used for the Maxwell capacitance matrix.

Spice Matrix for 2D Extractor

The entries of the Spice capacitance matrix represent the values of a network of 2-terminal capacitors. Imagine that between every pair of conductors (say, i and j, i different from j), you have a capacitor of value $C^S_{i,j}$. In addition, let every conductor (say the i^{th}) have a capacitor $C^S_{i,0}$ connected to "ground" (the values of these grounded capacitors are the entries on the main diagonal of the Spice capacitance matrix). The total charge on a given conductor will be found by summing up the charges on the individual capacitor plates connected to it.

$$Q_i = C^S_{i,0} * Vi + C^S_{i,1} * (V_i - V_1) + C^S_{i,2} * (V_i - V_2) + ... + C^S_{i,N} * (V_i - V_N)$$

Relationship between Maxwell Matrix and Spice Matrix for 2D Extractor

Because there are two different expressions for the same thing, the entries of the Maxwell matrix and Spice capacitance matrices can relate to each another.

Matching the coefficients of the voltages results in:

$(1) C^{M}_{i,i} = C^{S}_{i,0} + C^{S}_{i,1} + C^{S}_{i,2} + \ldots + C^{S}_{i,N-1}$

$(2) C^{M}_{i,j} = - C^{S}_{i,j}$ *(for i not equal to j)*

So the off-diagonal entries of the Maxwell capacitance matrix (which are generally negative) are the opposite of the off-diagonal entries of the Spice capacitance matrix (which are positive.) And the main-diagonal entries of the Maxwell matrix are the sum across an entire row of the Spice capacitance matrix.

Alternatively, you could express the grounded capacitor in the Spice matrix as the sum across a whole row of the Maxwell capacitance matrix:

$C^{S}_{i,0} = C^{M}_{i,1} + C^{M}_{i,2} + \ldots + C^{M}_{i,N}$

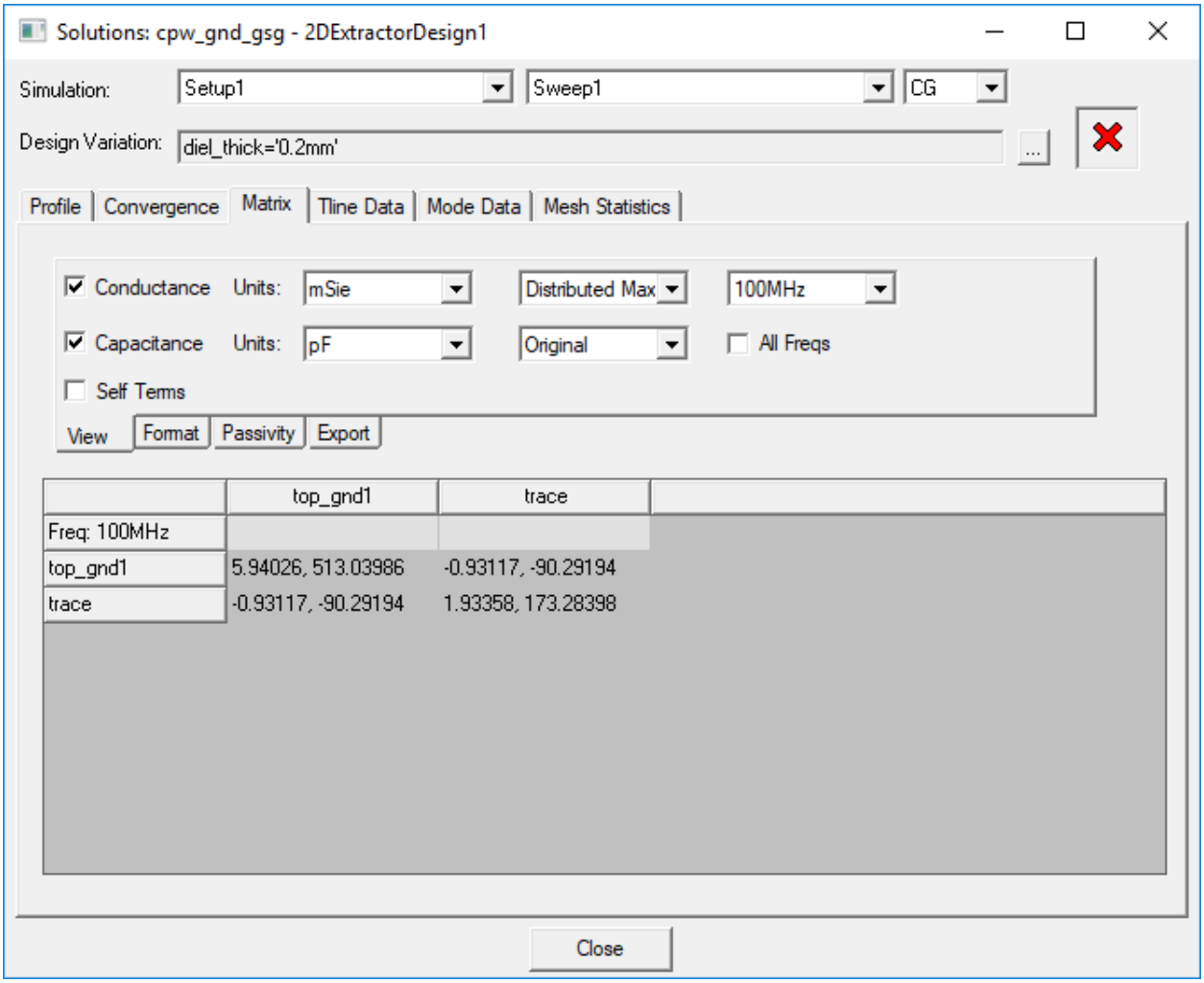
> **Note:**
>
> Since the off-diagonal entries of the Maxwell matrix are negative, the diagonal entry of the Spice matrix is generally smaller than the diagonal entry of the Maxwell matrix. It can even go to zero if the conductor is well shielded from the ground at infinity.

Exporting RLGC Matrix Data for 2D Extractor

To export RLGC matrix data:

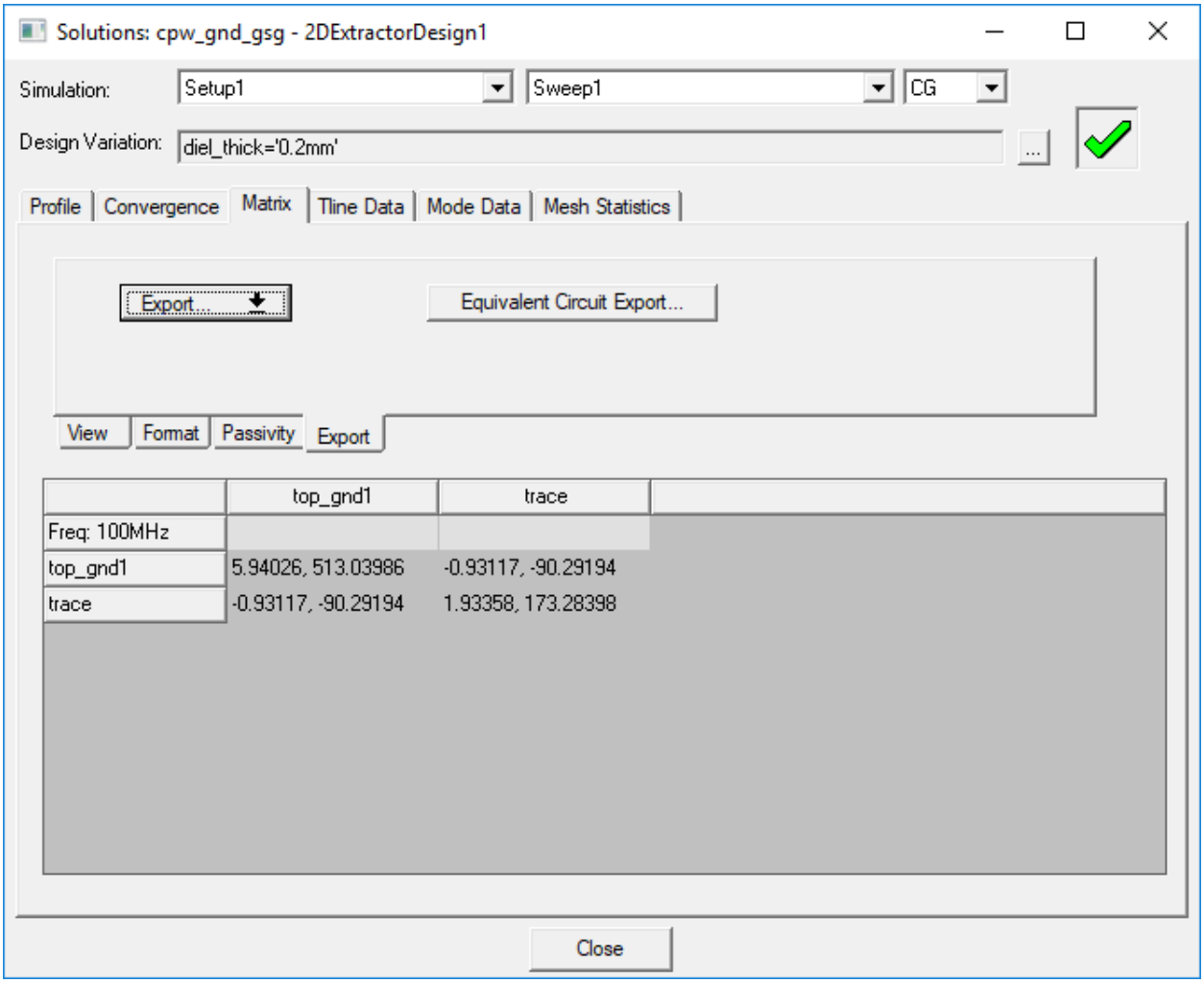
1. In the project tree, right-click the solution setup and select **Matrix** from the shortcut menu. Alternatively, click **Results** > **Solution Data** and select the **Matrix** tab.

   The **Solutions** window appears with the **Matrix** tab selected.

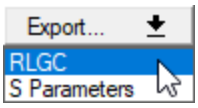


2. Click the **Export** tab.

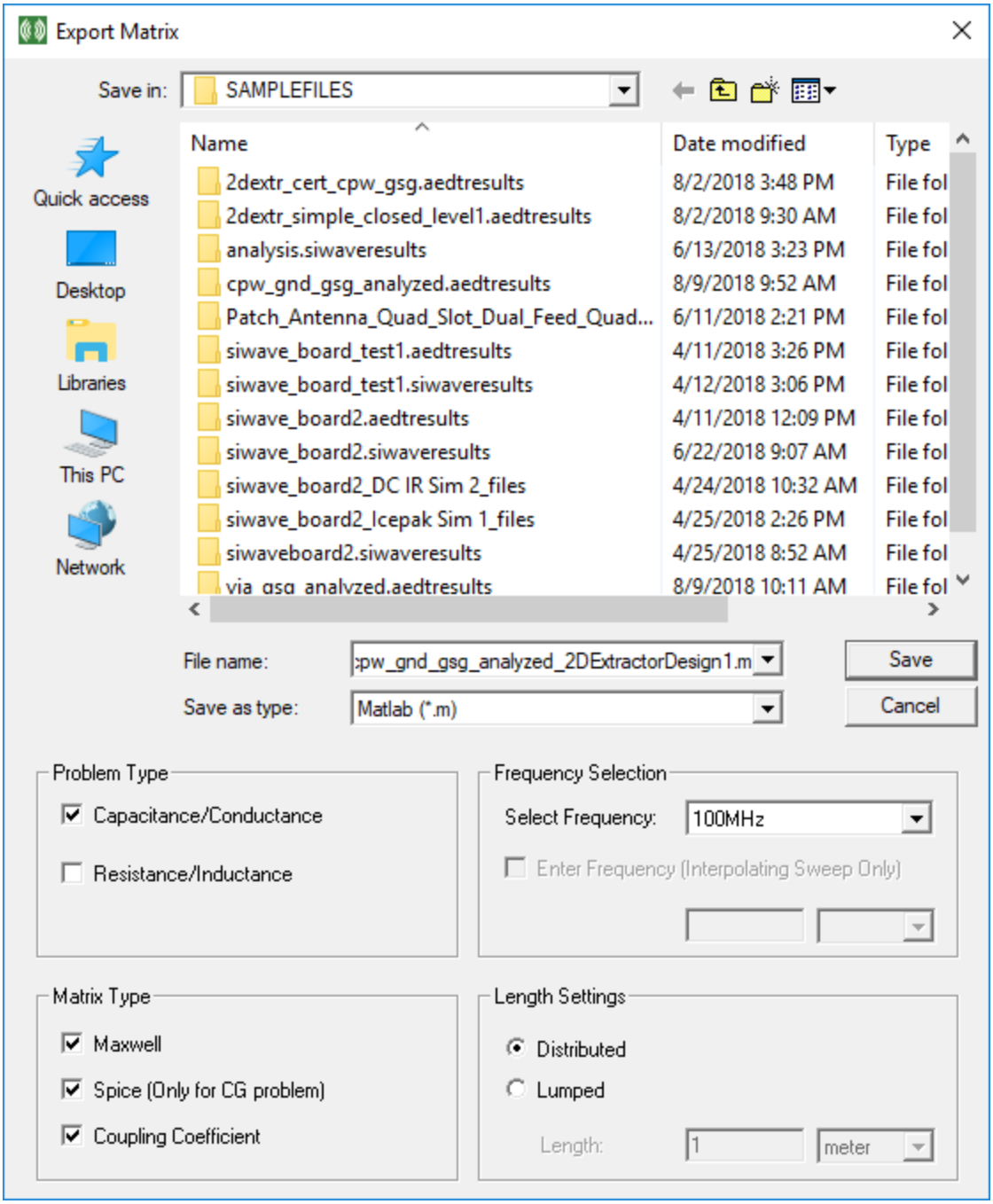
   The **Export** tab appears.



3. Click **Export** and select **RLGC** from the drop-down menu.



The **Export Matrix** window appears.



4. Enter the name of the file you are exporting to in the **File name** field.
5. Select one the following file formats from the **Save as type** drop-down menu:
    - Ansys EM Legacy Format Files (*.lvl)
    - Data Table (spreadsheet) (*.txt)

- Matlab (*.m)
- Ansys EM Legacy Capacitance Matrix (*.cap)
- Ansys EM Legacy Inductance Matrix (*.ind)
- Ansys EM Legacy Admittance Matrix (*.adm)
- Ansys EM Legacy Impedance Matrix (*.imp)

The default file type is .**txt**.

If you select the data table format, a text file will be created. The elements of the RLGC-matrix are arranged in a series of columns that are tab-separated and include a first row of headings.

If you select the Matlab format, the elements of the R-,L-, G-, or C- matrix will be arranged in a series of rows.

6. Select the **Problem Type** of the RLGC data to be exported. You can select **Capacitance/Conductance** or **AC Resistance/Inductance**.

   The convention of entry names remains the same as in the previous version:

   - For RL problem types, the return path name is specified along with entry name. For example: “Object1-Ground1”.
   - If there is a default return path or surface ground is defined, then return path name is left empty. For example “Object1-”.
   - If any reduce matrix is specified, then no return path is specified. For example: “Object1”.

   > **Note:**
   >
   > RLGC matrix data is always exported in SI units just like in lvl format.

7. Select the **Matrix Type**.
8. Select or enter the **Frequency**.
9. Select the **Length Settings** to be applied to the matrices. In the previous version, matrices were always in SI units and always in distributed settings with length units in SI. To have the same matrices as the previous version, click **Distributed**.

   If you select **Lumped**, you can enter the value and units for the **Length**.

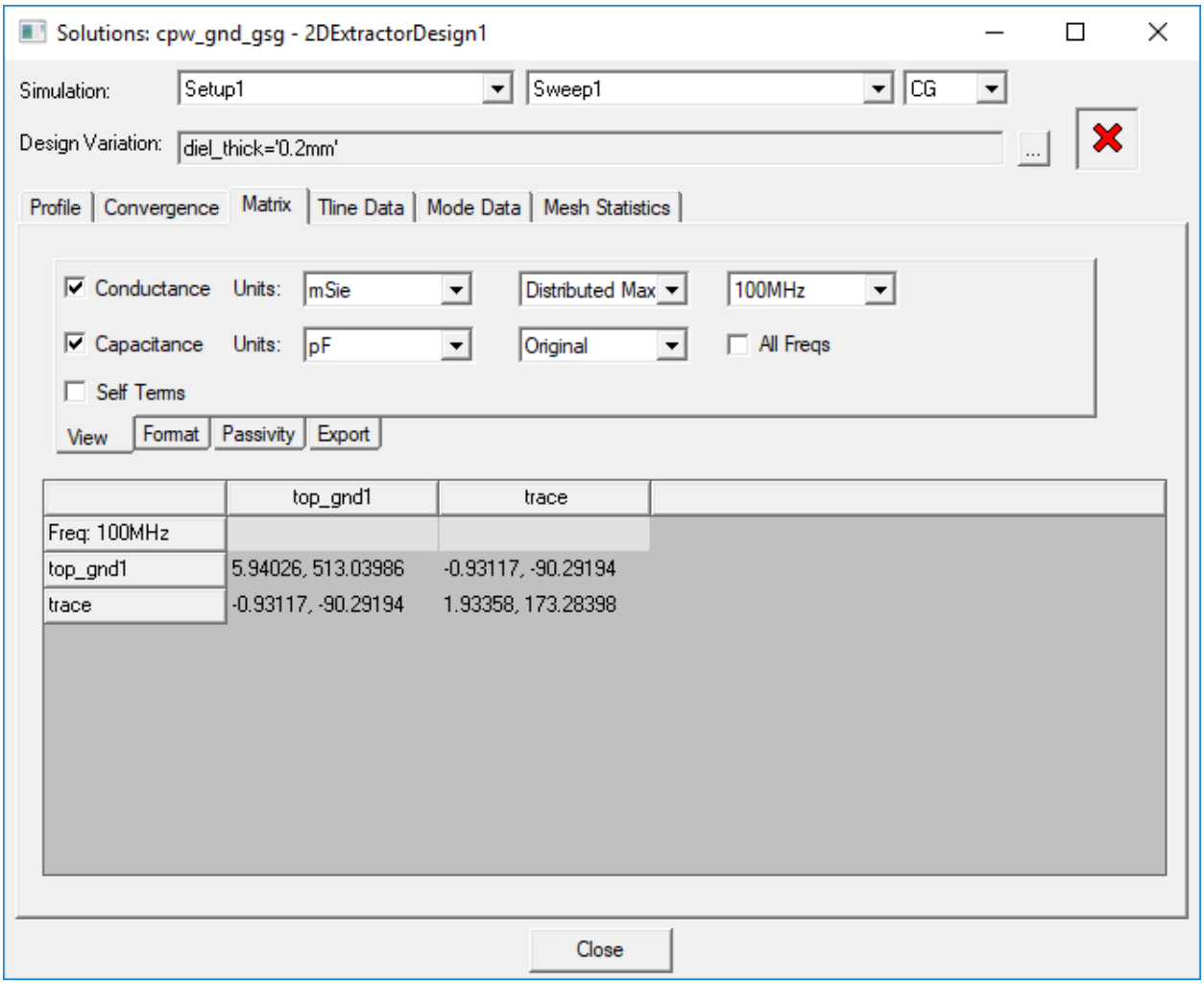
9. Click **Save**.

Exporting S Parameter Data for 2D Extractor

To export S parameter data:

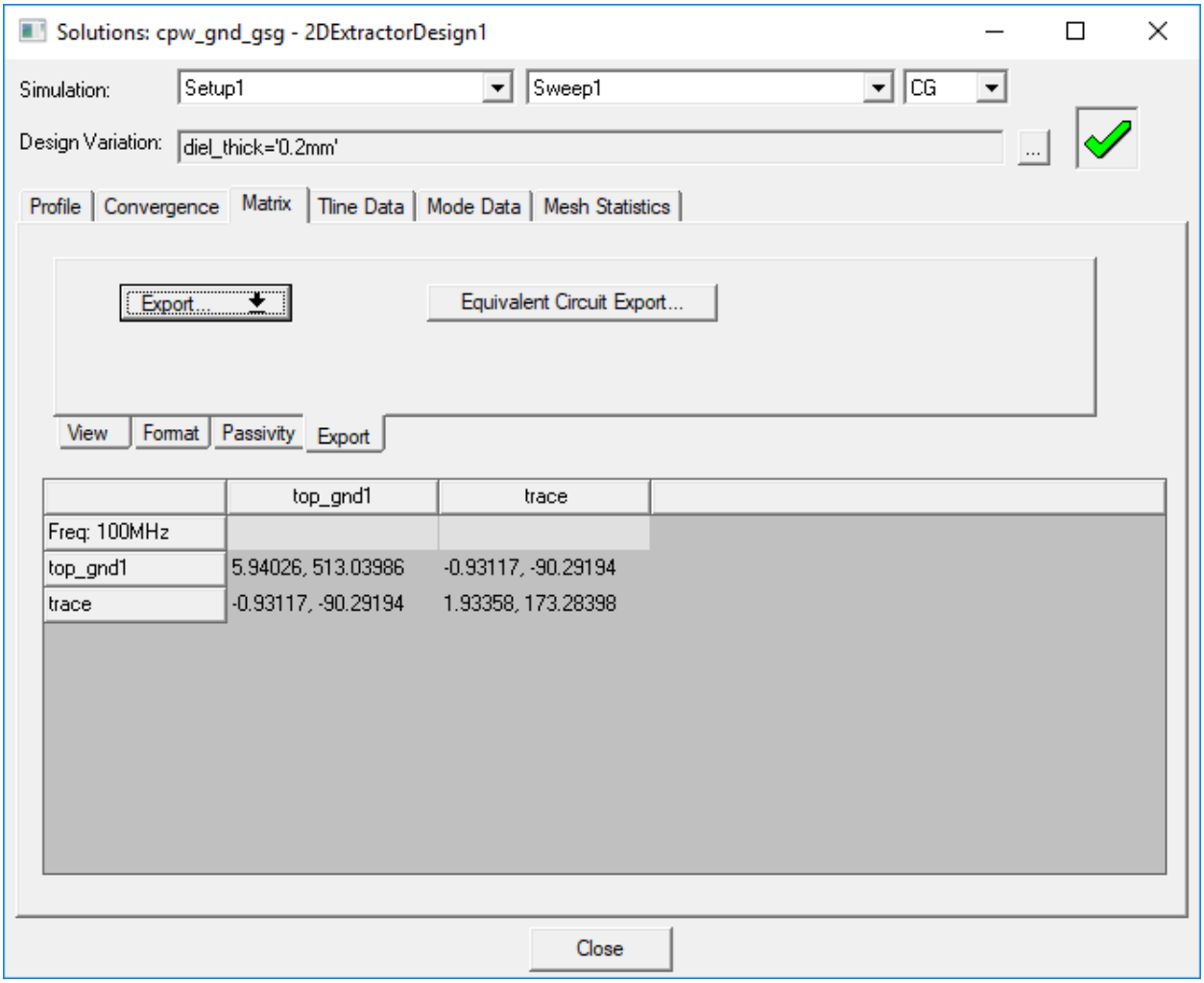
1. In the project tree, right-click the solution setup and select **Matrix** from the shortcut menu. Alternatively, click **Results** > **Solution Data** and select the **Matrix** tab.

   The **Solutions** window appears with the **Matrix** tab selected.

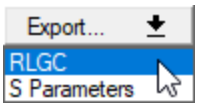


2. Click the **Export** tab.

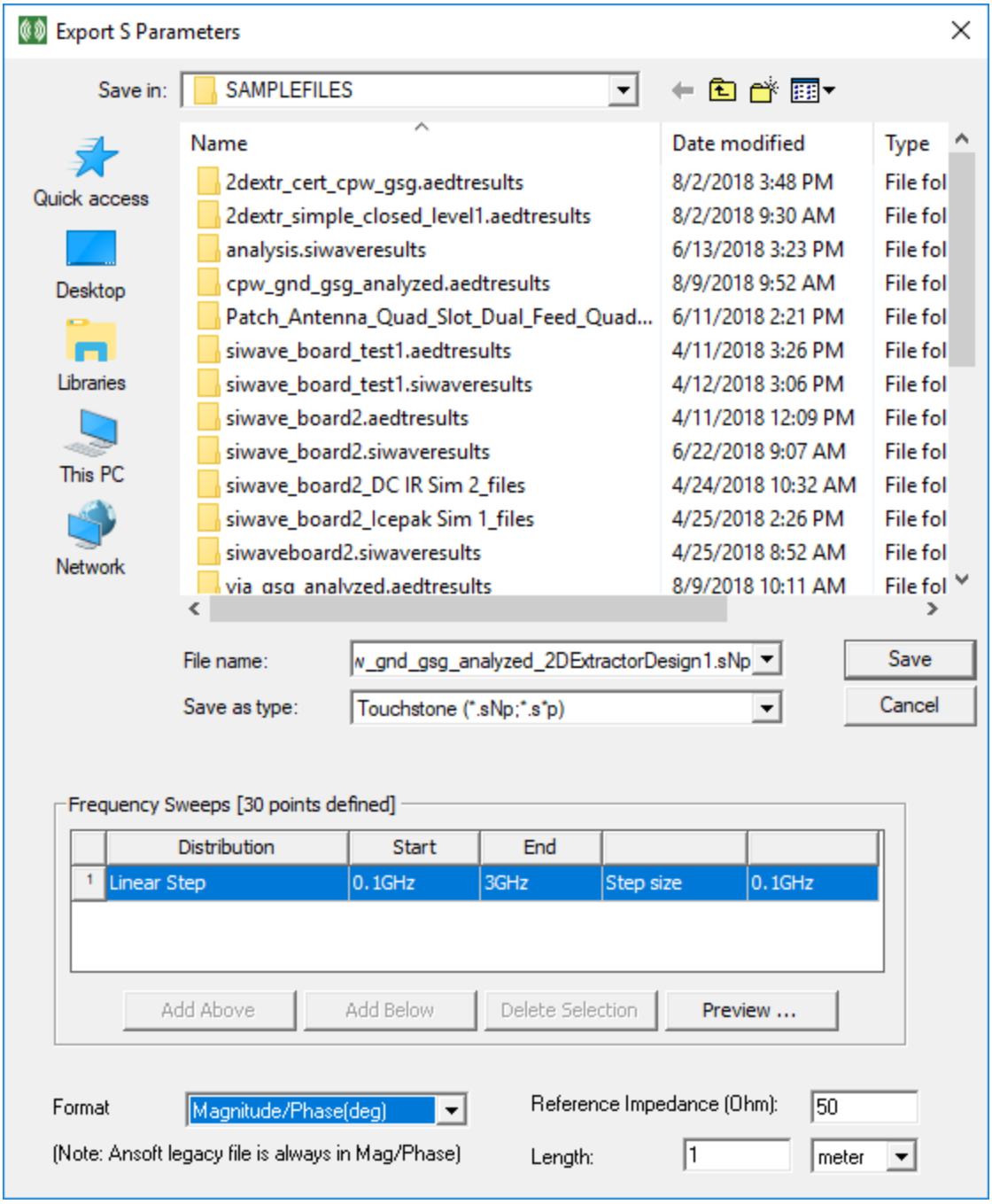
   The **Export** tab appears.



3. Click **Export** and select **S Parameter** from the drop-down menu.



The **Export S Parameters** window appears.



4. Enter the name of the file you are exporting to in the **File name** field.
5. Select one the following file formats from the **Save as type** drop-down menu:
    - Ansys EM Legacy Format Files (*.szg)
    - Citi Files (*.cit)

- Data Table (spreadsheet) (*.txt)
- Matlab (*.m)
- Neutral Model Format (*.nmf)
- Touchstone (*.sNp, *.s*p)

If you select the data table format, a text file will be created. The elements of the S-matrix are arranged in a series of columns that are tab-separated and include a first row of headings.

If you select the Matlab format, the elements of the S-,Y-, or Z- matrix will be arranged in a series of rows.

If you select the Touchstone format, a Touchstone S parameter file will be created. The number of ports is indicated by *N*. For example, a Touchstone file with one port would have the extension *.s2p.

6. Specify the **Frequency Setup**:
    a. Specify the **Start** frequency, **Stop** frequency, and **Step** size.
    b. Select the units.
    c. Click **Display** to see the selected values.
7. Select the format in which the data is to be exported. The available formats depend on the matrix type being displayed.
8. Enter a value for the **Reference Impedance**.
9. Click **Save**.

Viewing Mesh Statistics for 2D Extractor

To view an adaptive solution's mesh information, either during or after the solution process:

- From the **Project Manager**, right-click a solution setup and select **Mesh Statistics**.

The **Solutions** dialog box appears, with the **Mesh Statistics** tab selected.

The table lists the design elements and depending on the solution type may include: Num Elements, Min edge length, Max edge length, RMS edge length, min elem area, max elem area, mean elem area, and standard deviation.

If mesh repairs have been performed, two additional columns appear in the table: **Recovered %**, and **Repaired %**. These columns indicate the fraction of an object that was successfully recovered and the fraction that needed some repair.

Viewing Mode Data

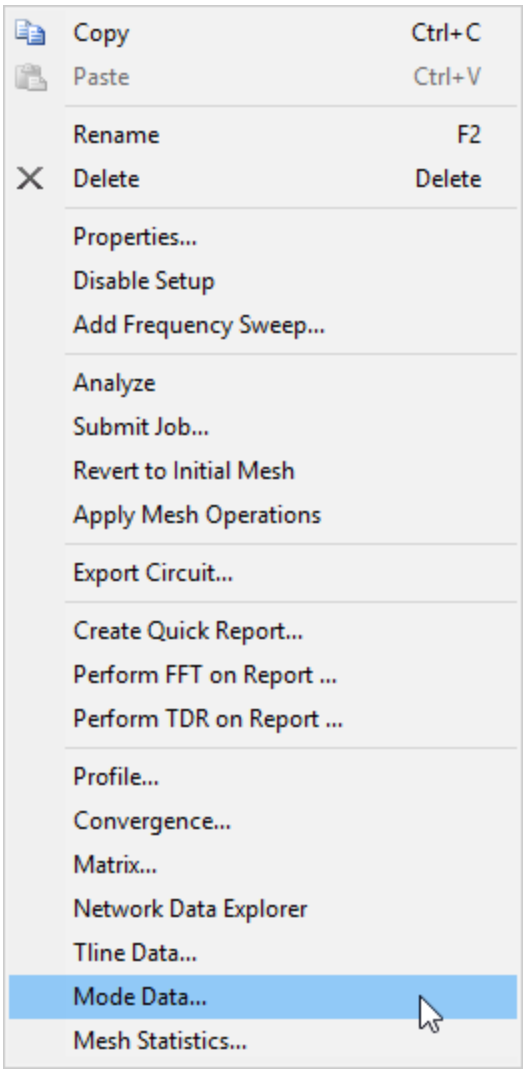
The 2D Extractor computes *distributed* values of capacitance and inductance along the length of each line, which would require a ladder network model consisting of a large number of

inductances and capacitances. To model the transmission line, the software uses modal transformation matrices to transform an *n*-conductor coupled line into an *n*-mode decoupled line, where each mode can be visualized as a specific pattern of voltages on the physical conductors.

To view Mode Data, either during or after the solution process:

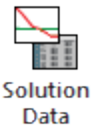
1. Access the **Solutions** window one of two ways:
    - In the project tree, under **Analysis**, right-click the solution setup.

      The right-click menu appears.



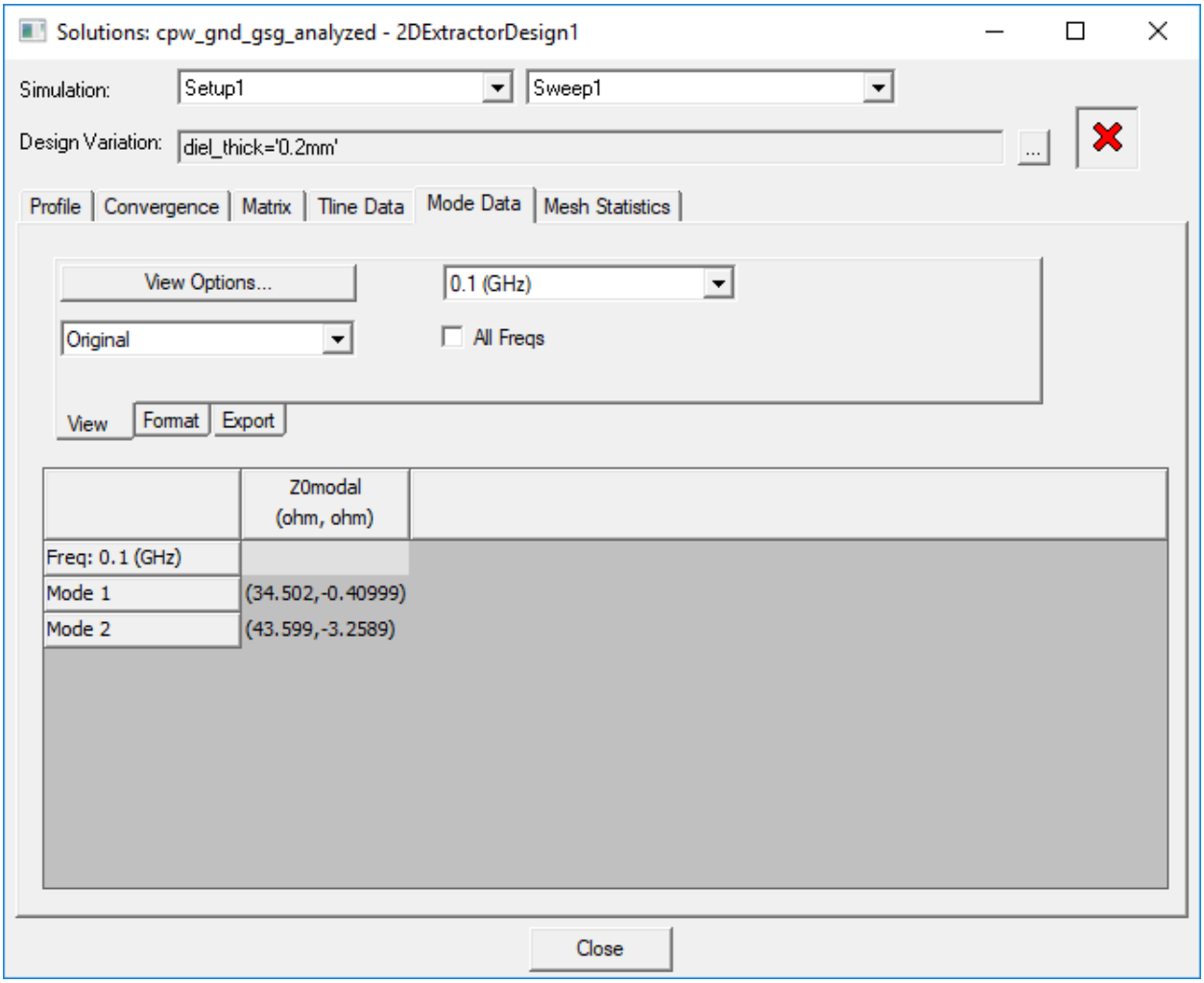
      Select **Mode Data...**

    - Or, from the ribbon, click **Results > Solution Data**.



Select the **Mode Data** tab.

The **Solutions** window appears, on the **Mode Data** tab.

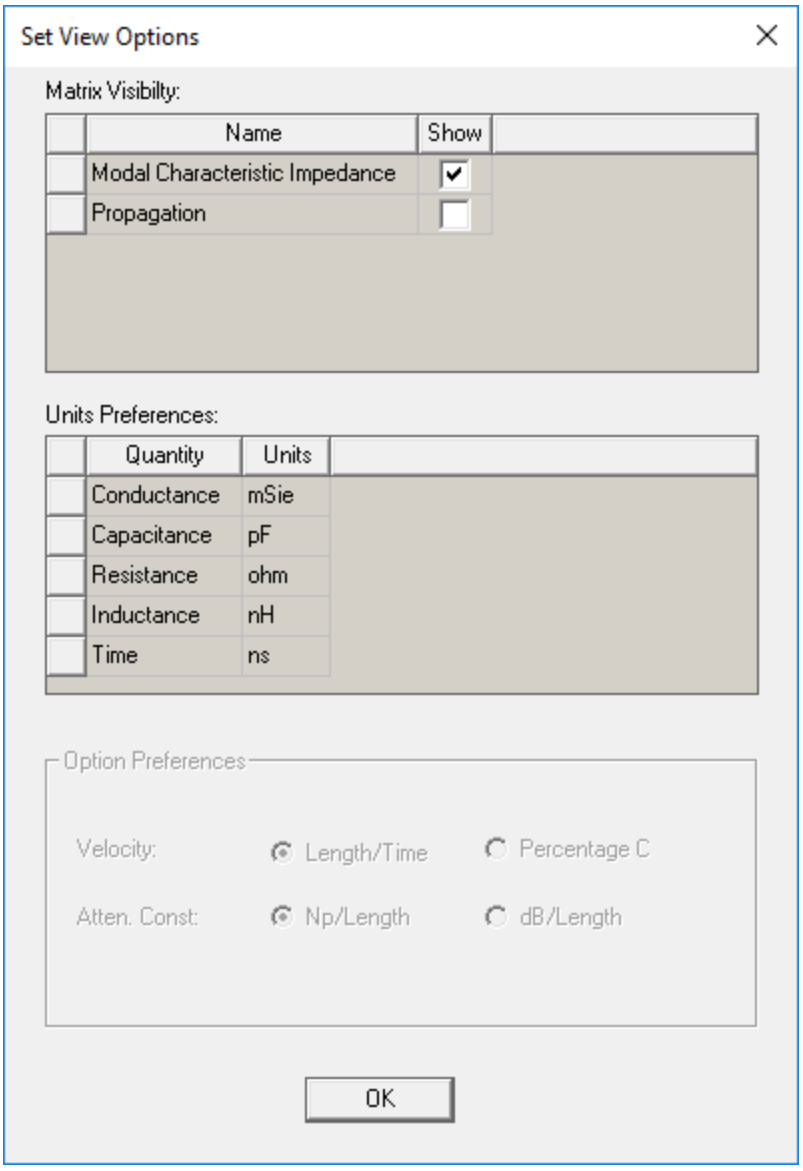


2. The **Mode Data** tab contains three sub-tabs: **View**, **Format**, and **Export**. The **View** tab displays by default.

   From the **View** tab:

- Use the drop-down menu to choose from **Original** or **Reduced** matrices.

- Use the drop-down menu to select a frequency, or select the **All Freqs** check box.
- Click **View Options** to open the **Set View Options** window.

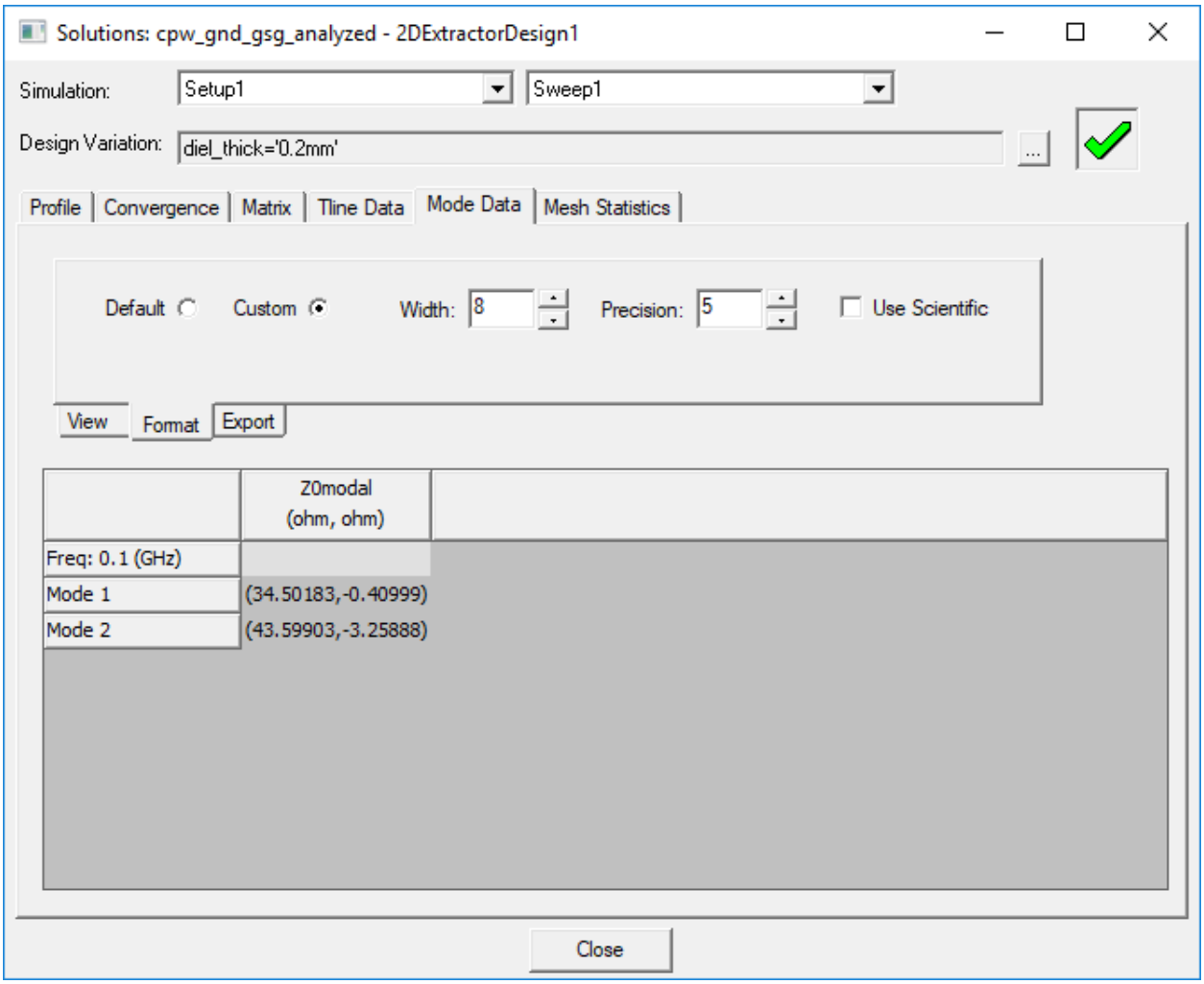


- From this window, you can set **Matrix Visibility** for Modal Characteristic Impedance and Propagation Velocities, as well as preferred units of measurement.
- Click **OK** to apply changes.

> **Note:**
>
> Selections persist during execution, but are reset to their defaults on exit.

3. Click the **Format** tab.

   The **Format** tab displays, with additional options.

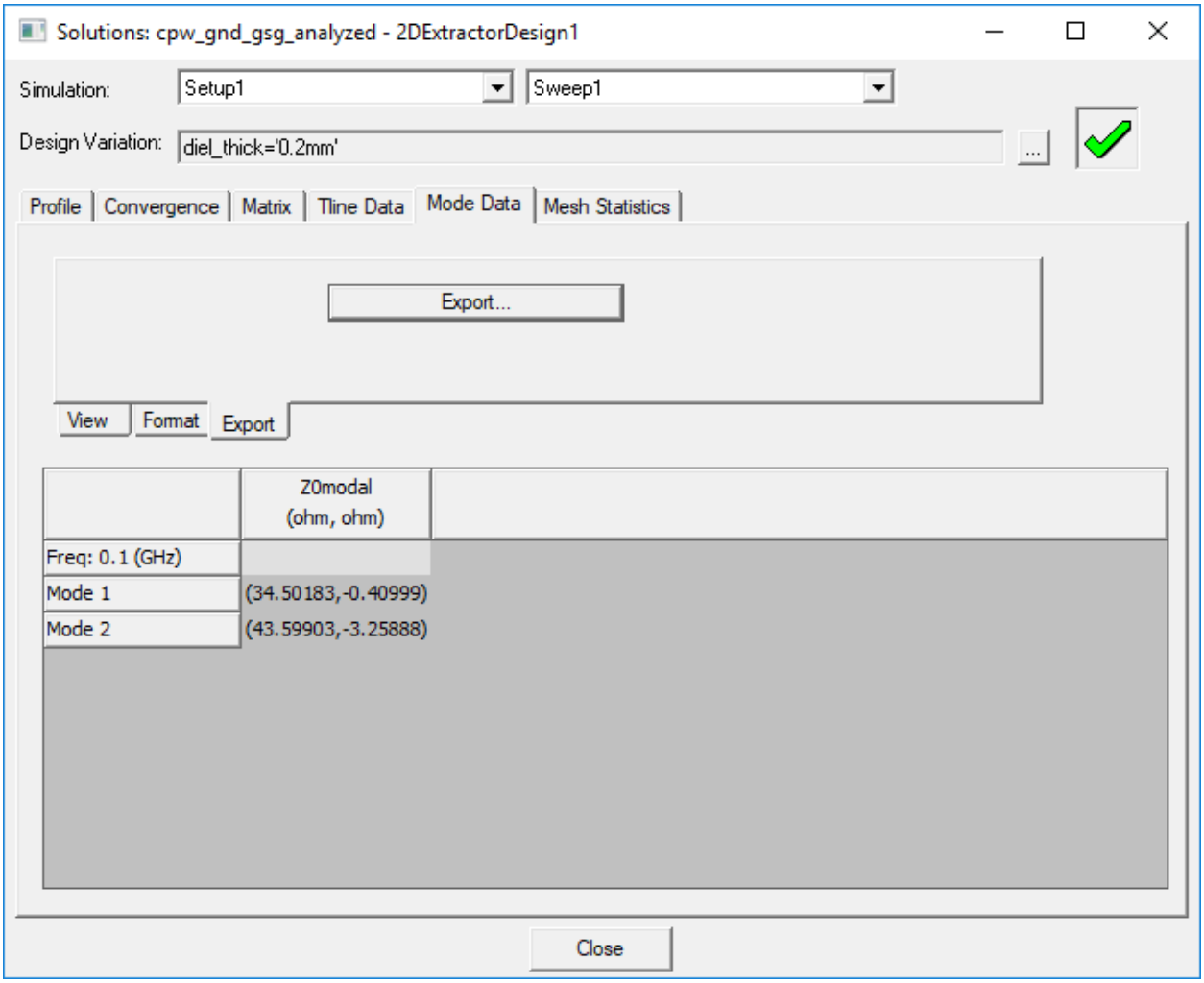


   From the **Format** tab:

   - Select either the **Default** or **Custom** display.
   - If you select **Custom**, specify the matrix field **Width** and **Precision** in digits.
   - Select the **Use Scientific** check box to display Mode data using scientific notation.

4. Click the **Export** tab.

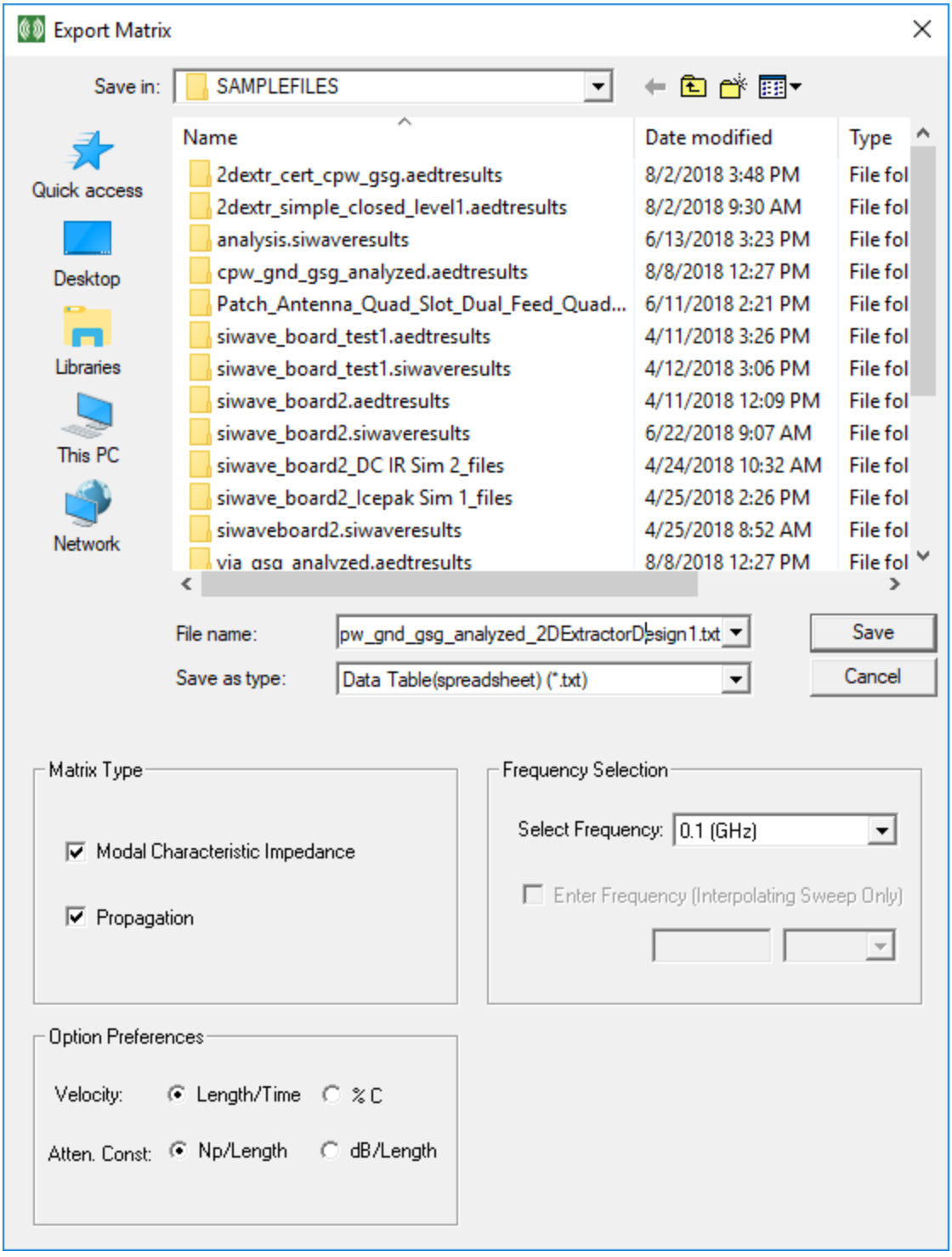
   The **Export** tab displays.



From the **Export** tab:

- To export Mode data, click **Export**.

  The **Export Matrix** window appears.



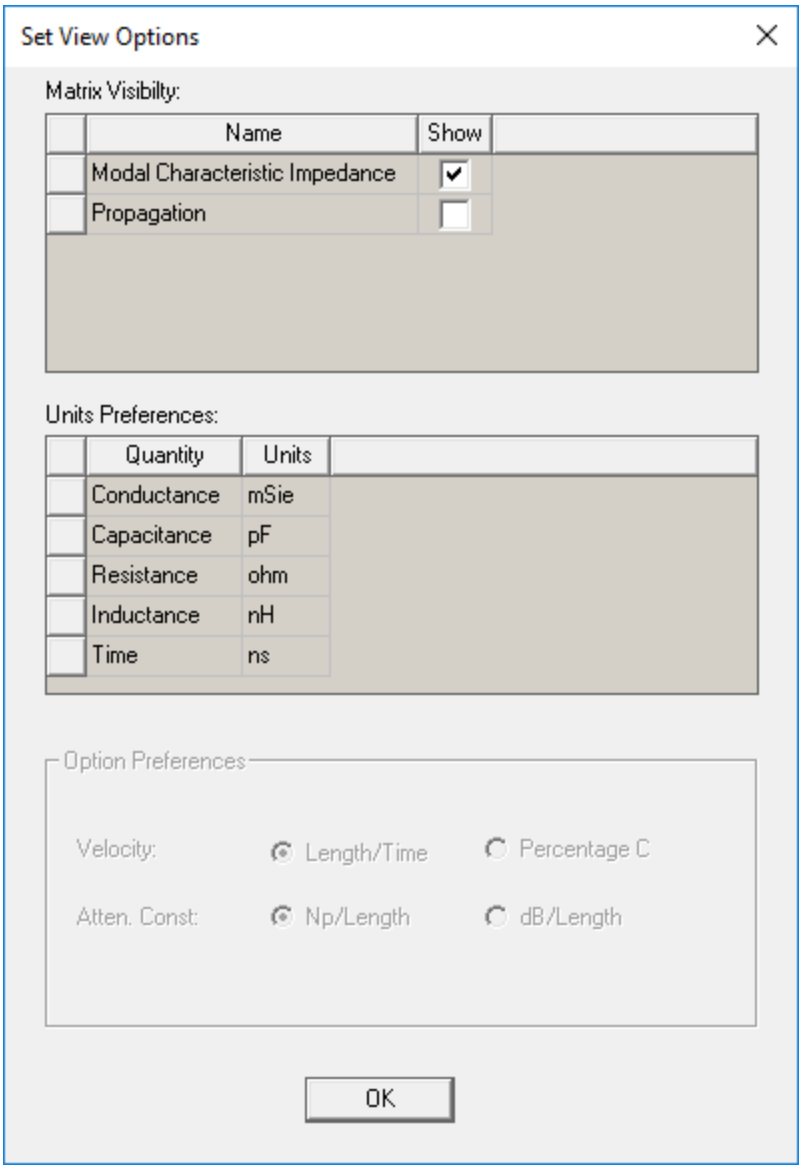
- Select your desired options and click **Save**.

5. Click **Close** to exit the **Solutions** window.

Modal Characteristic Impedance

2D Extractor can display characteristic impedances resulting from a system transformed using the modal transformation matrices.

From the **Set View Options** window, you can select **Modal Characteristic Impedance**.



Note:

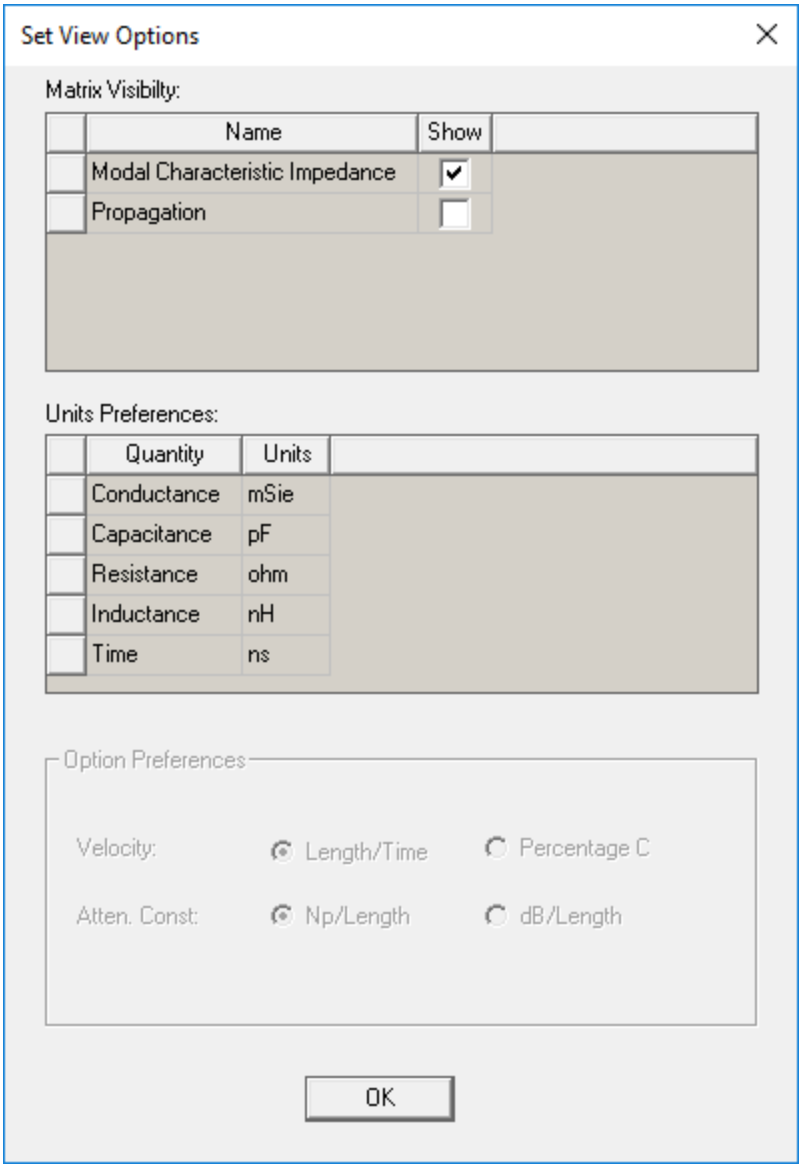
You must solve for both admittance and impedance to activate this option.

The modal characteristic impedance is *not* the characteristic impedance matrix for the physical transmission structure, and is generally only directly useful in simple differential problems. It is used by the software to calculate the characteristic impedance matrix for the structure.

Propagation

2D Extractor can display signal propagation velocities associated with each mode.

From the **Set View Options** window, you can select **Propagation**.



Some important notes:

- You must solve for both admittance and impedance).
- Modal transformation matrices are used to decouple the lines in the problem and to view the resulting values for the decoupled lines.

You can set the **Option Preferences** to view the **Velocity** and **Attn. Const**.

Propagation allows you to view the following:

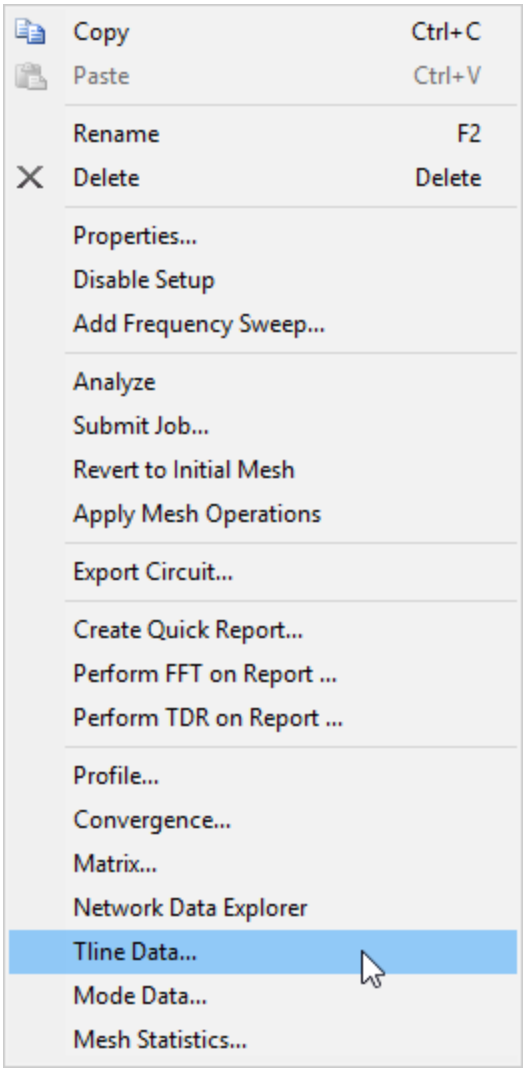
Velocity	Propagation velocity, given in units of length/second or as a percent of the speed of light.
Atten. Const	Attenuation constant, a, given in nepers/unit of length.
Eps_eff	Effective dielectric constant, e.
Lambda_ eff	Effective wavelength, l, given in units of length.
Delay	Inverse of velocity.

Viewing Tline Data

To view Tline Data, either during or after the solution process:

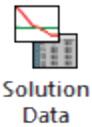
1. Access the **Solutions** window one of two ways:
    - In the project tree, under **Analysis**, right-click the solution setup.

      The right-click menu appears.



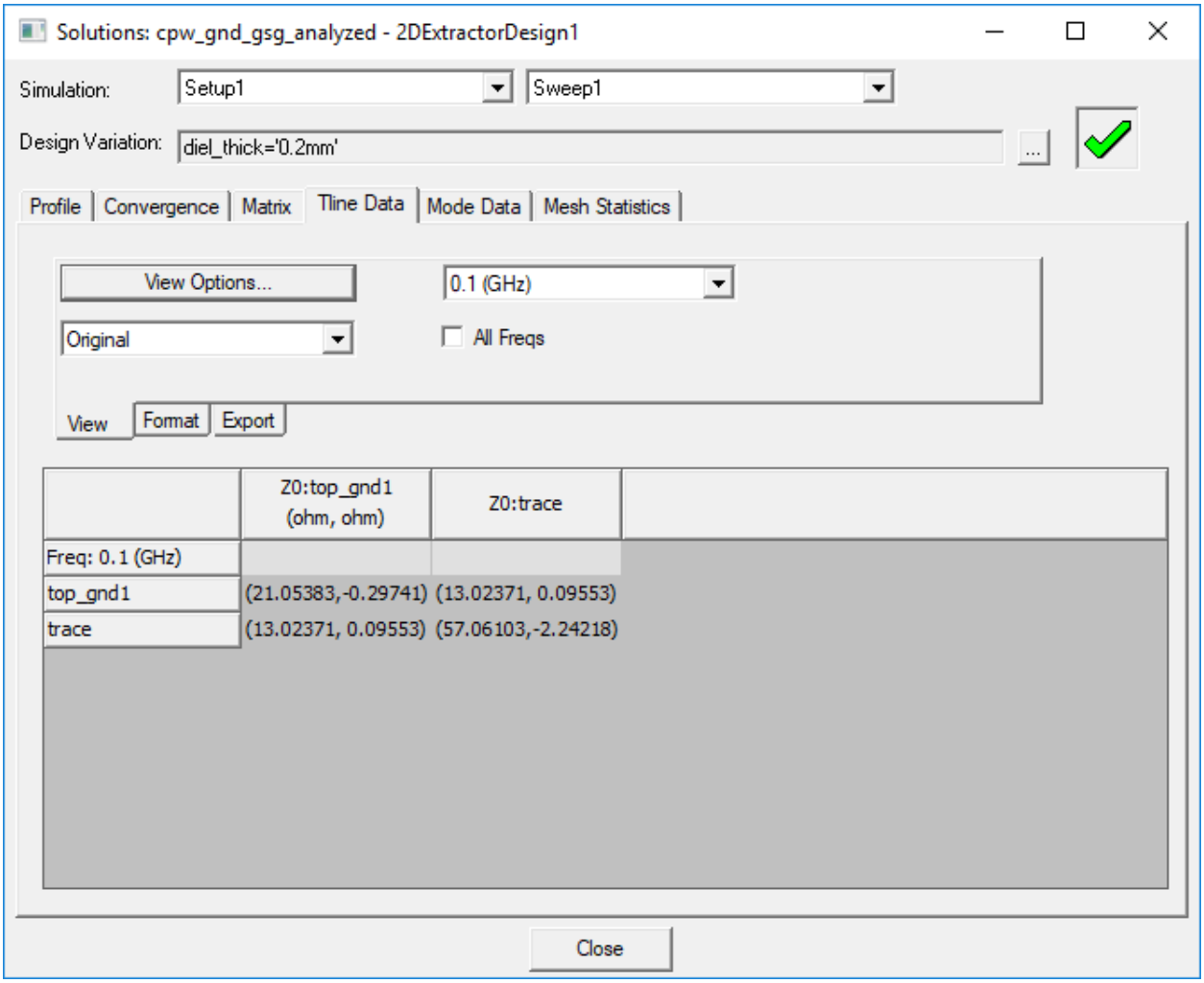
Select **Tline Data...**

- Or, from the ribbon, click **Results > Solution Data**.



Select the **Tline Data** tab.

The **Solutions** window appears, on the **Tline Data** tab.

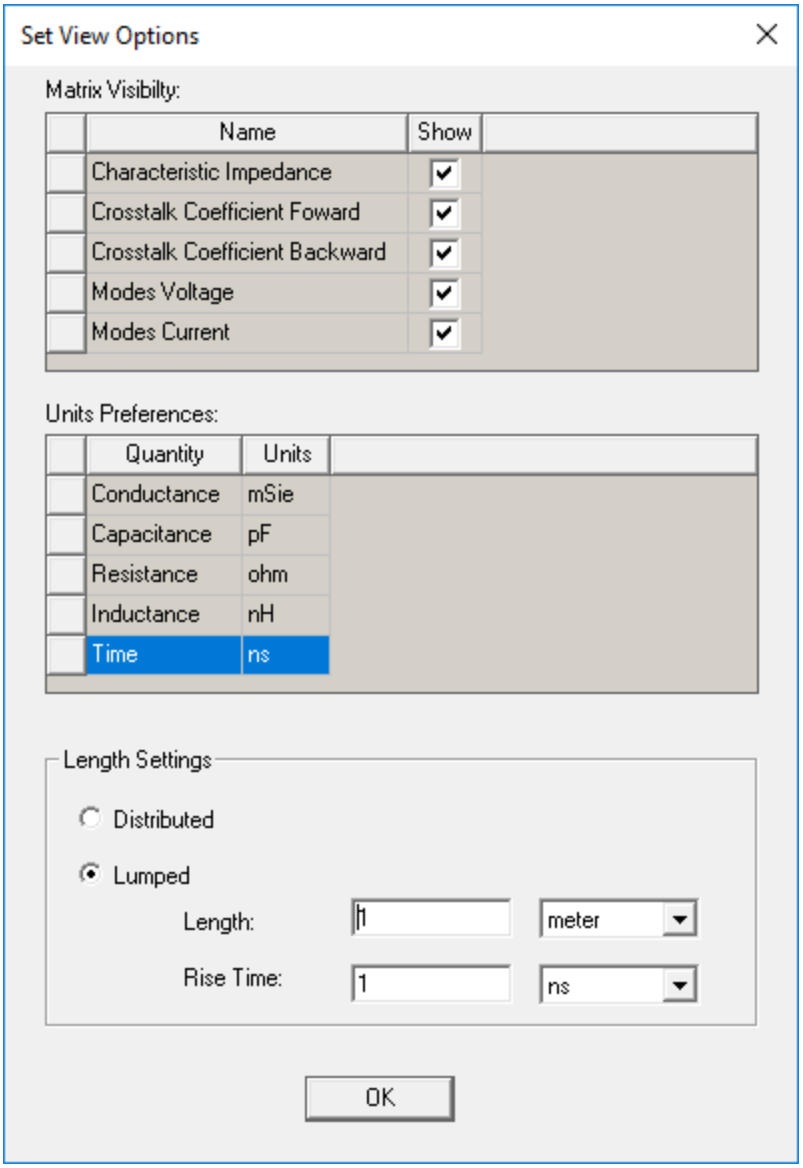


2. The **Tline Data** tab contains three sub-tabs: **View**, **Format**, and **Export**. The **View** tab displays by default.

   From the **View** tab:

   - Use the drop-down menu to choose from **Original** or **Reduced** matrices.

   - Use the drop-down menu to select a frequency, or select the **All Freqs** check box.
   - Click **View Options** to open the **Set View Options** window.

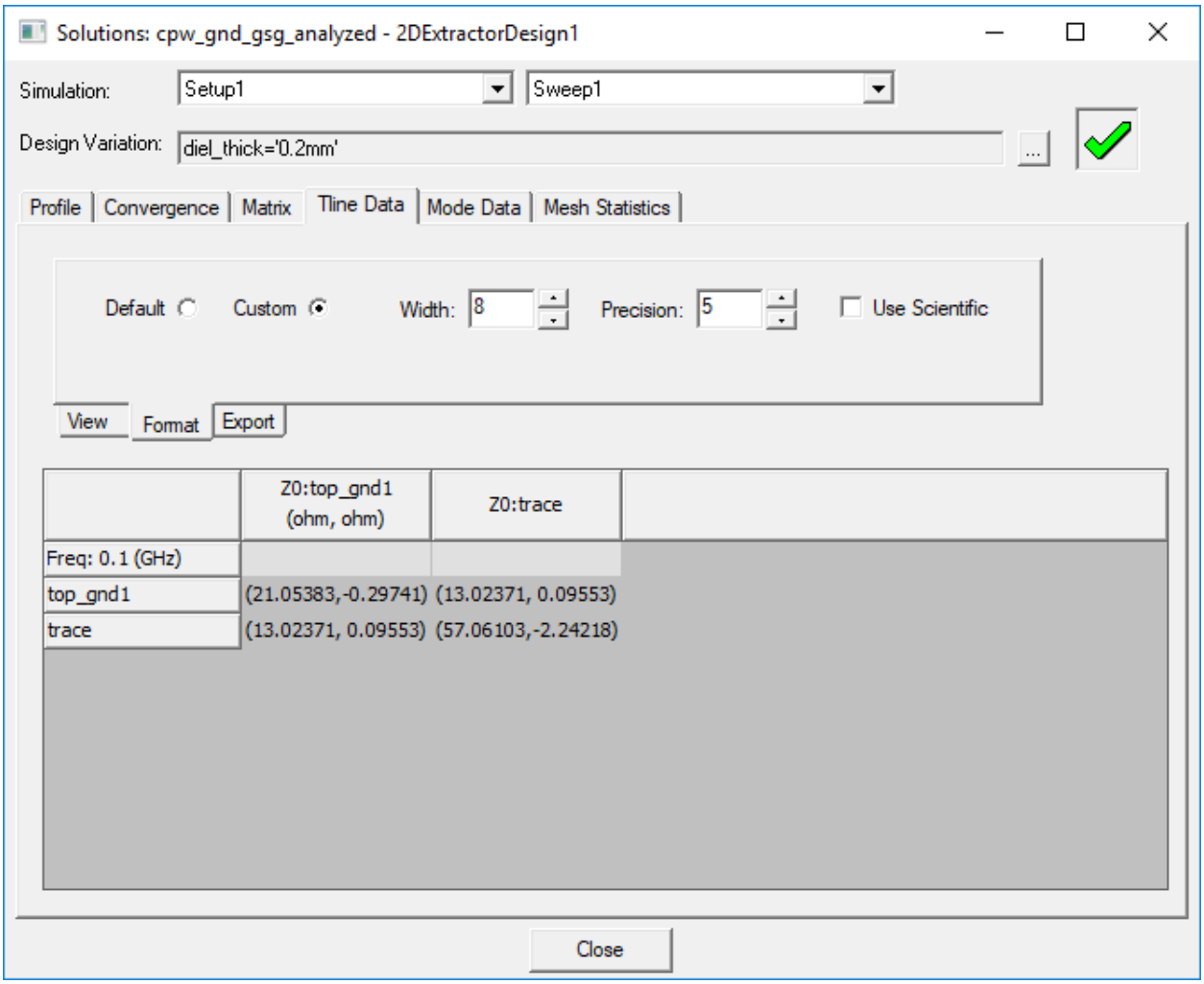


- From this window, you can set **Matrix Visibility** for characteristic impedance, crosstalk coefficients, and mode voltages and currents, as well as preferred units of measurement.
- Click **OK** to apply changes.

> **Note:**
>
> Selections persist during execution, but are reset to their defaults on exit.

3. Click the **Format** tab.

The **Format** tab displays, with additional options.

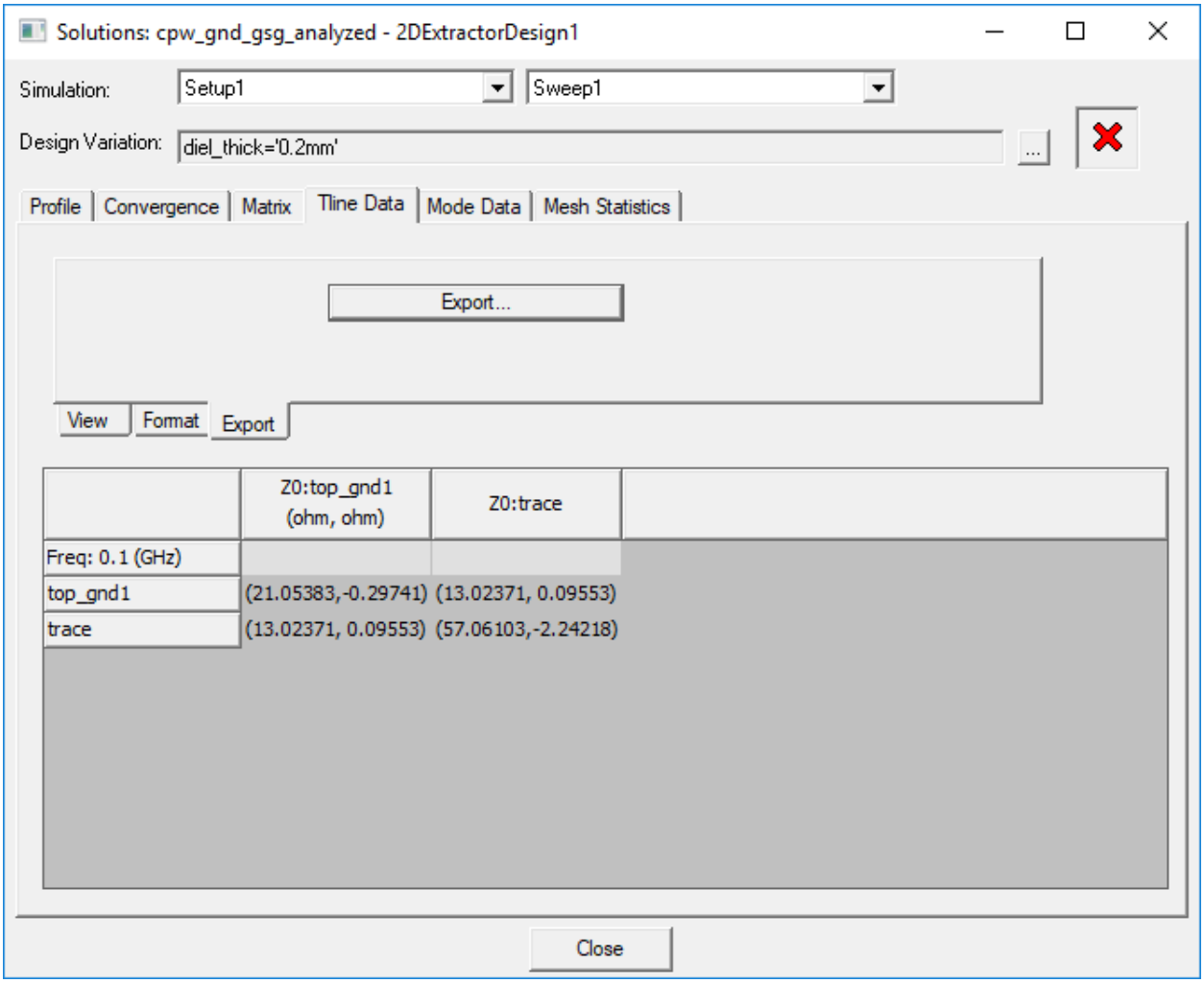


From the **Format** tab:

- Select either the **Default** or **Custom** display.
- If you select **Custom**, specify the matrix field **Width** and **Precision** in digits.
- Select the **Use Scientific** check box to display Tline data using scientific notation.

4. Click the **Export** tab.

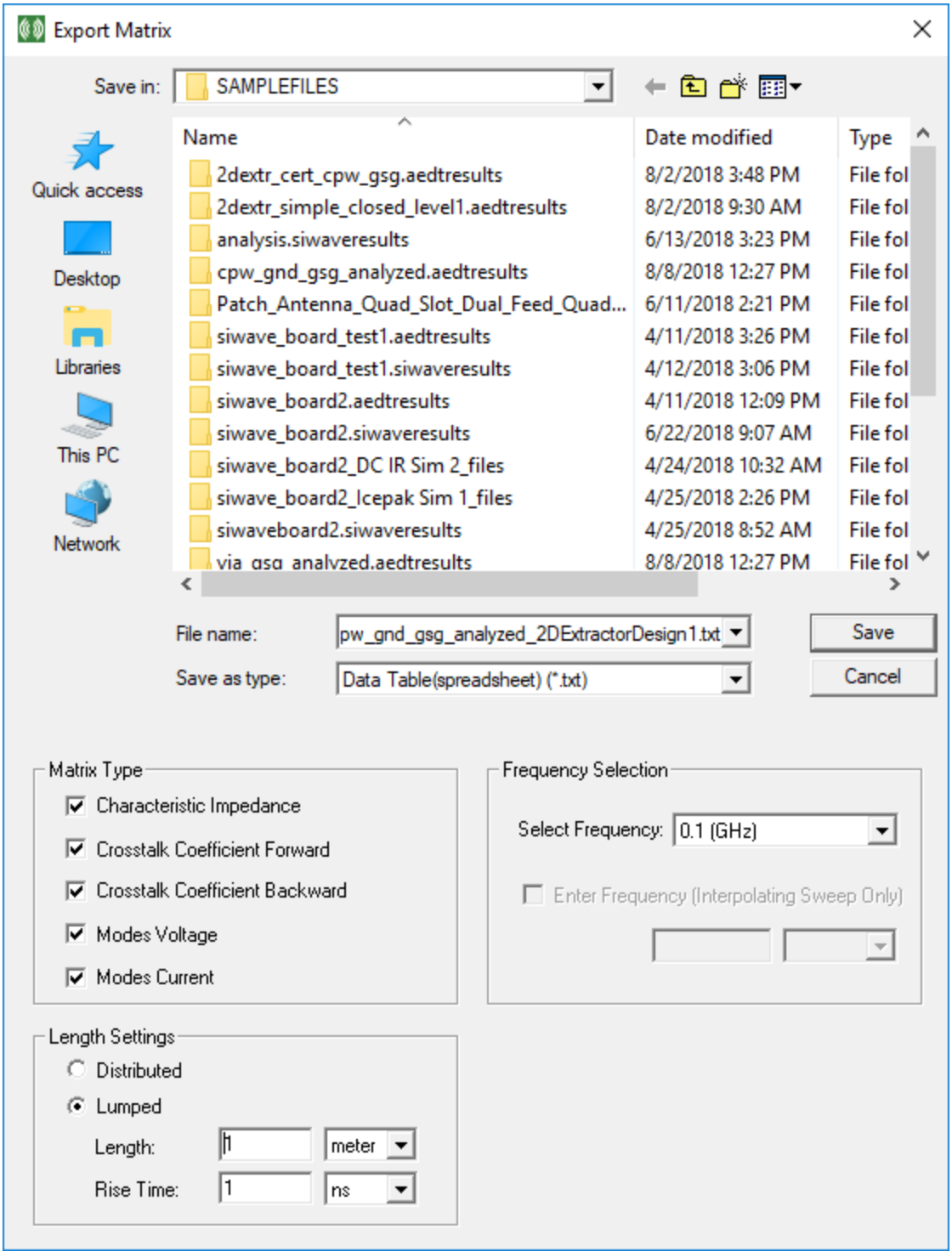
   The **Export** tab displays.



From the **Export** tab:

- To export Tline data, click **Export**.

  The **Export Matrix** window appears.



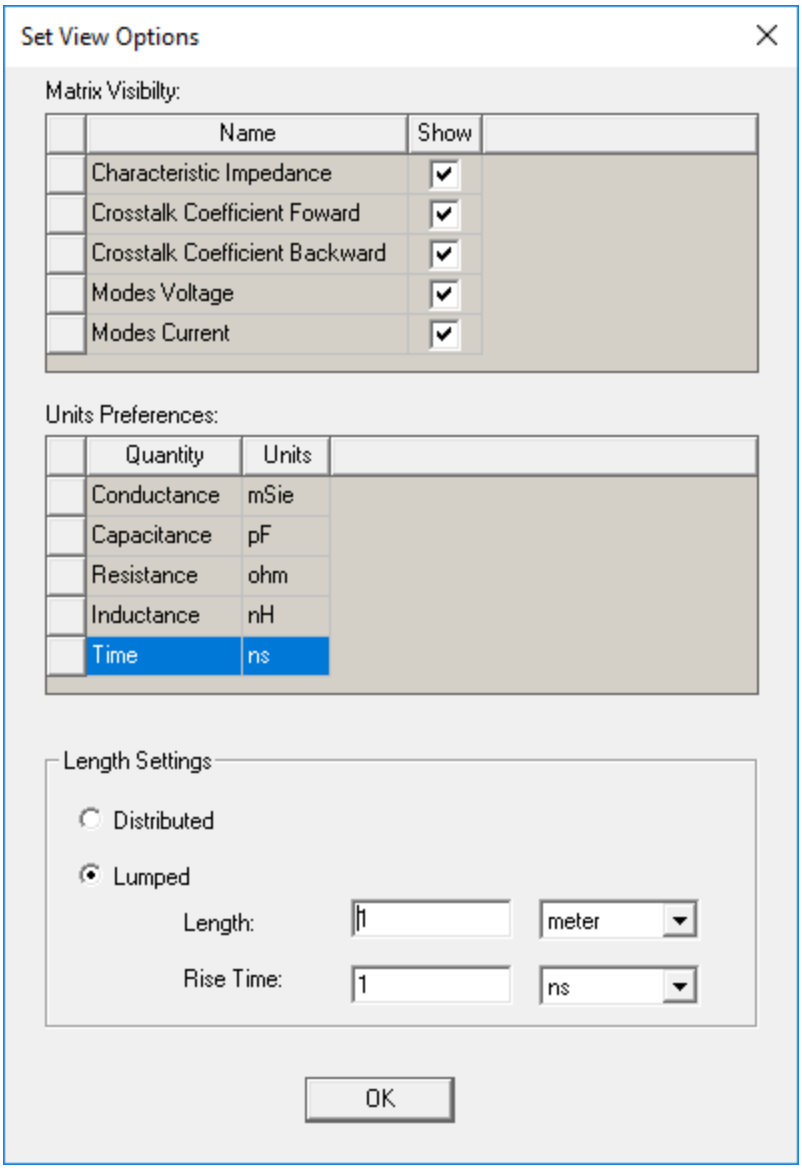
- Select your desired options and click **Save**.

5. Click **Close** to exit the **Solutions** window.

Characteristic Impedance

2D Extractor can display a characteristic impedance (Z_0) matrix for a transmission line. These matrices specify the relationship between voltages and currents on a multiple conductor transmission line.

From the **Set View Options** window, you can select **Characteristic Impedance**.



Some important notes:

- You must solve for at least capacitance and inductance.
- Modal transformation matrices are used to decouple the lines in the problem. The **Mode Data** tab displays resulting values for the decoupled lines.

For a decoupled line, the software defines characteristic impedance as the following:

$$\mathbf{Z_0} = \sqrt{\frac{\mathrm{R}+j\omega L}{\mathrm{G}+j\omega C}}$$

with values given in ohms. Thus, you must solve for both admittance and impedance.

Lossless Characteristic Impedance

For a decoupled lossless line, the software assumes that R ≪ jωL and G ≪ jωC (that is, losses are small), and defines characteristic impedance as $\mathbf{Z_0} = \sqrt{L/C}$ with values given in ohms. When you design a circuit using the model of a single transmission line, this is the appropriate value to use as a resistance termination for that line.

The decoupled Z_o values are used to compute the coupled characteristic impedance.

The matrix row and column headings are the conductor names.

For a set of coupled lines, the diagonal entries of the matrix (both approximately 91Ω here) are the appropriate termination values for both lines. The off-diagonal entries (both approximately 1.9 Ω here) give information about the crosstalk at the near end. The near end voltage on line *i* caused by a current in line *j* is represented by:

$$\mathbf{V_{0j}} = \mathbf{Z_{0ij}} \times /\mathbf{j}$$

Thus, if line *j* carries 1 mA as it switches, there would be (1.9 Ω)(1 mA) = 1.9 mV of crosstalk on line *i*.

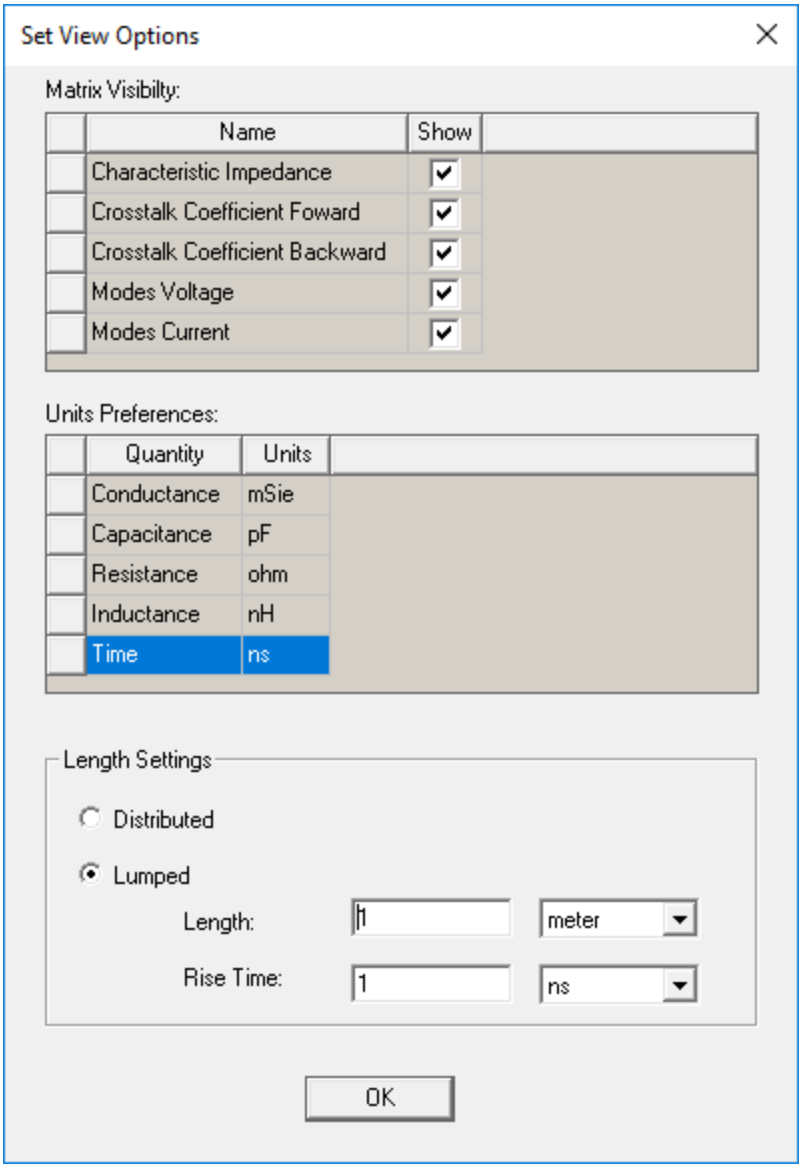
Lossy Characteristic Impedance

This displays the characteristic impedance for a lossy line. Characteristic impedance values are given as (R, X), where R and X are both in ohms. In this case, characteristic impedance is a complex number in the form Z0 =R + jX, where X is the reactance j(wL-1/(wC)) with w being 2p times the frequency (1kHz in the example) of the AC voltages during the solution.

Crosstalk Coefficients

2D Extractor can calculate crosstalk coefficients if you have solved for both the inductance and capacitance matrices.

From the **Set View Options** window, you can select **Crosstalk Coefficients Forward** and **Crosstalk Coefficients Backward**.



For Crosstalk Coefficients Forward, you can specify:

- **Distributed** – for crosstalk per unit length/rise time.
- **Lumped** – for the total crosstalk. This option requires you to enter the **Length** of the coupled transmission lines and the signal **Rise Time**. See: Editing the Length Settings.

The following describes these coefficients in greater detail.

Background

Crosstalk coefficients were originally introduced in a classic paper by D. B. Jarvis [IEEE Trans. on Electronic Computers, Oct 1963, p. 476]. Jarvis analyzed the case of two weakly coupled,

lossless, transmission lines having the same self-capacitance and inductance values. His paper gave formulas to estimate the crosstalk that would result when one of the lines is driven by a pulse with a finite rise time. Versions 16 and earlier of 2D Extractor implemented Jarvis' formulas in essentially their original form. In version 17 and later, 2D Extractor implements a new set of crosstalk formulas based on an extension of Jarvis' approach to the more general case of transmission lines with different self-inductance and capacitance values. These formulas provide higher accuracy when calculating crosstalk between lines with different LC parameters. The new formulas reduce to Jarvis' original equations when the inductance and capacitance values of the transmission lines are the same.

The formula for the backward crosstalk on line 2 due to a 1-Volt pulse on line 1:

$$K_B = \frac{l}{2(T_1+T_2)} \cdot \left(\frac{L_{21}}{\zeta_1} - \zeta_2 C_{21} \right) \cdot \min \left(1, \frac{T_1+T_2}{t_r} \right)$$

The formula for the forward crosstalk on line 2 due to a 1-Volt pulse on line 1:

$$K_F = -\frac{l}{2 \cdot \max(|T_1-T_2|, t_r)} \left(\frac{L_{21}}{\zeta_1} + \zeta_2 C_{21} \right)$$

It is important to note that C_{21} is the off-diagonal term of the per-unit-length Maxwell capacitance matrix. It is typically *negative* for transmission lines operated in single-ended mode (This might not be true for differential mode). Here *l* is the length of the lines, T_1 and T_2 are the time of flight delays, t_r is the input signal rise time, and ζ_1 and ζ_2 are the characteristic impedances.

The derivation of these formulas assumes the driven line is excited by an ideal voltage source at the near end and terminated at the far end in its characteristic impedance. The victim line is assumed to be terminated at both ends in its characteristic impedance. If different terminations or excitations are used, then the crosstalk coefficients may not correctly predict the actual crosstalk. Also note that the formulas assume weak coupling. If coupling is very strong then accuracy may start to degrade. The crosstalk coefficients should only be used as an estimate of the actual crosstalk; they are not a replacement for detailed transient simulation.

Pre-R17 Crosstalk Formulas

If the you set the environment variable ANSYS_R16_CROSSTALK to a non-zero value before invoking the software, then 2D Extractor will revert to using the older version of the crosstalk formulas. These are essentially the original formulas derived by Jarvis.

The pre-R17 formula for backward crosstalk was:

$$K_b = \frac{1}{4} \left\{ \frac{|{C_{ij}}^m|}{\sqrt{{C_{ii}}^m {C_{jj}}^m}} + \frac{|L_{ij}|}{\sqrt{L_{ii} L_{jj}}} \right\}$$

The pre-R17 formula for forward crosstalk coefficient K_f was:

$$K_f = \frac{1}{2}\left\{\frac{|C_{ij}{}^m|}{\sqrt{C_{ii}{}^m C_{jj}{}^m}} - \frac{|L_{ij}|}{\sqrt{L_{ii}L_{jj}}}\right\} \times \sqrt{\left\{\sqrt{L_{ii}C_{ii}{}^m}\sqrt{L_{jj}C_{jj}{}^m}\right\}}$$

The value of K_f has units of seconds per meter, and is displayed in this form when the **Distributed** option is selected in the **Set View Options** dialog box.

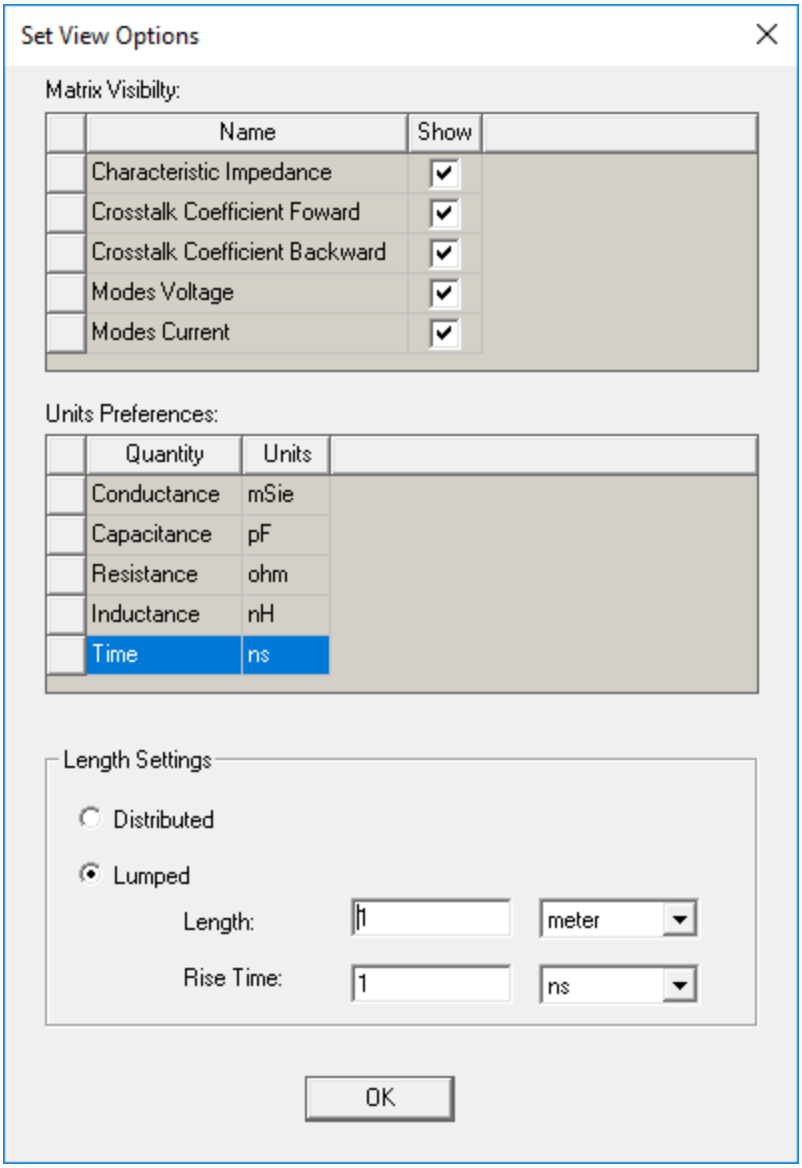
The actual forward crosstalk is strongly dependent on the length of the transmission line and the rise time.

To obtain the actual crosstalk the "distributed" K_f value must be multiplied by the line length and divided by the signal rise time. If **Lumped** crosstalk is selected, then the software performs this multiplication for you and displays a dimensionless number representing the total far end crosstalk.

Mode Voltages and Currents

2D Extractor can display modal transformation matrices.

From the **Set View Options** window, you can select **Modes Voltage** and **Modes Current**.



Modes Voltage converts voltage signals to voltage modes. Select this option to display the transformation matrix for converting a system of *n* individual voltage signals to a system of *n* voltage modes.

Modes Current converts current signals to current modes. Select this option to display the transformation matrix for converting a system of *n* individual currents to a system of *n* current modes.

Viewing Optimetrics Analysis Results in 2D Extractor

To view data specific to an Optimetrics solution:

- From the **Project Manager**, expand **Optimetrics**.
- Right-click the Optimetrics setup and select **View Analysis Result**.

  The **Post Analysis Display** dialog box appears, on the **Result** tab.

Plot View:

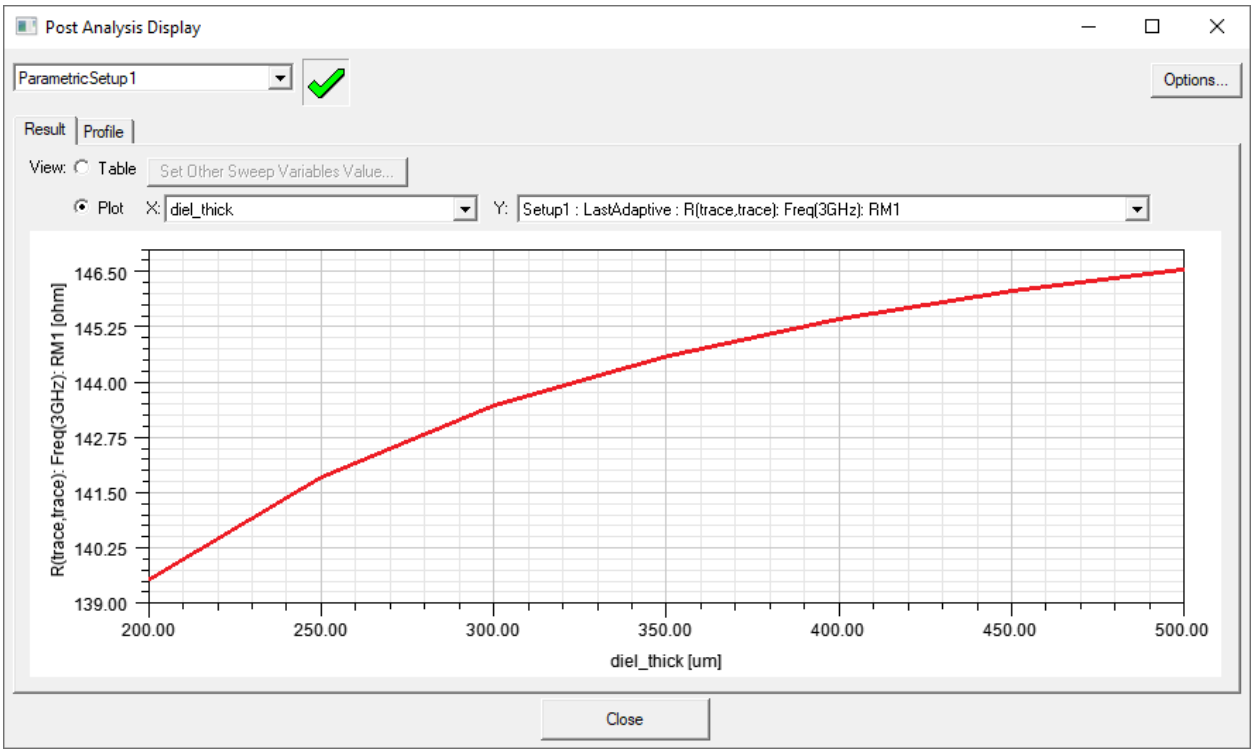
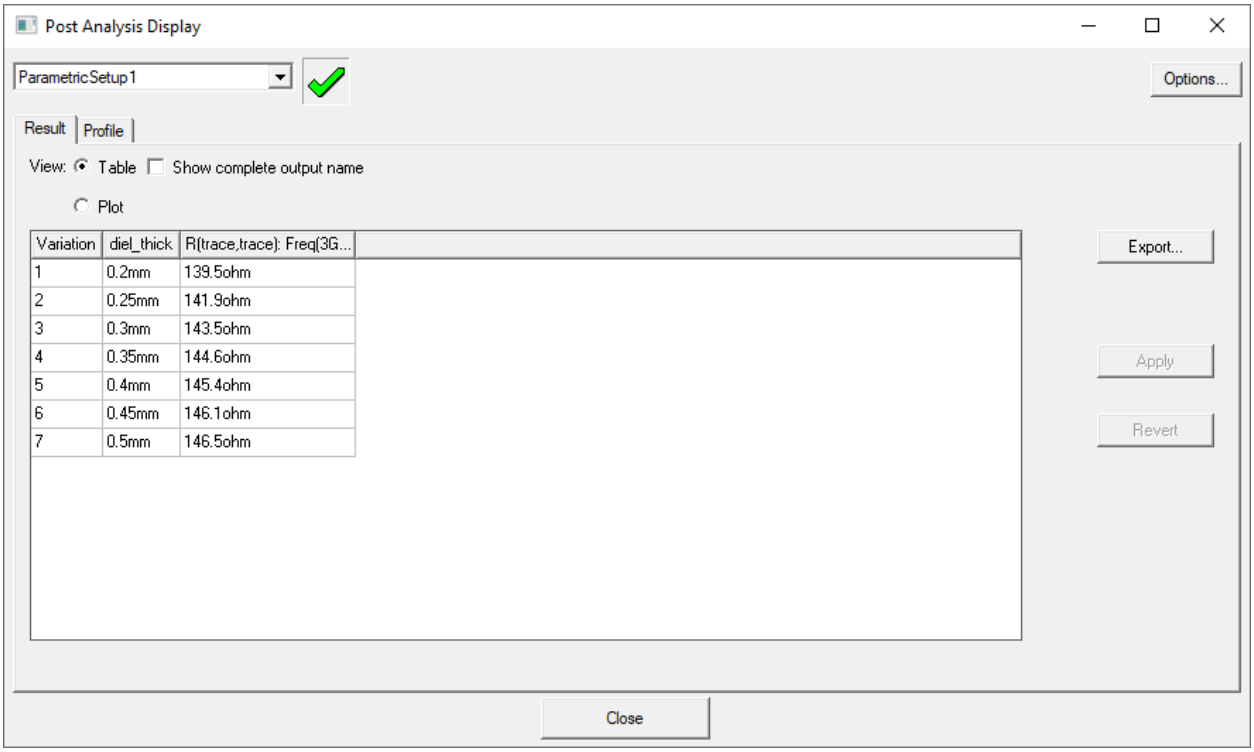


Table View:



- Use the drop-down menu at the top of the window to select the Optimetrics setup for which you want to view results.
- Under **View**, select either **Table** or **Plot** to change how the results are displayed.
  - In **Table** view, select a design variation from the table and click **Apply** to apply that variation.
  - In **Table** view, select **Show Complete Output Name** to view full solution names.
  - In **Plot** view, use the **X** and **Y** drop-down menus to select which data is plotted.
- Click **Options** to select the number of significant digits to display.
- Use the **Profile** tab to view the Optimetrics solution profile.

Some Optimetrics analysis types have additional options in this window:

- **Log Scale** – for applicable analysis types, select this check box to select logarithmic scaling.
- **Num. of Bins on X Axis** – for statistical analysis, type the number of bins you want to plot along the X axis.

Note:

For Design Xplorer Analysis:

- Plot view is not available.

For Sensitivity Analysis:

- 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.

For Parametric Analysis:

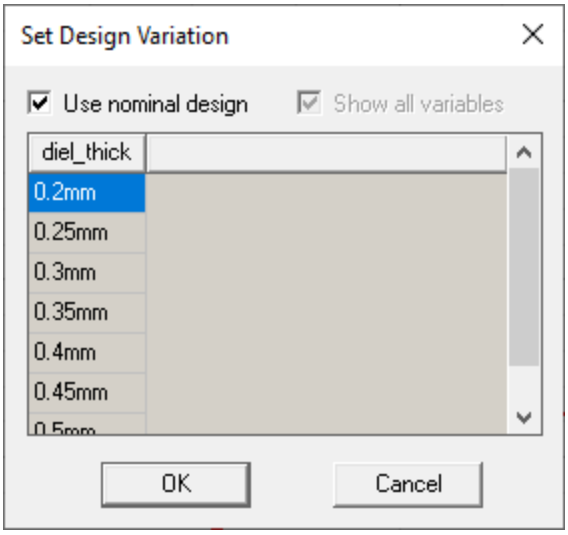
- Only one sweep variable at a time can be plotted against solution quantity results. Any other variables that were swept during the parametric analysis remain constant.

Viewing Solution Data for an Optimetrics Design Variation in 2D Extractor

To view the convergence information, computing resources, or matrices computed for any design variation solved during an Optimetrics analysis, you must first select the design variation in the **Set Design Variation** window.

To access the **Set Design Variation** window:

- From the **Project Manager**, right-click **Results** and select **Apply Solved Variation**.
- Click **2D Extractor** > **Results** > **Apply Solved Variation**.
- Click **2D Extractor** > **Results** > **Solution Data** to open the **Solutions** window. Then click the ellipsis (**...**) button next to **Design Variation**.



By default, the **Use Nominal Design** check box is selected. Select any variation from the list, and the check box clears.

Click **OK** to view solution data for the selected variation.

Viewing an Optimetrics Solution Profile in 2D Extractor

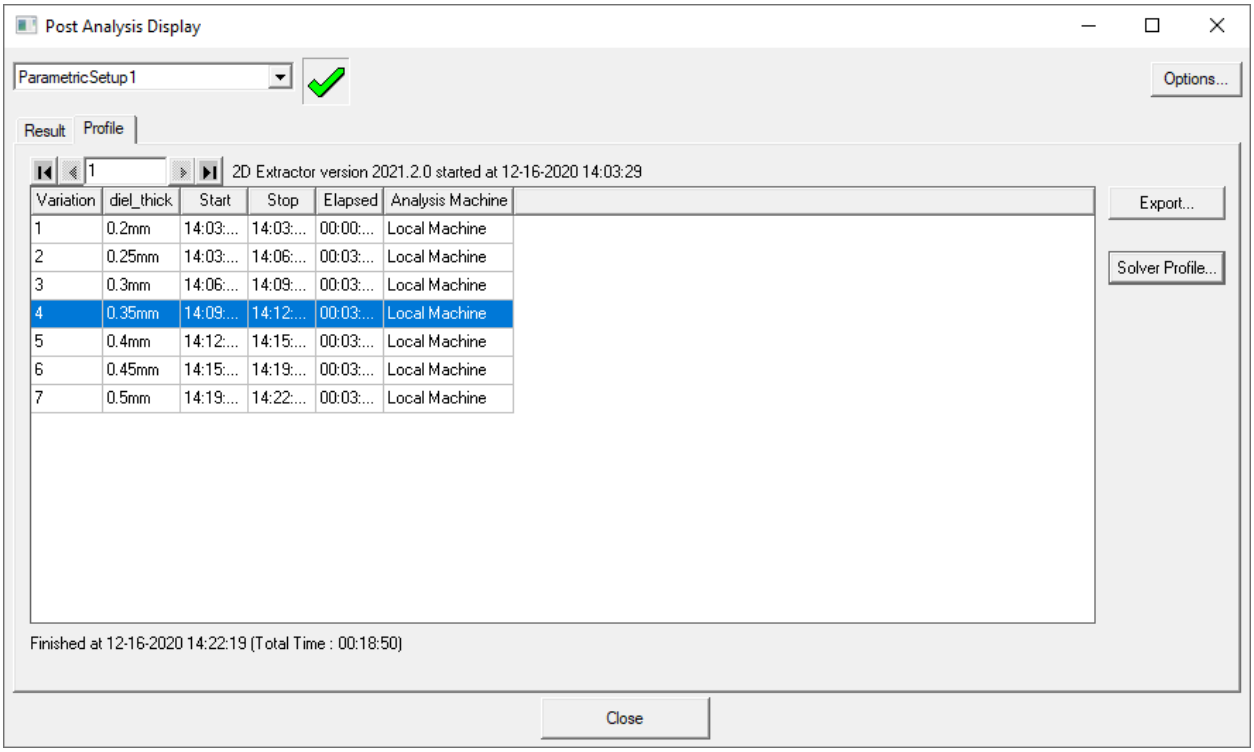
At any time during or after the solution process, you can examine the computing resources or profile data used for an optimetrics analysis. The profile data is essentially a log of the tasks performed by during the solution. The log indicates the length of time each task took and how much physical/disk memory was required.

To view profile data:

1. From the **Project Manager**, right-click an **Optimetrics** solution setup and select **View Analysis Result**.

   The **Post Analysis Display** dialog box appears.

2. Select the **Profile** tab.



3. Use the drop-down menu at the top of the window to select the Optimetrics setup of interest.
4. In the table, select a design variation.
5. Click **Solver Profile**.

   The **Solutions** dialog box appears, showing the profile data.

Editing Length Settings in 2D Extractor

To edit the length settings for the post-processing options:

1. Click **2D Extractor** > **Edit Length Settings**.

   The **Edit Length Settings** dialog box appears.

2. Specify the **Units**. All solutions in 2D Extractor are per unit length. For example, capacitance extracted by the solver is actually in "<SI units of C>/<SI units of Length>". You can choose the distributed units and view the solutions in the selected length unit.

3. Specify the options for **Lumped Length**:
    - Enter a value in the **Length** text box and select the units.
    - Enter a value in the **Rise Time** text box and select the units. This value is used when exporting the equivalent circuit and calculating lumped crosstalk coefficient forward.
4. If you want to save these settings as defaults and use them for other projects, select the **Save As Default** check box.
5. Click **OK**.

Editing Sources in 2D Extractor

Change the magnitude of voltage and current sources in the **Edit Sources** dialog box.

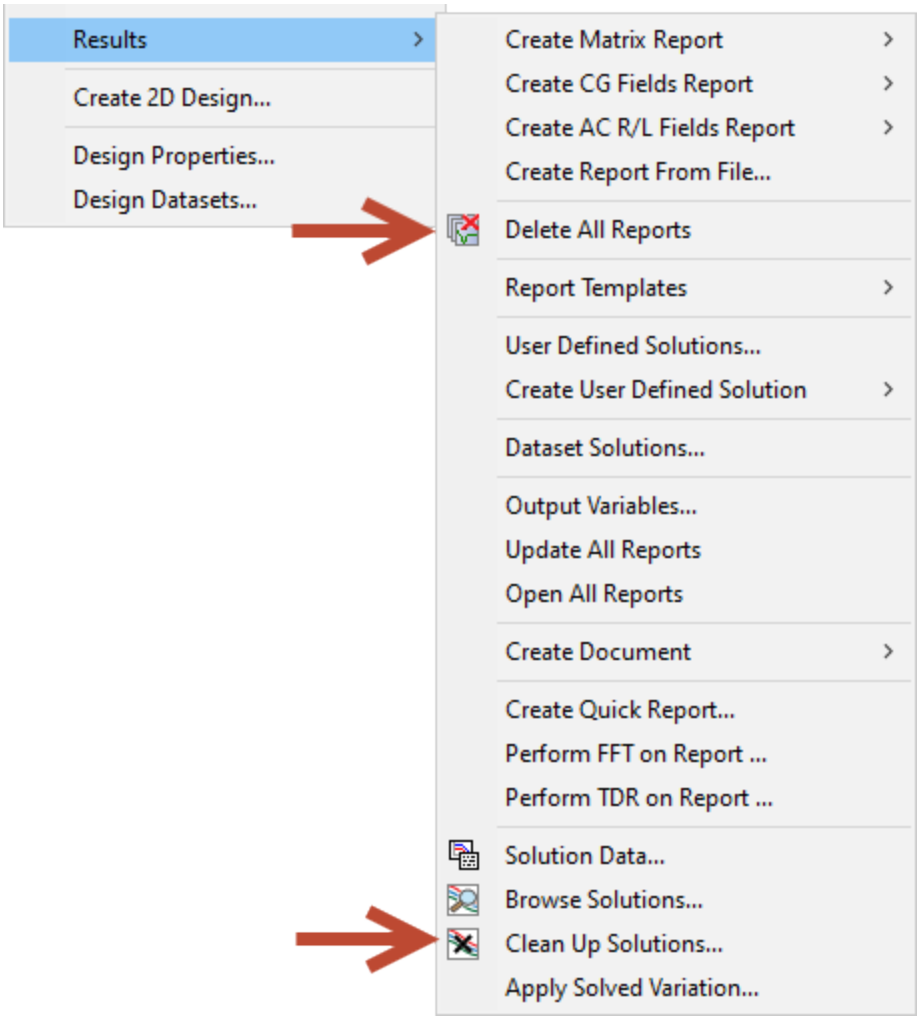
1. Click **2D Extractor** > **Fields** > **Edit Sources**.

   The **Edit Sources** dialog box appears.

2. Select the tab whose magnitude you want to edit - CG or RL. AC RL voltage and current sources are assumed to be peak quantities.
3. Click **OK**.

Deleting Solution Data

From the **Q3D Extractor** or **2D Extractor** menu, you can use **Clean Up Solutions** to selectively make deletions, or **Delete All Reports** to remove all solutions from the results.

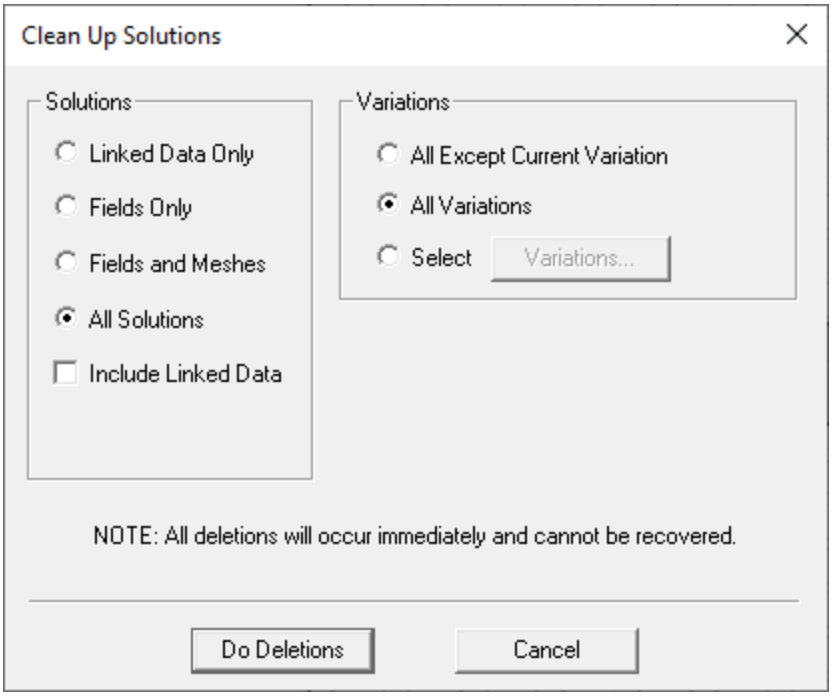


Delete All Reports works instantly, with no additional input required.

To clean up solutions:

1. Click [**Q3D Extractor** or **2D Extractor**] > **Results** > **Clean Up Solutions**.

   The **Clean Up Solutions** window appears.



2. Under **Solutions**, select the data you want to delete:
    - **Linked Data Only** – only linked data.
    - **Fields Only** – only fields data.
    - **Fields and Meshes** – only fields and mesh data.
    - **All Solutions** – deletes all mesh, matrix, and fields data for all adaptive passes and frequency sweeps.
    - **Include Linked Data** – use the check box to select whether to include linked data.
3. Under **Variations**, select the variation(s) for which you want to delete data:
    - **All Except Current Variation** – all solution data that does not correspond to the current project and design variable values for the current design.
    - **All Variations** – all solution data for the current design.
    - **Select** – click Variations to select specific variations.
4. Click **Do Deletions**.

The solution data is deleted.

Any post-processing reports or field overlays that included data you deleted will be invalidated until new solution data is generated.

Deleting Reports

You can delete one or all reports.

> **Warning:**
>
> Solution data that has been deleted cannot be recovered!

To delete a selected report:

- From the **Project Manager**, right-click a report icon and select **Delete**.

To delete all reports:

- Click [**Q3D Extractor** or **2D Extractor**] > **Results** > **Delete All Reports**

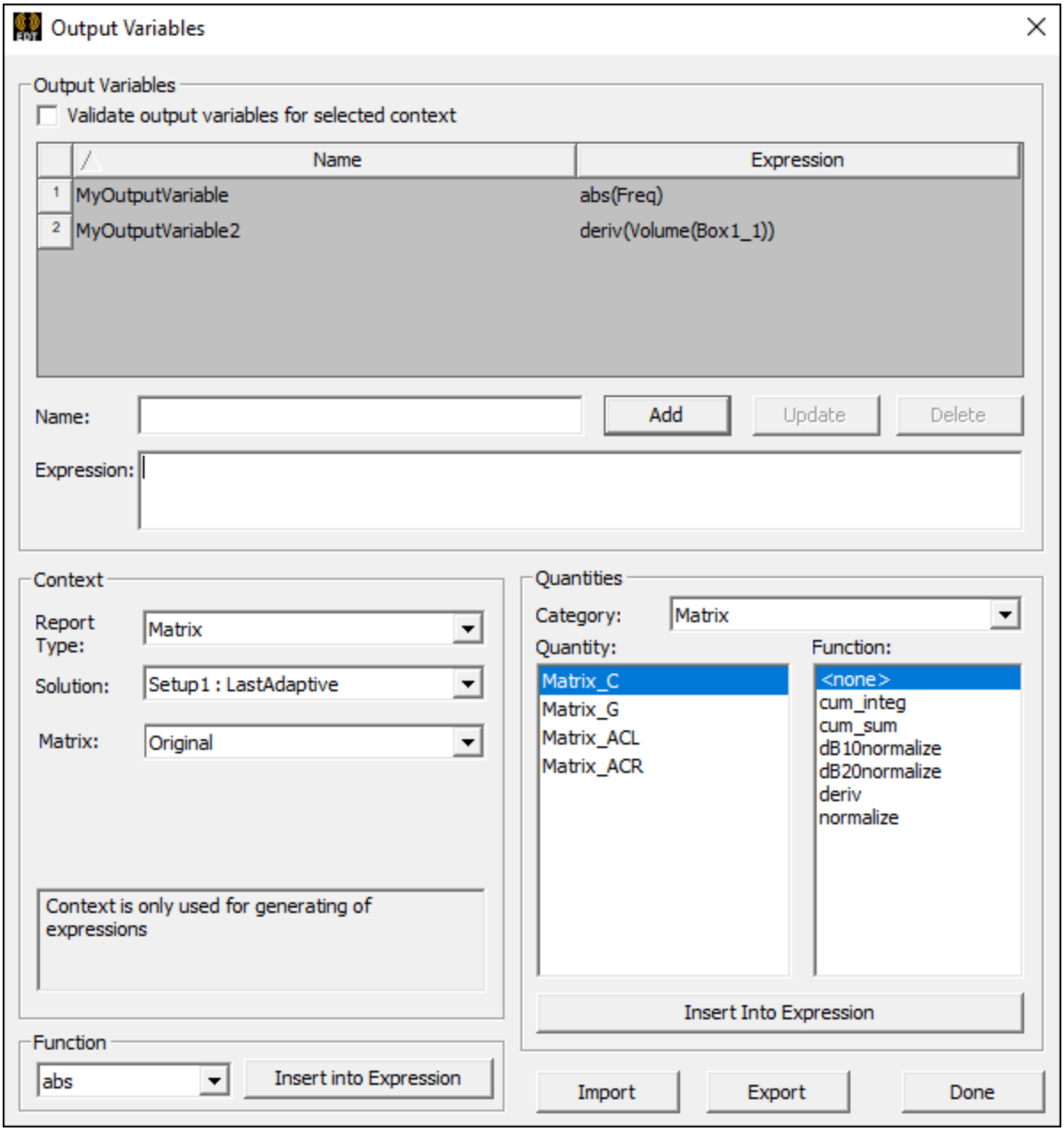
  In the **Project Manager**, all items under the **Results** folder are removed.

Specifying Output Variables

You can define output variables to use calculated expressions of results as adaptive convergence goals and for reports. You can use output variables in the Expression Cache tab of the **Solution Setup** dialog box and select them as Categories in the **Reports** dialog box, as well as the **Output Variables** window.

Access the **Output Variables** window one of several ways:

- Click **Q3D Extractor** > **Results** > **Output Variables**.
- In the Project Manager, right-click **Results** and select **Output Variables** from the shortcut menu.
- In the **Solution Setup** dialog box, select the **Expression Cache** tab, click **Add...** to display the **Add to expression cache** dialog box, and click the **Output Variables** button.
- Click the **Output Variables** button in the **Report** dialog box.



The **Output Variables** window contains four sections:

- **Output Variables** – contains a list of existing output variables, and allows you to specify the name and expression for a new output variable. The **Validate output variable...** check box allows you to validate an expression before adding it. Valid expressions appear in blue and invalid expressions appear in red.
- **Context** – specifies the Report Type and Solution. Depending on the type of report selected, additional fields may appear. All selections affect the functions and quantities

listed.

> **Note:**
>
> The context is used only for generating expressions.

- **Quantities** – allows you to insert quantities into the Expression area of the **Output Variables** section.
- **Function** – allows you to insert completed expressions into the Expression area of the **Output Variables** section.

To add an output variable:

1. Enter a **Name** for the variable.
2. Specify the **Context**. This impacts the available quantities and functions.
3. Specify the **Quantities** (required) and **Functions** (optional) using the **Insert Into Expression** buttons.
4. Review the **Expression**. When it is ready, click **Add**.

   The new output variable appears in the **Output Variables** list.

To delete an output variable:

1. Remove all references to the output variable from the project.
2. Save the project to erase the command history.
3. Open the **Output Variables** window using one of the methods above.
4. Select the variable from the **Output Variables** list and click **Delete**.

Note:

The evaluated value of an expression is always interpreted in SI units. However, when an angle quantity is plotted in a report, you have the option to plot values in units other than SI. For example, if you want to plot the polar angle of a complex simulation result S_{11}, you can choose between ang_deg(S_{11}) and ang_rad(S_{11}). Both of these return the exact same angle quantity, but in degree and radian units respectively.

When using non-SI units in expressions, surprising outcomes might result. For example, the expression 1+ang_deg(S_{11}) represents an angle and the number 1 is treated as 1 rad. The angle SI unit is attached to any unitless number that is added/subtracted from an angle value. If you want to treat 1 as degrees, make it explicit and use 1deg + ang_deg(S_{11}) instead.

If you are interested in unitless degree values, two additional functions exist: ang_deg_val(S_{11}) and cang_deg_val(S_{11}). These return simple numbers and are treated as such by any expression. For example, if the complex S_{11} lies on the positive Y axis, ang_deg_val(S_{11}) would be 90 and 1 + ang_deg_val(S_{11}) would be 91.

Function List for Output Variables

The **Output Variables** window 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: min, max, integ, avg, rms, pk2pk, cang_deg and cang_rad.

You can select from the functions in the **Output Variables** dialog box's **Function** list or type them directly into the Expression field, if necessary.

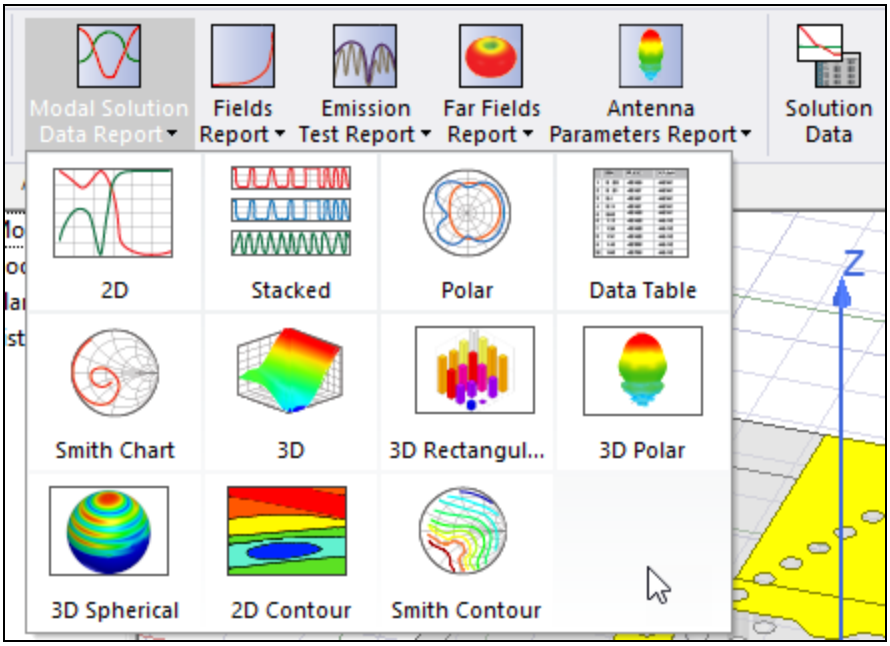
Creating Reports

After a solver generates a solution, you can analyze all the results for that solution. Ansys Electronics Desktop lets you create 2D or 3D plots. A 2D or 3D plot shows the relationship between a design's values and the corresponding results of the analysis. You can create reports using either the **Create Quick Report** option 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>** (for instance, Modal, Terminal, Transient, Eigenmode, Characteristic mode, Fields, Far Fields, Emission test, Partial Discharge, Multipaction, and so forth), 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 Eigenmode solution type, the **Results** menu contains templates for Eigenmode Parameters and for Fields. These appear on the menus as **Create Eigenmode Parameters Report** and **Create Fields Report**. For the Modal and Terminal Solution types, several different types appear, appropriate to each solution type. 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.

The **Results** tab for the Ribbon will show icons and drop-down menus for available report types for the active project.

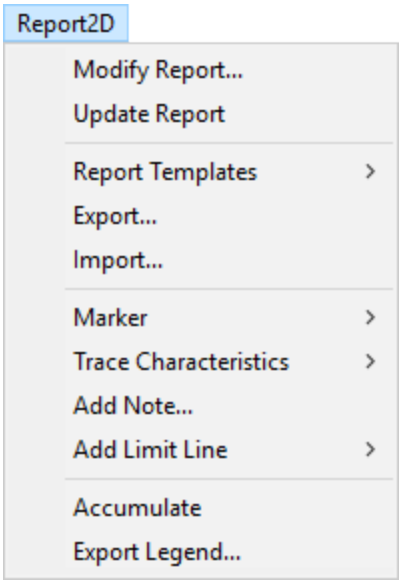


If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **Q3D Extractor** > **Results** > **Report Templates** > <***templateName***>. You can also access previously defined 2D templates 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** or **Report 3D** >**Export** feature, select the *Ansoft ReportData Files (.rdat)* format, and **Save** the file, which you can later select to Create Report from File.

Importing Plot Data

When a report is open in Ansys Electronics Desktop, the **Report2D** menu appears.



> **Note:**
>
> You *must* have a report open in order to import plot data.

The **Report2D**> **Import Data** command lets you import plot data from comma delimited files (*.csv), tab delimited files (*.tab), Ansoft PlotData files (*.dat), post-processor files (*.txt), or Ansoft ReportData files (*.rdat).

1. From the top menu bar, click **Report2D** > **Import**.

   The **Open** window appears.

2. Browse for and select the desired file.
3. Click **Open** to import the file into the current report.

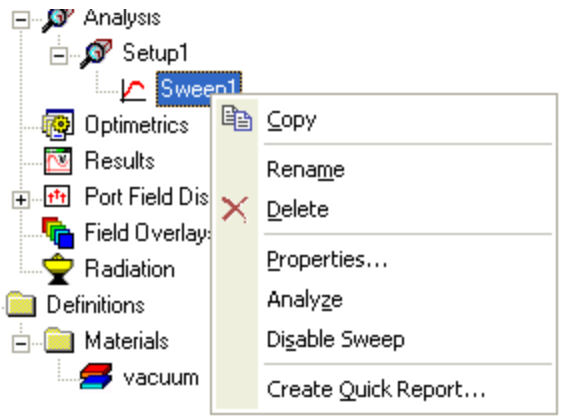
   The imported traces appear in the Project tree under the current report.

   > **Note:**
   >
   > For a report trace where the primary sweep is NOT the same as the x-component, an export report then import report may not produce the same curve. In this circumstance, only *.rdat import the same curve/trace. For other formats, the import produces two separated traces.

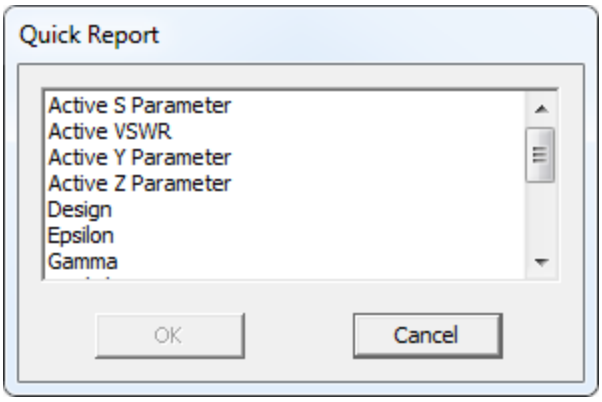
Creating a Quick Report

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.



The list of available reports differs depending on the Solution type. The figure shows reports for a Modal solution. Eigen mode solutions and Terminal solutions provide different selections.

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 derives from the report type specified in the **Quick Report** dialog box.

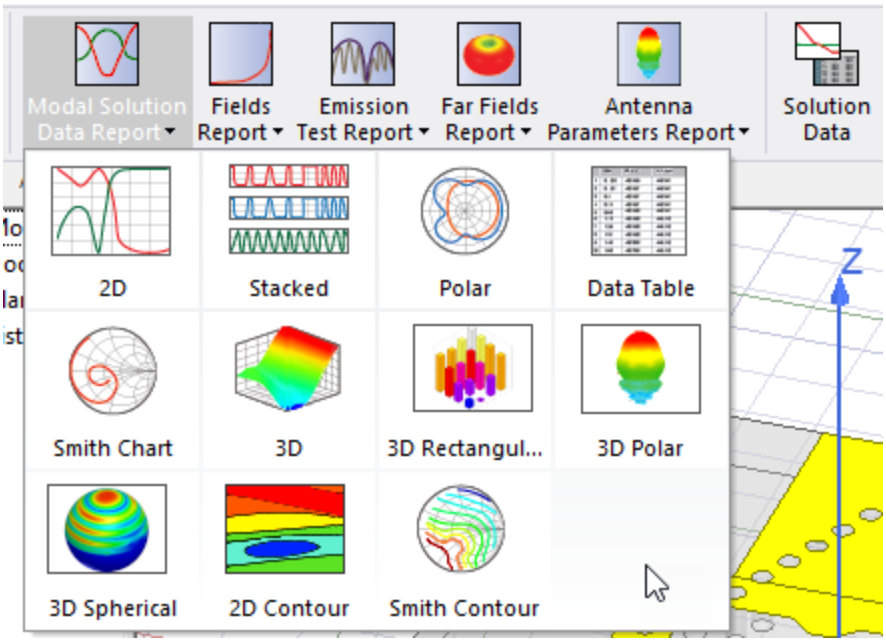
Creating a New Report

Following is the general procedure for creating a new report:

1. On the **Q3D Extractor** menu or the Project Manager, point to **Results**, select **Create *<type>* Report**, and select the **Display Type** template. There are more **Report Type**

templates available for terminal solutions (terminal, modal, fields, near fields, and far fields) and for modal solutions (modal and fields). For Eigenmode solutions, the **Report Type** ***<templates>*** are for Eigenmode Parameters and for Fields. Characteristic Modes solution reports include quantities for Significance, Value, and Angle.

The **Results** tab for the Ribbon will show icons and drop-down menus for available report types for the active project.



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 **Q3D Extractor** > **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 report and display type from the **Results** menu, the **Report** dialog box appears, with the **Trace** tab selected by default.

2. In the Context section, you make selections depending on the design and solution type.
3. In the **Y** Component section of the dialog box, make selections for the following:
    a. Categories – those depend on the Solution type and the design. For example, Eigenmode quantities include Eigenmodes, variables, output variables, and the design. Driven solutions include such categories as S parameters. Report categories for Transient designs include Spectral and Transient. For a Transient Network design with differential pairs defined, the Reporter interface allows selection of single-ended or differential signals just as for driven terminal. Characteristic Mode Data Reports include Characteristic Mode quantities for

Significance, Value, and Angle. Report categories for SBR+ designs that include an Incident Plane wave and an RCS selection as Monostatic do not require a geometry selection and include a range of Monostatic Quantities when select Monostatic RCS as the Report Category. For SBR+ designs and RCS Monostatic you can choose between Freq, IWaveTheta and IWavePhi variables for specifying sweeps. The selected Category provides the default plot name. 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.

> **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. See Report Setup Options.

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.

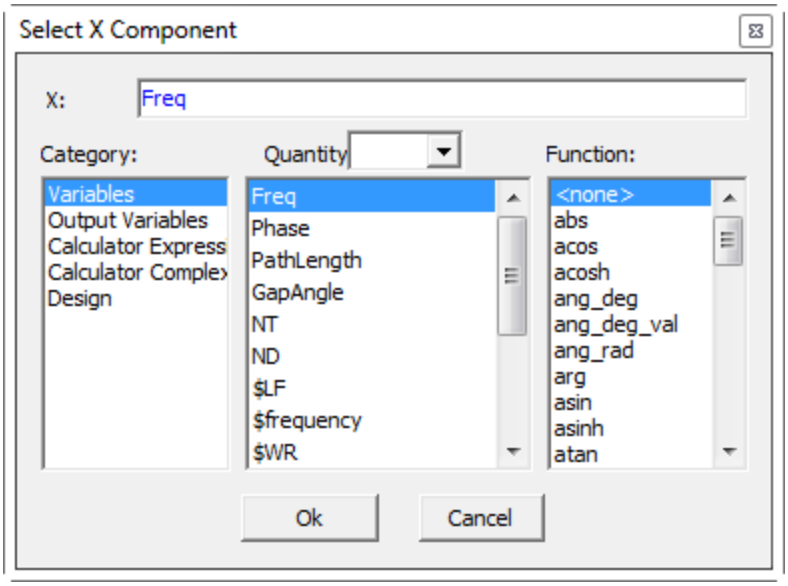
e. **Range Function** button – opens the **Set Range Function** dialog box. This applies to the currently specified Quantity and Function.

f. To use a dataset in this report, set the **Y** component to a Piecewise Linear Function `pwl(dataset_expression, primary_sweep)` where is the `primary_sweep` is what you set in the next step.

4. In the **X (Primary Sweep)** section, make selections for the following:

   a. Select the Primary value(s) from the drop-down menu.

   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 **Browse** [...] to display the **Select X Component** dialog.



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 box to assign the X component.

b. If sweeps are available, you can select **Browse** [...] to display a panel that lets you select Use all values, or selected sweep or sweeps, or access an Edit Sweep dialog box with further editing options. Post-Processing variables are Post-Processing sweeps/editable sweeps, so you can use the **Edit Sweep** dialog box to create your own sweep.

c. 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 Manager.

   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, S Parameter 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. For example if re(S_{11})*re(S_{22}) is requested over multiple widths, each of the S_{11} and S_{22} are stored when you save. If you do not do a save of a changed plot, the changed version is not stored.

Note:

Remember that the evaluated value of an expression is always interpreted in SI units. However, when an angle quantity is plotted in a report, you have the option to plot values in units other than SI. If you want to plot the polar angle of a complex simulation result, S11 say, you can choose between ang_deg(S_{11}) and ang_rad(S_{11}). Both of these return the exact same angle quantity but in degree and radian units respectively.

Note that when used in expressions, some surprising outcomes might result. For example, the expression "1+ang_deg(S11)" represents an 'angle' and the number "1" is treated as "1 rad". The angle SI unit is attached to any unitless number that is added/subtracted from an angle value. If you want to treat "1" as degrees, make it explicit and use "1deg + ang_deg(S_{11})" instead.

If you are interested in unitless degree values, two additional functions exist: ang_deg_val(S_{11}) and cang_deg_val(S_{11}). These return simple numbers and are treated as such by any expression. If the complex S_{11} lies on the positive Y axis say, ang_deg_val (S_{11}) would be 90 and "1 + ang_deg_val(S_{11})" will be 91.

Context Section for Reports

In the **Context** section, make selections from the following field or fields, depending on the design and solution type.

1. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
2. Domain field with a drop down selection list. Whether this field appears, and domains are listed, depends on the Solution type and the report ***<type>*** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

   Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   For Near Field or Far Field report, for a Rectangular Contour Plot, the Domain can be Theta, Phi, or Sine Space. Before you can create a Sine Space plot, you must create the appropriate Radiation Setups. If you manually enter Phi or Theta component of a far field quantity when you using Az/El or El/Az far field infinite sphere definition, then the Report uses the Z axis of the Az/El or El/Az coordinate system definition as pole to calculate theta/phi component for rE.

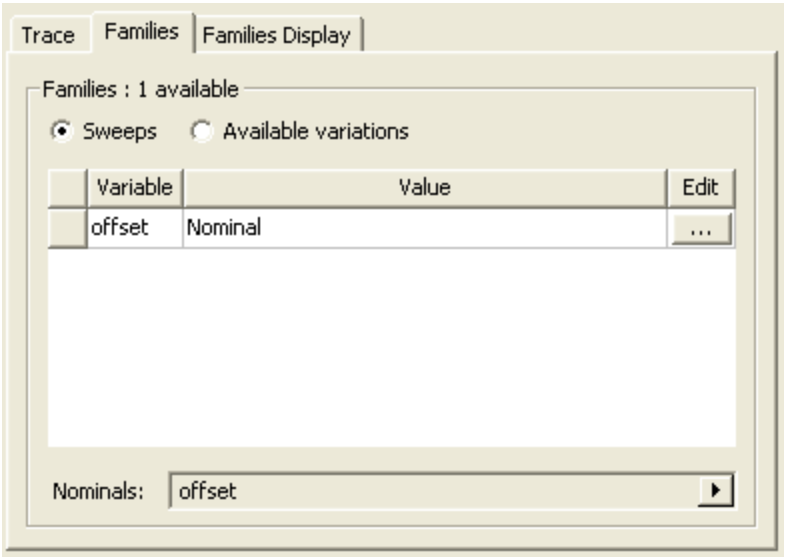
3. Geometry field with a drop-down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup. For SBR+ designs with an

Incident Plane Wave and Monostatic RCS selected, the Geometry field is unavailable. In this case, the solver computes the scattered field in the direction of the plane wave.

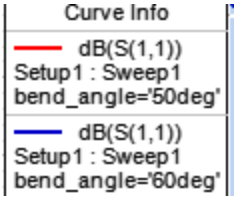
4. Derivative field with a drop-drown selection list of none, all, and specific variables for which you specified **Use** on the Derivatives tab of the solution setup. You can use derivatives in some Optimetrics situations, Far Field reports, non-Port excitations including incident wave, linked field, voltage source, current source, and magnetic bias with the Derivative Tuning feature in the Reporter.
5. A Sources combo box appears in the Reporter when you have specified at least one source in the **Edit Sources** dialog box. See Specifying Source Contexts for Creating Radiated Field Reports. For a usage example, see: *User Defined Solution for MIMO Calculations*.

Using Families Tab for Reports

The **Families** tab of the **Report** dialog box 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 or re-running an analysis with a different variable value). If no variables are defined, or none have solutions for different values, 0 families will be available. If so, the variables other than 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.

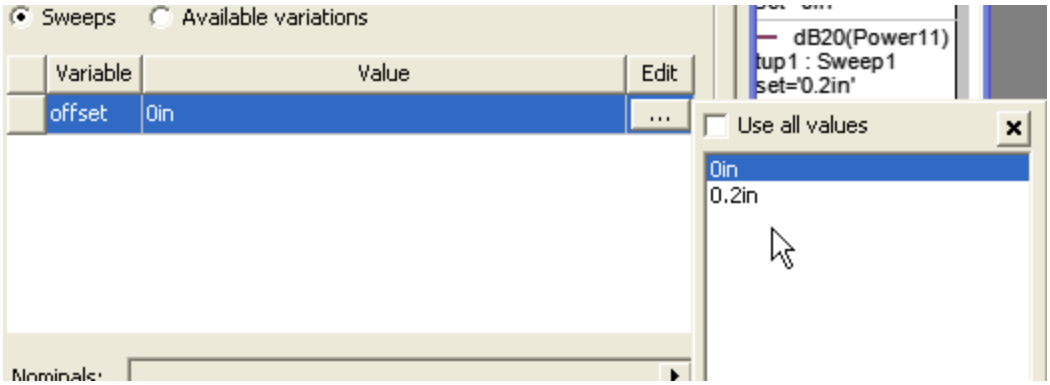


When you select a variable with multiple solved values, a trace for each solved value appears in the Report, with the variable value appended to the trace name in the Report legend.

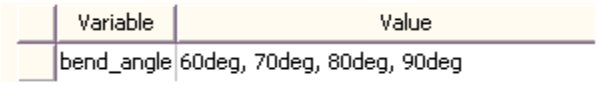


When families are available, you can make selections for the following:

1. Select the **Sweeps** radio button (the default) to list the swept variables you can select or the Available variations button to list and select variation values for which solutions exist.
2. With the **Sweeps** radio button selected, click the ellipsis [**...**] button to display a list of variable values for a particular variable. If many variables exist, you can use a scroll bar to navigate the list.

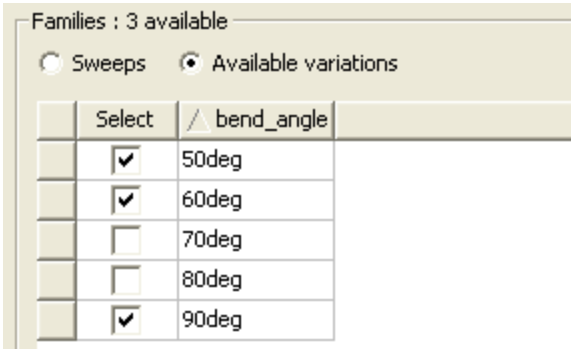


- To select all values, click the check box for **Use all** values. This writes "All" in the value field for that variable. You can also select individual values by clicking on them.
- To select a range of values, hold down the shift key, and click again.
- To select intermittent additional values, hold 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.

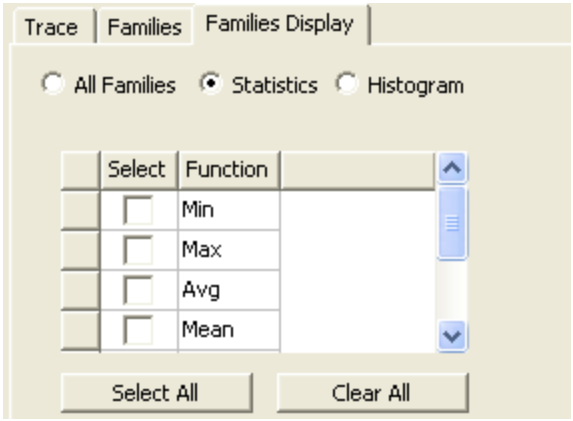
Select the **Available variations** radio button to list the choices that derive from variable combination.



a. To select individual variations, check the select box.
b. To check or clear all variations at once, click the **Select** button at the top of the column.
c. To invert the list order, click the triangle beside the variable name.

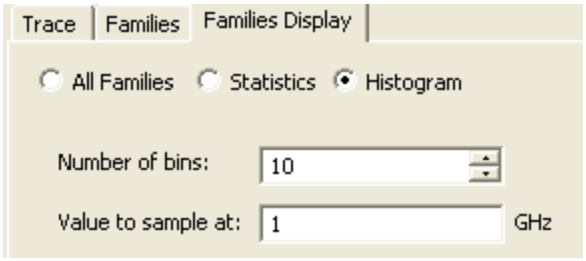
The **Families Display** tab has three radio button selections.

a. **All Families**
b. **Statistics** 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.



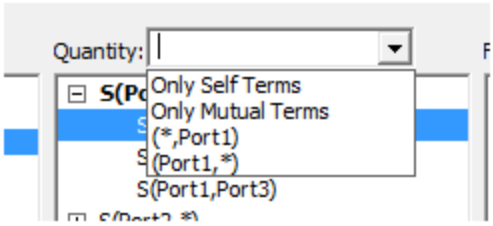
c. **Histogram** lets you select the number of bins to use for a histogram plot, and the

sampling frequency to use.



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.

Modifying Reports

To modify the data that is plotted in a report:

1. In the Project Manager, click the report you want to modify.
2. Right-click **Modify Report**.

   The **Report** dialog box appears.

3. The **Report** dialog box 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 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 box.

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.
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, Eigenmode quantities include Eigenmodes, variables, output variables, and the design. Driven solutions include such categories as S parameters. Report categories for Transient designs include Spectral and Transient. For a Transient Network design with differential pairs defined, the Reporter interface allows selection of single-ended or differential signals just as for driven terminal. Report categories for SBR+ designs that include an Incident Plane wave and an RCS selection as Monostatic do not require a geometry selection and include a range of Monostatic Quantities when select Monostatic RCS as the Report Category. For SBR+ designs and RCS Monostatic you can choose between Freq, IWaveTheta and IWavePhi variables for specifying sweeps. The selected category provides the default name of the plot, for instance S Parameter 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.

> **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. See Report Setup Options.

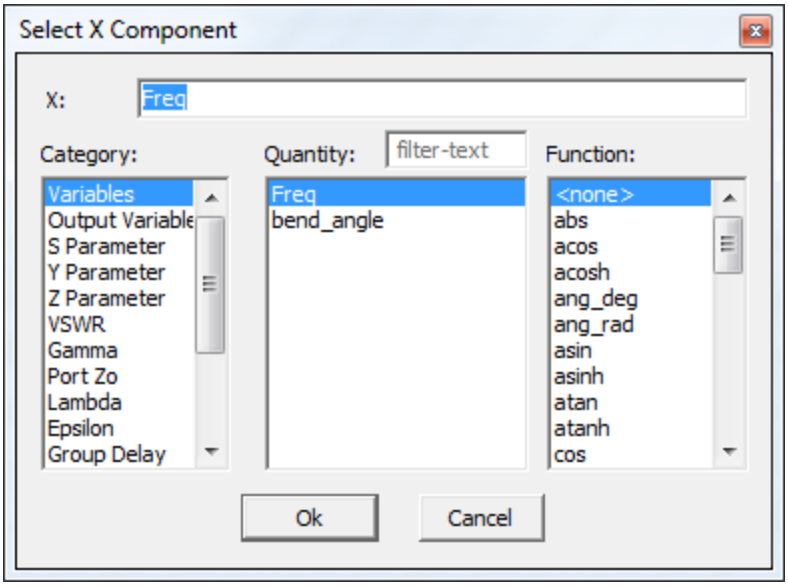
c. **Function** list to apply to the Y quantities.
d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

> **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 value(s) from the drop-down menu.

      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 **Browse** [...] to display the **Select X Component** dialog box.



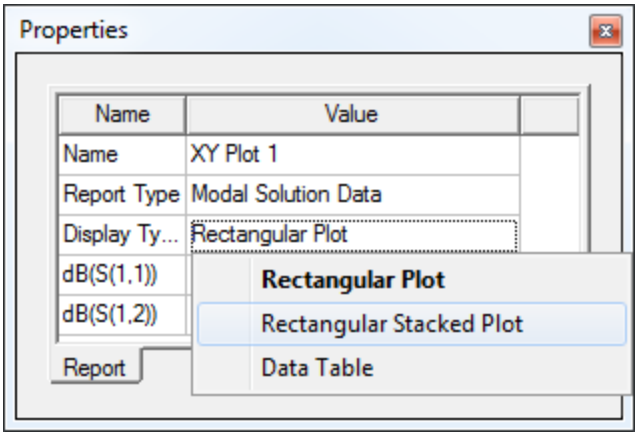
This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

b. If sweeps are available, you can select **Browse** [...] to display a panel that lets you select Use all values, or selected sweep or sweeps, or access an Edit Sweep dialog box with further editing options. Post-Processing variables are Post-Processing sweeps/editable sweeps, so you can use the *Edit Sweep* dialog box to create your own sweep.

c. 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 box for that plot. Select the Report icon in the Project Manager to display the **Properties** dialog box. Selecting the Display Type field displays a menu with selections available for that plot.

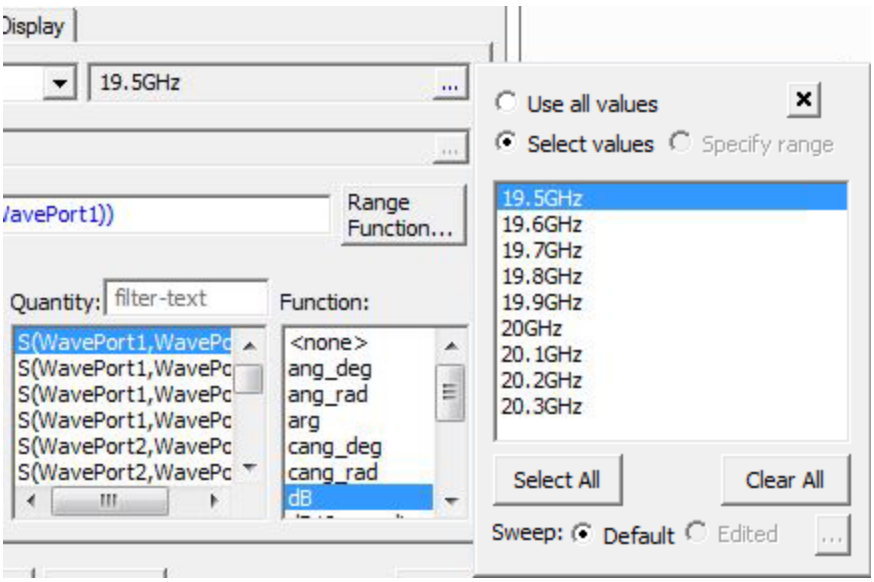


Once you make a selection, the plot display updates for the current selection.

> **Note:**
>
> Remember that for many excitations of interest for plotting, you can control the default base names through the dialog box described here: Setting Default Boundary/Excitation Base Names. This may save you the need to edit individual names in the plots.

Modify Report: Selecting Use All Values or Making Selection

Clicking the browse button on **Primary Sweep** line shows the default selection of **Use all values**. Choose **Select values** to display the sweeps and enable editing, including the **Select All** and **Clear All** buttons.



With **Select values** selected, you can select one or more by clicking an individual value, dragging to select multiple values, or using Alt+click to select specific values.

You select either the Sweep radio button for Default or Edited selection.

You can also select **Browse** [...] here to display the Edit Sweep dialog for Modify Reports, which includes additional editing features.

Modify Report: Using the Edit Sweep Dialog Box

Clicking **Browse** [...] at the lower right corner of the **Use all values** pane opens the **Edit Sweep** dialog box for Modify Report. This 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.

Creating a Report from an Ansoft Report Data File

If you have previously saved an Ansoft Report Data Format (using Report2D > Export), you can create a report from that rdat file. This provides a way to reuse the data and/or the format of a

previously created report.

1. Right-click the Results icon in the Project Manager to display the short cut menu and select **Create Report From File**, or click **Q3D** > **Results** > **Create Report From File**.

   A file browser displays.

2. Select an .rdat format file, and click **Open**.

   The report is created. If it contains data, it displays the report with traces. If not, the report uses the format exported to the .rdat file.

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.

Modifying the Background Properties of a Report

To modify the appearance of a report, or to modify the display properties any object in a report, including traces, axis labels, grids, colors, fonts, legends, color maps, contour color, and so forth:

1. Open the report you want to modify.
2. 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 the selected object, or select **Edit** > **Properties** from the menu bar.

   The **Properties** tabs and options displayed for editable plot objects varies depending on the report type (for example, whether 2D rectangular, 2D polar, Smith, Stacked, or 3D), but can include the following:

   - **Cartesian** – controls the scroll bar and thumb properties for 2D rectangular plots.
   - **Header** – controls 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 box as Subtitle. Edits to the Subtitle Font Property affects both of them.
   - **General** – controls the background color (the perimeter around the trace display) for the plot, the contrast color (the trace display background), the Field width, the Precision, 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 Fonts property is on by default and scales text in plots and colorkey (contour plot, field plots in 3D modeler) for high resolution screens.

- **Legend** – controls the properties for whether to include a Legend Name, Show Trace Name, Solution Name, and Variation Key. At least one of these must be selected. Legend Name is blank by default. When non-empty, a header row for the Legend in plot shows up with that string. You can also specify the File Name Display as Full Path, File Name without Path, or as an Array Index. 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. See: Modifying the Legend in a Report.
- **Color Key** – for 3D plots, controls the appearance of the color key (colors, transparency, border appearance, fonts, number format, field width and precision.
- **Contour** – for 3D plots, controls the appearance of the color map, including map type, ramp color, spectrum, IsoValType, levels, number of contours, and values shown.
- **Radiation Pattern** – for 2D polar plots, controls whether to show the circular grid and angle lines.
- **Stacked** – for stacked plots, controls properties for X scrollbar, thumb properties, and stack layout, auto fit, and stack height.
- **Smith** – for Smith charts, controls whether to show grids for Imp., Adm., Cir, and angle lines.
- **Traces** – controls the properties for traces, including: 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. You can select traces either in the Legend or on the plot. See: Editing the Display Properties of Traces.
- **Axis** for X, Y or Z, or for Phi, Theta or Rho, and circular – the defaults for most of these values (applying to 2D and 3D both) 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 box.
    - 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 box lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog box also contains a preview field. OK the selections to apply the font edits and to close the dialog box.
    - Display Units – this specifies whether to display the axis 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 or Y or Z **Scaling Tab** – 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. For 3D plots, scaling is on the Axis tabs
    - 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.

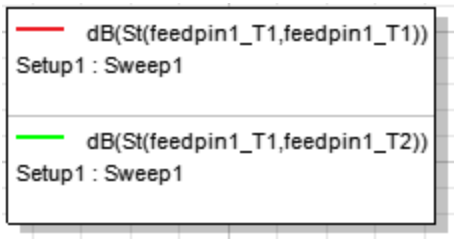
        Each axis can be set to treat infinity values in a user defined way. When you check the Map Infinity Mode, any infinity values in the input data get the infinityMap value (negative infinity get the value*-1 and positive infinity the positive value specified). This can be useful if there are zeros, or very small values that Q3D Extractortreats 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, major and minor circles on polar grids, and scaling. For the 3D rectangular plots, there are separate tabs for the XY, YZ and ZX axes, and for 3D Polar plots, tabs for phi-rho, or theta-rho grids.
3. Edit the properties, and **OK** the dialog box to apply the changes.

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.

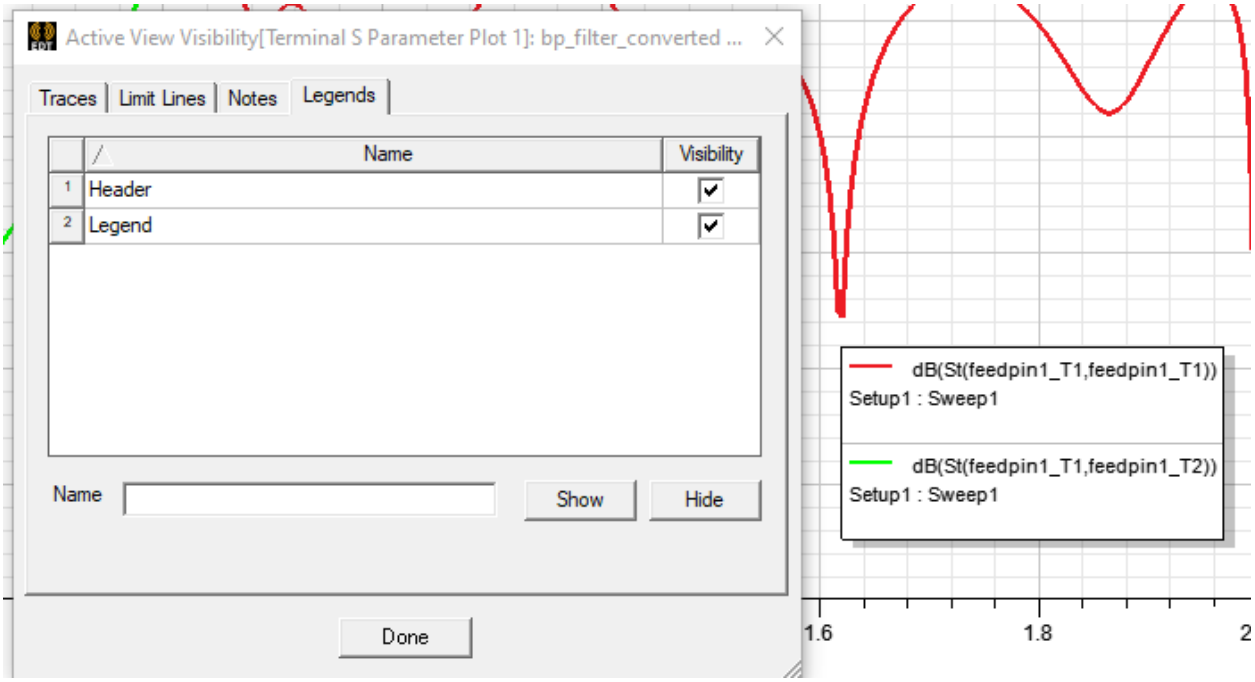


To show or hide a legend in a report:

1. Make the report the active view.
2. Use **View** > **Active View Visibility** or the **Show/Hide** icons on the toolbar to display or hide the report.

   Either command displays the **Active View** dialog box.
3. Select the **Legend** tab.

   This lists the legend (or legends) in the report.

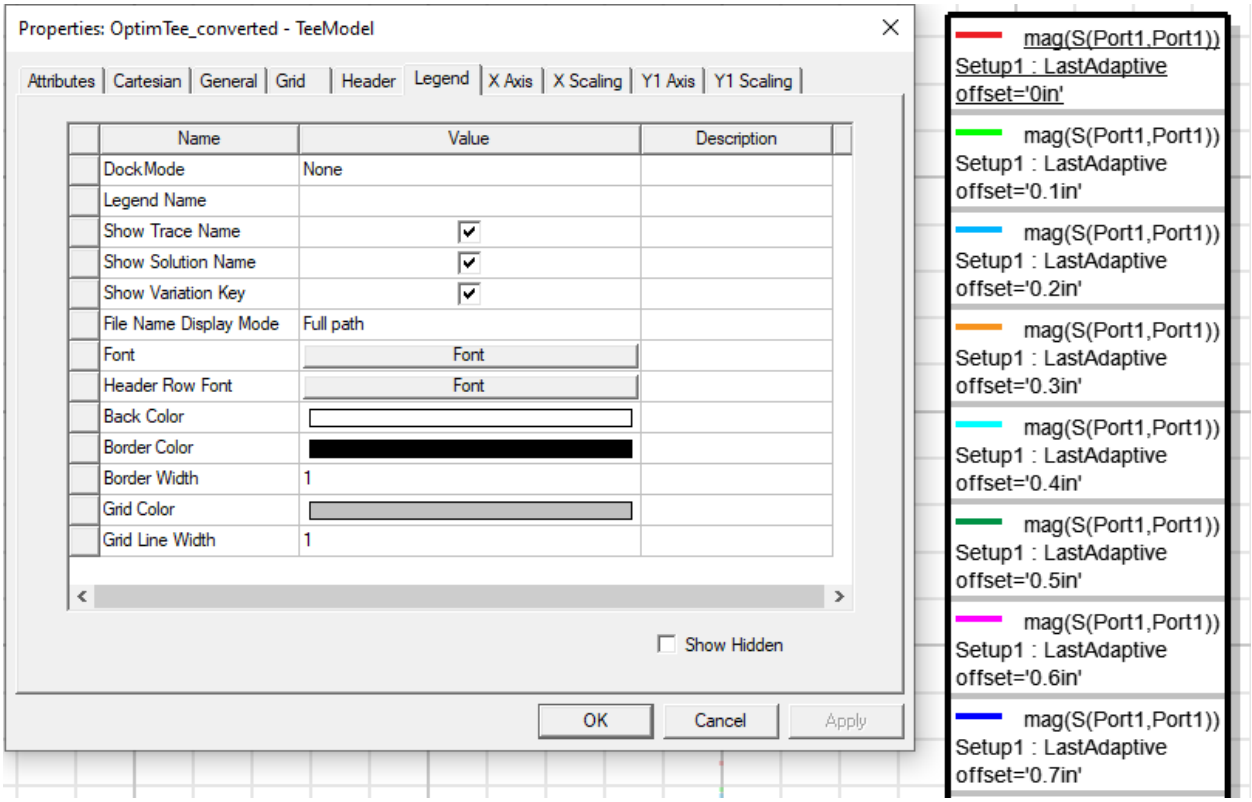


4. Enable the visibility check box, and click **OK** to close the dialog box and apply the change.

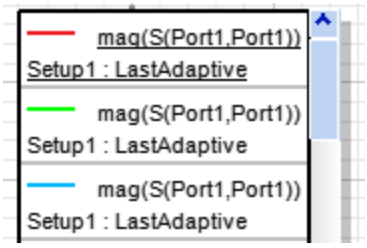
To edit the display properties of a legend:

1. Double-click a legend in a report to display a docked properties window, or right-click on the legend and select **Edit** > **Properties** to display the floating **Properties** dialog box.

   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 box 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** dialog box 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.

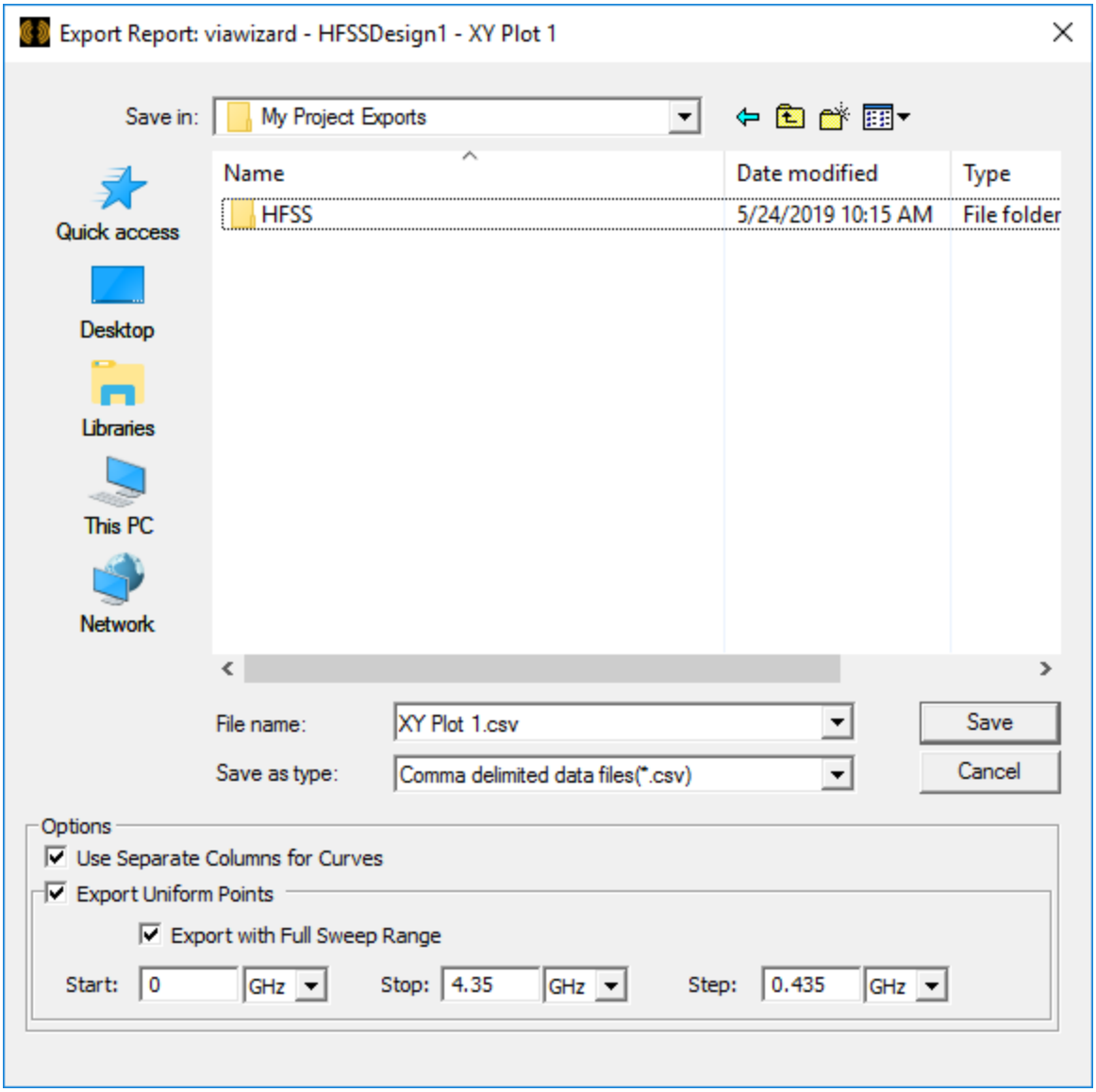
2. Click and drag the horizontal or vertical edge to the desire size.
3. Release.

Exporting Ansoft Report Data Format Files

Ansoft Report Data format (*.rdat) files provide a way to export reports or report formats, which you can then import using **Reports> Create Report From File**. This can save repeated editing of properties (for example, the company name or color schemes) when you create other reports. You must have an existing plot open to see the **Report2D** menu.

1. Click **Report2D** > **Export...** or **Report 3D**> **Export...**

   The **Export Report** dialog box appears:



2. Use the **Save as type** drop-down menu to select **Ansoft Report Data Files (*.rdat)** format.
3. Browse to a destination folder and enter a name for your file in the **File name** field.
4. The **Export Uniform Points** option, if available for your report type, allows the *.rdat file to contain points for the start, stop, and step at the given frequencies. If you do not select this option, the file contains only the current file format, including any modifications you have applied.
5. Click **Save**.

   The file is exported to the specified location as an Ansoft Report Data file (*.rdat).

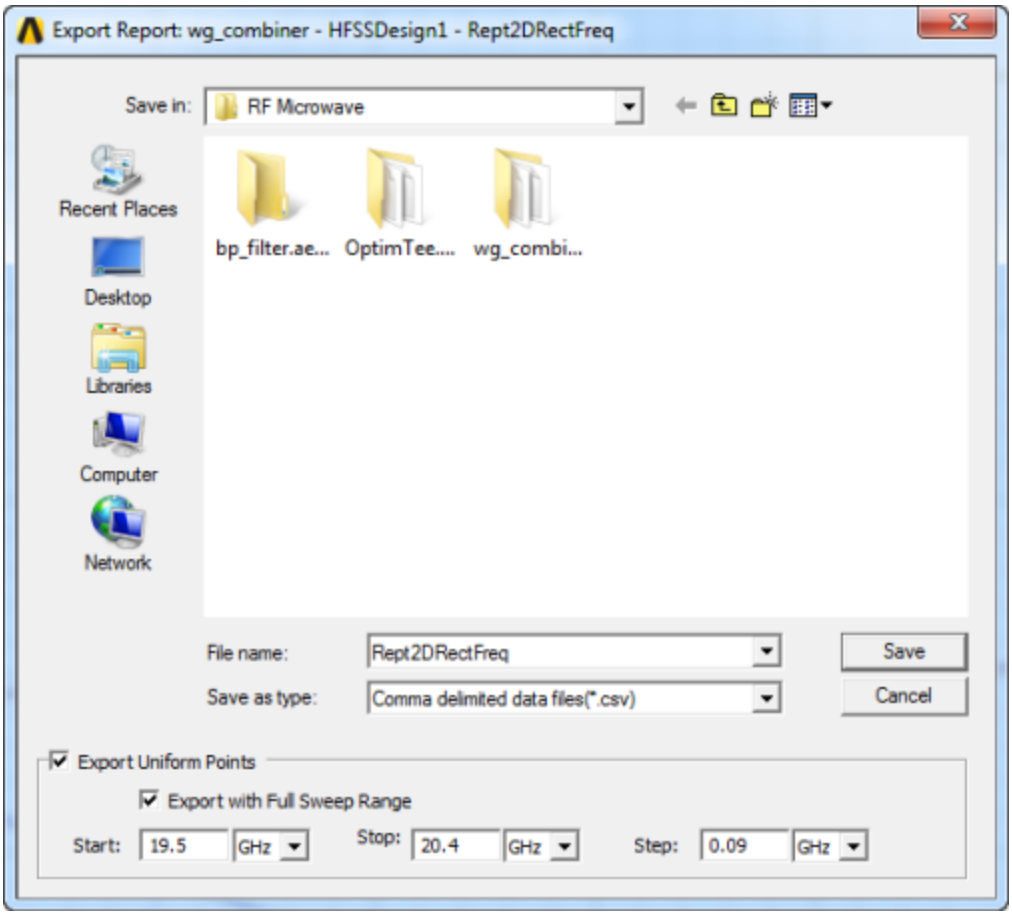
The file will then be available for import using Create Report from file.

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.

1. With a report open, click **Report2D** > **Export...** or **Report3D** > **Export...**

   The **Export Report** dialog box appears:

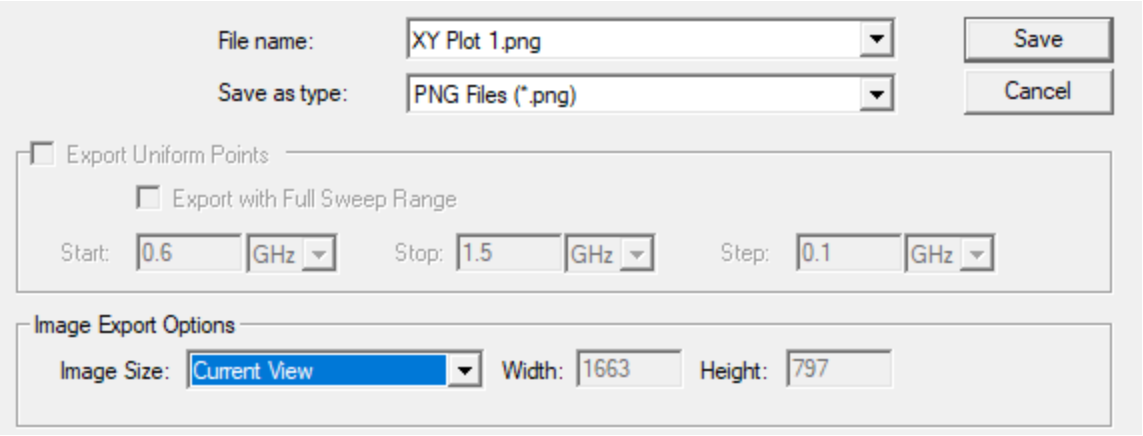


2. Click **Browse...** to open the **Export Report** browser window.
3. Specify the file location and name, and select a graphics format from the drop-down menu:

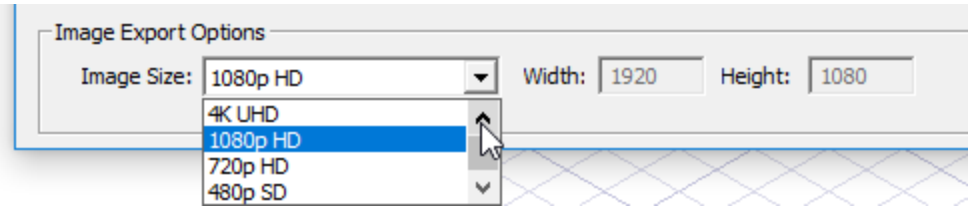
   - .bmp
   - .gif
   - .jpeg

- .png
- .tiff
- .wrl

4. When you select an image format, the **Image Export Options** appear at the bottom of the dialog box:



5. Specify the **Image Export Options** by selecting from the drop-down menu. You may need to scroll or click the up or down arrows to see all selections. These include: Current View, User Defined (which enables the Width and Height fields), Full Screen, 8K UHD, 4K UHD, 1080p HD, 720p HD, and 480p SD. The resolution for the current selection is displayed in the Width and Height fields.



6. Click **Save** to close the browser window, and then **OK** to close the **Export** window.

To export an image file of a report to specified high resolution, use the scripting commands. The image can be created at a specified resolution. Fonts and line thickness is not scaled. Only the image. You will have to iteratively increase font sizes until you find a suitable output.

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 from Ansys products.

File Extension	Information
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.

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 and plot attribute settings) when you create other reports. You can prepare a template by copy/pasting plots settings from one plot to another of the same display type. Once you create templates, you can access them from the **Results** > **Report Templates** > menu and the **Report2D** > **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 the report name of interest to display the shortcut menu and click **Report Templates** > **Save...**. You can also click **Report2D** > **Report templates** > **Save...** or **Report3D** > **Save As Template...**

   This displays the **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.

   The **Save As Type** field currently supports the Ansoft Report Format (*rpt) format.

4. Click **Save** to save the template to the PersonalLib menu.

All *.rpt templates in the userLib 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 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**.

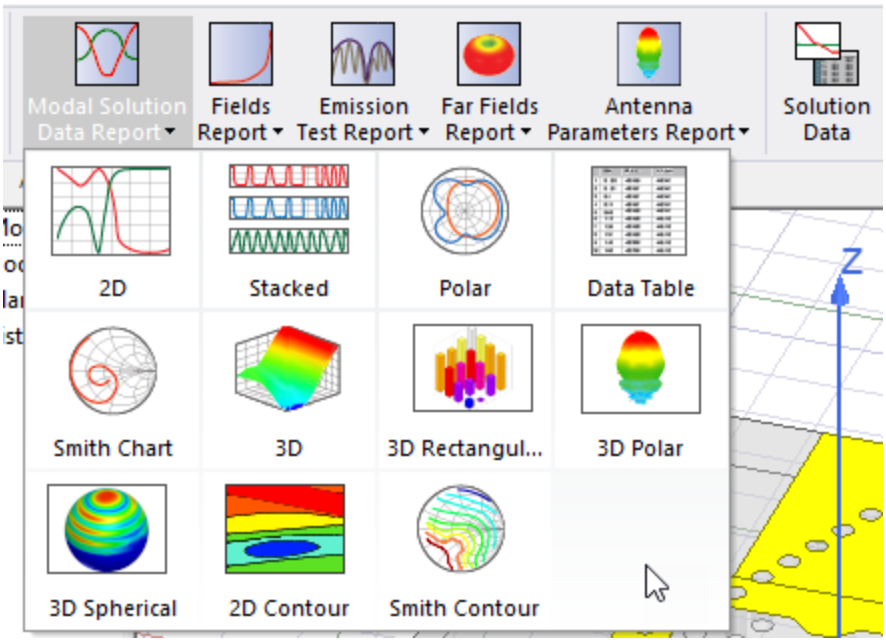
Selecting the Report Type

The **Report Types** available for creating a report depend 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, S Parameters as the Category type causes the default report name to be S Parameter Plot *n*. Depending on the setup, you can make a selection from the following report types.

Modal Solution Data	S-, Y-, and Z-parameter data are available to plot, along with propagation constant, characteristic port impedance, reflection/transmission coefficients for HFSS designs, and Voltage Standing Wave Ratio (VSWR) data. For HFSS calculations, phase is currently assigned zero value.
Terminal Solution Data	This solution type results in a terminal-based description in terms of voltages and currents. Some modal data is also available. The terminal-based S-, Y-, and Z-parameters, VSWR, Port Zo, and Active S-, Y-, Z- and VSWR parameters are available to plot.
Eigenmode Parameters	Eigen Modes and Eigen Q data are available to plot.
Fields	Basic or derived field quantities calculated on lines or integrated over surfaces or objects are available to plot. Displacement magnitude is available as a predefined modal field result. Predefined thermal field results include temperature, heat flux magnitude, and EM loss densities.
Far Fields	Radiated fields computed in the far-field region. The following quantities are available to plot: rE, gain, realized gain, beam area, directivity, axial ratio, polarization ratio, and normalized antenna calculated by HFSS. You can do Contour Plots with a Domain of either Theta, Phi, or Sine Space. You must have defined an infinite sphere geometry and at least one radiation or PML boundary to create a far-fields report. If you manually enter Phi or Theta component of a far field quantity when you using Az/El or El/Az far field definition, then the Report uses the Z axis of the Az/El or El/Az coordinate system definition as pole to calculate theta/phi component for rE.

Near Fields	Radiated fields computed in the near-field region. These include: variables, output variables, near E, max near field parameters, and near normalized antenna. You can create Contour Plots with a Domain of either Theta, Phi, or Sine Space. You must have defined a near-field line, box, rectangle, or near-field sphere and at least one radiation or PML boundary to create a near-fields report. Field components and positions in reports are in the local coordinate system of the near field domain, not global coordinates.
Antenna Parameters	Whereas far field reports are computed as points around an infinite sphere, antenna parameters provide one value per quantity for the entire sphere. You must have defined an infinite sphere geometry and at least one radiation or PML boundary to create an antenna parameters report.
Emission Test	You can conduct an emission test under the same conditions as for a near field report except that.an emission test cannot be conducted for a ports-only solution. You must have defined a near-field line or near-field sphere and at least one radiation or PML boundary.

When you click **Product > Results** or right-click **Results** in the Project tree, or select the **Results** tab of the ribbon, one or more of the following report types and displays are available. *Available options depend on the nature of your installation and project(s)*.

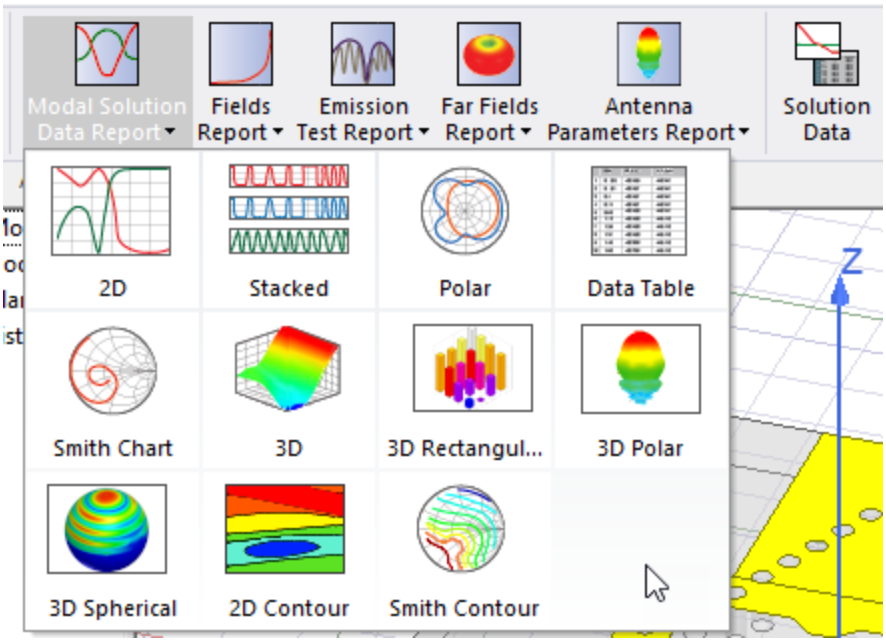


Report Type	Display Type

Create Standard Report	A Standard report can be depicted in any of the following display types: • Rectangular Plot • Data Table • Polar Plot • Smith Chart • Rectangular Stacked Plot • 3D Rectangular Plot • 3D Rectangular Bar Plot • 3D Polar Plot • 3D Spherical Plot • Rectangular Contour Plot • Smith Contour Plot
Create Eye Diagram Report	An Eye Diagram report can be depicted in any of the following display types: • Rectangular Plot • Data Table • Statistical Eye Plot
Create Constellation Report	A Constellation report can be depicted in any of the following display types: • Rectangular Plot • Data Table
Create Statistical Eye Report	A Statistical Eye report can be depicted in any of the following display types: • Rectangular Plot • Data Table
Create Report From File	A Data File report can be created in any of the available display types using a previous report that has been saved to a file.

Selecting the Display Type

The information in a report can be displayed in several formats. When you select the **Results** tab in the ribbon, depending on reports available for a given design and Solution type, you see a selection of report type groups (for example, Modal Solution Data, or Fields Report, Emission Test, Antenna Parameters, Near Fields, Terminal Solution Report, Far Fields Report, Characteristic Mode Data, etc.), with drop-down menus that displays the kinds of reports available for the design in that group:

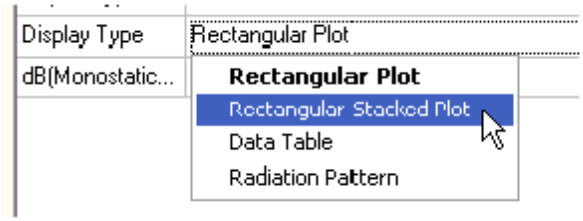


For the initial plot, you can also select from the following **Display Type** formats in the **Create<*type*> Report** submenu:

Rectangular Plot	A 2D rectangular (x-y) graph.
Rectangular Stacked Plot	This choice puts each trace into its own 2D rectangular plot, and stacks each plot, rather than overlaying the traces on the same plot.
3D Rectangular Plot	A 3D rectangular (x-y-z) graph.
3D Rectangular Bar Plot	A 3D plot of rectangular bars.
Rectangular Contour Plot	A rectangular (x-y-z) graph. Contour plots are useful to visualize surfaces (e.g., Directivity as a function of phi/theta).
Polar Plot	A 2D circular chart divided by spherical coordinates.
3D Polar Plot	A 3D circular plot divided by spherical coordinates.
3D Spherical Plot	A 3D circular plot where the radius (rho) of the plot at all points is uniform and is equal to the maximum of all rhos at each (theta, phi) points.
Smith Chart	A 2D polar chart of S-parameters upon which a normalized impedance grid has been superimposed.
Smith	A polar chart. Contour plots are useful to visualize surfaces.

Contour Plot	
Data Table	A grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or another quantity.
Radiation Pattern	A 2D polar plot of radiated fields.

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.

Creating 2D Rectangular Plots

A rectangular plot is a 2D, x-y graph of results.

1. On the **Results** menu (under the **Q3D Extractor** menu, or right-click **Results** in the Project Manager), click **Create < *type* > Report**, and select **Rectangular Plot**.

   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. Whether this field appears, and the domains listed, depend on the Solution type and the report <*type*> selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   c. 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, **Y** component section, specify the information to plot along the y-axis:
    a. In the **Category** list, click the type of information to plot. The category selected 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. Value field displays the currently specified Quantity and Function. You can edit this field directly.

    > **Note:**
    >
    > Color shows valid expression.

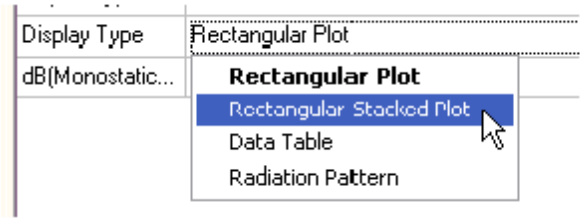
    e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:
    - Select the sweep variable to use from the drop-down list.
    - If sweeps are available, you can select the browse button to display a dialog that lets you select particular sweep or sweeps, or all sweeps. 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 **Add Trace** 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. The default name is based on the Report Category you selected, (for example, S Parameter Plot *n* or rE Plott *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes. 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**.

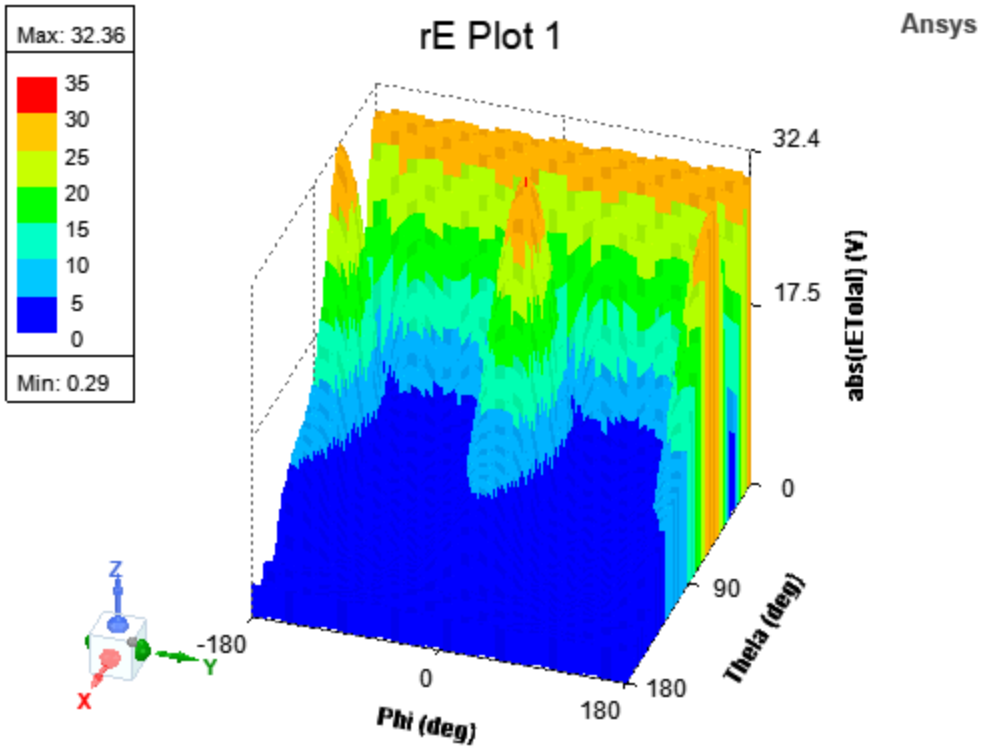
You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Creating 3D Rectangular Bar Plots

This is a 3D, x-y-z graph of results as rectangular bars.

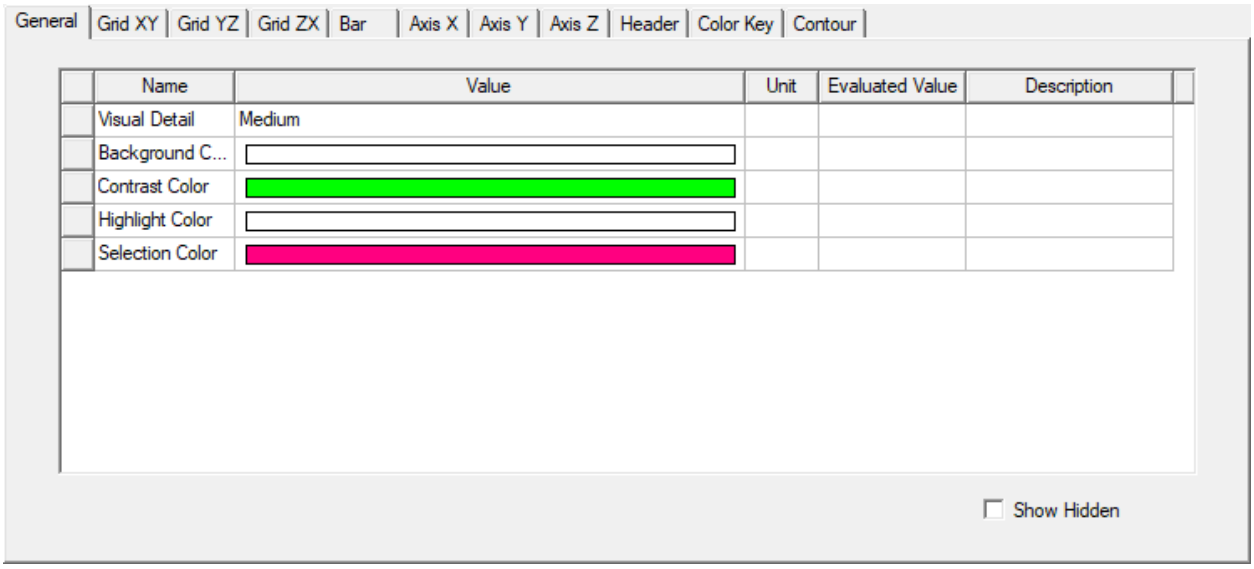


Working with a 3D Rectangular Plot

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.



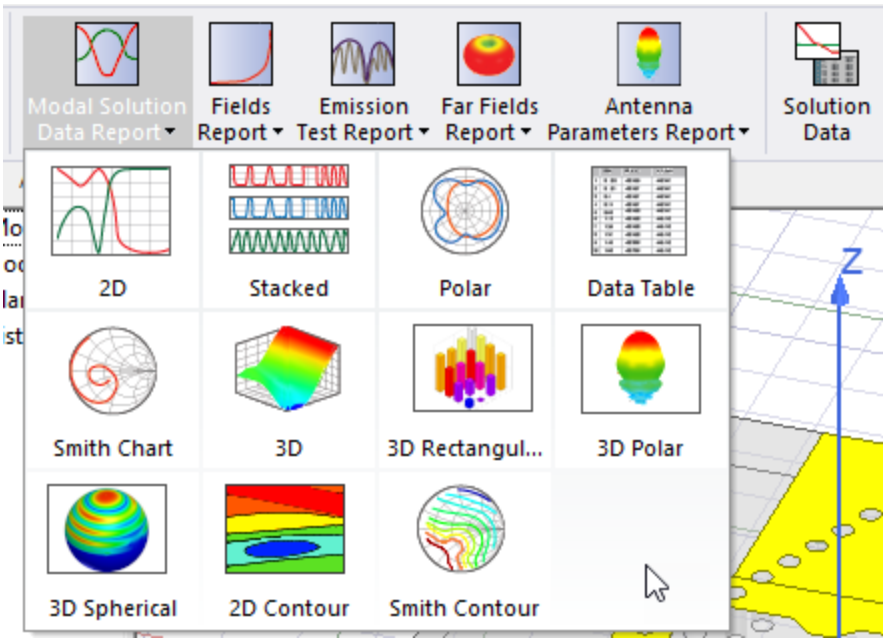
The properties tabs for a 3D Rectangular Bar plot are:

- 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 17-127.

Creating a 3D Rectangular Bar Plot

1. Access the Results menu one of two ways:
   - From the **Q3D Extractor** menu.
   - From the **Project Manager**, right-click **Results**.
2. From the **Results** menu, click **Create *<type>* Report**, and select **3D Rectangular Bar plot**, or click the **Results** ribbon tab and select the 3D Rectangular Bar icon for the Report type you want to create.



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.
    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.
4. 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.

5. 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.
6. 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 box that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
7. 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.

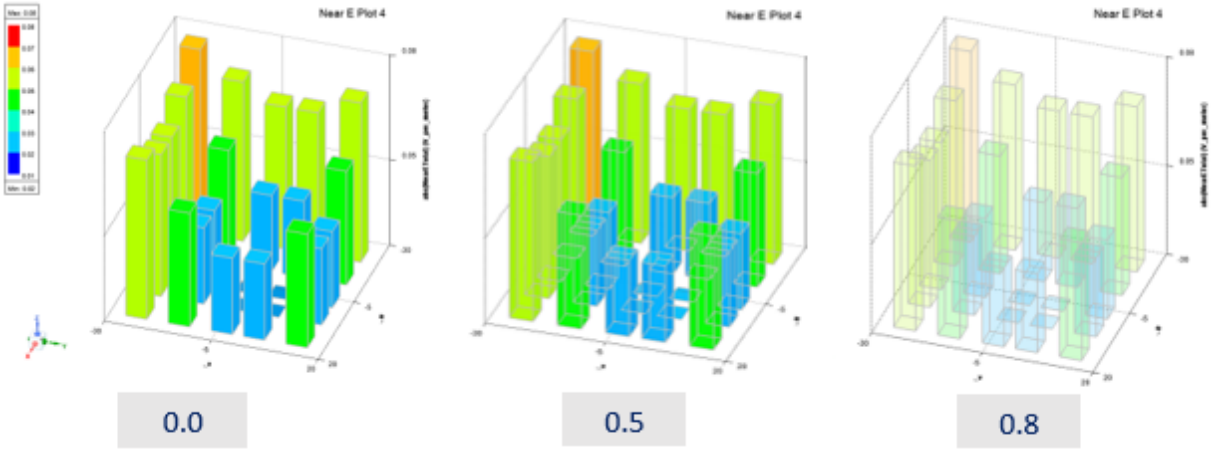
8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

3D Rectangular Bar Customization

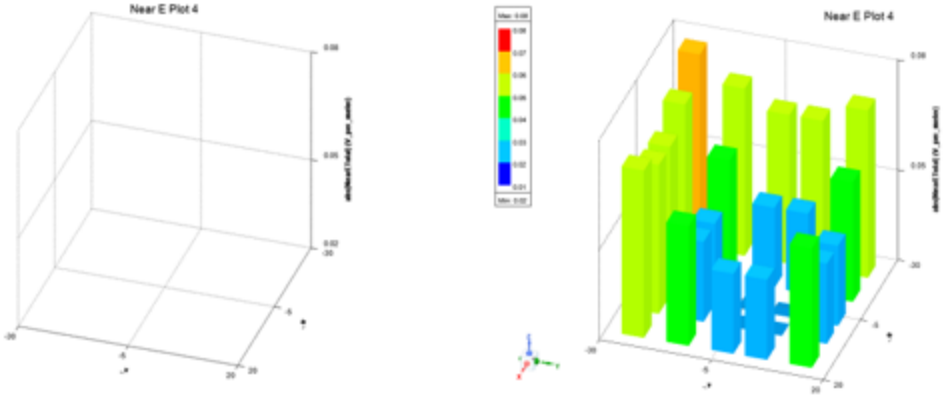
Selecting a bar in the plot shows the customizable attributes of bars in a 3D Rectangular Bar plot.

The customizable attributes of the bars in a 3D Rectangular Bar plot include:

- Transparency



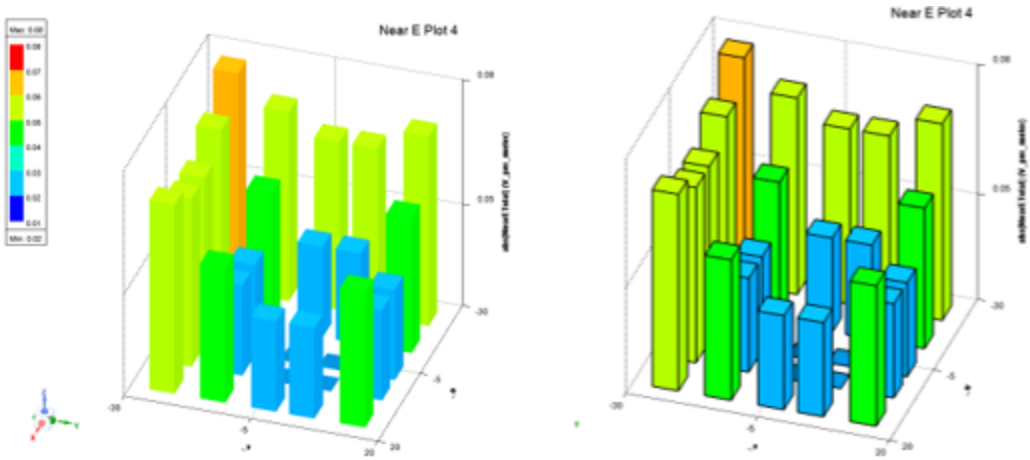
- Filled



Filled [ ]

Filled [√]

- Show Outline

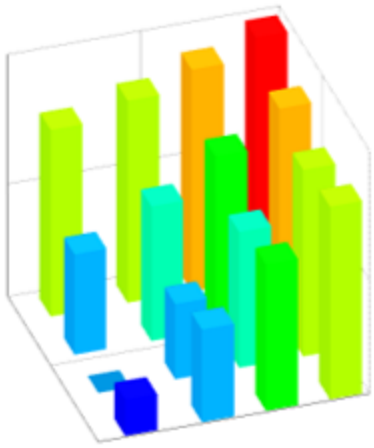


Show Outline []

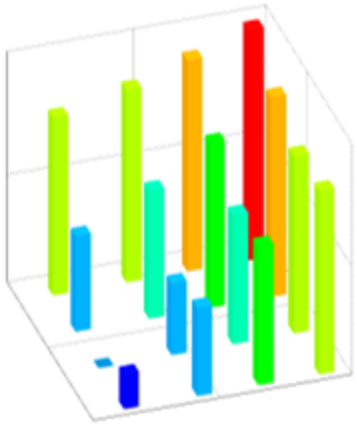
Show Outline [√]

- Outline Color (customizable, see plot above)
- Outline Width (customizable, see plot above)

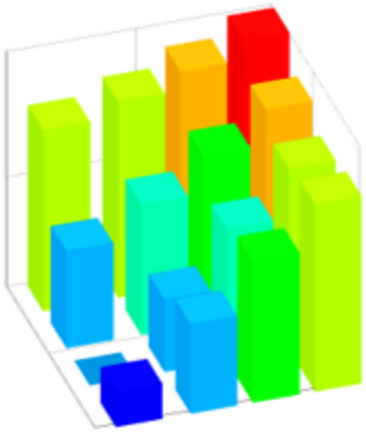
- Bar Width, controls the thickness of the bar. Medium (default), Wide and Narrow are the width that equals to ½, ¾ and ¼ of the min spacing between adjacent points respectively.



Medium

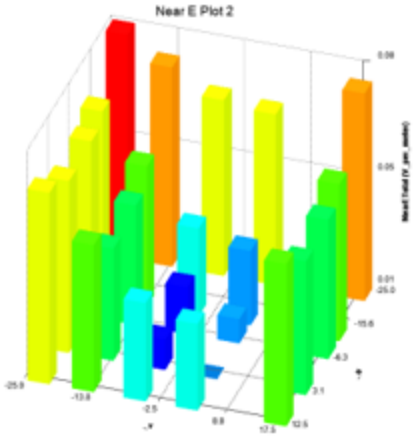


Narrow

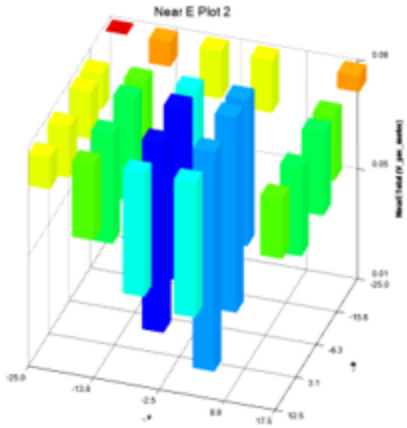


Wide

- Bar Infinity represents the third coordinate (z) using the bottom of the bar.



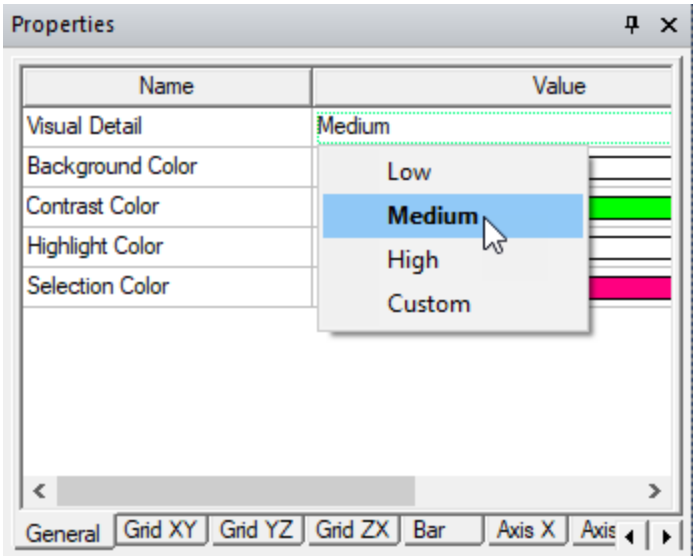
Bar Infinity []



Bar Infinity [V]

Controlling Visual Detail in a 3D Rectangular Bar Plot

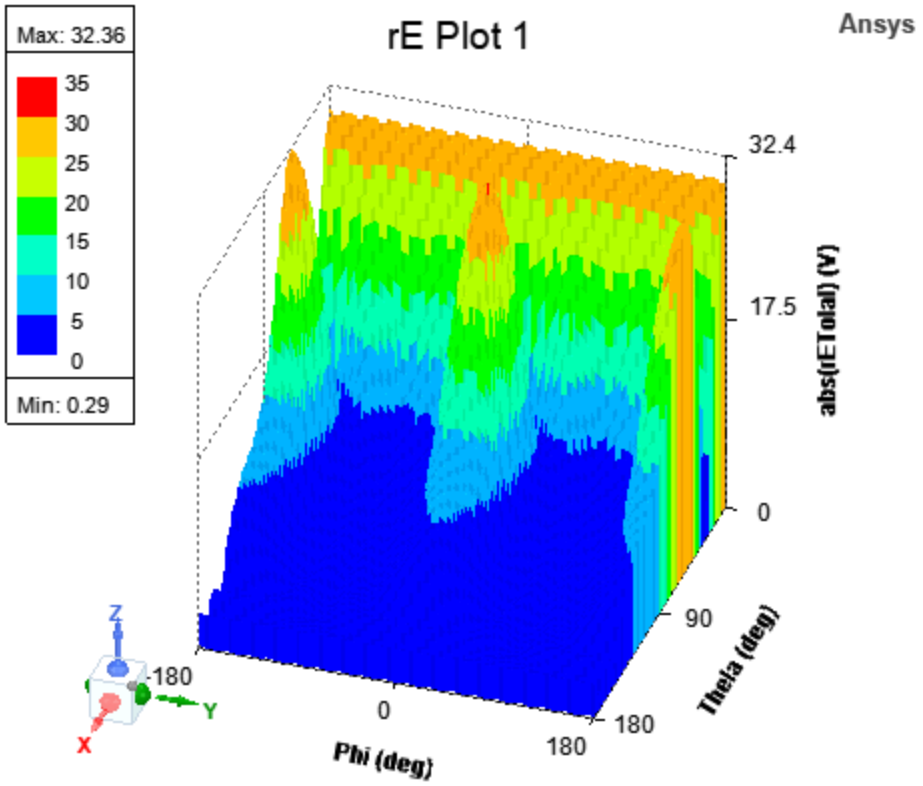
If a particular plot seems busy with information, you can edit plot properties, such as **Axis** and **Grid Attributes** for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the **Properties** dialog box. The **Visual Detail** property on the **General** 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).

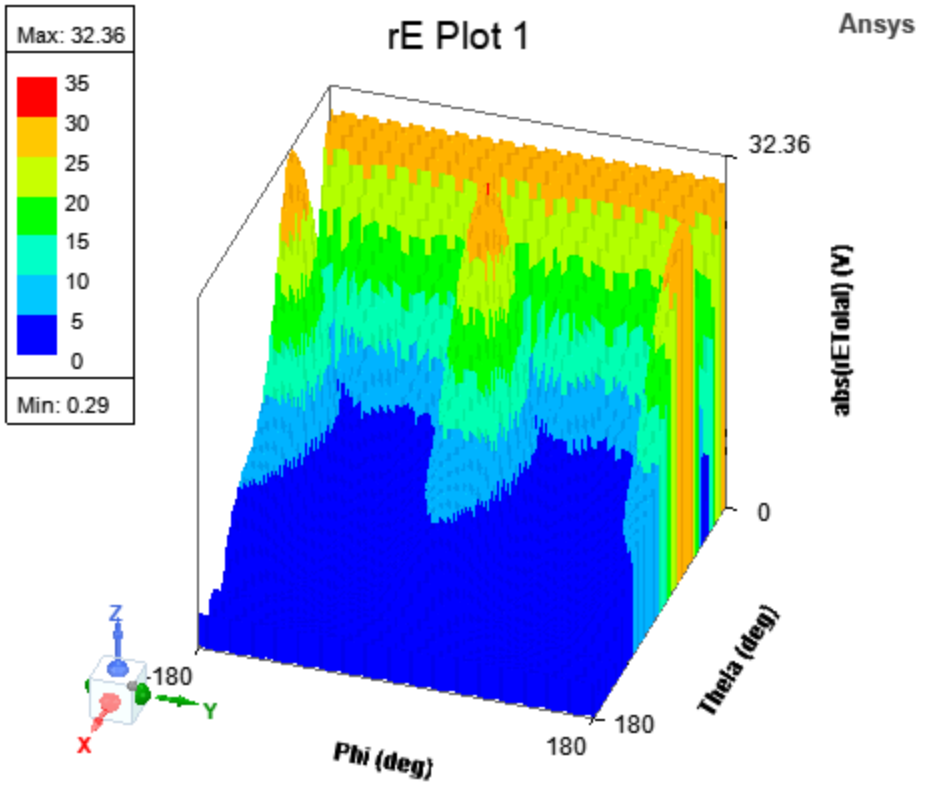
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



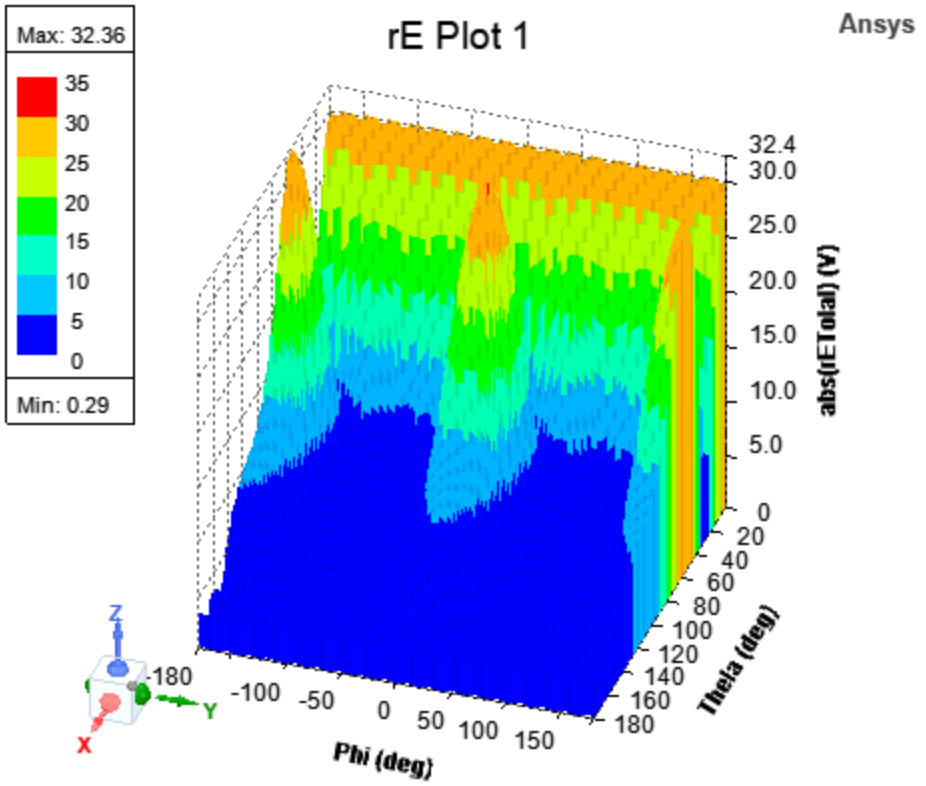
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

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.

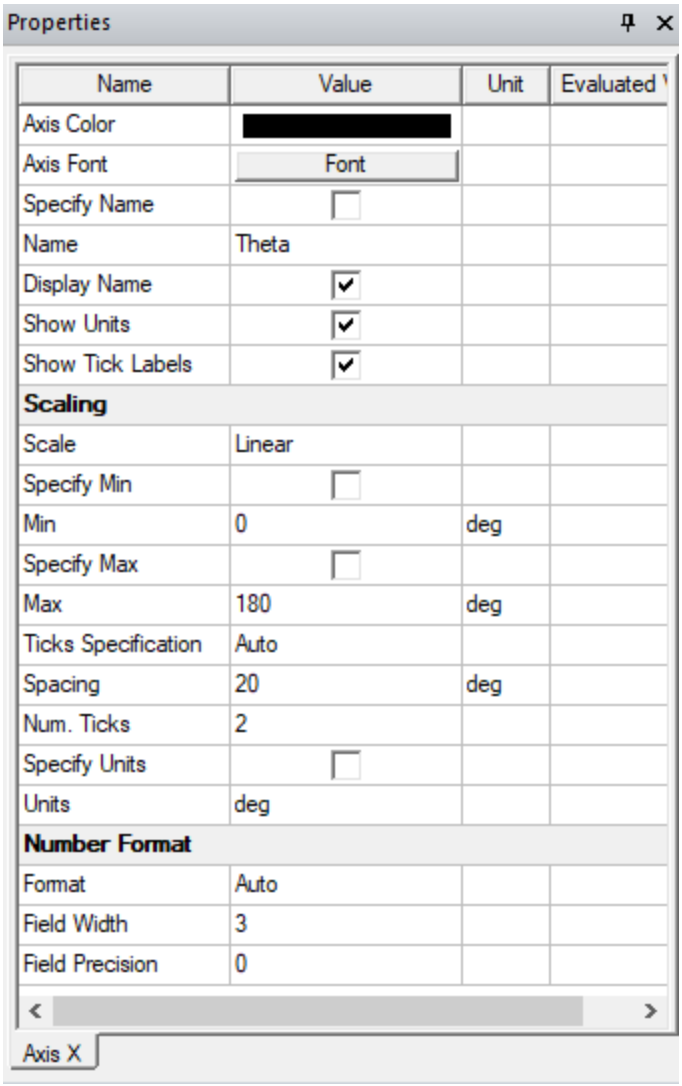
3D Rectangular Bar Plot with High Visual Detail



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:

--Scaling			
Scale	Linear		
Specify Min	☐		
Min	0	deg	
Specify Max	☐		
Max	375	deg	
Ticks Specification	Auto		
Spacing	125	deg	
Num. Ticks	2		

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.

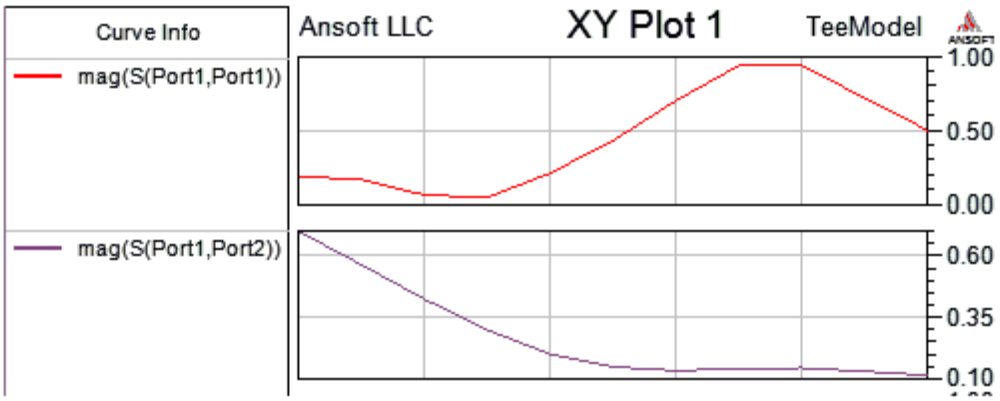
Valid Num. Ticks are between 0 and 100, including 0 and 100. If you enter an invalid value, an error message is shown. If you enter a spacing value that results in number of ticks greater than 100, then an appropriate value is shown.

- 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.

Creating 2D Rectangular Stacked Plots

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 (under the **Q3D Extractor** menu or right-click **Results** in the Project Manager), click **Create <*type*> Report**, and select **Rectangular Stacked Plot**.

   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. Whether this field appears, and the domains listed, depend on the Solution type and the report ***<type>*** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   c. 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, **Y** component section, specify the information to plot along the y-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. 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.

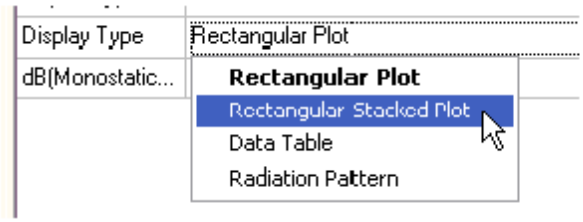
4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular sweep or sweeps, or all sweeps. 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 **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 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**.

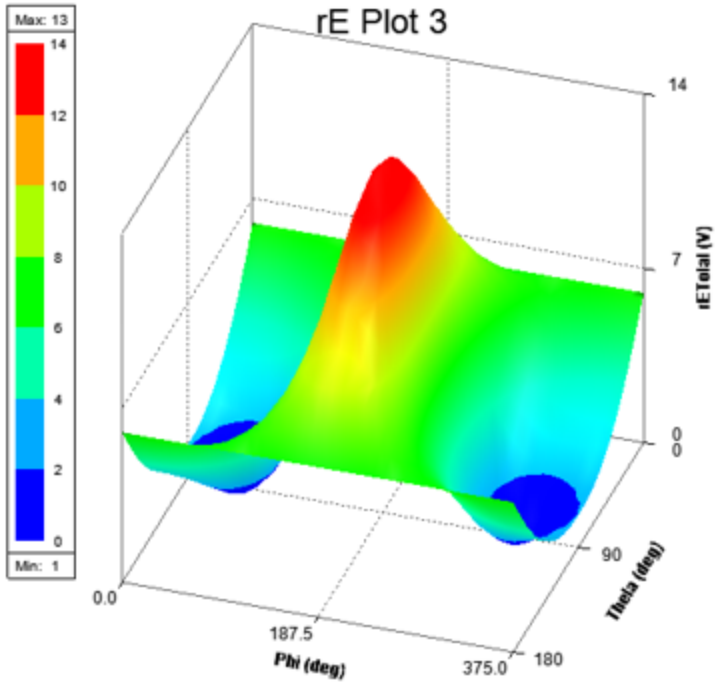
You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Creating 3D Rectangular Plots

This is a 3D, x-y-z graph of results.



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 the next page.

Creating a 3D Rectangular Plot

1. On the **Results** menu (under the **Q3D Extractor** menu or right-click **Results** in the Project Manager), click **Create *<type>* Report**, and select **3D Rectangular plot** from the report type menu.

   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. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report.**

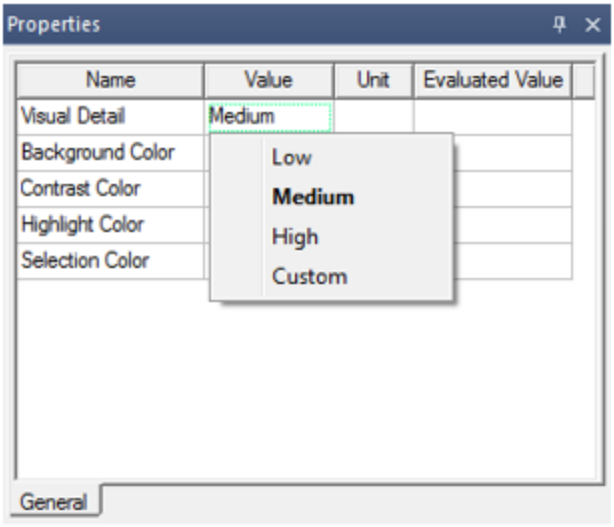
   This creates a new report in Project tree, displays the report with the defined trace, and enables **Add Trace** on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, S Parameter Plot *n* or rE Plot *n*). You can edit the plot names in the project 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.

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

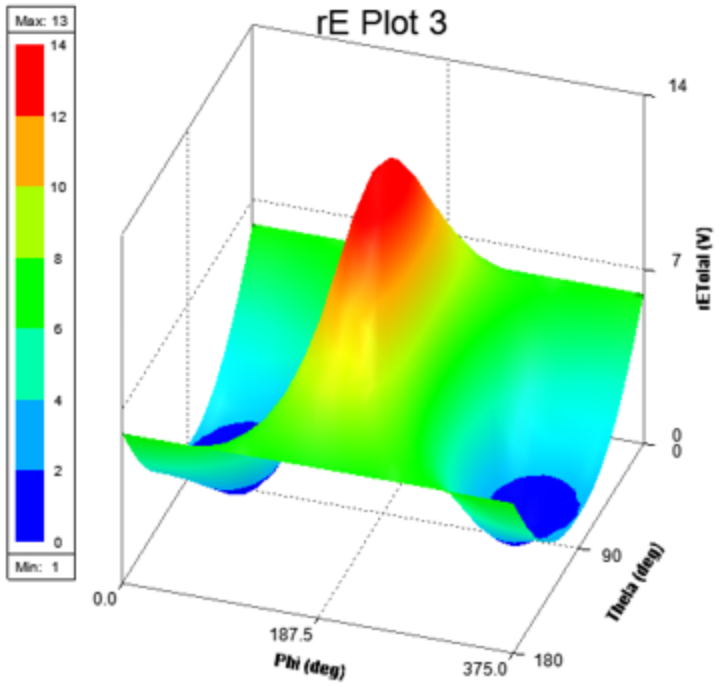
If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides 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).

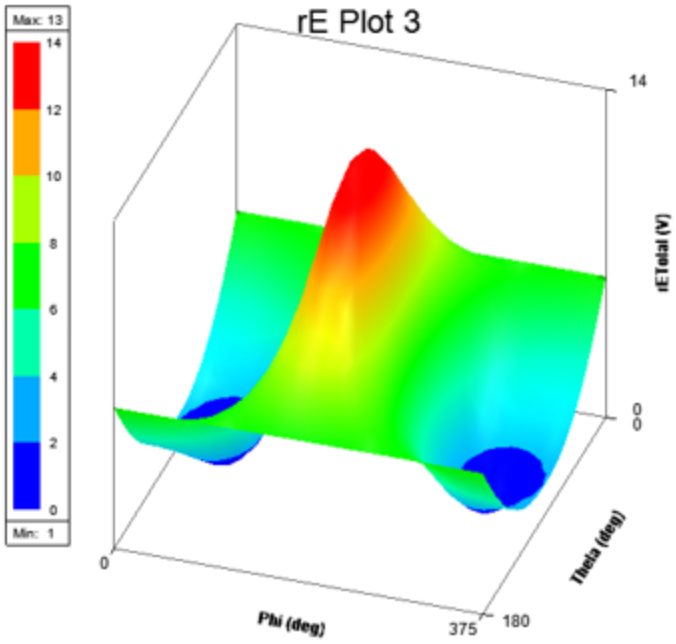
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



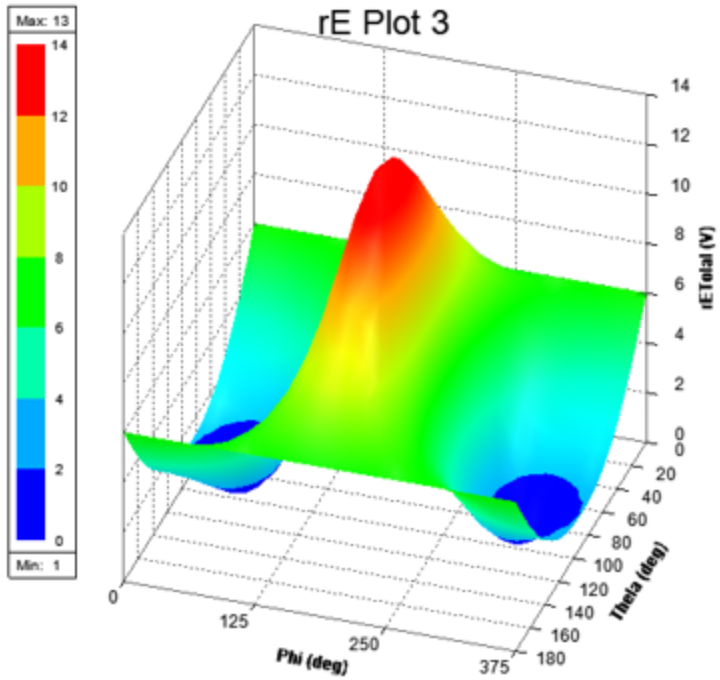
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

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.

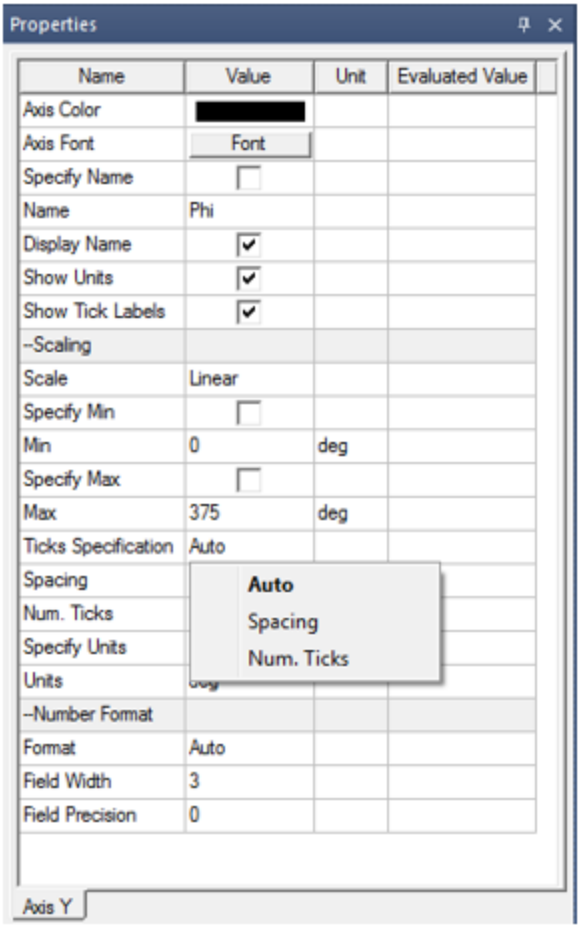
3D Rectangular Plot with High Visual Detail



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:

--Scaling			
Scale	Linear		
Specify Min	☐		
Min	0	deg	
Specify Max	☐		
Max	375	deg	
Ticks Specification	Auto		
Spacing	125	deg	
Num. Ticks	2		

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.

Creating Rectangular Contour Plots

This is an x-y-z graph of results. Any data that you can currently plot in 3D (as 3D Cartesian or 3D polar) is a candidate for a contour plot.

1. On the **Results** menu (**Q3D Extractor** menu or right-click **Results** in the Project Manager), click **Create *<type>* Report**, and select **Rectangular Contour plot** from the report type menu.

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.
    c. Domain field with a drop-down selection list. For Near and Far Field reports, if you have defined the respective Radiation Setups, a Domain field lists Theta, Phi, and Sine Space.

    For details, see Creating Sine Space plots.

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. The selected Category 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 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 Manager. 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.

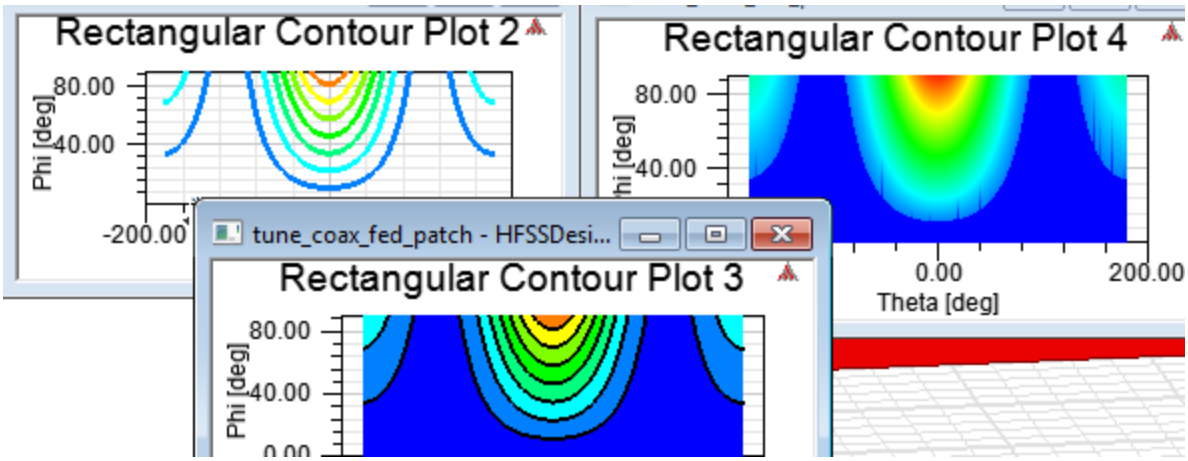
   The Trace Properties for Cartesian plots include tabs for:

   - Attributes tab, including Name, Line Style, Line Width, and Color.
   - Contour tab, including Colormap Type, Color, Spectrum, IsoValType, whether to Overlay Contour Lines (on Fringe or Tone plots), Scaling parameters, and Number of contours and spacing.

   Colormap Type can be Spectrum, Ramp, or Uniform.

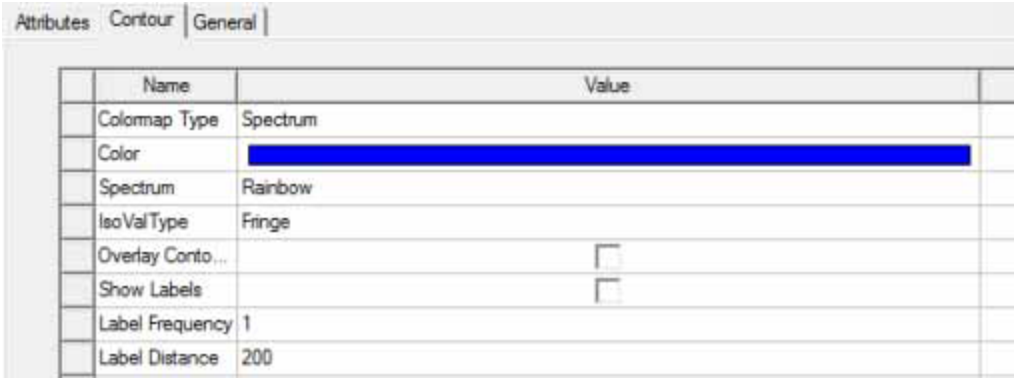
   Spectrum can be Rainbow, Temperature, Magenta, or Gray

   IsoValType can be Line, Fringe, or Tone.



   General tab, including Back Color, Plot Area Color, Enable Y Axis Stripes, Field Width, Prevision, and whether to Use Scientific Notation.

   You can also access these properties by double-clicking on the Contour plot and viewing the **Contour** tab.



7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Creating Sine Space Plots

Sine Space plots provide an alternate way to view radiated field quantities. The transformation from theta, phi space to u, v sine space is defined as follows:

u = sin(theta)*cos(phi)

v = sin(theta)*sin(phi)

Note that this is similar to a polar coordinate system, where the radius is given by sin(theta) and the angle is given by phi. Another way to think about this: Given theta and phi that define a unit sphere, then these u,v coordinates are like viewing the unit sphere from above or below the XY plane. So, a sine space plot is a 2D plot that lies on a unit circle in the u,v space, that shows the values of radiated field quantities projected onto the upper or lower half of a full 3D sphere.

Sine Space Plots

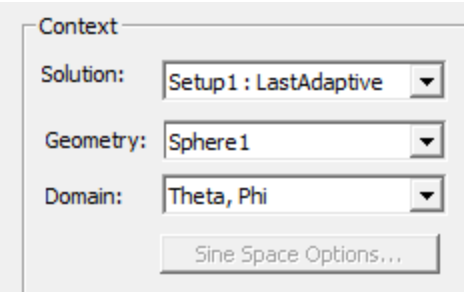
Sine Space plots are implemented as a new Context in Reporter. They will leverage the existing radiated field setups and calculations.

To create a Sine Space plot, after you have specified radiated field Setups.

1. Click **Results> Create Far Fields Report> Rectangular Contour Plot** or **Data Table**

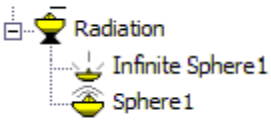
   Results> Create Near Fields Report> Rectangular Contour Plot or **Data Table**.

   The *Report* dialog box displays and shows choices for Domain in the Context field.



"Theta, Phi" is selected by default and gives the existing functionality.

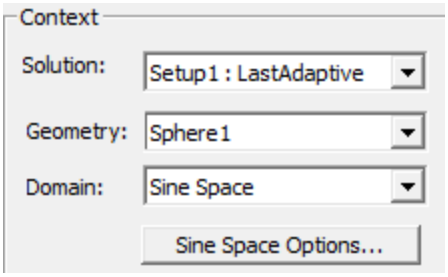
Notice that the Context shown above already includes selection of a Geometry, which corresponds to a Radiation setup.



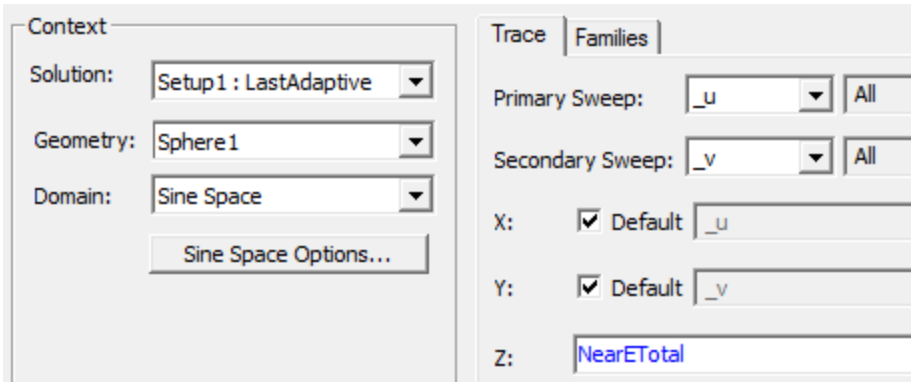
This radiation setup is used to calculate the fields for the Sine Space plot. Internally, this computes the values of the complex E vector based on the theta, phi sampling in the radiation setup. When data is requested for a sine space plot, it will correspond to some arbitrary theta, phi, and the fields will be interpolated to this location on the sphere. Note that this interpolation will occur on the complex E vector, and can be more detailed than a simple linear interpolation between sample points.

In order to support sine space plots, the radiation setup must have appropriate sampling in theta and phi. By default, a sphere is defined with phi from 0 to 360 (angle of rotation about Z), and theta from 0 to 180 (angle of rotation away from Z). These settings support a full unit circle representing the upper (theta from 0 to 90) or lower (theta from 90 to 180) half space. If the range of theta is reduced to 0 to 90, it would be possible to support upper half space only. Other ranges of theta and phi will likely create odd sine space plots, such as a partial circle, and these are not allowed.

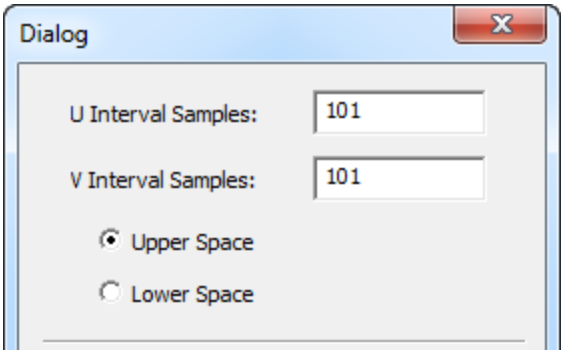
2. Select Sine Space as the Domain. This also enables the **Sine Space** options button.



Selecting "Sine Space" changes the Primary Sweep to U, and the Secondary Sweep to V.



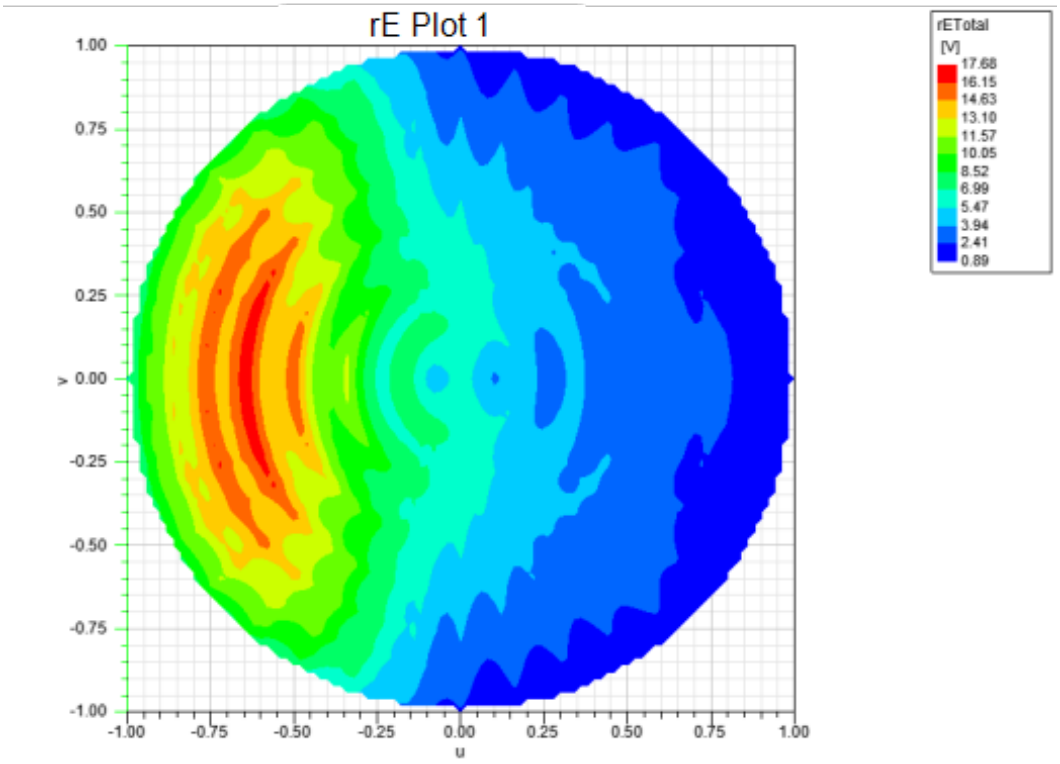
The Sine Space Options provide additional settings, including the sampling interval for the u,v coordinates (default of .05 would provide 40 samples over each axis, -1 to 1) and selection of upper or lower half space (default to upper).



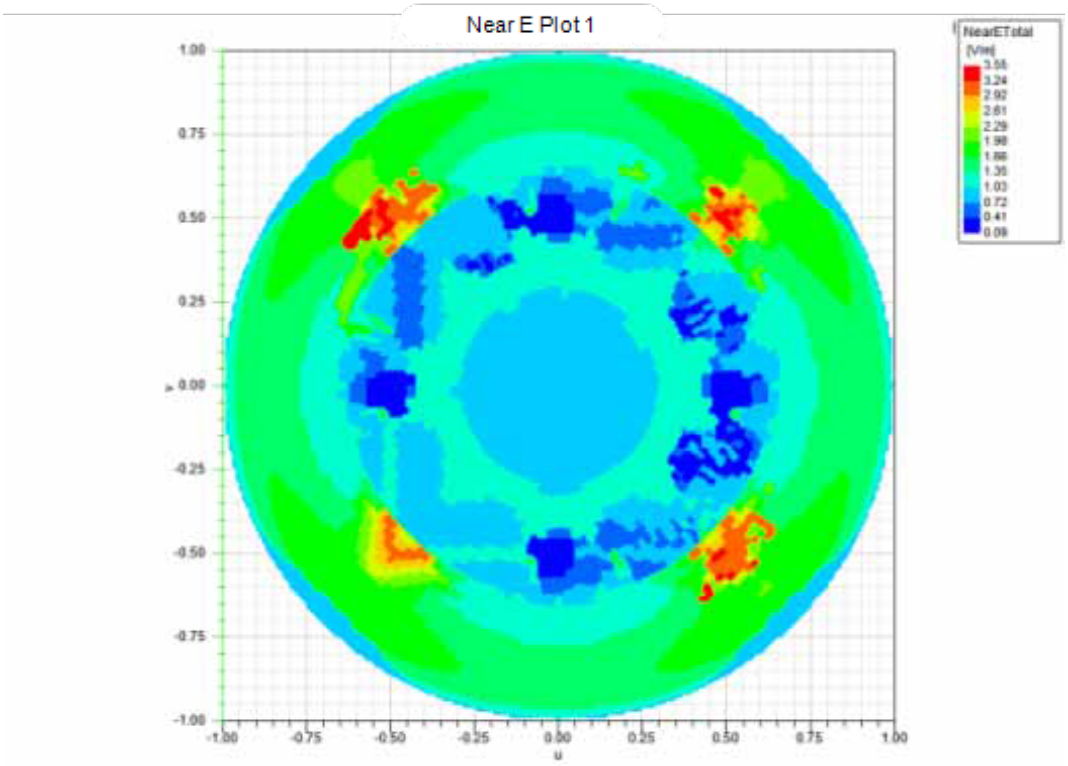
Once you have set up the domain, you define the Sine Space plot, selecting Category, Quantity and Function settings as appropriate. The Report Category you select provides the default report name, for example, S Parameter Plot *n* or Output Variables Plot *n*. Given

that the u,v sampling is set in the Sine Space Options, it should not be necessary to modify the sweeps in the Trace area of the dialog box.

Here is an example Sine Space plot of Far Field rETotal.



And here is an example Sine Space plot of Near field E Total.



You can use the **Contour** tab of the *Properties* dialog box to modify the appearance of the plot. See Creating Rectangular Contour Plots for discussion and examples.

Creating 2D Polar Plots

In Q3D Extractor, a polar plot is a 2D circular chart divided by the spherical coordinates R and theta, where R is the radius, or distance from the origin, and theta is the angle from the x-axis. Following is the general procedure for drawing a polar graph of results:

1. On the **Results** menu (Q3D Extractor menu or right-click **Results** on the Project tree), click **Create <*type*> Report**, and select **Polar 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.
    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. Whether this field appears, and the domains listed depend on the Solution type and the ***<type>*** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

c. 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. In the **Trace** tab **PolarComponent** area, specify the information to plot:
   a. On the **Category** drop-down menu, 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.
4. 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 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.

5. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Reviewing 2D Polar Plots

For a polar plot of S-parameters, HFSS displays in the lower-left corner the following derived information about the cursor's location:

MP The magnitude and phase of the point.

RX The normalized resistance (**R**) and reactance (**X**).

GB An alternate view of the normalized resistance and reactance in the form of

$$R + jX = \frac{1}{G + jB}$$

where

- ***G*** = conductance
- ***B*** = susceptance

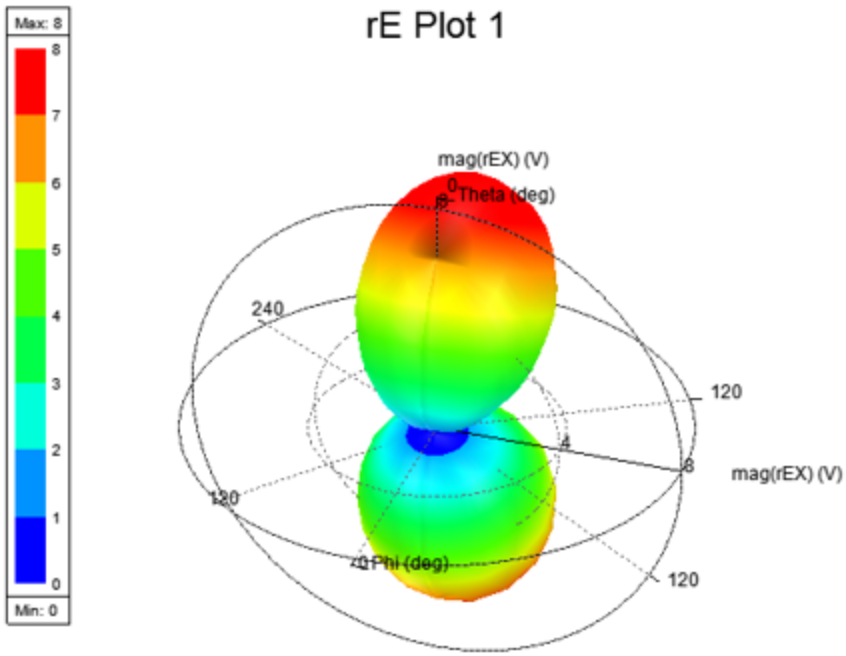
Q The quality factor.

VSWR The voltage standing wave ratio, calculated from the equation $\frac{1 + |S_{ij}|}{1 - |S_{ij}|}$.

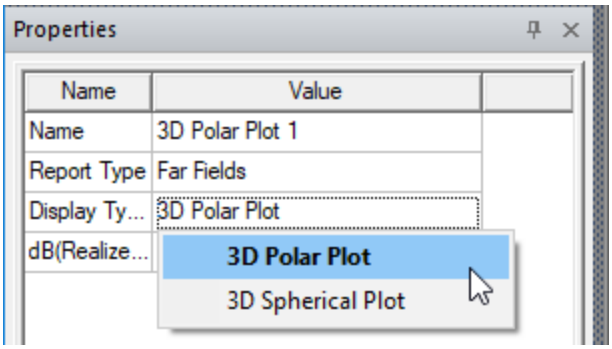
A scale below the plot displays the scale of points along the R-axis.

Creating 3D Polar Plots

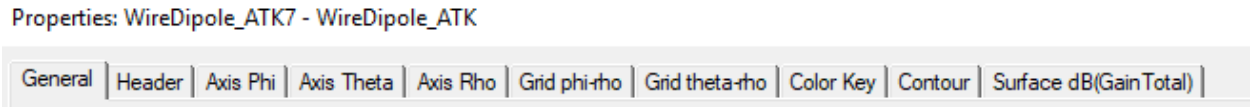
A 3D polar plot is a 3D circular chart divided by the spherical coordinates R, theta, and phi, where R is the radius, or distance from the origin, theta is the angle from the x-axis, and phi is the angle from the origin in the z direction.



To convert a 3D Polar plot to a 3D Spherical Plot, select the plot in the Project tree, and select from the Display type menu in the Properties window:



Double-clicking anywhere in the plot brings up the display **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

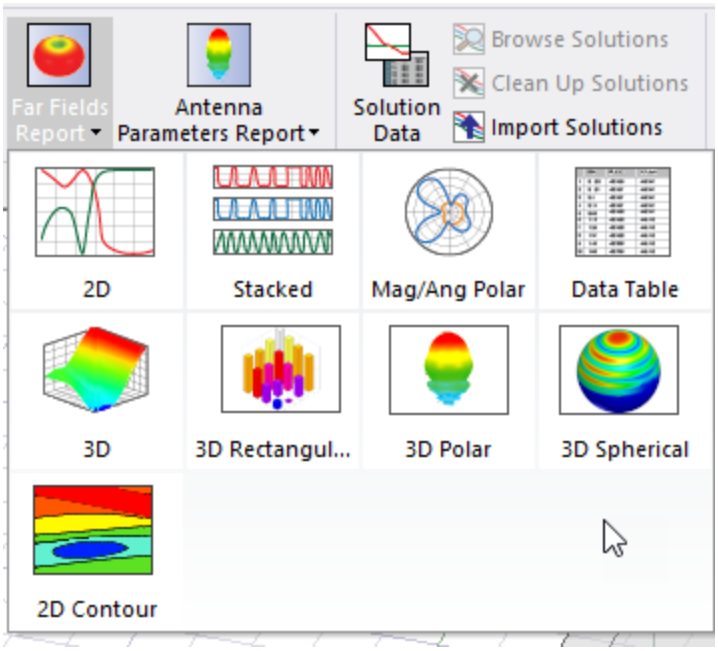
You can edit the properties on these tabs to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Polar Plot" on page 17-155 below.

Creating 3D Polar Plots

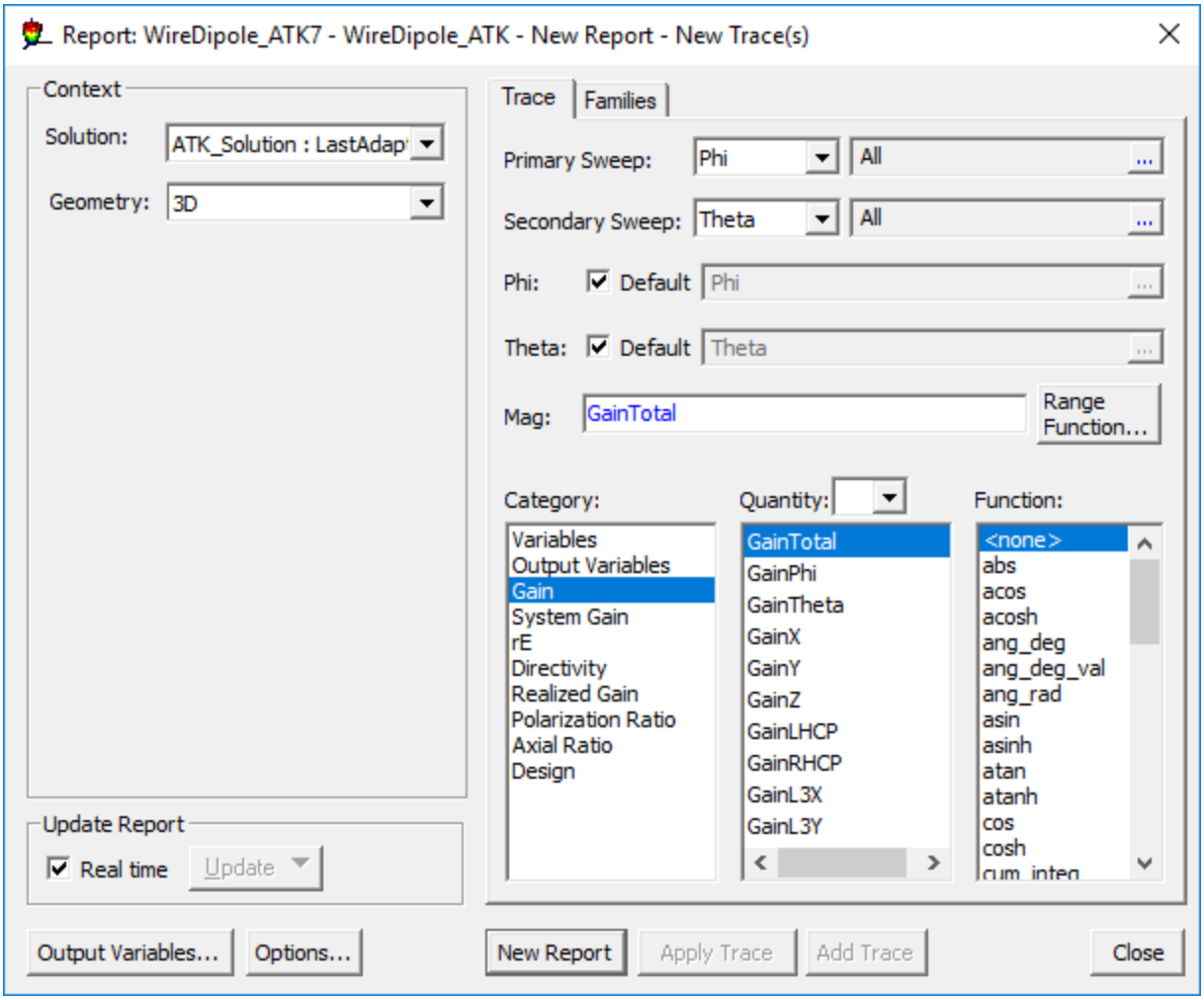
Following is the general procedure for drawing a 3D polar plot of results:

1. On the **Results** menu (HFSS menu or right-click **Results** on the Project tree), click **Create <*type*> Report**, and select **3D Polar Plot** from the report type menu.

You can also select 3D Polar Plots from the **Results** tab of the ribbon if such plots are appropriate for the current design.



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. In the **Trace** tab **Mag** area, specify the information to plot along the R-axis, or the axis measuring magnitude:
    a. From the **Category** drop-down menu, select the type of information to plot. The selected category provides the default plot name.
    b. In the **Quantity** list, click the values to plot. 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.

4. On the **Trace** tab **Theta (Secondary Sweep)** line, select the sweep variable from the drop-down menu and specify all values or select values to plot along the theta-axis.
5. On the **Trace** tab **Phi(Primary Sweep)** line, select the sweep variable from the drop-down menu, and specify all values or select values to plot along the phi-axis.
6. Click **New Report.**

   This button 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 R-, phi-, and theta-axes on a 3D polar 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

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

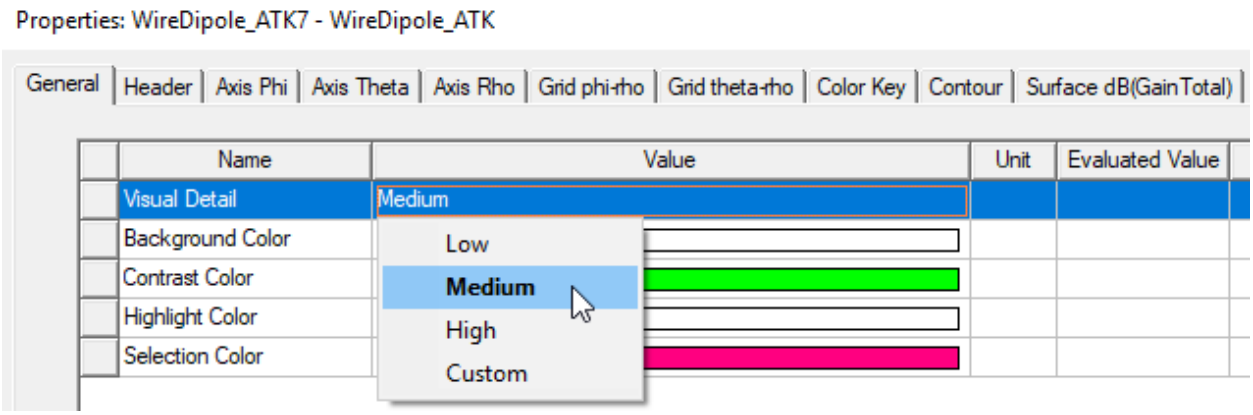
Interacting with 3D Polar Plots

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. Also, see Using the Orientation Gadget.

Controlling Visual Detail in a 3D Polar 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. Clicking on a plot entity selects it, highlighting the selected entity in bold.

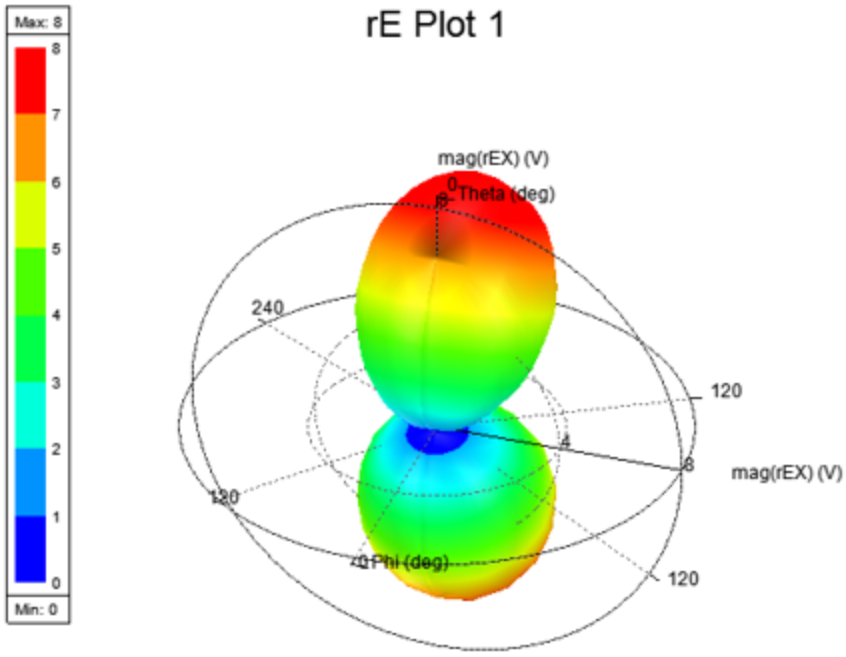
The Visual Detail property on the **General** tab also provides control suited to different screen and plot sizes.



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).

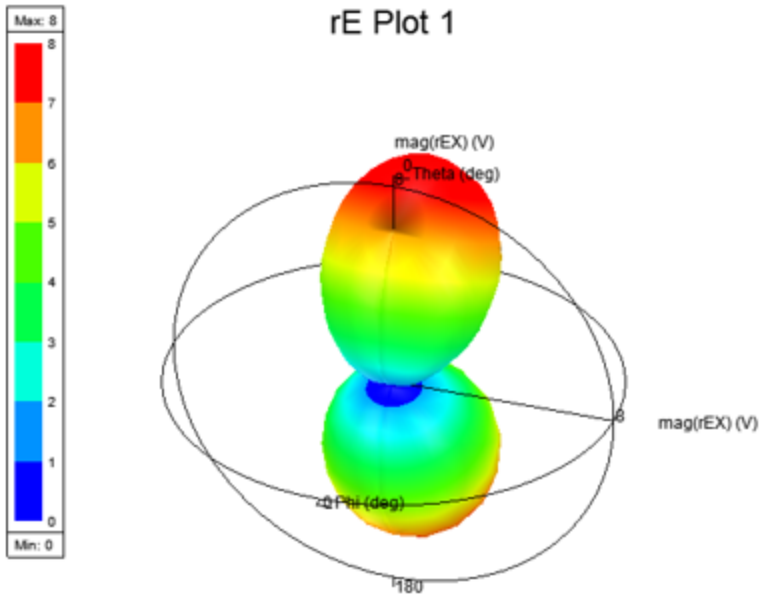
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Polar Plot with Medium Visual Detail



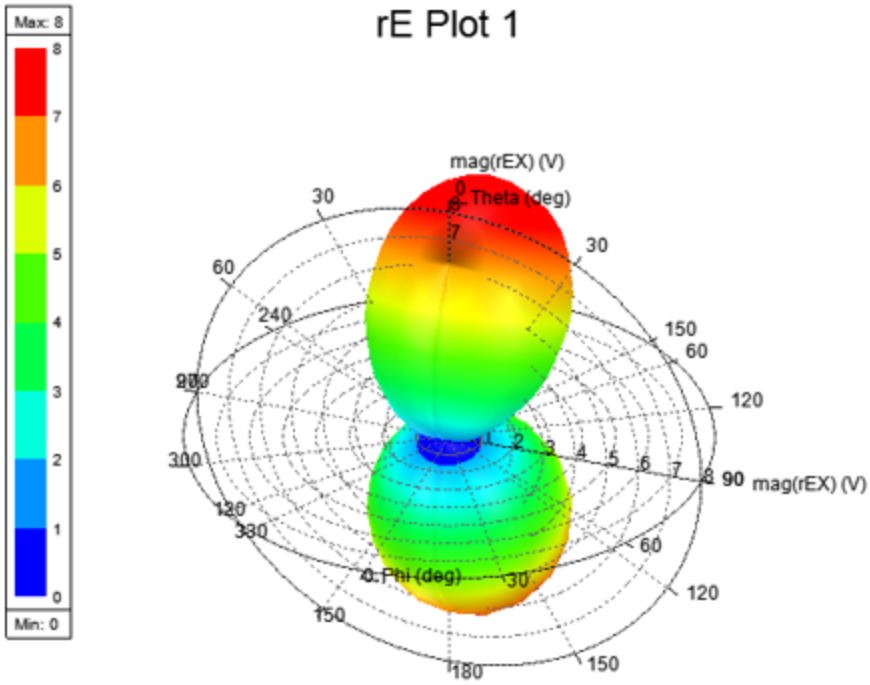
On creation, a 3D Polar Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under the Medium Visual Detail level, 3D Polar Plot has 3 ticks per axis (phi, theta and rho axis) which show min, max and middle value. This setting also shows axes labels.

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.

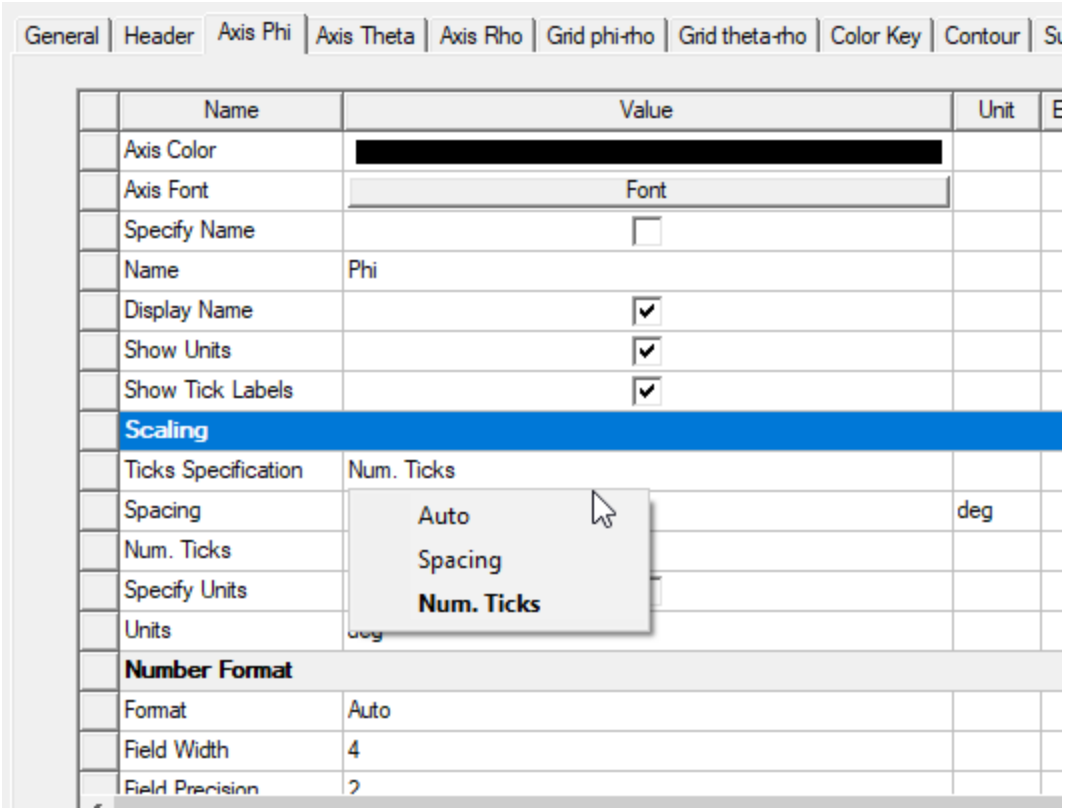
3D Polar Plot with High Visual Detail



With Visual Detail set to High, a 3D Polar Plot shows all polar axes and grids together with all nice 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:

--Scaling			
Scale	Linear		
Specify Min			
Min	0	deg	
Specify Max			
Max	375	deg	
Ticks Specification	Auto		
Spacing	125	deg	
Num. Ticks	2		

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** Unchecked, Ticks Specification is set to Auto.
> - If an R18.0 or R18.1 project is opened with **Specify Spacing** Checked, Ticks Specification is set to Spacing.

Grid Properties for Phi-Rho and Theta-Rho

You can control a range of display properties for the pho-rho and theta-rho grids, including line styles and colors for major and minor grids.

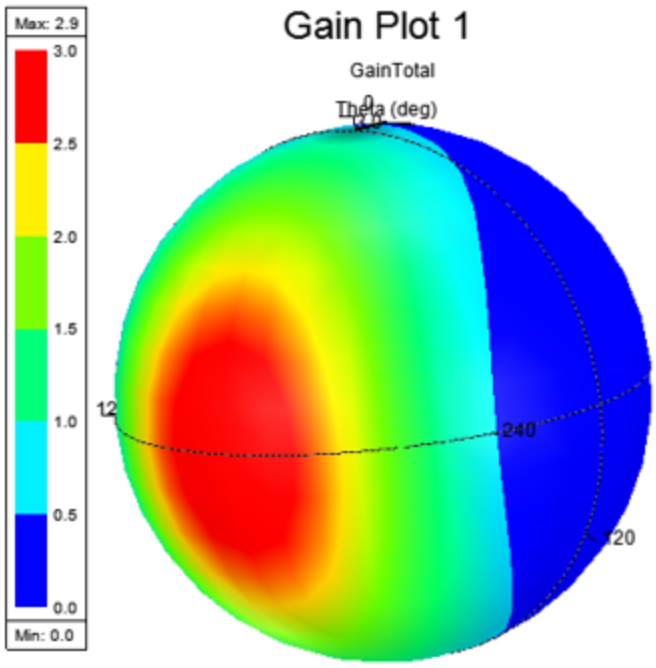
Overlay 3D Polar Plot on Model Window

For a 3D polar plot to be eligible for overlay, it must have its primary and secondary sweep from variables Phi and Theta or IWavePhi and IWave Theta in that order. If the plot is unsuitable, the **Overlay** commands are disabled.

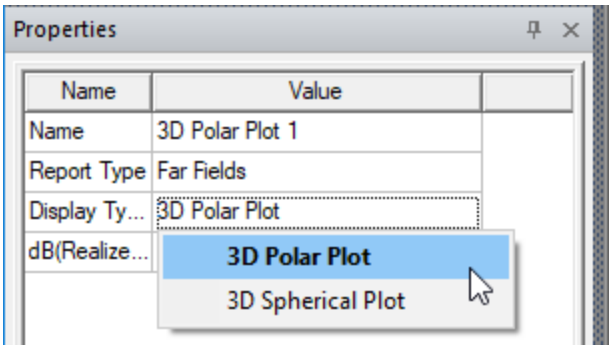
Once you create a suitable plot, you can overlay the 3D polar plot on the model window.

Creating 3D Spherical Plots

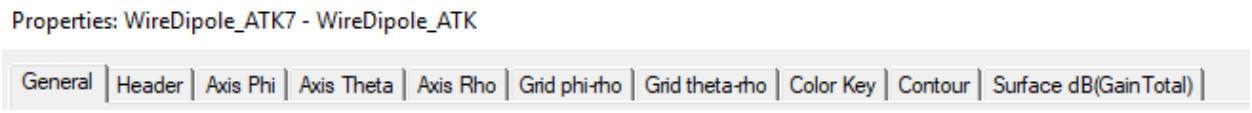
A 3D spherical plot resembles a 3D Polar Plot in all aspects except that the radius (rho) of the plot at all points is uniform and is equal to the maximum of all rhos at each (theta, phi) points.



To convert a 3D Spherical plot to a 3D Polar Plot, select the plot in the Project tree, and select from the Display type menu in the Properties window:



Double-clicking anywhere in the plot brings up the display **Properties** dialog box. The display 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

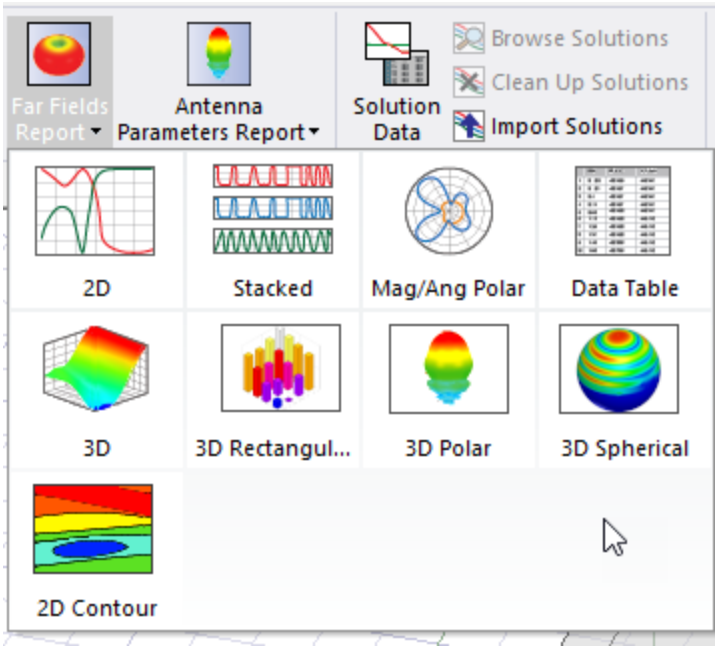
You can edit the properties on these tabs to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Spherical Plot" on page 17-164 below.

Creating 3D Polar Plots

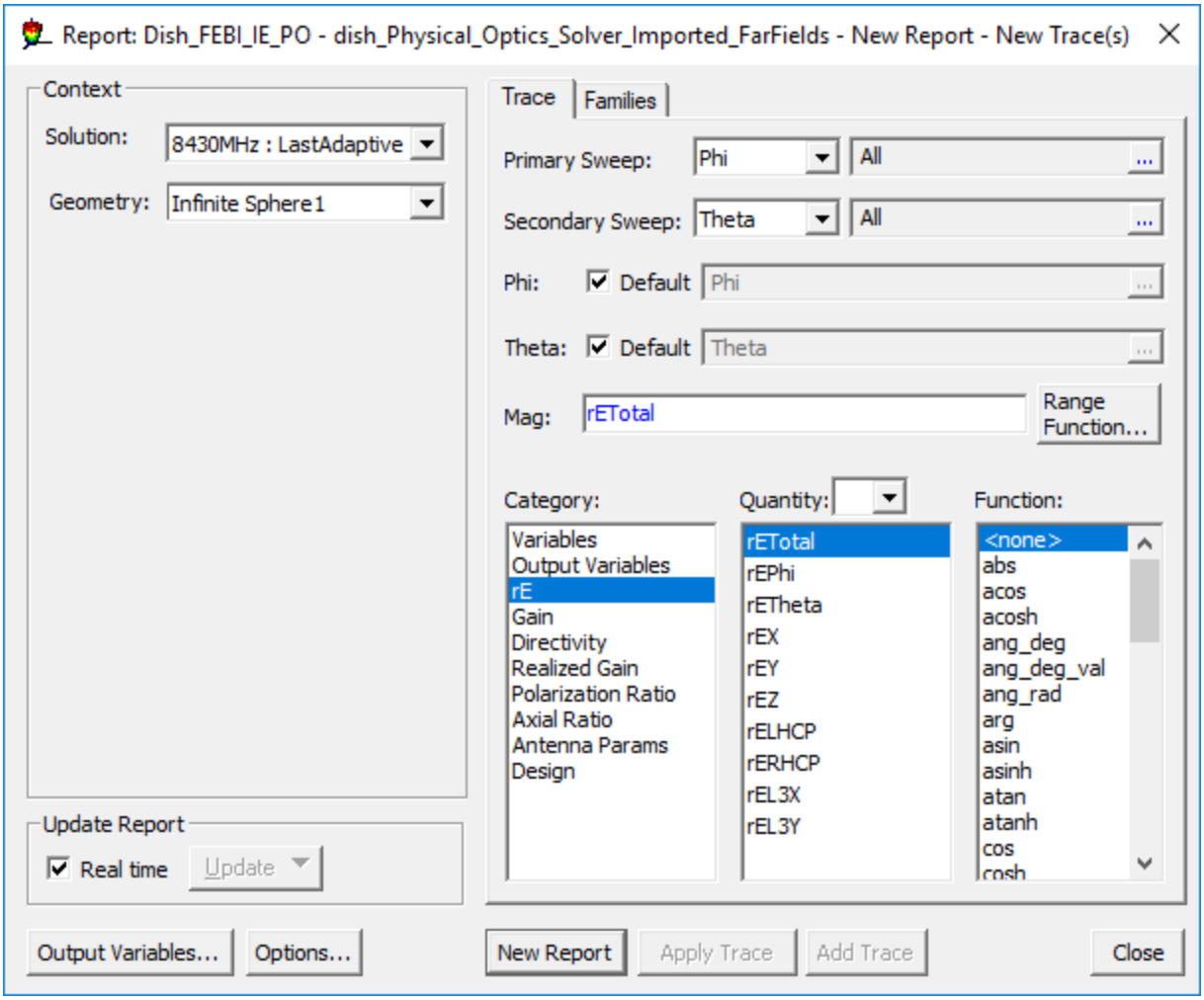
Following is the general procedure for drawing a 3D polar plot of results:

1. On the **Results** menu (HFSS menu or right-click **Results** on the Project tree), click **Create *<type>* Report**, and select **3D Spherical plot** from the report type menu.

   You can also select 3D Polar Plots from the **Results** tab of the ribbon if such plots are appropriate for the current design.



   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. In the **Trace** tab **Mag** area, specify the information to plot along the R-axis, or the axis measuring magnitude:
    a. On the **Category** drop-down menu, click the type of information to plot. The category you select provides the default plot name.
    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.

4. On the **Trace** tab **Theta(Secondary Sweep)** line, select the sweep variable from the drop-down menu and specify all values or select values to plot along the theta-axis.
5. On the **Trace** tab **Phi (Primary Sweep)** line, select the sweep variable from the drop-down menu, and specify all values or select values to plot along the phi-axis.
6. Click **New Report.**

   This button 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 R-, phi-, and theta-axes on a 3D polar 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.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

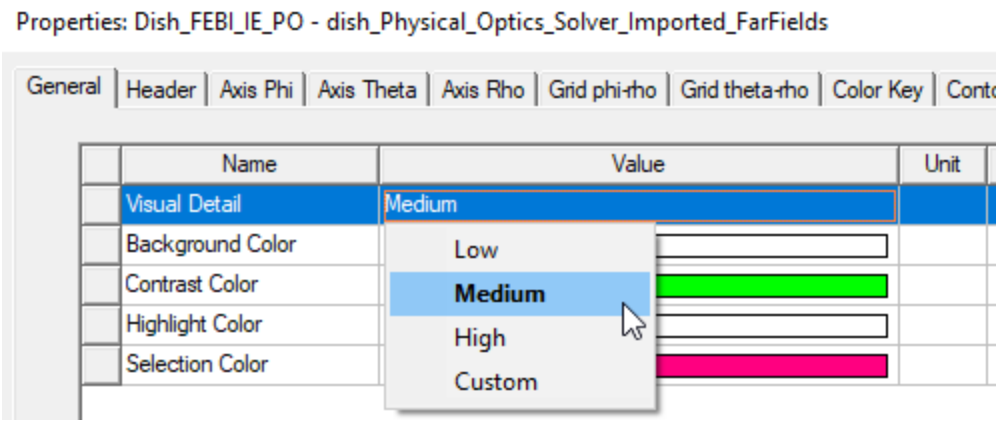
Interacting with 3D Spherical Plots

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. Also, see Using the Orientation Gadget.

Controlling Visual Detail in a 3D Spherical 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. Clicking on a plot entity selects it, highlighting the selected entity in bold.

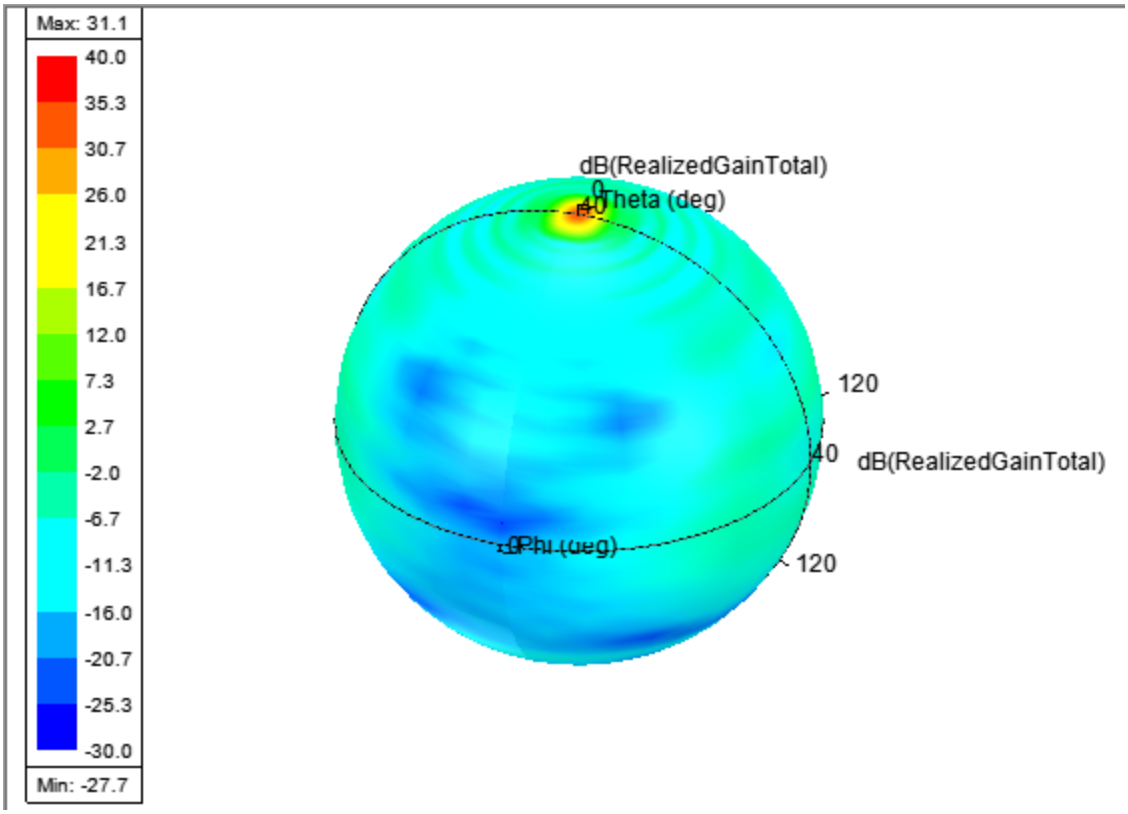
The Visual Detail property on the **General** tab also provides control suited to different screen and plot sizes.



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).

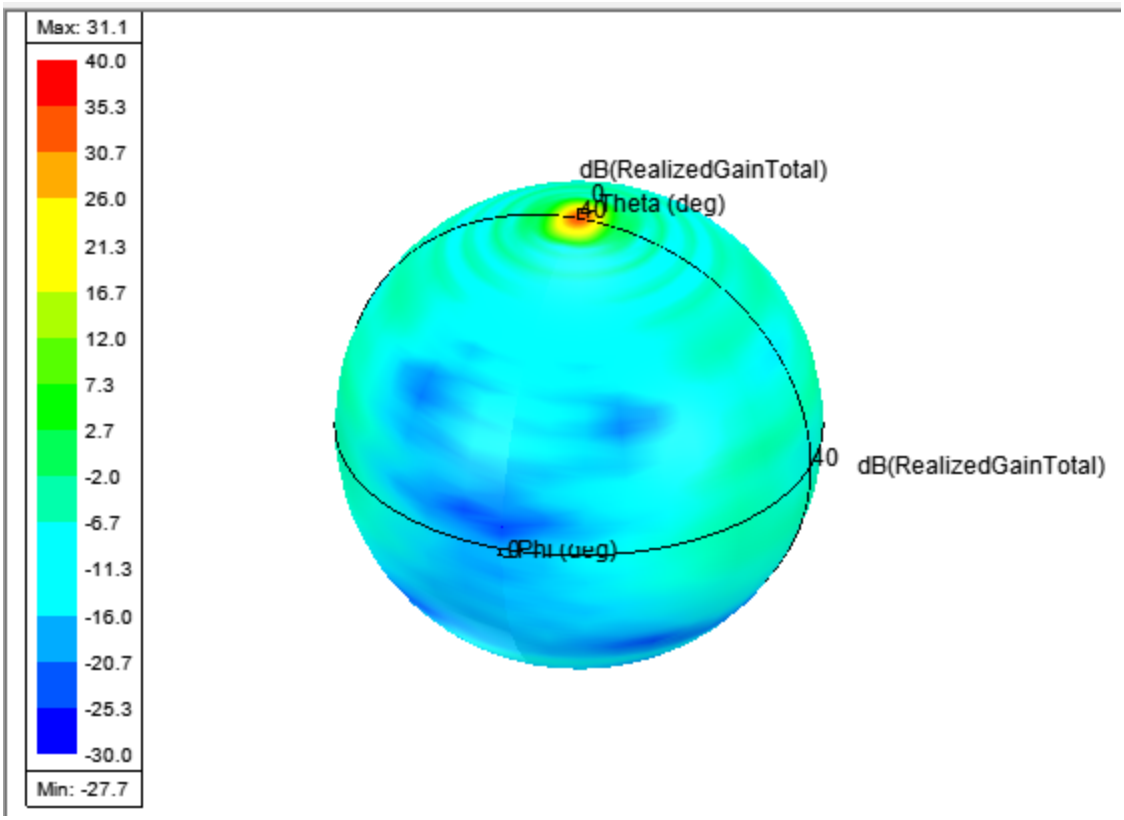
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Spherical Plot with Medium Visual Detail



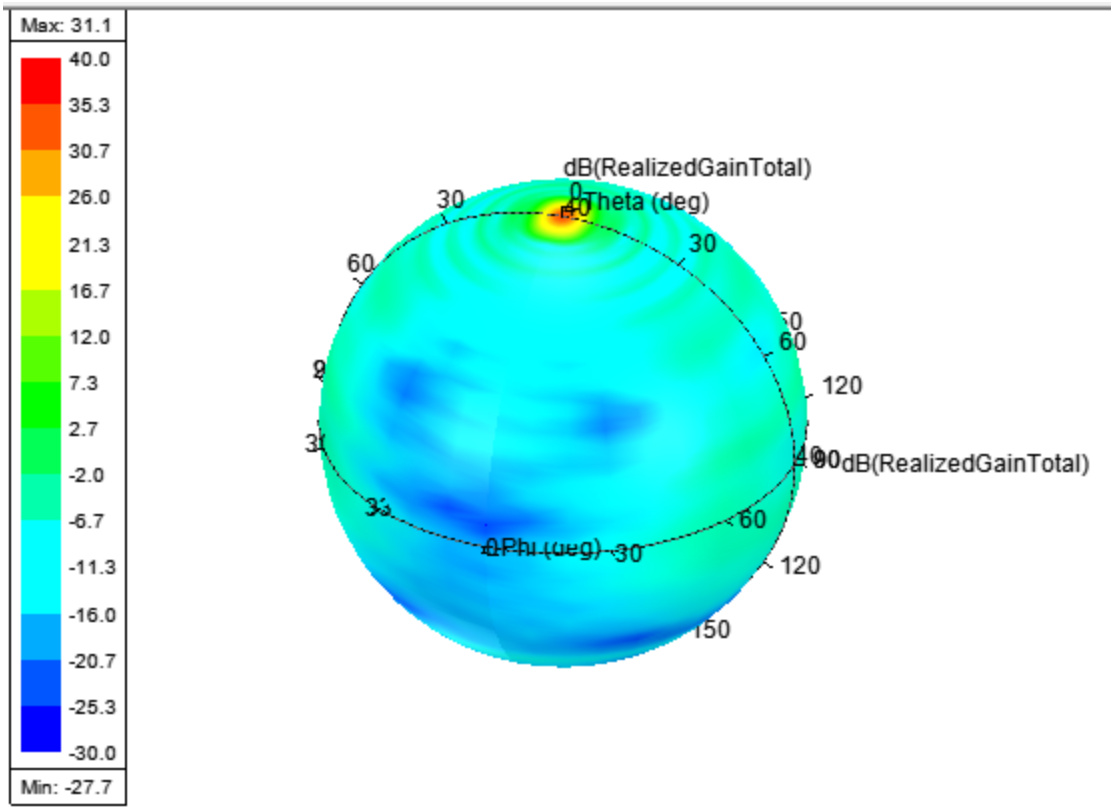
On creation, a 3D Polar Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under the Medium Visual Detail level, 3D Spherical Plot has 3 ticks per axis (phi, theta and rho axis) which show min, max and middle value. This setting also shows axes labels.

3D Spherical Plot with Low Visual Detail



With Visual Detail set to Low, 3D Spherical 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.

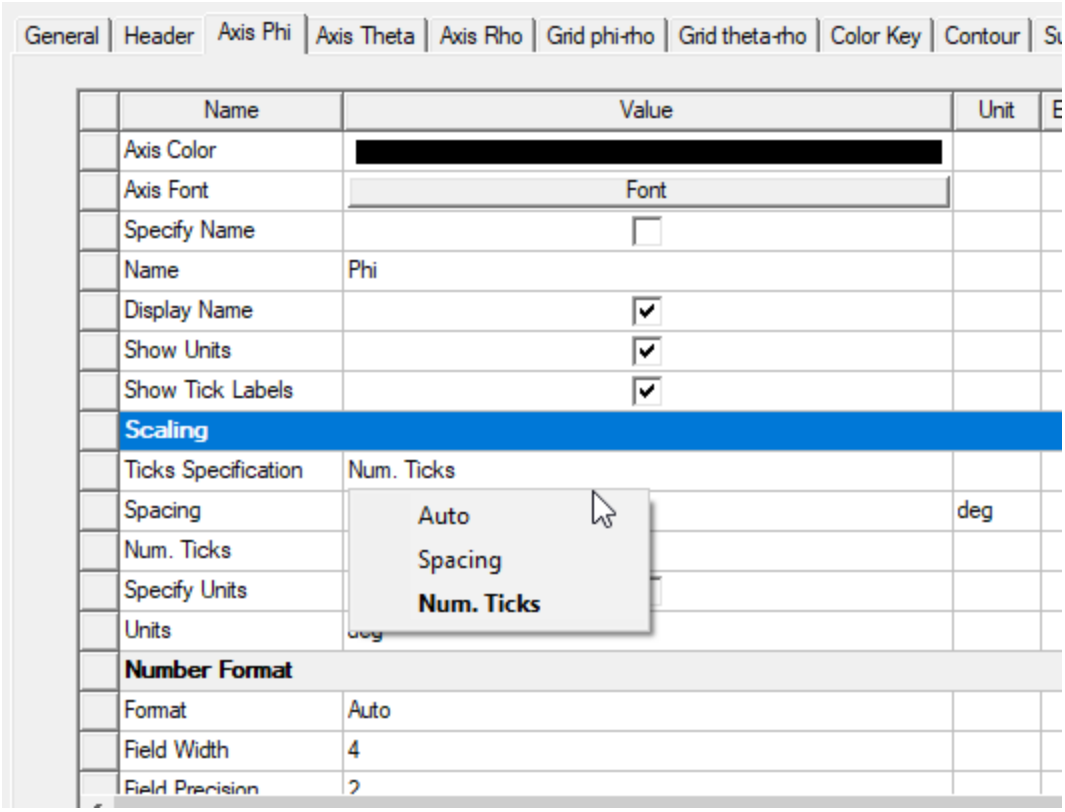
3D Spherical Plot with High Visual Detail



With Visual Detail set to High, a 3D Spherical Plot shows all polar axes and grids together with all nice 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:

--Scaling			
Scale	Linear		
Specify Min			
Min	0	deg	
Specify Max			
Max	375	deg	
Ticks Specification	Auto		
Spacing	125	deg	
Num. Ticks	2		

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** Unchecked, Ticks Specification is set to Auto.
> - If an R18.0 or R18.1 project is opened with **Specify Spacing** Checked, Ticks Specification is set to Spacing.

Grid Properties for Phi-Rho and Theta-Rho

You can control a range of display properties for the pho-rho and theta-rho grids, including line styles and colors for major and minor grids.

Overlay 3D Spherical Plot on Model Window

For a 3D Spherical plot to be eligible for overlay, it must have its primary and secondary sweep from variables Phi and Theta or IWavePhi and IWave Theta in that order. If the plot is unsuitable, the **Overlay** commands are disabled.

Once you create a suitable plot, you can overlay the 3D spherical plot on the model window.

Creating Smith Charts

A Smith chart is a 2D polar plot of S-parameters upon which a normalized impedance grid has been superimposed. Following is the general procedure for creating a Smith chart of results:

1. On the **Results** menu (HFSS menu or right-click **Results** on the Project tree), click **Create <*type*> Report**, and select **Smith Chart** from the report type menu.

   The **Report** dialog appears.

2. In the **Trace** tab **PolarComponent** area, specify the information to plot:

a. On the **Category** drop-down menu, click the type of information to plot. The category selected provides the default plot name.
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.

3. 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 will be plotted against the values you specified on a polar plot. In addition, each circle on the plot is labeled with values of *R*, measuring normalized resistance, and each line is labeled with values of *X*, measuring normalized reactance. 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.

4. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Creating Smith Contour Charts

A Smith contour chart is a polar plot of S-parameters upon which a normalized impedance grid has been superimposed. Following is the general procedure for creating a Smith chart of results:

1. On the **Results** menu (HFSS menu or right-click **Results** on the Project tree), click **Create <*type*> Report**, and select **Smith Chart** from the report type menu.

The **Report** dialog appears.

2. In the **Trace** tab **Mag** area, specify the information to plot:
   a. On the **Category** drop-down menu, click the type of information to plot. The Category selection provides the default plot name.

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.

3. On the **Trace** tab **(Secondary Sweep)** line,select the sweep variable from the drop-down menu and specify all values or select values to plot along the theta-axis:

To select an Secondary sweep component that is different from the default, uncheck the Default field to enable the X field and **Browse** [...] button. Click **Browse** [...] 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.

a. If sweeps are available, you can select **Browse** [...] to display a dialog that lets you select particular sweep or sweeps, or all sweeps.
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.

4. On the **Trace** tab **(Primary Sweep)** line,select the sweep variable from the drop-down menu, and specify all values or select values to plot along the phi-axis:

To select an X component that is different from the default, uncheck the Default field to enable the X field and browse [...] button. Click **Browse** [...] 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.

a. If sweeps are available, you can select **Browse** [...] to display a dialog that lets you select particular sweep or sweeps, or all sweeps.
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. 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 will be plotted against the values you specified on a polar plot. In addition, each circle on the plot is labeled with values of *R*, measuring normalized resistance, and each line is labeled with values of *X*, measuring normalized reactance. 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**.

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. Click **Q3D Extractor**> **Results**> **Create *<type>* Report** or right-click **Results** in the Project Manager 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.
   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. Whether this field appears, and the domains listed depend on the Solution type and the report ***<type>*** selected. For modal and terminal solution data reports, the domain can be **Sweep** or **Time**.

      Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal). If you select **Time**, the **TDR Options** button is enabled. Select it and follow the directions for time-domain plotting.

   c. 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.
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 menu, click the type of information to plot. The category selected provides the default name for the 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.

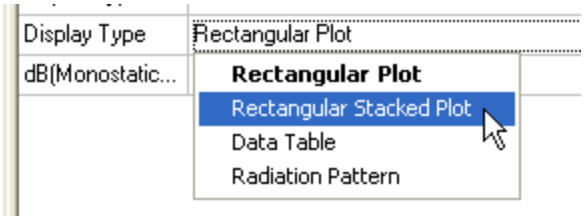
5. On the **Trace** tab **X (Primary sweep)** line, select the sweep variable from the drop-down menu and specify all values or select values.
6. Click **New Report**.

   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 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.

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.

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 scroll bar), printing first scrolls right, prints until no more scrolling and then scroll down.
- The Page number appears at the bottom of the page, aligned at center.
- The table layout of each page follows the screen, but with no scroll bar will be printed, and no data page bar as on screen.

Creating Radiation Patterns

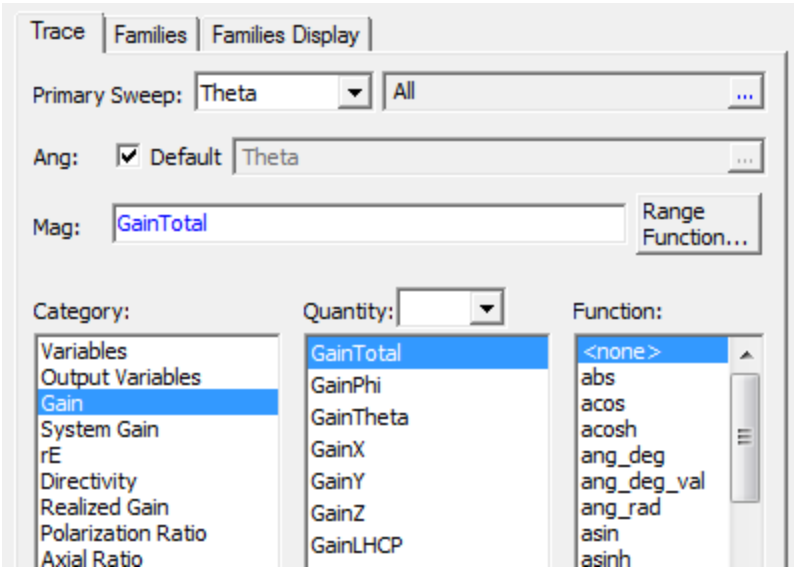
A radiation pattern is a 2D polar plot displaying the intensity of near- or far-field radiation patterns. It is divided by the spherical coordinates R and theta, where R is the radius, or distance from the origin, and theta is the angle from the x-axis. Following is the general procedure for drawing a radiation pattern of results:

1. Click **HFSS>Results>Create Far Fields Report>Radiation Pattern**, or right-click the **Results** icon in the **Project tree** and click **Create Far Fields Report>Radiation Pattern**.

The **Report** dialog box appears, and a Radiation Pattern Plots icon appears under **Results** in the Project tree.

3. 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.
4. In the **Trace** tab area, specify the Primary Sweep and Ang information to plot along the R-axis, or the axis measuring magnitude. The Primary Sweep field has a drop-down menu of current choices for Theta, Phi, and defined variables. Click the ellipsis button if you want to select from available values.

5. In the **Ang**, if you want to change the default, uncheck to enable the field and ellipsis button.



   a. On the **Category** drop-down menu, click the type of information to plot. The selected category also provides the default plot name.
   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 **Mag** 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.

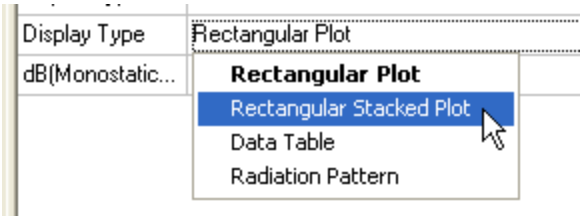
5. The **Families** tab is helpful if you want to use the plot as an overlay and you need to restrict the Phi (default) or Theta values to make the plot appropriate for overlay. See Overlaying 2D Radiation Field Plots on Models.
6. Click **New Report.**

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, System Gain Plot *n* or Directivity 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 a 2D polar plot. 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.

Delta Markers in 2D Reports

To view the difference between any two marker points in a report:

1. Set the first marker by clicking and holding the mouse button.
2. Move the mouse without releasing left button to another position, and then release the left button to create second marker.

In the marker text window, you see the difference between the two markers instead of the X, Y value of marker.

Plotting in the Time Domain

The idea behind Time-Domain Reflectometry (TDR) is to excite a structure with a step function, and inspect the reflections as a function of time. Before you can examine the time domain, you must perform an Interpolating sweep for a driven solution (Modal or Terminal or Transient). You can then select **Time** from the **Domain** list in the **Report** dialog box. You also need to specify the input signal, whether step or impulse.

With **Time** selected as the domain, you can select from several Categories and associated Quantities to plot, for example mag(S_{11}). When you plot in the Time domain, every frequency domain quantity is first converted to the time-domain before the formula is evaluated. For example, if you type in

$S_{11} / (1 - S_{11})$

and plot it in the time domain the reporter will plot

$IFFT(S_{11} * input) / (1 - IFFT(S_{11} * input))$

It will NOT plot

$IFFT(S_{11}/ (1 - S_{11}) * input)$

The two expressions are not equivalent.

If you select Time Domain Impedance as the Category, you can select the TDRZ quantity. This is defined as

$TDRZ(t) = Zref * (1 + IFFT(S_{11} * input)) / (1 - IFFT(S_{11} * input))$

where "input" denotes the Fourier transform of the input signal (step or impulse) and "IFFT(.)" denotes the inverse FFT.

This equation is the instantaneous ratio of the time-domain voltage v(t) to the time-domain current i(t). That is because voltage and current are defined (in the frequency domain) in terms of the incident and reflected waves a and b, respectively, as

$V = sqrt(Zo) * (a + b) = sqrt(Zo) * (1 + S_{ii}) * a$

$I = 1/sqrt(Zo) * (a - b) = 1/sqrt(Zo) * (1 - S_{ii}) * a$

This lets the incident wave be the input step signal, and so when we take the inverse FFT of V and I, we get v(t) and i(t) in the time domain. Taking their ratio as a function of time then yields TDRZ(t). By default, Zo is equal to 50 Ohm.

For HFSS, HFSS 3D or Layout Linear Network Analysis on Circuit, to create a plot in the **Time Domain**:

1. For a design with an existing sweep setup, follow steps 1 - 4 for creating a report for design.
2. In the **Report** dialog box, in the **Domain** list, click **Time**.

   This enables the **TDR Options** button and for terminal solution data reports includes the Terminal TDR Impedance in the Category list.

3. Click the **TDR Options button**.

   The **TDR Options** dialog box appears.

4. Select the input signal type, **Step** or **Impulse**.

   A **Step** describes a sustained change in the signal, whereas the **Impulse** is a brief excitation. **Impulse** is a very narrow rectangular pulse, with zero rise and fall time, width of 1 time step, and height of 1/(time step).

   Selecting **Step** enables the **Rise Time** field, and **Impulse** disables it.

5. If you selected **Step**, enter the rise time of the pulse in the **Rise Time** text box.

The rise time should be appropriate for the frequency context.

With a band width from DC to f_{max}, the best time resolution that can be achieved is $1/(2f_{max})$.

A rise time of $1/(2f_{max})$ is the shortest rise time that can be resolved. However, a rise time of 0 s gives equally valuable information, so 0 is the default in this panel. See the example plot.

6. Enter the total time on the plot in the **Maximum Plot Time** text box.

The default maximum plot time in the **TDR Options** dialog is related to the delta frequency df in the frequency sweep: it is 1/2df, since that is the extent of time for which the IFFT gives information. This is often very long relative to the time delay that corresponds to the length of your device under test, so you may want to reduce this value. Alternatively, you can adjust the time axis of your TDR plot after it has been created.

7. Set the number of time points to plot in the **Delta Time** text box. By default, this is set to the number of points in the frequency sweep.

The delta time is based on the bandwidth of the sweep: with a frequency sweep from DC to f_{max}, the smallest time resolution you can obtain is given by $1/(2f_{max})$. The IFFT algorithm provides data points as a spacing of $1/(2f_{max})$, but you can smoothly interpolate between points by setting a finer resolution, e.g., $1/(10f_{max})$, at the expense of extra computation time.

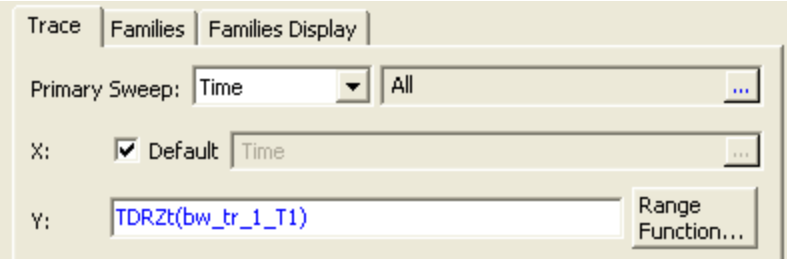
8. Optionally, under **TDR Window**, modify the window type and width.
9. You can use the **Save as Default** to set the current values as a default, and the **Use Defaults** button to use previously saved options. Note that when you select a trace, the initial displayed values are those of the selected trace.
10. Click **OK**.

Optionally, to plot Terminal TDR impedance (that is, rather than calculate the S-parameter for waveport1 versus frequency, instead calculate the delay versus time at a particular impedance), do the following:

a. In the **Category** list, click **Terminal TDR Impedance**.
b. In the **Quantity** list, click a quantity to plot.

The default impedance (Zo) for the TDRZ quantity is 50 Ohms, unless you specified differently when you Set Renormalizing Impedance for Terminals when you created the terminals in the model. If you need a different impedance value, you can either edit the value in the **Report** dialog (as shown below), or you can create an Output Variable representing $Zo \times (1+S_{ii})/(1-S_{ii})$ with the Zo of your choice. To edit the Zo value in the **Report** dialog:

1. For the Category, select Terminal TDR Impedance, and the Port and Function of interest.



2. Edit the value by placing the cursor in the Value field.

   In this example, the value for Zo is changed from the default to 75 Ohms by typing ',Zo=75ohm' in the Y-column field.

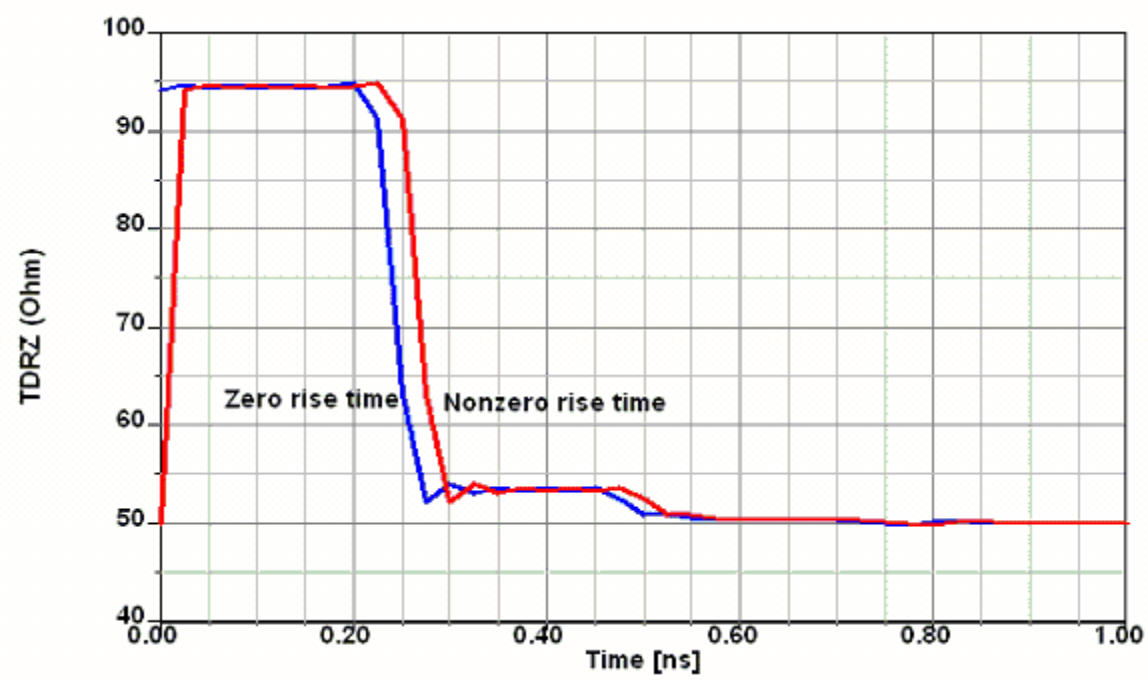
   c. In the **Function** list, click the mathematical function of the quantity to plot.

3. Click **Done**.

The report appears in the view window. It will be listed in the project tree.

If S_{11} = 0 at DC, the time-domain step response will settle to zero and the TDRZ step response settles to Zref. If S_{11} is nonzero at DC, the time-domain step response will settle to a nonzero value and TDRZ will settle to a value different from Zref. The time-domain impulse response will always settle to zero, since it can be seen as the derivative of the step response. The TDRZ impulse response will always settle to Zref.

The plot below shows the difference between a short nonzero rise time and zero rise time for a transmission line segment of 94 Ohm. Note that the trace with zero rise time starts at the correct line impedance while the other starts at the renormalizing impedance. Other than that, one trace is a shifted version of the other. The reason the plot with finite rise time starts at 50 ohms is that the time-domain voltage and current are still at their steady state values, so v = Zref * i. As the pulse arrives, the TDRZ response changes from the steady-state behavior because there's a reflection from the transmission line back to the exciting source, which has a different renormalizing impedance from the characteristic impedance of the transmission line.



Some things to keep in mind with TDR:

$$Spatial resolution \Delta x = c/(2B) \quad (1)$$

where c is the speed of light in the medium and B is the bandwidth of the signal. Since TDR is usually based on a frequency band that starts at DC, the spatial resolution becomes

$$\Delta x = c/(2F_{max}) \quad (2)$$

where F_{max} is the highest frequency in the frequency sweep. For example, if F_{max} = 15 GHz and the medium has ε_r=4, the spatial resolution will be (1.5E8 m/s)/(3E10 /s) = 5 mm.

A spatial resolution of $c/(2F_{max})$ corresponds to a resolution in time

$$\Delta t = 1/(2F_{max}) \quad (3)$$

Let N be the number of points in the IFFT. N equals the number of time samples, and it also equals twice the number of frequency samples. The density of frequency samples in the frequency sweep influences the total time T as follows:

$$2F_{max}/(\Delta f) = N(\text{number of points in IFFT}) = T/\Delta t \quad (4)$$

So increasing the density of the frequency samples leads to an increase in total time T. In practical case, this often leads to a long tail in the TDR plot with little useful information. Therefore, the TDR Options interface lets you set the maximum plot time to a smaller value.

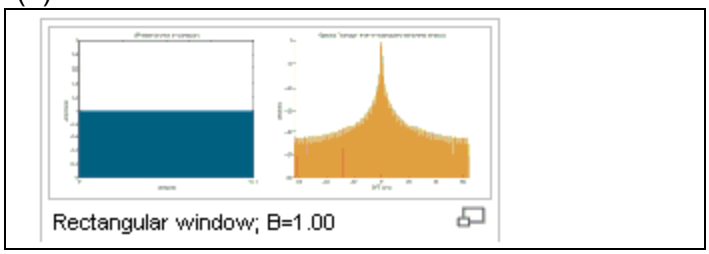
The TDR Options interface also lets you choose a smaller Δt than given by equation (3) above. When you choose a smaller Δt, you increase F_{max} by "zero padding" (adding zero values for S_{11} beyond the calculated frequency sweep). Whether this is justified depends on your judgment. It leads in practice to a smoother TDR signal.

HFSS also lets you set the rise time of your input signal. The rise time should be at least $1/(2F_{max})$. Even this rise time is a bit short for comfort, as it equals the duration of only one time sample. An input signal with a longer rise time has a smaller high-frequency content and will lead to reduced "ringing" in the TDR response.

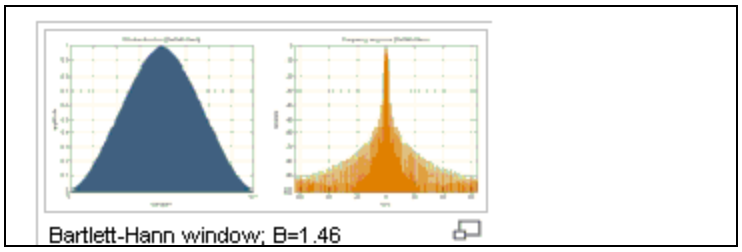
A Hamming or Hann filter will also reduce the high-frequency content and tends to lead to a smoother TDR response. With these filters, one can select a width. A width of 100% is often a good choice.

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 the following:

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 applications. $w(n) = a_0 - a_1\left\lvert\frac{n}{N-1} - \frac{1}{2}\right\rvert - a_2\cos\left(\frac{2\pi n}{N-1}\right)$

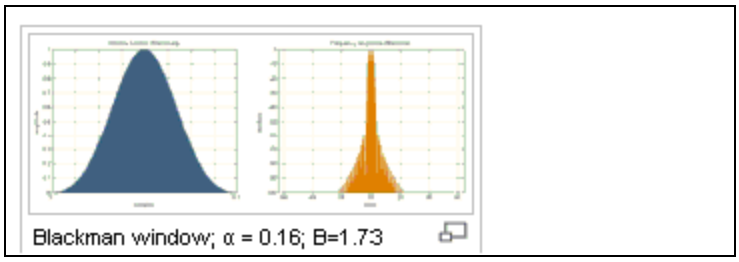
Window Function	Preferred Use
	where a_0=0.62; a_1=0.48; a_2=0.38



Blackman — A high dynamic range function, with lower resolution, designed for wide band applications.

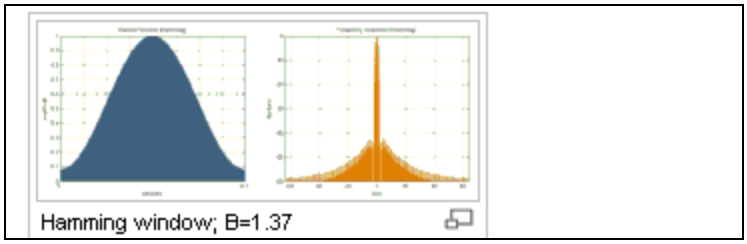
$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right)$$

where a_0 =(1-α)/2; α_1=1/2; α_2=α/2



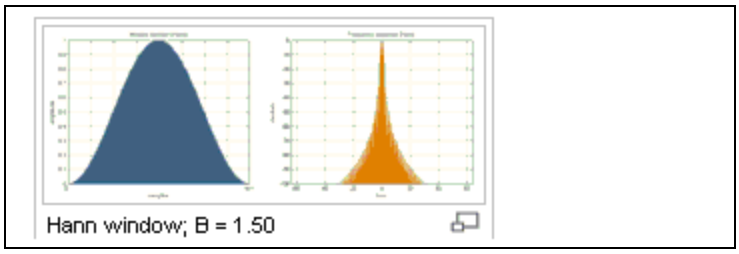
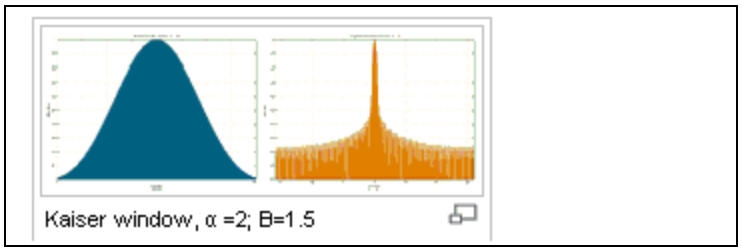
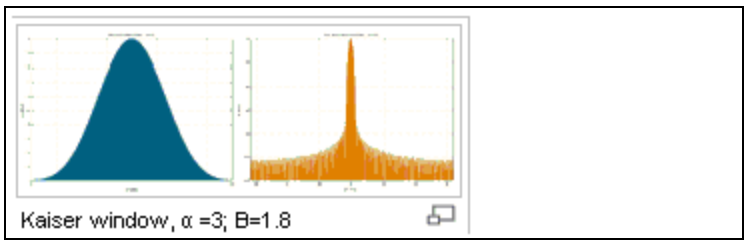
Hamming — A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$



Hanning (default) — A moderate dynamic range function, designed for narrow band applications.

$$w(n) = 0.5\left(1 - \cos\frac{2\pi n}{N-1}\right)$$

Window Function	Preferred Use
	
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 parameter controls the trade off between width of the central lobe and the area of the side lobes.  
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.

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 rectangular graphs, 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 Linux 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 box, 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 the currently specified Quantity and Function.
3. In the *Report* dialog box, 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. Selecting a trace in the report or legend displays the display 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 Manager.

By the 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 respective Properties windows or edited in the *Report* dialog box. To change the name or definition of a trace, see Editing Trace Properties. To edit other display properties of a trace, see Editing the Display Properties of Traces

Editing Trace Properties

To edit trace properties such as the name, 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 Manager.

   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 Manager.
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 twenty independent Y axes can be added to a plot.

To edit a **trace component definition**:

1. Select the trace in the Project Manager.
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 edit Component field window from which you can edit the category, quantity and function.

3. Click **OK** to apply the changes and close the dialog box.

To edit a **trace Context**:

1. Select the trace in the Project Manager 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. Select the trace in the Project Manager to display the docked properties window.
2. Under the -Variables category, on the Families line, click the Edit button to display the Edit families dialog box.

   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.

Editing the Display Properties of Traces

Editing the display properties of traces differs for 2D and 3D reports. To edit the display properties of a trace for a 2D report:

1. Select a trace in an open **Report** window.
2. Click once on the trace to view settings in the docked *Properties* window, or double-click to open *Properties* dialog box.

   The display properties window for a 2D trace includes a **General** tab and an **Attributes** tab.

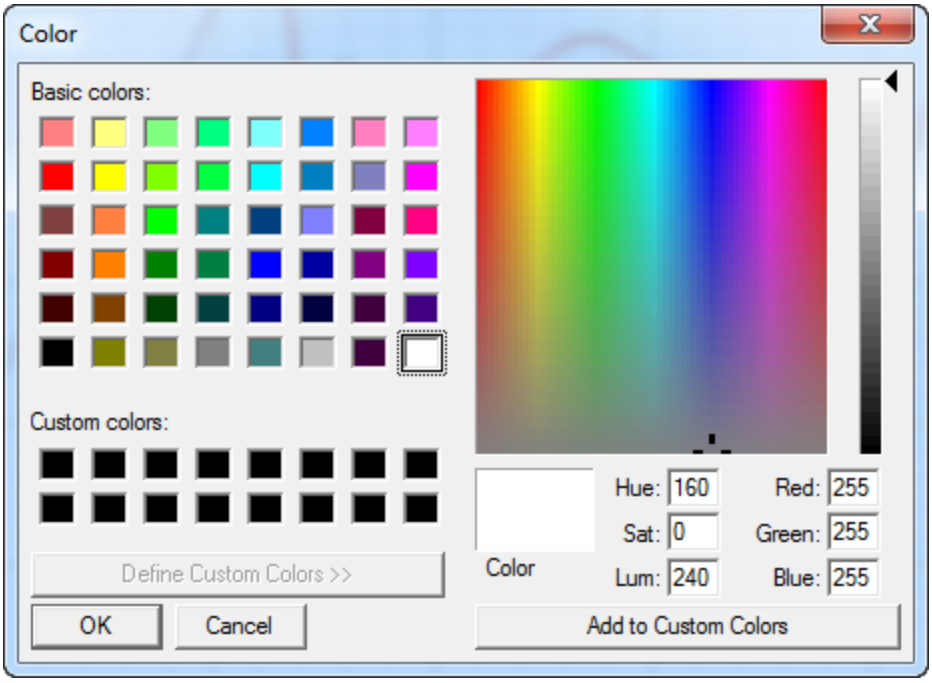
   The *General* tab properties apply to the general appearance of the plot. They include the Background color, Contrast color, Field width, and Whether to use Scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.)

   The *Attributes* tab properties apply specifically to the trace. The defaults are set in the Report2D options. They include:

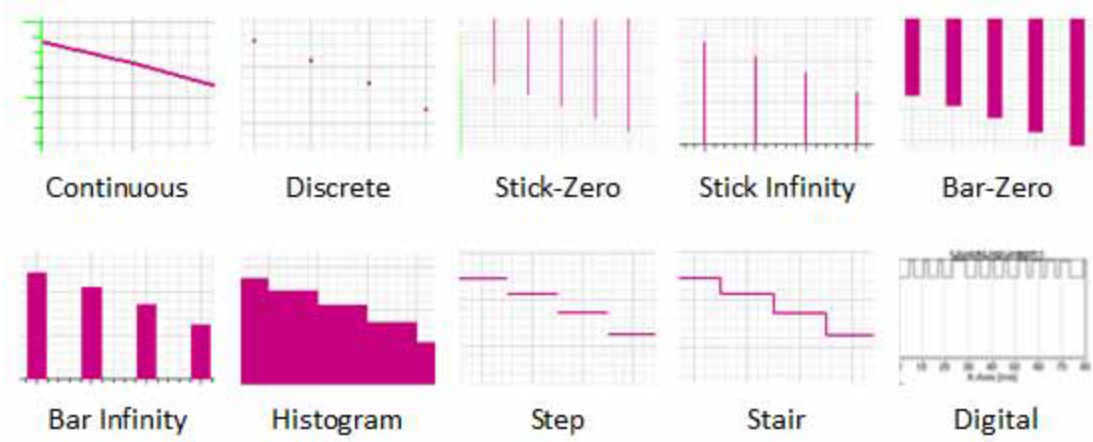
   - Name -- not editable by selecting the trace from the Report. It shows the characteristics of the trace as defined in the *Report* dialog box.

     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 values for Hue, Saturation, Luminescence, and the Red, Green, and Blue.

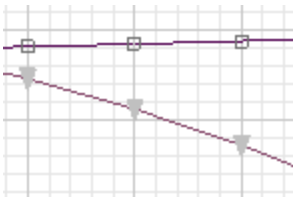


- Line style -- a drop-down menu lets you select Solid, Dot, Dash, or Dot-dash.
- Line width -- a text field lets you edit the numeric value.
- Trace type -- the drop-down menu contains entries for Continuous, Discrete, Bar-Zero, Bar Infinity, Stick Zero, Stick Infinity, Histogram, Step, and Stair.



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.

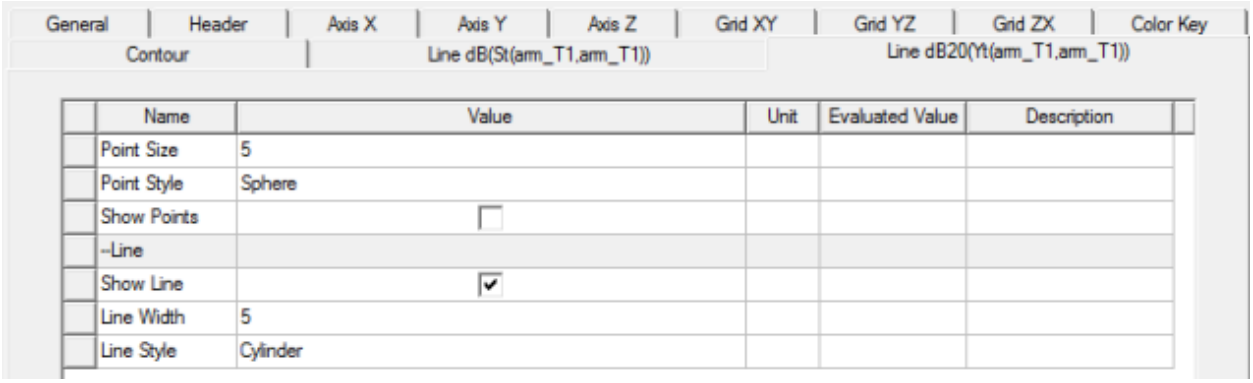
> **Note:**
>
> So that curves with single points always appear, Box is the default symbol.

3. Edit the properties, if needed. Click OK to apply the changes and close the window.

To edit the display properties of a trace in a 3D report:

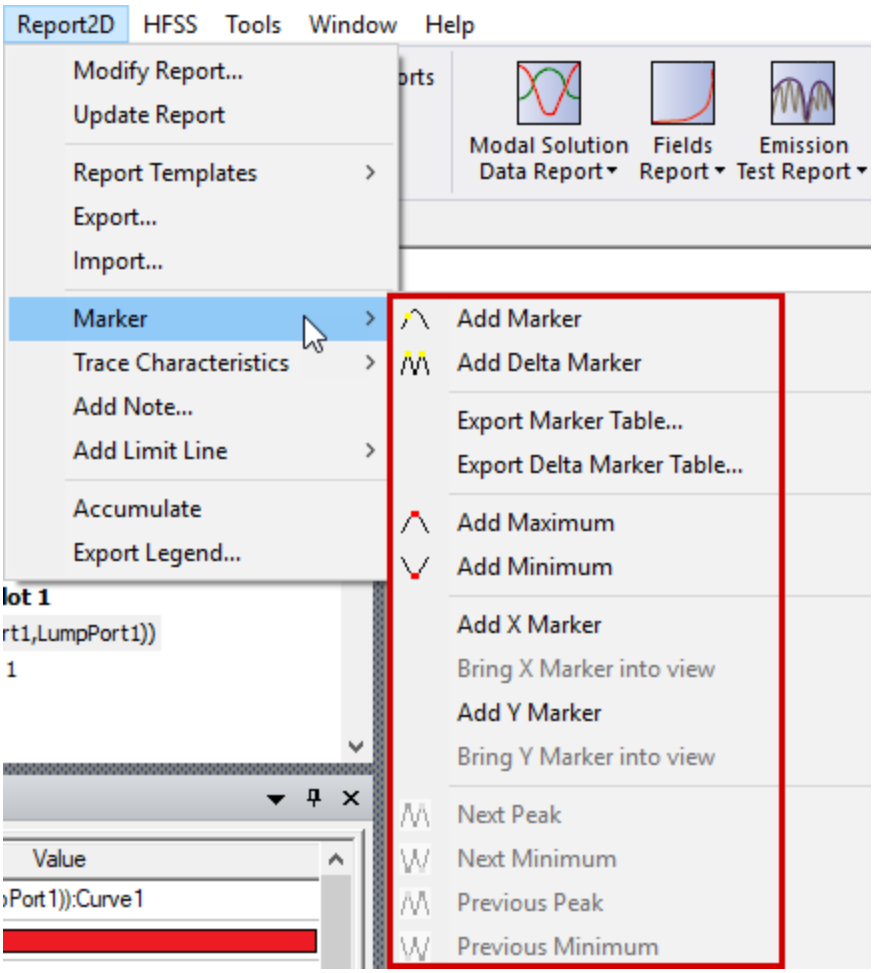
1. Click on the trace. This opens a Properties dialog for the plot with a tab named for the trace selected.
2. The editable properties include Point size, Point Style, whether to show points, whether to

show line, line width, and line style.



Adding Data Markers to Traces

The Reporter includes **Report 2D** > **Marker** > menu commands and icons:



You can also access these commands from the shortcut menu that appears when you right-click inside a report plot window.

These command 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:

- **Marker**  – this command lets you place a marker at an arbitrary point on a 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. 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 flag on the vertical line identifies the Y coordinate on the trace. A trace property lets you lock the drag feature to leave the marker in place. 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 Y Markers with a horizontal line extending from the Y axis. 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.
- **Maximum**  – places a marker at the Maximum value on the selected trace.
- **Minimum**  – places a marker at the Minimum value on the selected trace.
- **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

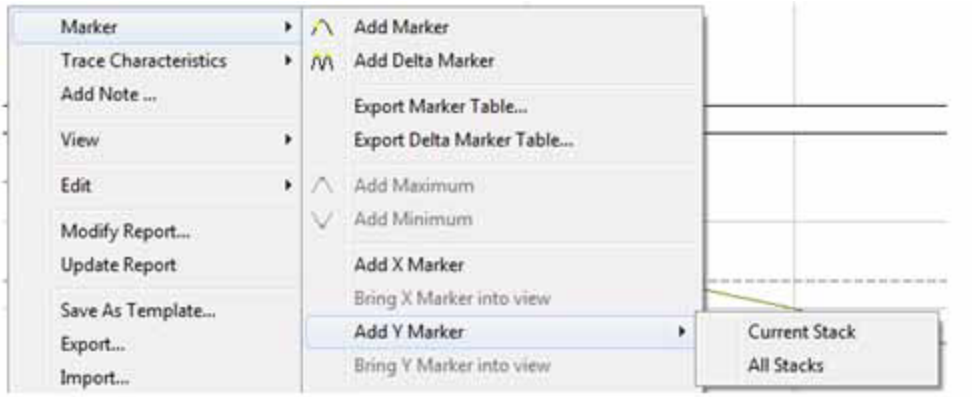
- **Next Peak** – moves a selected marker on the next peak on a trace. You must exit marker mode and select a marker to enable this command.
- **Next Minimum** – moves a selected marker to the next minimum on a selected trace. You must exit marker mode and select a marker to enable this command.
- **Previous 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.

Y Markers in Stacked XY Plots

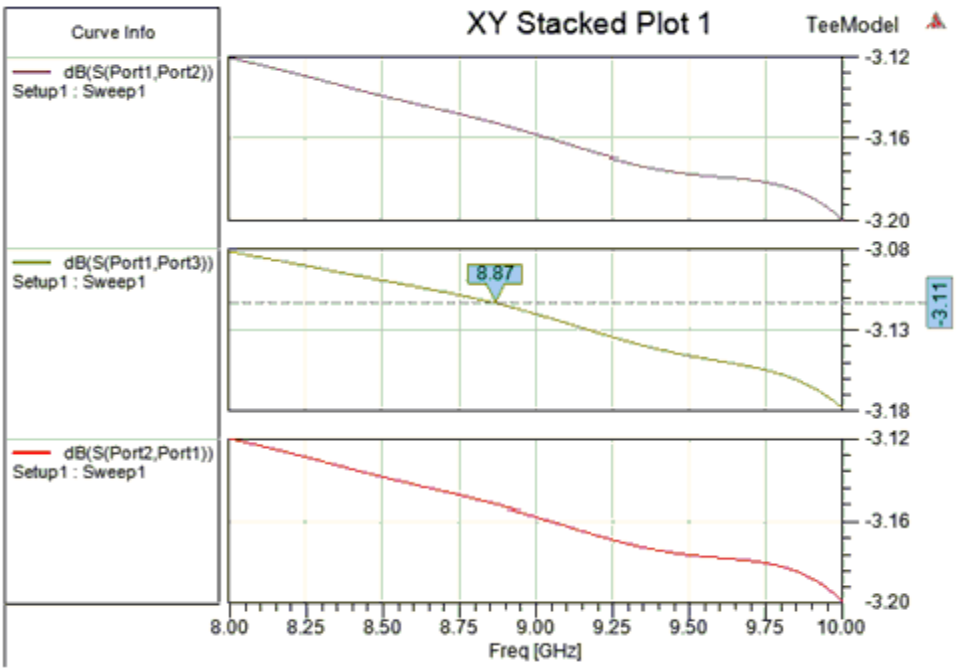
Y Markers allows for easy analysis and comparison of curves at a particular y-coordinate. Y Markers can be used to compare stacked curves.

Creating Y Markers in Stacked Plots

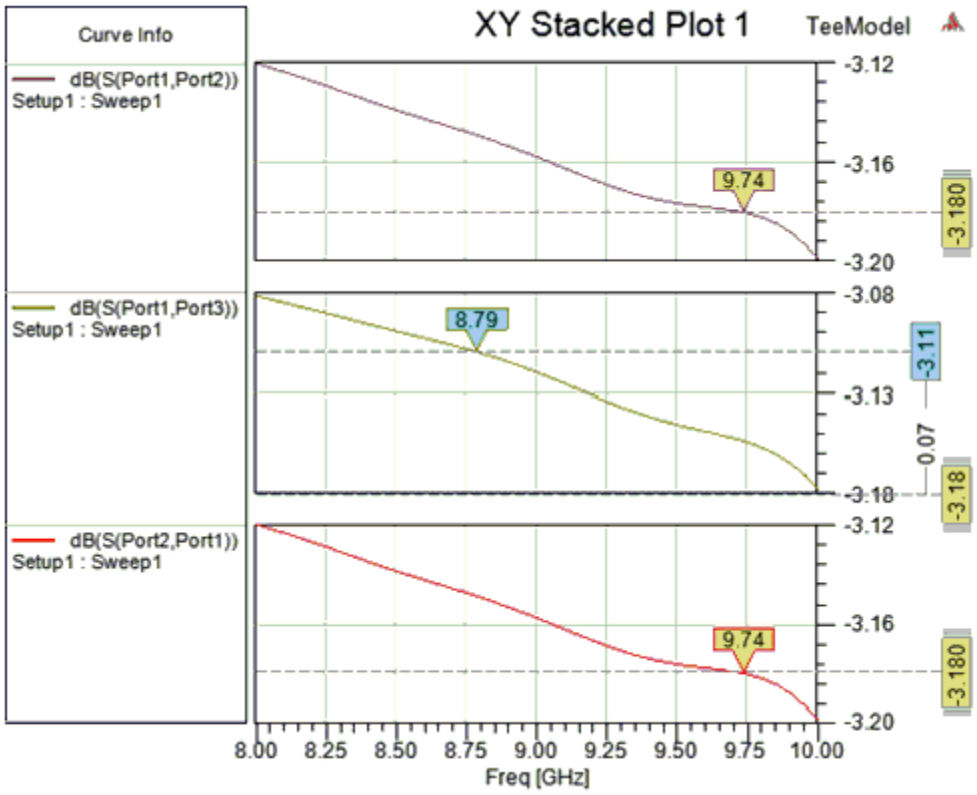
There are two ways to create Y Markers in Stacked Plots. You can create a Y Marker for a particular stack or for all stacks. Right-clicking on any stack shows the following shortcut menu:



Add Y Marker > **Current Stack** creates a Y marker for the stack on which user performed right mouse button click. The following figure shows that a Y Marker was added to second stack only:



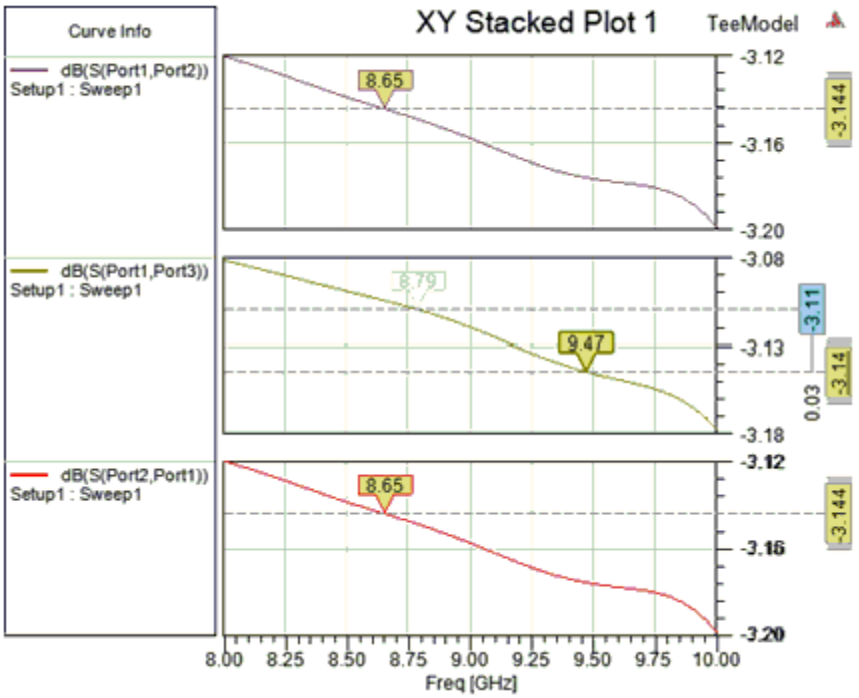
Add Y Marker > **All Stacks** creates one Y marker in each stack with same value. Initially this value is the minimum Y value of the Y ranges in all the stacks. This is shown is figure below:



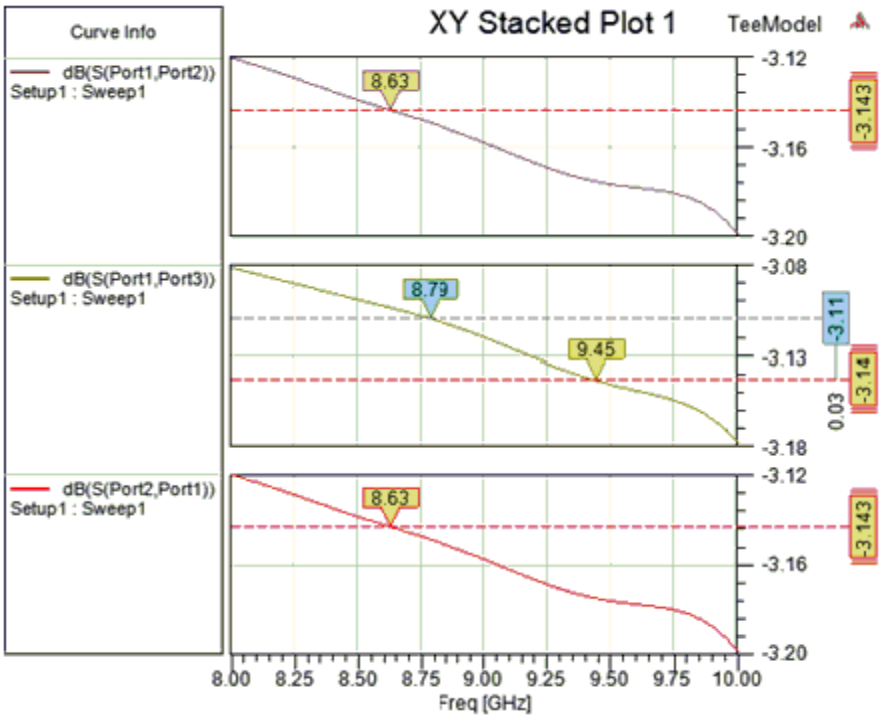
Notice that the Y Marker for All Stacks has a different appearance than the Y Marker for a particular stack, that is, it has double parallel lines above and below the Y Marker textbox.

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 it's value is changed, all the "same" Y markers in all the stacks will change their position too. The following figure shows that when Y marker in bottom stack was dragged, Y marker in top stacks moved as well:

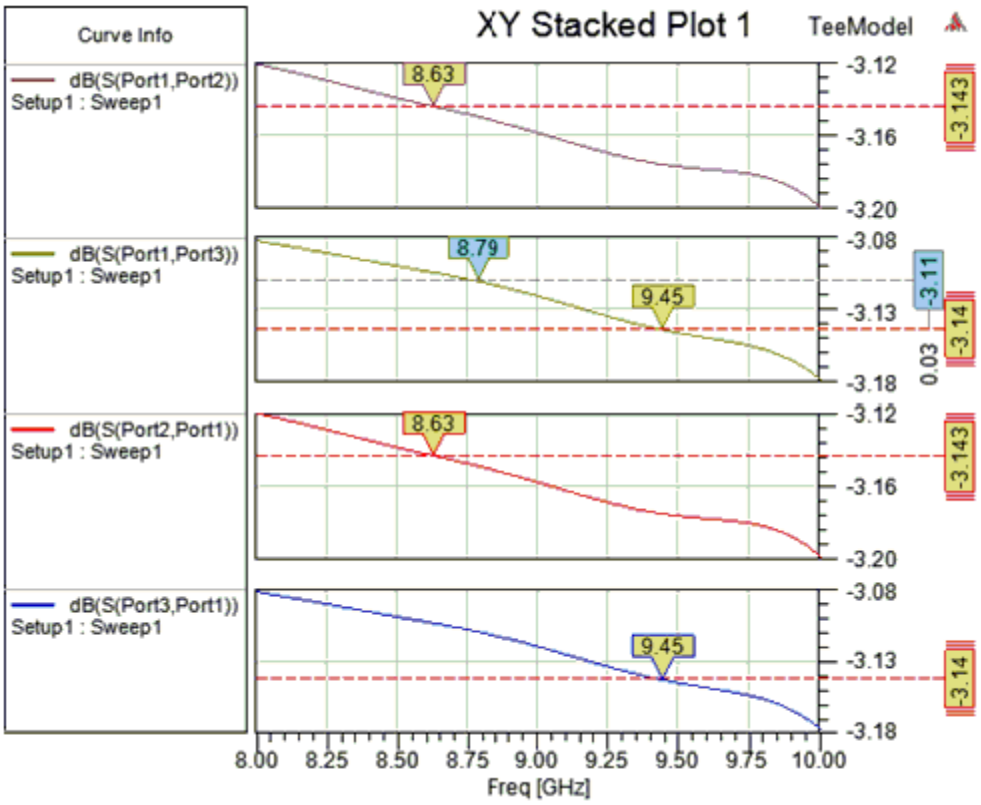


Also if a property of any one Y Marker is changed, all the "same" Y Markers show the change in property as well. For example the following figure shows that when the line color of a Y Marker in the top stack was changed to red color, a Y Marker in bottom stack show the same line color as well:



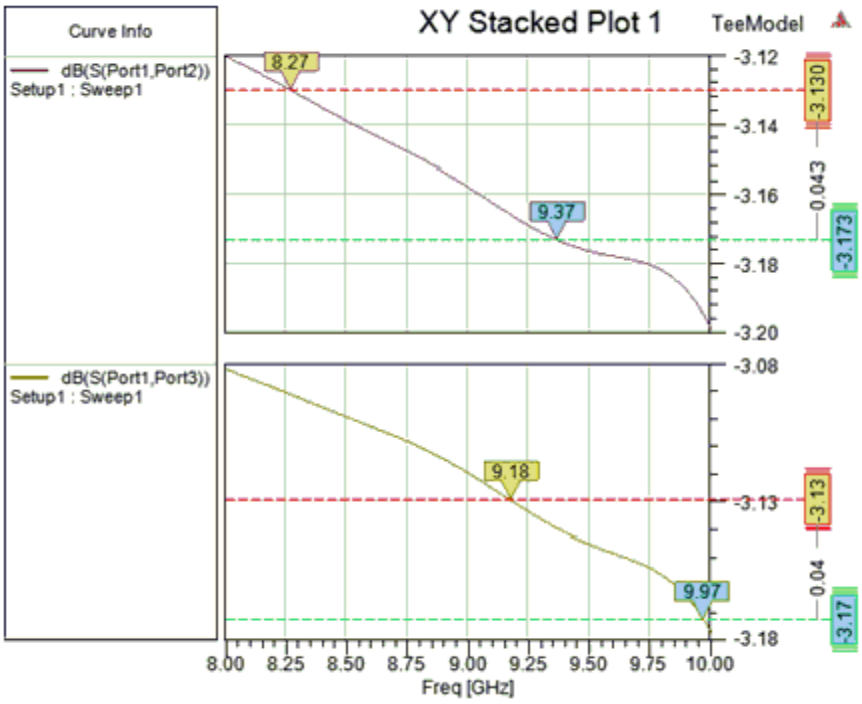
Automatic Y Markers for the New Stack

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:



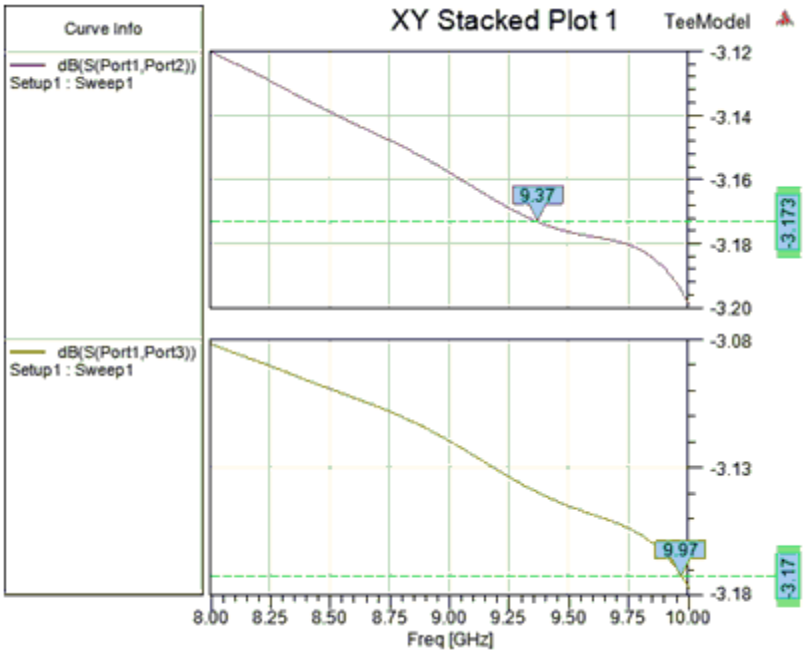
Y Marker Delta Annotations

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 figure below:



Deleting a Y Marker

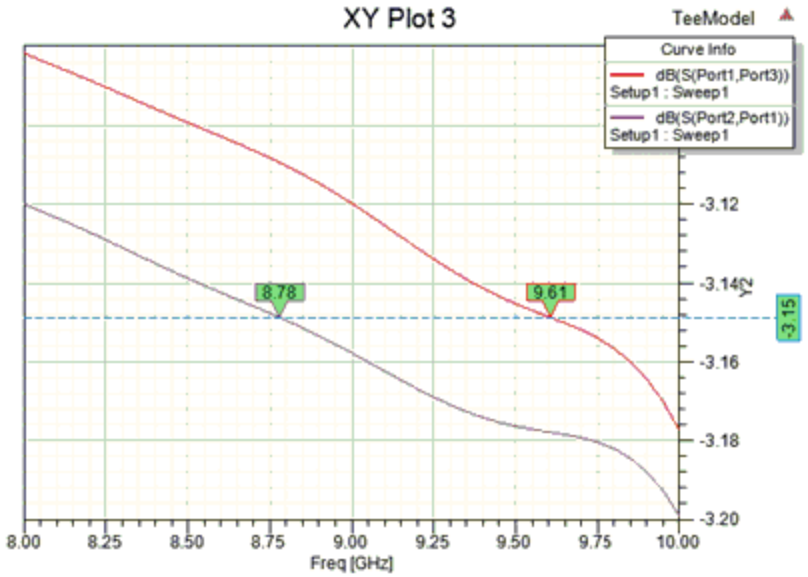
To delete a Y Marker, select a Y Marker in any stack and press the Delete key. This action will also delete all the corresponding Y Markers in all the stacks. For example, when the Y Marker with value -3.13 (red Y Marker) was deleted from the bottom most stack, all of the corresponding Y Markers were also deleted:



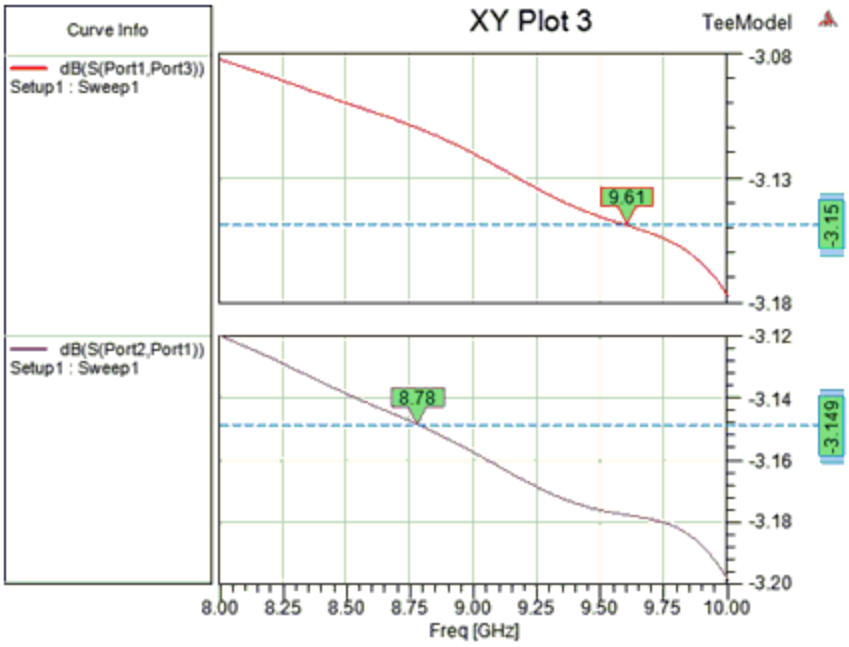
Note that on deleting a stack, Y Markers in other stacks are not affected.

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):

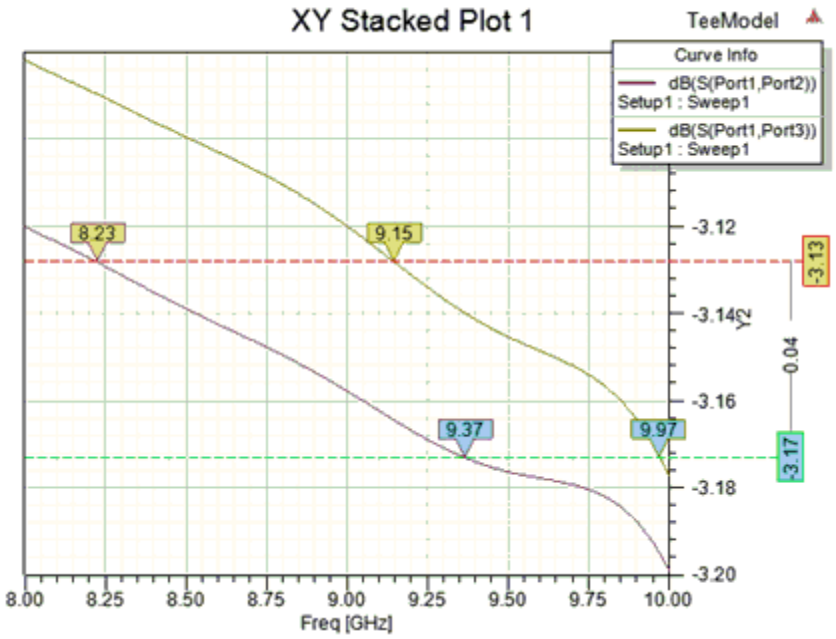


If you change the Display Type property of this plot to Rectangular Stacked Plot then a Rectangular Stacked XY Plot is created with each curve in its own stack and a Y Marker is shown in each stack with value -3.15 (blue Y Marker):



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 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:



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).

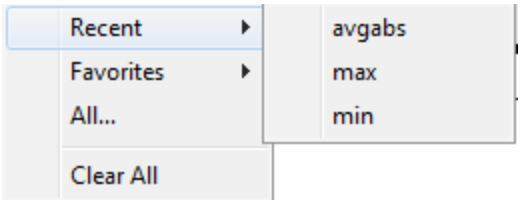
1. Double-click the X or Y axis of interest on an open plot display.

   This opens the **Properties** window for the Axis.

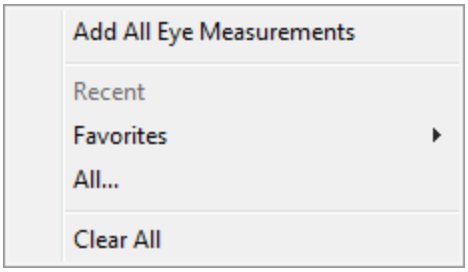
2. Under the **Axis** tab, use the scroll bar to find the **Specify Discard Values** property.
3. Click the check box to enable the property.
4. Enter a value in the **Discard Below** field. Units specified elsewhere in the Axis property are applied to this value. The Discard Below text box is inactive if the Specify Discard Values check box is not enabled.
5. Click **OK** to apply the Discard Values to the report.

Adding Characteristics to a Trace

There are several options for adding characteristics to a trace. When you click **Report2D** > **Trace Characteristics**, or right-click a selected trace, the shortcut menu is displayed. The following example shows the menu with expanded **Recent** selections.



The following shows the shortcut menu for Eye Measurements, which includes the **Add All Eye Measurements** option. You can use this option to add all eye measurements at once.



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 **Report2D** > **Trace Characteristics**, or right-click 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.

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 **Report2D** > **Trace Characteristics**, or right-click 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.

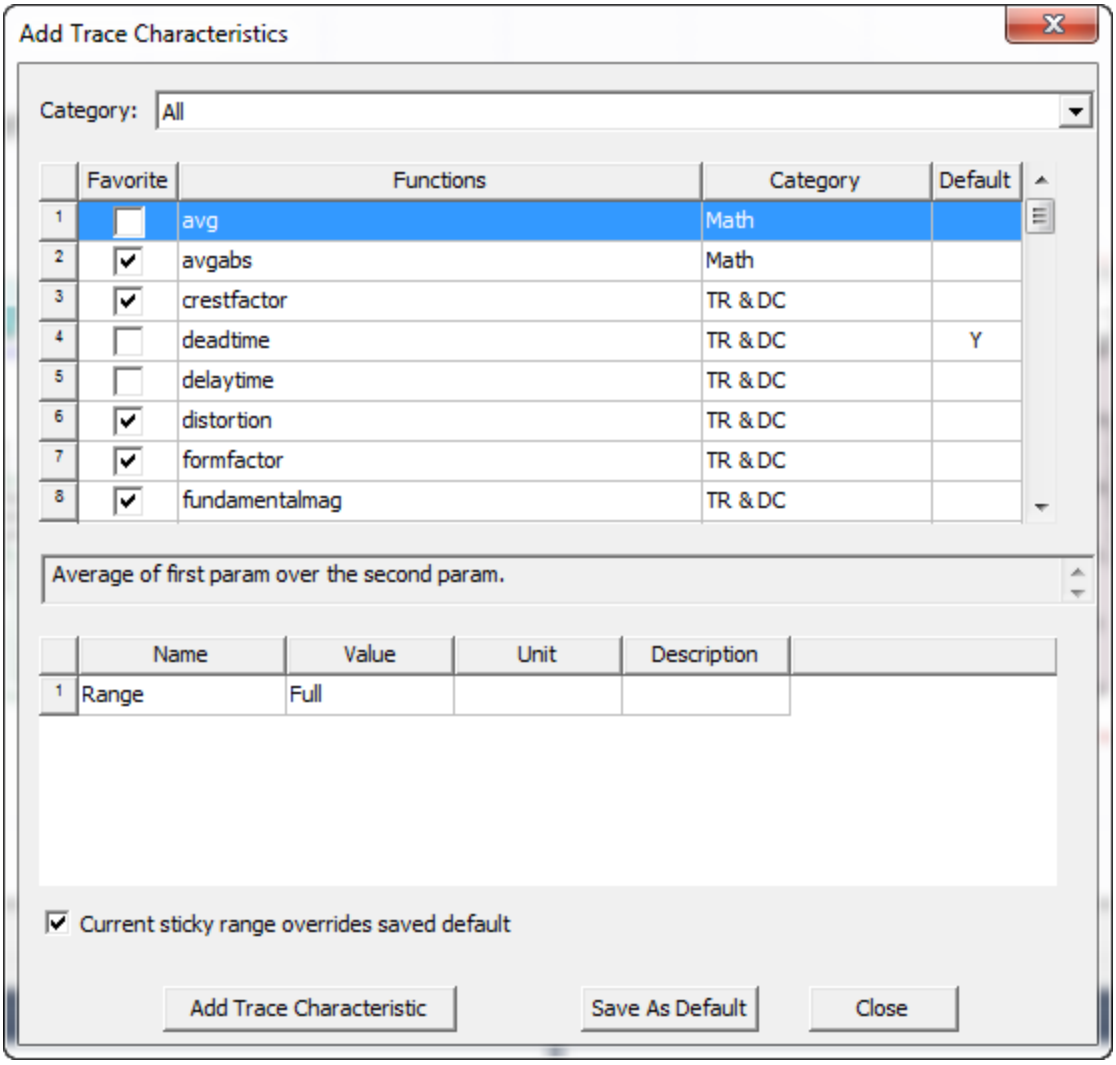
Adding Trace Characteristics to your Favorites

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 **Report2D** > **Trace Characteristics** or right-click the selected trace to display the short cut menu.
3. Select **All**.

   The *Add Trace Characteristics* dialog box displays.



4. Click the **Favorite** check box in front of any function you want to add to your Favorites. You can define as many favorites as you need, but no more than ten 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 menu.

> **Note:**
>
> You can remove favorites by clearing the **Favorite** check box for one or more functions, and clicking **Close**.

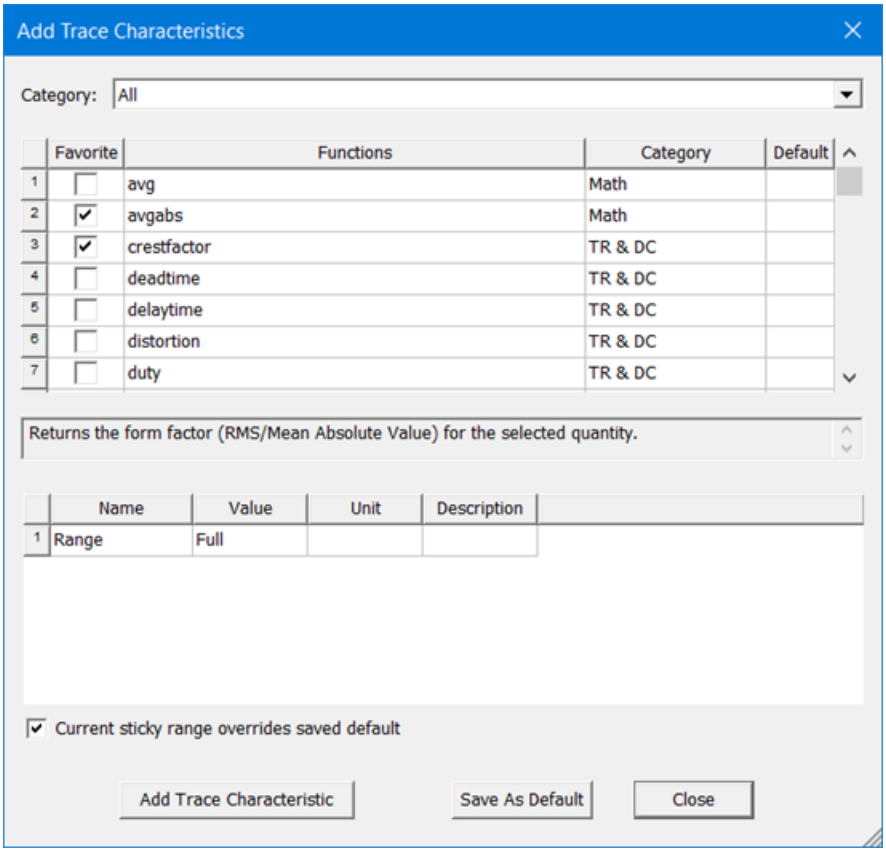
Adding Characteristics Using the Add Trace Characteristics Window

You can add characteristics to a selected trace by selecting from the **Add Trace Characteristics** window.

To add additional characteristics to a selected trace:

1. Select a trace in a report plot or legend.
2. Click **Report2D** > **Trace Characteristics**, or right-click the selected trace to display the short cut menu.
3. Select **All**.

   The **Add Trace Characteristics** window displays:



4. Select the desired **Category**.

   The available categories depend on the plot, and the selecting of a category displays its associated functions.

> **Note: Categories** and **Functions** only appear if they are available for that plot.

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 box displays the function's purpose in a text field. For a list of functions and their definitions, see the table in Defining Traces Using Range Functions.

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**.

Removing All Trace Characteristics

1. Select a trace in a report plot or legend.
2. Click **Report2D** > **Trace Characteristics** or right-click the selected trace to display the shortcut menu.
3. Select **Trace Characteristics** > **Clear All**.

Trace characteristics are cleared from the selected trace.

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 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 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 the trace or traces under a report name in the project tree and select **Copy Data**.

To paste copied trace data:

Right-click 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.

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
> - You must copy and paste trace definitions between the same report types. For example, you cannot copy a trace from a Modal Solution Data report and paste it in a Far Fields report.

To copy a Report Definition:

Right-click the report name in the Project Manager and select **Copy Definition** from the shortcut menu.

To paste the Report Definition:

Right-click Results in the Project Manager 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 the trace or traces under a report name in the project tree and select **Copy Definition**.

To paste the Trace Definition(s):

Right-click 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 that contains a definition with the same name, then an incremented number is appended to the pasted report or trace name.

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**.

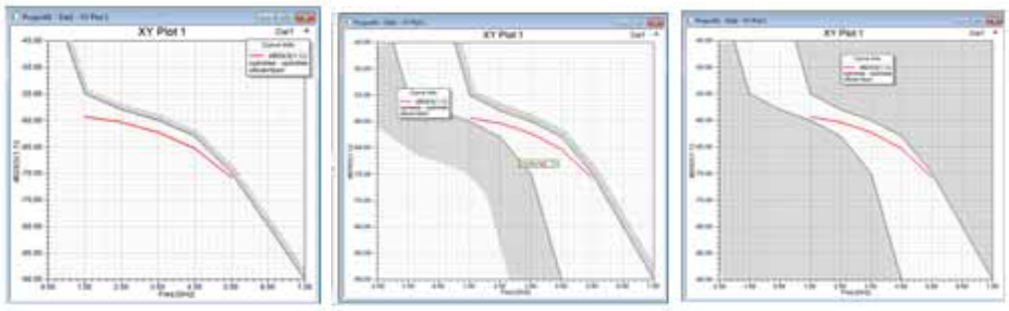
To *remove all traces* from the report:

- Select all the traces and click **Delete**.

Limit Lines in Cartesian Plots

Limit lines are simple graphical representation of constraints on XY plots. These are modeled as a sequence of XY point pairs, or as offsets from a selected curve. You can designate a single limit line to delineate an upper limit, or two lines to delineate upper and lower limits, or upper and lower offset lines simultaneously.

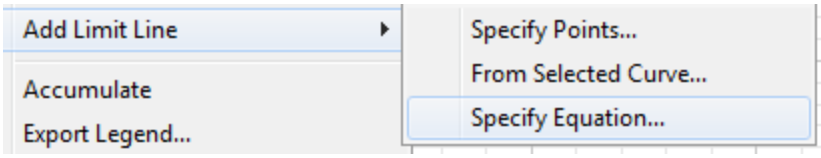
You can control the display properties of the line including color and hatch width in pixels. These lines are available only on XY plots (and not on the XY-like plots: bode, stacked etc)



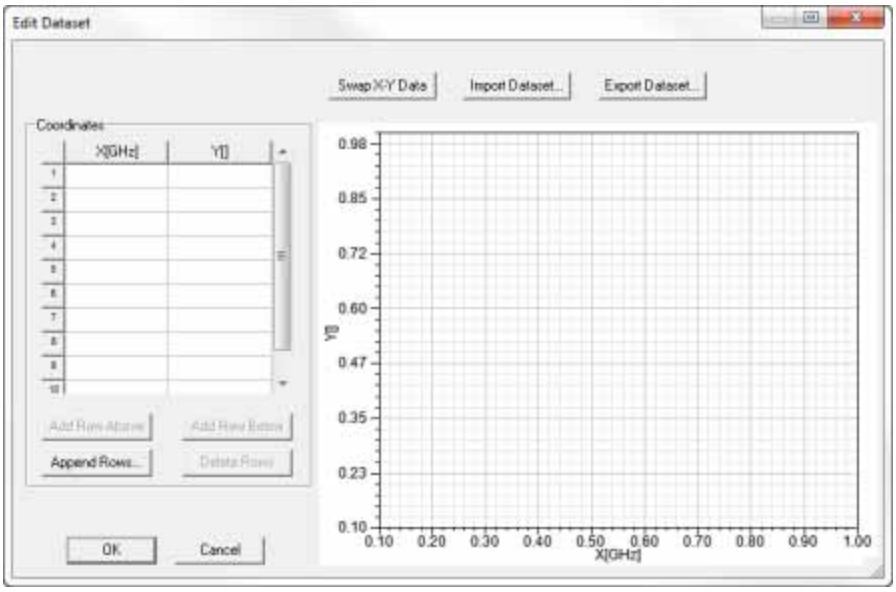
> **Note:**
>
> Limit lines are available only on Rectangular (XY) plots, not on XY-like plots such as Bode or Rectangular Stacked. On 2D Plots, the axes extents are based on the extents of the curves.

To create a limit line:

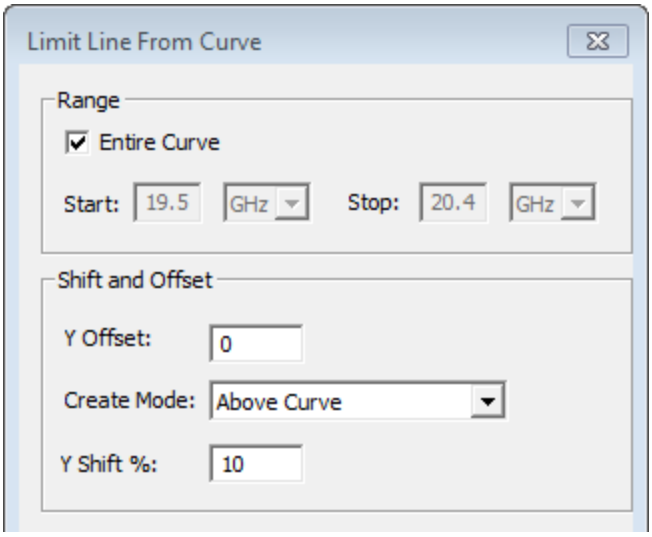
1. Click **Report2D** > **Add Limit Line** or right-click an XY plot and select **Add Limit Line...** from the Context menu. You then select whether to **Specify Points**, or **From a Selected Curve**, or **Specify Equation**.



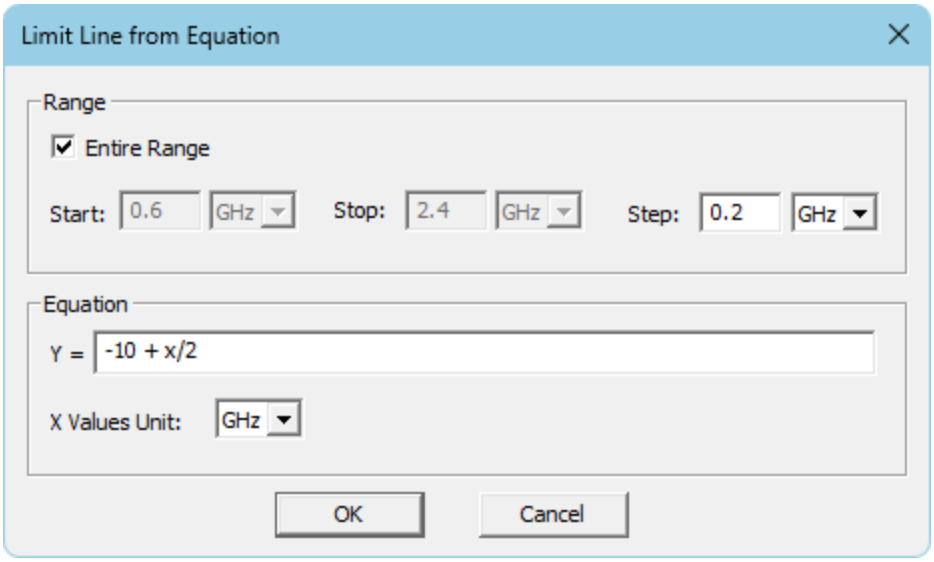
Select **Specify Points** to open an *Edit Dataset* dialog box so that you can specify points.



Select **From Selected Curve** to open the *Limit Line From Curve* dialog box.



Select **Specify Equation** to open the *Limit Line from Equation* dialog box. Specify the desired **Equation** (y as a function of x). Choose the **X Values Unit** from the drop-down menu. The available units depend on the type of sweep quantity (frequency, distance, and so on). The default choice is the current X axis unit for the plot.



2. You can use the *Edit Dataset* dialog box to
   - enter the EY values directly.
   - import XY values from a .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 Row Above**, **Add Row Below** the selected rows, or **Delete Rows**.

   You can use Shift+click to select multiple adjacent rows, or Ctrl+click to select any rows for deletion.

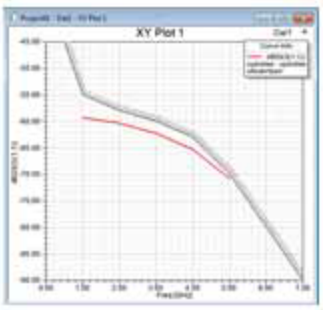
   > **Note:**
   >
   > Each limit line is associated with a particular Y axis (because it has to be scaled the same way as all the curves associated with the axis, follow its log/linear scale and so on). This Y axis association defaults to the first available Y axis when the limit line is created. However, if the plot contains multiple Y axes, it can be associated with a different Y axis later via its properties tab.

3. You can use the *Limit Line From Curve* dialog box to create a limit line with a:
   - Range using the Entire Curve, or a specified Start and Stop.

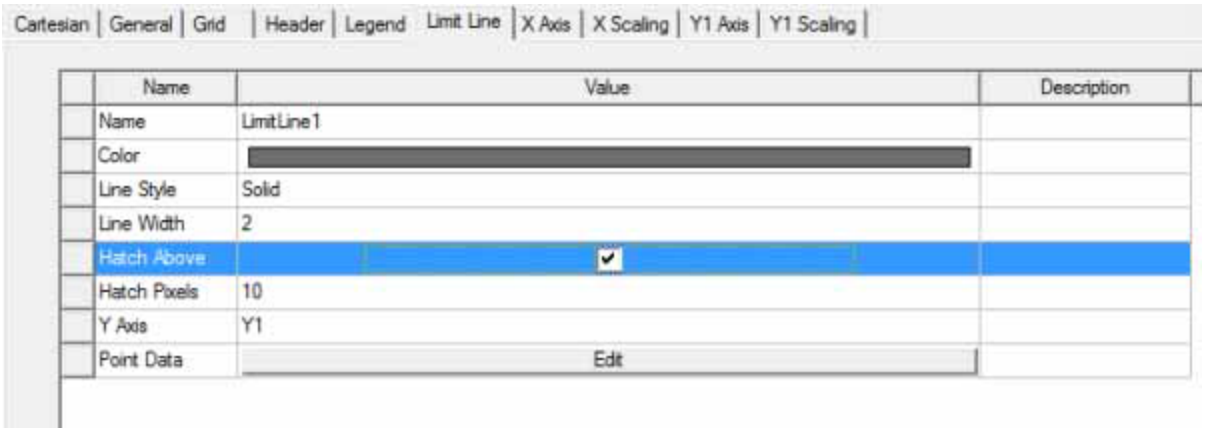
If you uncheck “Entire Curve,” the Start and Stop fields are enabled and initialized based on the zoom level.

- Shift and Offset relative to the Y, either as a Y Offset value or as a Y Shift %.
- Create Mode as Above Curve, Below Curve, or Above and Below Curve.

4. You can use the *Limit Line from Equation* dialog box to create a limit line based on an equation in the form Y = f(x).
   - Only x variable is allowed
   - x value is linearly sampled and f(x) is evaluated to get y. By default, the entire range for plot is covered. You can uncheck **Entire Range** to specify Start and Stop. You can always edit Step and unit values for x.
   - If y results in NaN, an error is indicated and limit line is shown.
   - If y results in Inf value, then it maps to the MapInfValue specified on its associated y axis.
   - Selecting the limit line added in such a way should show: Start, Stop, Step, and Equation in limit line property window. You can edit these values in the Properties window.
5. Once you click **OK**, the limit line, or lines, you define are added to the plot. Each limit line divides the plot into two regions within the context of its length. By default, the upper region is hatched to designate constraint violation.



6. You can select the limit line in the plot to edit its properties via the **Limit Line** tab of the plot properties.

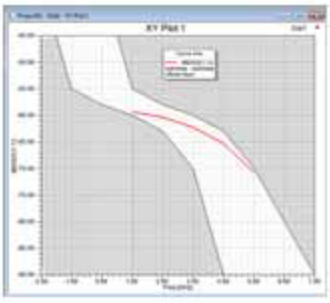


- Line properties: Color, Style and Width
- Y axis association
- Hatch properties
- Hatch width in pixels
- Hatch direction (hatch above or below the Limit line)
- The point definition of the limit line itself.

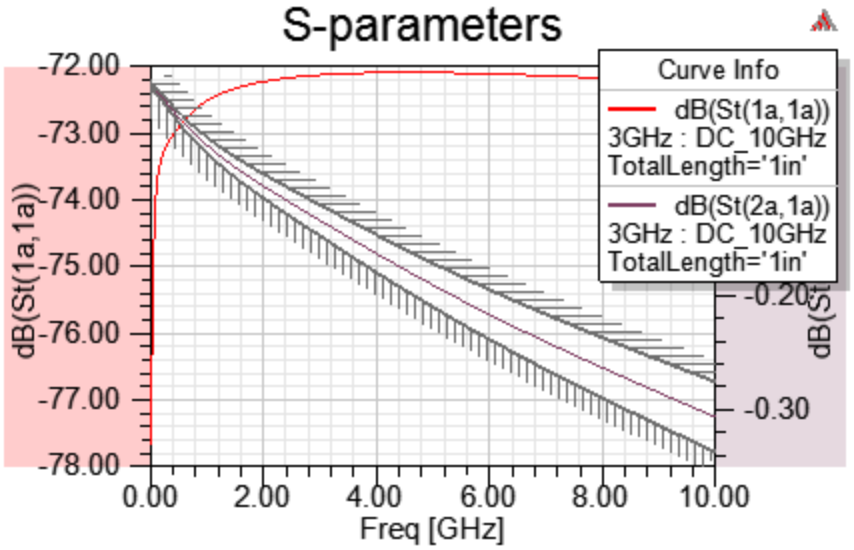
For a limit line specified as an Equation, the Properties also include:

- Start, Stop, Step, and Equation values.

7. If you add a second limit line, you can designate it as hatch below by unchecking **Hatch Above** to produce a tunnel marking upper and lower constraints.

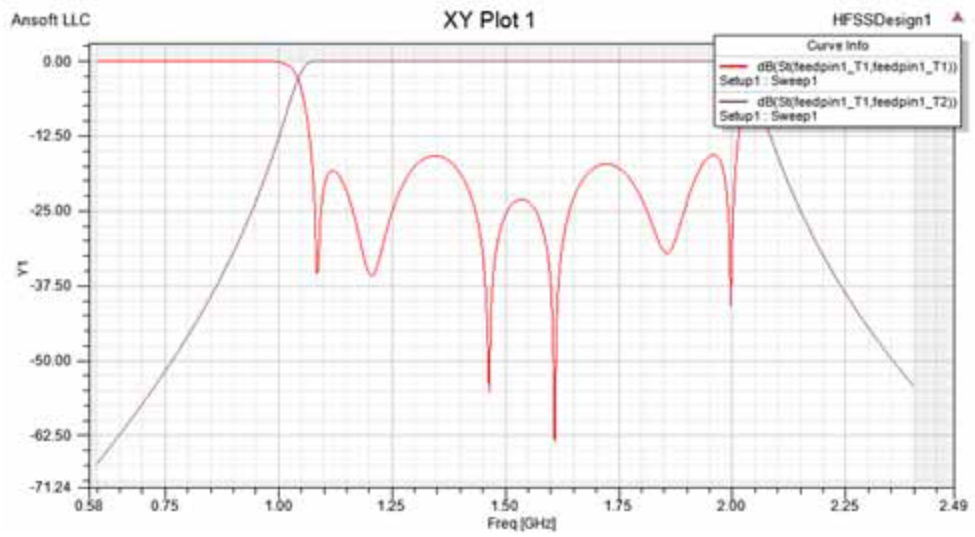


The following example shows the a limit line from curve plot, where the Hatch Above property for the lower limit line has been unchecked.

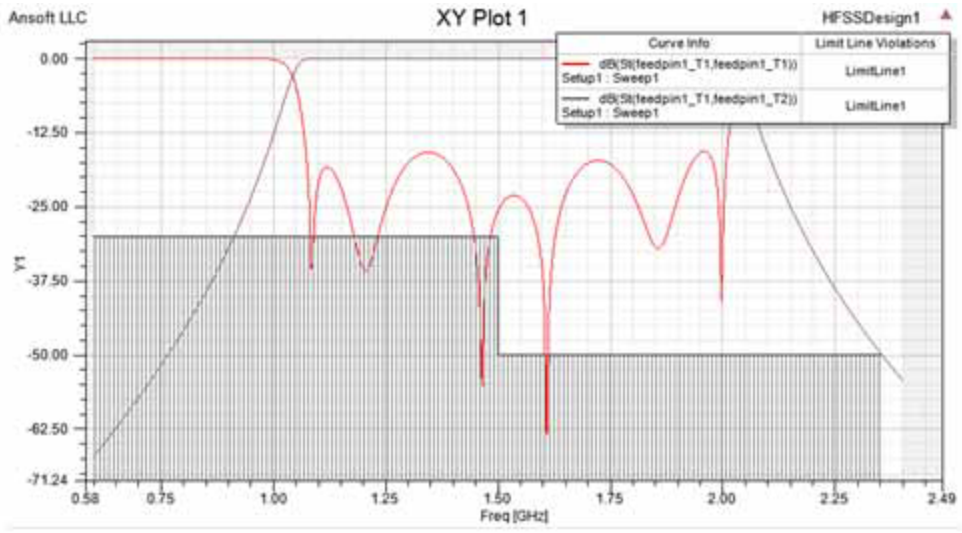


Limit Line Violations

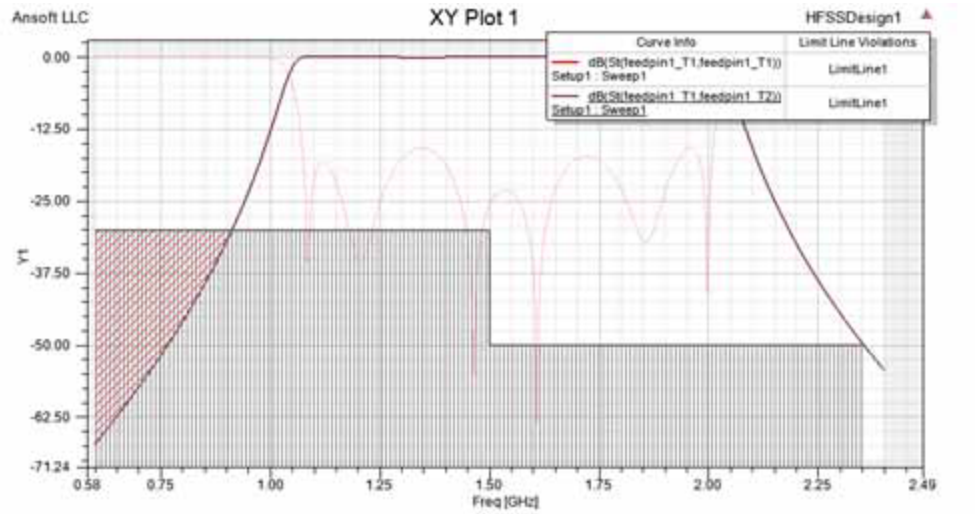
You can use a plotting feature to help you discern whether a curve violates a limit line or not. Consider following plot which shows two curves:



Suppose that the response cannot be below -30 dB until 1.5 GHz, and cannot be below -50 dB at 1.5 GHz or higher. You add a limit line for this requirement in the plot using 'Add a Limit Line' functionality. The plot automatically calculates whether a curve violates this requirement, that is, the limit line and show it in the legends window, as shown in following figure:

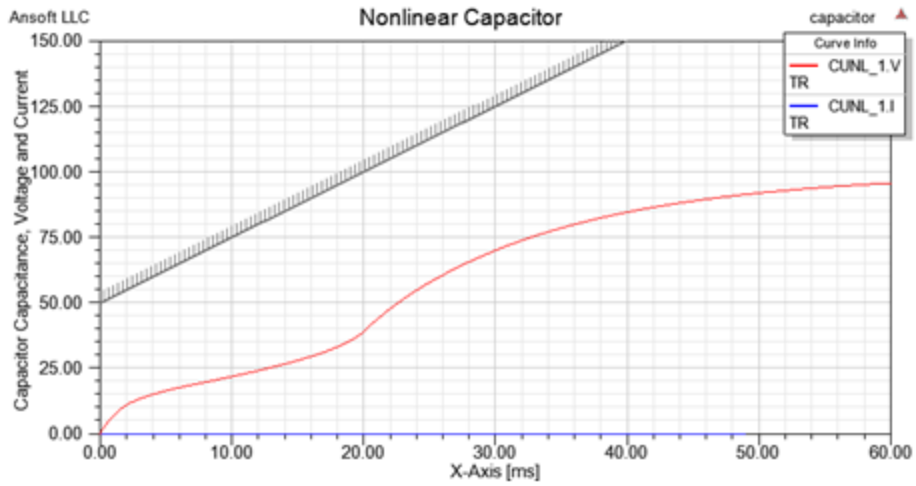


If a curve is selected, then the plot shows the region of the curve that violates the limit line (shaded with slanted red lines), as shown in following figures:



Note:

As shown in the following figure, if no curve violates a limit line, then the limit line hatching is restrained to 10 pixels in length. The minimal hatching keeps the focus on the curve traces rather than the limit lines.



Error Handling

If an error is encountered while calculating Limit Line Violations for a particular plot trace, the *Curve Info* window shows "NaN (*<Limit Line name>*)" under the *Limit Line Violations* column for that curve. For example, if a value is not available or goes to infinity at a particular point, the program is unable to evaluate that point. One of the coordinates is "Not-a-Number" (NaN).

Sweeping a Variable in a Report

In Q3D Extractor, a swept variable is a 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.

For large projects or projects with many variables, you may obtain faster post processing before generating a solution by selecting which variables function as Sweep variables. Only the variables with Sweep enabled are indexed for post processing. See Adding a Design Variable and Adding a Project variable.

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) drop-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 pop-up list of the possible values.
    b. Select **Use all values** or 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.

Sweeping Values Across a Distance

1. If you are plotting a field quantity along a line, define a polyline object in the problem region.

   If you are plotting a near-field quantity along a line, set up a near-field line.

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.

5. For Near field, to plot the selected quantity or quantities at every point on the line, select **All Values**.

   For Near field, to plot the selected quantity, or quantities, at specific points on the line, clear the **All Values** option, and then select the point values on which you want to plot.

> **Note:**
>
> All maximum near-field data calculated by Q3D Extractor is at their maximum over the selected line object; if you plot the parameter over a sweep of values, the parameter will have the same value at each point on the plot.

Sweeping Values Across a Sphere

1. Set up a near-field sphere or a far-field infinite sphere.
2. In the **Report** dialog box, click the sphere geometry of interest in the **Geometry** list.
3. For the **Sweeps** variable corresponding to **phi**, select the ellipsis [...] button.

This displays a small dialog.

4. Clear the Use all values check box to enable selection and editing of the sweep values.

All of the possible values for the phi variable are listed in the dialog box. The values are the result of the range of phi you specified during the infinite sphere's setup. To modify the values of phi to be plotted across the sphere, do the following:

a. Click **Edit Sweep**.
b. Specify the following information:

Step or Count	Whether to sweep by steps, or by linear count, decade count, octave count, or exponential count.

Start Value	The point where the rotation of phi begins.
End Value	The point where the rotation of phi ends.
Step or Count	The number of values between the start value and the end value.

c. Click **Update Values**, and then click **OK**.

The values listed are updated to reflect the new number of points.

5. To plot the selected quantity or quantities at every value of phi, select **All Values**.

To plot the selected quantity or quantities at specific values of phi, clear the **All Values** option, and then select the phi values at which you want to plot.

6. For the **Sweeps** variable corresponding to **theta**, follow steps 4 and 5 for modifying the values of theta, if necessary, and specifying the theta values at which to plot the selected quantity or quantities.

> **Note:**
>
> All antenna parameters and maximum far-field data calculated by HFSS is at their maximum over the selected object; if you plot the parameter over a sweep of values, the parameter will have the same value at each point on the plot.

Selecting a Function for a Plot

The value of a quantity being plotted depends upon its mathematical function, which you select from the **Trace** tab **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, "S(11)" is the value of the S-parameter for every swept combination of variables (e.g., "height", "frequency"). (A smaller set of functions appears for the Function list in the Output Variables dialog.)

These functions can also be applied to previously specified Quantities and Functions as **Range Functions** when using the Set Range Function dialog box.

Some of these functions can operate along an entire curve. These are: min, max, integ, avg, 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 Yor 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 quantities) is computed as: 20.0*log10(x) + 60.0
dBu	(for electric field quantities) 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.
fmod	nReturns the double precision remainder of x/y.
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 double precision modulus (x - floor(x/y)*y). Not supported in solvers.
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	Returns the fractional part of a decimal number such that rem(x) = x-int (x) Syntax: rem(x)

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 Solution Quantities to Plot

When you create a report, each trace in the report includes a quantity that is plotted along an axis. The quantity being plotted can be a value that was calculated by Q3D Extractor, a value from a calculated expression, or an intrinsic variable value (such as frequency or theta).

> **Important:**
>
> Available solution quantity categories depend on the design type, solution type, setup parameters, and plot domain. The selected category provides the default name for the plot, such as "S Parameter Plot *n*". You can edit plot names in the project tree and plot header text in the Properties window.

To select solution quantities to plot:

1. Click **Results** > **Standard Report** and select the type of report you would like to generate.

   The **Report** window appears.

2. On the **Trace** tab, select a **Category** from the list.

Category	Description
Variables	Intrinsic variables (such as frequency or theta), or user-defined project variables (such as the length of a quarter-wave transformer).
Output Variables	User-defined expressions applied to derive quantities from the original field solution.

Category	Description
S-Parameter	S-parameters from the S-matrix. For designs which include a Frequency Selective Surface (FSS)-referenced radiation boundary, S_{11} and S_{21} represent the extracted reflection and transmission coefficients, respectively.
Y-Parameter	Admittance matrix parameters computed from the S-parameters and port impedances.
Z-Parameter	Impedance matrix parameters computed from the S-parameters and port impedances.
Group Delay	Quantity calculated as rate of change of the total phase shift with respect to angular frequency, $\frac{d(\Phi)}{d(\omega)}$
VSWR	Voltage standing wave ratio, calculated from the equation: $\frac{1+\lvert S_{ij}\rvert}{1-\lvert S_{ij}\rvert}$
Gamma	Propagation constants for the S-parameters.
Port Z_o	Characteristic port impedances.
Matched S Parameter	Matched S is the S-matrix produced by solver. It becomes useful to view when passive ports exist and the matched S-matrix is rectangular instead of square. When passive ports exist, S is the square matrix of the active ports and the passive ports represent a loss mechanism. Matched S-matrix provides coupling information to the passive ports. In Transient, we show Matched S entries while the simulation is in process which is not clean for regular S when port post processing exists.
Short Circuit Z	Short circuit impedance, calculated as: $\mathrm{ShortCircuitZ}\,(\mathrm{P1}) = \mathrm{mag}\left(\frac{1}{\mathrm{Y(P1,\ P1)}}\right)$
Loop Inductance	LoopL computes with all other ports shorted, even ports on other nets. These short circuits permit induced currents to flow in nearby nets, meaning LoopL will always be lower than a more rigorous loop inductance computation. $\mathrm{LoopL}\,(\mathrm{P1}) = \frac{\mathrm{im}\left(\frac{1}{\mathrm{Y(P1,\ P1)}}\right)}{(2 \times \pi \times \mathrm{freq})}$

Category	Description
Loop Resistance	Calculated as: $\mathbf{LoopR}\left(\mathbf{P1}\right) = \mathbf{re}\left(\frac{1}{\mathbf{Y(P1,\ P1)}}\right)$
Capacitance	This capacitance calculation leaves other nets open-circuited, whereas a rigorous capacitance calculation would short them to ground, meaning "Capacitance" computed here will always be too low. $\mathbf{C}\left(\mathbf{P1}\right) = \frac{\mathbf{im}\left(\frac{1}{\mathbf{Z(P1,\ P1)}}\right)}{(2 \times \pi \times \mathbf{freq})}$
Lambda	Guided wavelength.
Epsilon	Effective permittivity.
TDR Impedance	Time-Domain Reflectometry (TDR) impedance for non-terminal problems. The idea is to excite a structure with a step function, and inspect the reflections as a function of time. This quantity is available only when the **Time** domain is selected. Selecting the TDR Impedance category displays the TDRZ of every terminal or mode in the ports. The list of available Functions includes those that can operate on the TDRZ values.

Category	Description
Active S-Parameter	Active S parameters represent the reflection coefficient of one port when another port is excited. HFSS offers passive S-parameters as well as active S-Parameters. If you plot Active S-Parameters, HFSS includes the effects of mutual coupling with other ports or elements. One use is in Optimization, to view how S-parameters change as design parameters change. In this case you may plot Active S Parameters. In such cases they allow you to forecast design behavior in the presence of an ideal feed network. A 0w value is a matched load, but 1w values take that into account. Active S-parameters are given as a linear combination of the regular S-parameters, with some weights 'a', based on the matrix equation $\mathbf{b} = \mathbf{S} \times \mathbf{a}$. In the post-processor, the active S-parameter for port*n* is displayed to the user as ActiveS (P*n*:*m*) where "P*n*" indicates the port number and ":*m*" indicates the mode number. Active S, Y, Z, VSWR are supported only for driven modal projects. Power is proportional to the square of the fields. As a consequence, active S-parameters, which are field related, are proportional to the square root of what you set under Edit Sources. You can check this by setting all power to 0 except for one port of interest where you want to see the Active S-parameter. Then for a neighboring port, change the input power in steps, and monitor how the active S-parameter changes. Keep in mind that this involves complex numbers which complicates things. Given a driven model project with a total of *N*port modes, let a_k denote the complex excitation for the *k*-th mode specified in the Edit Sources dialog box. Also let S denote the computed *N* x *N*scattering matrix. If $a_m \neq 0$ define active-S_m, *m* = 1..., *N* by $$active\text{-}S_m \equiv \sum_{n=1}^{N} S_{mn} \frac{a_n}{a_m} \qquad m = 1, ..., N$$ If *S* is renormalized or deembedded, the interpretation is that the stimulations are applied to the renormalized external transmission lines at the plane of deembedding. The other relative active quantities are simply transformations on the active S_m.

Category	Description
Active Y-Parameter	$Y_0^{(M)}$ is the port admittance of the *m*-th mode. $$active\text{-}Y_m \equiv Y_0{}^{(m)} \frac{1 - active\text{-}S_m}{1 + active\text{-}S_m} \qquad m = 1, ..., N$$
Active Z-Parameter	$Z_0^{(m)}$ is the impedance of the *m*-th mode. $$active\text{-}Z_m \equiv Z_0{}^{(m)} \frac{1 + active\text{-}S_m}{1 - active\text{-}S_m} \qquad m = 1, ..., N$$
Active VSWR	Active Voltage standing wave ratio is supported only for driven modal projects. $$active\text{-}VSWR_m \equiv \frac{1 + \lvert active\text{-}S_m \rvert}{1 - \lvert active\text{-}S_m \rvert} \qquad m = 1, ..., N$$
Passivity	A scalar quantity based on the matrix Q = I - conjugate (transpose(S)) * S For every frequency, the value must be no larger than 1. A uniform renormalization of 50 ohms is performed on the solution data for Passivity checking.
Power	For Composite Excitation Solution types, selecting Power as Category lists Quantities for Accepted (Pacc), Reflected (Pref), and Incident (Pinc) powers/port based on your Edit Sources settings.
Design	Enables you to plot or tabulate properties of objects, such as their volumes.
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 Convergence	This is intended to plot convergence, as a function of adaptive pass, of expressions in the Expression cache of the Solution setup. In defining the report, for "Context", select Solution: SetupN:Adaptive Pass. ExprDelta will show the change in the value of the expression as a function of adaptive pass, while ExprGoal will show, for comparison, the convergence goal for this expression, as defined in the Analysis Setup under the panel Expression Cache.

3. Select a **Quantity** from the list.
4. If desired, select a **Function**.
5. Click **Close**.

Selecting a Field Quantity to Plot

When plotting field quantities, the quantity can be a value that was automatically calculated by Q3D Extractor such as the magnitude of S_{11}, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or phase.

To select a field quantity to plot:

1. When you create the report, specify the Report Type as "Fields" and the plot type (for example, radiation pattern.)
2. In the *Report* dialog box, select Geometry for the Context, unless you are plotting scalar (for example, integration). For example, to plot near-field values across a sphere, you select the sphere object from the **Geometry** list in the *Traces* dialog box when you create a report.
3. In the *Report* dialog box, select one of the following categories. The selected category provides the default plot name. You can edit the plot names in the project tree and the plot header text in the report synchronizes.

Variables	Intrinsic variables, such as frequency or phase, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	User defined expressions applied to derive quantities from the original field solution.
Calculator Expressions	Includes scalar and vector field quantities automatically calculated by Q3D Extractor, as well as derived field quantities that are defined by calculated expressions you set up in the Fields Calculator.

4. Select a quantity to plot from the **Quantity** list. The available quantities will depend upon the selected category and the setup of the design. See Field Quantities list for definitions.

Plotting Imported Solution Data

1. In the **Solution** drop-down menu of the *Report* dialog box, click the imported data you want to plot.
2. Follow the procedure for creating a report.

Setting a Range Function

To apply a range function to the Y, Z, or Mag component of a trace:

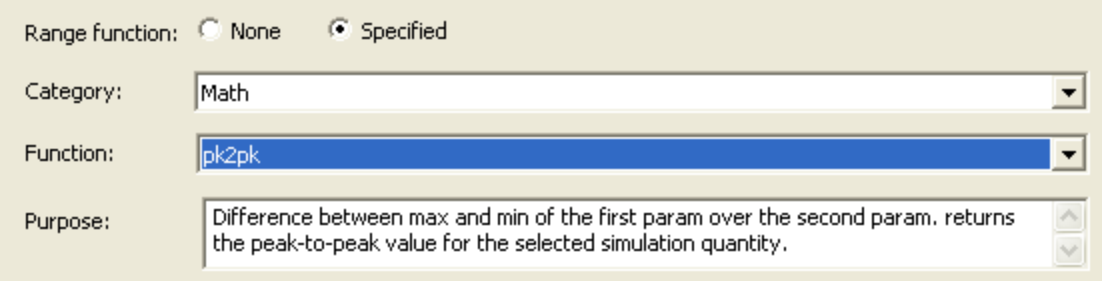
1. Click **Range Function** in the *Report* dialog box.

This opens the *Set Range Function* dialog box. 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. To enable Range function selection, click the Specified radio button.

   Selecting the None radio button disables 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.



   Given a selected Function, and Category, the **Set Range Function** dialog displays a text field that explains the Purpose of the function. See figure above.

   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, use the ellipsis [**...**] button and uncheck **Use All Sweeps**.

   This enables a list of the sweeps. The sweep(s) you select is displayed on the Over Sweep line. You can use the buttons to **Clear All Selections** or **Select All** sweeps.

6. Select the Sweeps **Default** or **Edited** radio buttons to specify whether to accept the default or edited sweeps.
7. 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.

8. Click **OK** to apply the range function.

Range Functions

The following table shows the **Functions** according to their **Categories**. The most commonly used range categories are **Math** and **Radiation**. Other functions could be used if needed. Use the category links to navigate to tables with definitions of functions.

Category	Functions
Math	max, min, pk2pk, rms, sum, mean, variance, stddev, integabs, avgabs, rmsAC, ripple, pkavg, XatYMin, XatYMax, YAtXMin, YAtXMax, XAtYVal, YAtXVal, XWidthAtYVal
PulseWidth	pulsemin, pulsemax, pulsemintime, pulsemaxtime, pulsefall9010, pulsefront1090, pulsefront3090, pulsetail50, pulsewidth5050, pw_plus, pw_minus, pw_plus_avg, pw_minus_avg, pw_plus_max, pw_minus_max, pw_plus_min, pw_minus_min, pw_plus_rms, 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
AC	gainmargin, phasemargin, gaincrossover, phasecrossover, lowercutoff, uppercutoff, bandwidth, peakgain, peakgainfreq.
Radiation	xdb10bandwidth, xdb20bandwidth, lSidelobeX, lSidelobeY, rSidelobeX, rSidelobeY
Eye Measurements	EyeLevelZero, EyeLevelOne, EyeAmplitude, EyeHeight, EyeSignalToNoise, EyeOpeningFactor, EyeWidth, EyeJitterP2P, EyeJitterRMS, EyeRiseTime, EyeFallTime, MinEyeWidth, MinEyeHeight

> **Note:**
>
> Refer to the SI Wave or Nexxim help for more information. The Purpose field offers brief descriptions of each.

Math Functions	
***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.

Math Functions	
integabs	Absolute value of integral.
max	Returns maximum value of the simulation quantity.
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.
rms	Returns the root mean square value of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
ripple	Returns the ripple factor (AC RMS/Mean) for the selected quantity.
pkavg	Returns the ratio of the peak to peak-to-average for the selected quantity.
pkp2pk	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.
sum	Returns the sum of the given values.
stddev	Returns the standard deviation of given values.
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 occurrences of Y value.
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.

Radiation Functions	
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.
rSidelobeX	Returns the X value of right side-lobe occurrence.
rSidelobeY	Returns the Y value of right side-lobe occurrence.
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.

Pulse Width Functions

> **Note:**
>
> In this table, the functions with the asterisk (*) do not appear on the Range Function drop-down menu. They can still be used via text entry.

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	Returns the rms of the negative pulse width input stream.

pw_plus	Returns the pulse width of the first postive pulse.
pw_plus_avg	Returns the average of the positive pulse width input stream.
pw_plus_max	Returns the maximum pulse width of the positive pulse of input stream.
pw_plus_min	Returns the minimum pulse width of the positive pulse of input stream.
pw_plus_rms	Returns the rms of the positive pulse width input stream.

Overshoot/Undershoot	
Overshoot	Calculates peak overshoot given a threshold value and number of evenly spaced points over entire time range.
Undershoot	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.

TR & DC Functions	
crestfactor	Returns the crest factor (peak/RMS) for the selected simulation quantity.
formfactor	Returns the form factor (RMS/Mean Absolute Value) for the selected quantity.
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.
fundamentalmag	Returns the RMS value of the fundamental frequency for the selected quantity, and an additional argument, Frequency, which specifies the fundamental frequency.
delaytime	Obtains the time from zero to 50% of the target point.
risetime	Obtains the time taken to go from 10% to 90% of target point.
deadtime	Obtains the latest time when the qtyl is within a tolerance of zero.
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 arguments.

Error Functions	
iae	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument.

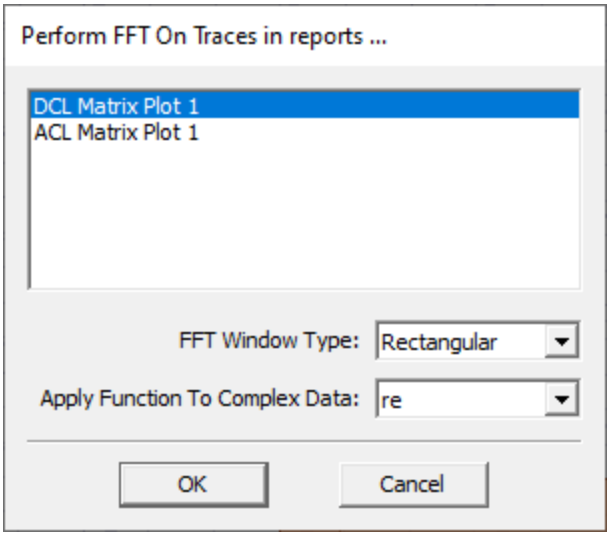
ise	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument.
itae	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument.
itse	Returns the time-weighted 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.

Periodic Functions	
per	Returns the period of a simulation quantity.
pmax	Max period of the selected simulation quantity.
pmin	Minimum period of the selected simulation quantity.
prms	Period Root Mean Square.

AC Functions	
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.
gaincrossover	Returns the gain crossover frequency (where the gain is 0 dB) of the selected simulation quantity in Hz.
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.
lowercutoff	Returns the lower 3dB frequency of the selected simulation channel in Hz.
uppercutoff	Returns the upper 3dB frequency of the selected simulation channel in Hz.
bandwidth	Returns the 3dB bandwidth of the selected simulation quantity. For bandwidth, the calculation is based on 3dB below the maximum peak.
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.

Perform FFT on a Report

You can perform FFT on an existing 2D plot by using the **Results** > **Perform FFT** command. You can perform TDR on an existing 3D plot by using **Q3D Extractor** > **Results** > **Perform FFT**. This opens the **Perform FFT on Traces in Reports** dialog box.



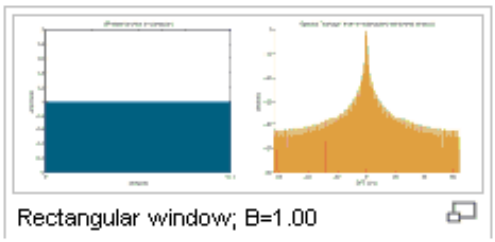
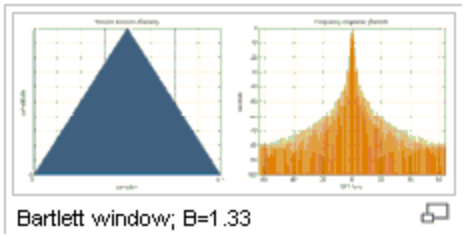
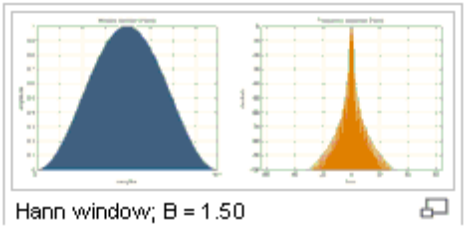
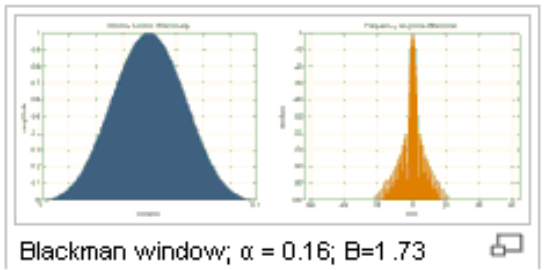
1. Select the report you want from the list.
2. Select the FFT window type to apply. Winding functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies.
3. Select the function to apply to complex data.

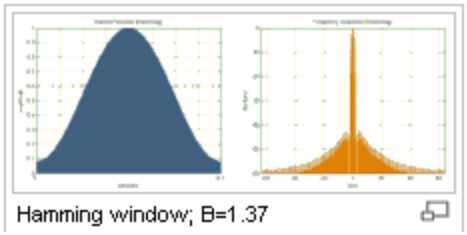
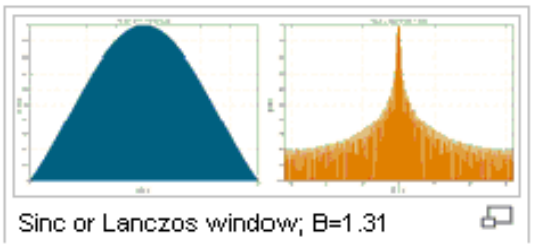
   The new report appears under **Results** in the **Project Manager**. The new report name prefixes FFT to the name of the original report. Trace names are also prefixed with FFT.

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. $w(n)=1$

Window Function	Preferred Use
	 Rectangular window; B=1.00
Tri	A Bartlett window with the endpoints valued at zero.  Bartlett window; B=1.33
Van Hann	A moderate dynamic range function, designed for narrow band applications. $$w(n) = 0.5(1 - cos\frac{2\pi n}{N-1})$$  Hann window; B = 1.50
Blackman	A high dynamic range function, with lower resolution, designed for wide band applications. $$w(n) = \alpha_0 - \alpha_1 cos(\frac{2\pi n}{N-1}) + \alpha_2 cos(\frac{4\pi n}{N-1})$$ where a_0 =(1-α)/2; α_1=1/2; α_2=α/2  Blackman window; α = 0.16; B=1.73
Hamming	A moderate dynamic range function, designed for narrow band applications. It

Window Function	Preferred Use
	minimizes the maximum sidelobe. $w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right)$ 
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. 
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.

Apply FFT to Report Functions

The choices in the Perform FFT on Traces in Reports dialog include:

ang_deg	**Angle (phase) of a complex number, cut at +/- 180**
ang_deg_val	**Angle (phase of a complex number in unitless degree values. Returns simple numbers.**
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_deg_val	**Cumulative angle (phase) of the first parameter of the selected simulation quantity in unitless degree 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.**
dB(x)	**20*log10(|x|)**
dB 10normalize	**10*log [normalize(mag(x))]**
dB 20normalize	**20*log [normalize(mag(x))]**
dBc	
GetGroupDelay	
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**

Note:

The evaluated value of an expression is always interpreted in SI units. However, when an angle quantity is plotted in a report, you have the option to plot values in units other than SI. If you want to plot the polar angle of a complex simulation result, S_{11} say, you can choose between ang_deg(S_{11}) and ang_rad(S_{11}). Both of these return the exact same angle quantity but in degree and radian units respectively.

Note that when used in expressions, some surprising outcomes might result. For example, the expression "1+ang_deg(S_{11})" represents an 'angle' and the number "1" is treated as "1 rad". The angle SI unit is attached to any unitless number that is added/subtracted from an angle value. If you want to treat "1" as degrees, make it explicit and use "1deg + ang_deg(S_{11})" instead.

If you are interested in unitless degree values, two additional functions exist: ang_deg_val(S_{11}) and cang_deg_val(S_{11}). These return simple numbers and are treated as such by any expression. If the complex S_{11} lies on the positive Y axis say, ang_deg_val (S_{11}) would be 90 and "1 + ang_deg_val(S_{11})" will be 91.

Perform TDR on Report

You can perform TDR on an existing 2D plot by using the **Results** > **Perform TDR on Report** command. You can perform TDR on an existing 3D plot by using the **HFSS 3D Layout** > **Results** > **Perform TDR on Report** command. This opens a **Perform TDR on Traces in reports** dialog box.

1. Select the report you want from the list in the dialog box.
2. Specify the input signal as **Step** or **Impulse**.
3. Set the **Rise Time**.
4. In the **Type** drop-down menu, select which 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.

5. To specify a window width, enter a percentage into the **Width (%)** field.
6. If you select the Kaiser function, specify a number in the **Kaiser Parameter**.

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.

Window Functions and Time Domain Plotting

HFSS and SIwave allow for time-domain plotting of S-parameters. Often, this feature is used to calculate a step response or time-domain reflectometry (TDR) plot of the structure being simulated. Fourier analysis provides the mathematical mechanism for transforming frequency sweep data to a time-domain plot, but two approximations are involved. First, the transform is between two sets of discrete data points, as opposed to continuous waveforms. Second, the frequency sweep data cannot have infinite bandwidth, but must truncate at some upper limit. This section discusses the implications of these approximations, and provide information for successful time-domain plotting.

Note: Frequency sweep data consists only of positive frequencies, but the negative frequencies are simply the complex conjugate of the positive: $S(-f) = S^*(f)$

This is true for any frequency-domain function when the corresponding time-domain waveform is real-valued.

Transforming Frequency- To Time-Domain

It is easier to make generalizations about the effect of finite bandwidth if we have continuous functions. Consequently, we will initially assume our frequency- and time- domain data is continuous, and defer discussion of the effects of discretization until later. With a continuous-time sweep over an infinite bandwidth, we could – at least in principle – calculate a time-domain response by multiplying our sweep data $S(f)$ with the spectrum of a time-domain excitation function and evaluating the inverse Fourier integral:

$$f(t) = \int_{-\infty}^{\infty} S(f)E(f)e^{2\pi jft}df$$

In practice, however, sweep data does not extend to infinite frequencies and is restricted to a bandwidth b. If we simply assume that the spectrum is zero-valued outside of the bandwidth, we can interpret the data as an infinite sweep that has been multiplied by a rectangular “window” function $W(f)$, with a value of 1 within the bandwidth and a value of 0 otherwise.

$$f(t) = \int_{-\infty}^{\infty} W(f)S(f)E(f)e^{2\pi jft}df = \int_{-b}^{b} S(f)E(f)e^{2\pi jft}df$$

This process is illustrated in Fig. 1, assuming that $S(f)\,E(f)$ corresponds to an ideal unit step function in the time domain. In Fig 1.a, the frequency spectrum is truncated beyond a certain upper limit. Since multiplication in the frequency domain corresponds to convolution in the time domain, this has the effect of convolving the time-domain step with a sinc function – the inverse

Fourier transform of the rectangle (Fig 1.b). The final result is an edge with a finite rise time and some oscillation.

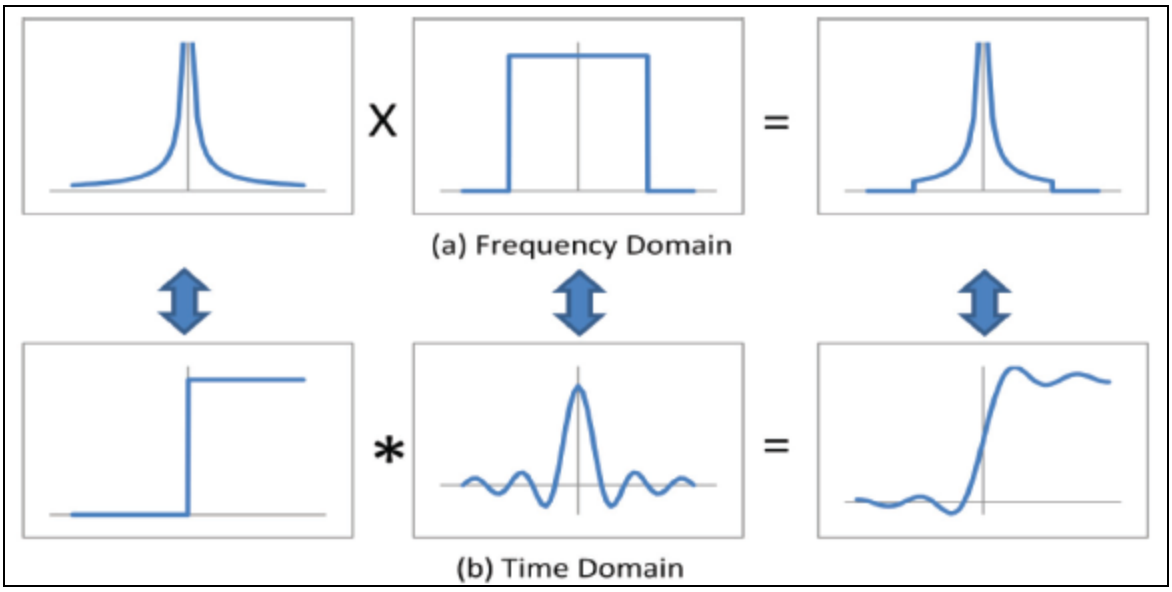


Figure 1. Multiplying the spectrum of a step function with a rectangular window produces a finite edge in the time domain

If the sweep is extended to higher frequencies – making the window function wider – the corresponding sinc pulse more closely approaches an impulse, and the time-domain edge becomes sharper. However, the oscillation never disappears for any finite sweep. Fig. 2 shows a step response for increasingly wider bandwidths.

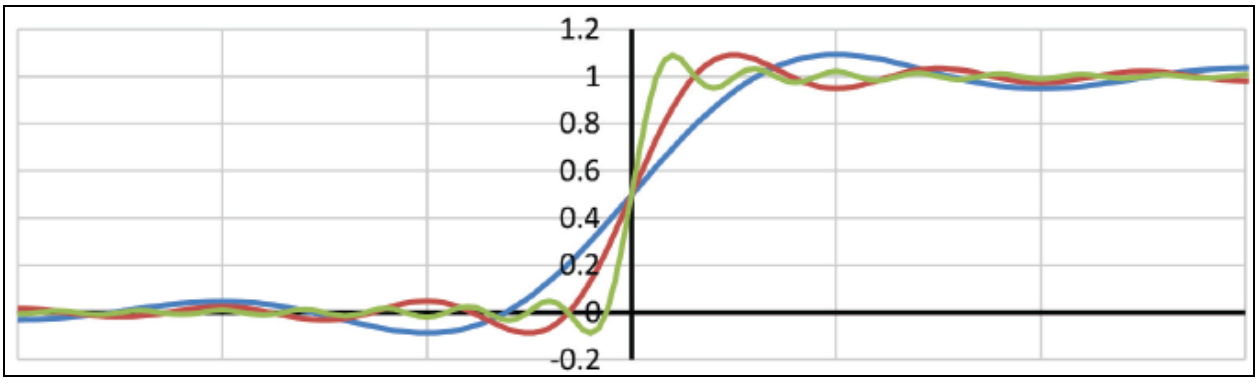


Figure 2. Increasing the width of the rectangular window makes the time-domain edge sharper, but does not eliminate the oscillation

Some distortion of the true time-domain waveform is unavoidable if the frequency sweep does not include the entire bandwidth of the signal, but there are other window functions besides the rectangle which distort the time waveform in ways which may be more desirable. In particular, it would be nice to reduce the spurious oscillation. The next sections will describe the window functions available and discuss their effects.

Window Functions

The window functions available are plotted in Figs. 3 and 4, and their expressions are given in the Appendix. All the window functions have a spectral width *w* and are zero-valued for |*f*| > *w*/2. In addition to truncating the data outside of the bandwidth, the non-rectangular windows filter the spectrum inside. The windows differ from each other in how strongly they attenuate the spectrum as the frequency approaches the upper limit. The Kaiser window has a parameter, which controls how sharply it decays. For $\alpha = 0$, the Kaiser window is equivalent to the rectangular window; for $\alpha = 5.4414$, it is equivalent to the Hamming window; and for $\alpha = 8.885$, the Blackman window.

Although the TDR Options dialog allows for windows that are narrower than the bandwidth of the simulation, it is generally best to set the window width to 100% and take full advantage of the available bandwidth.

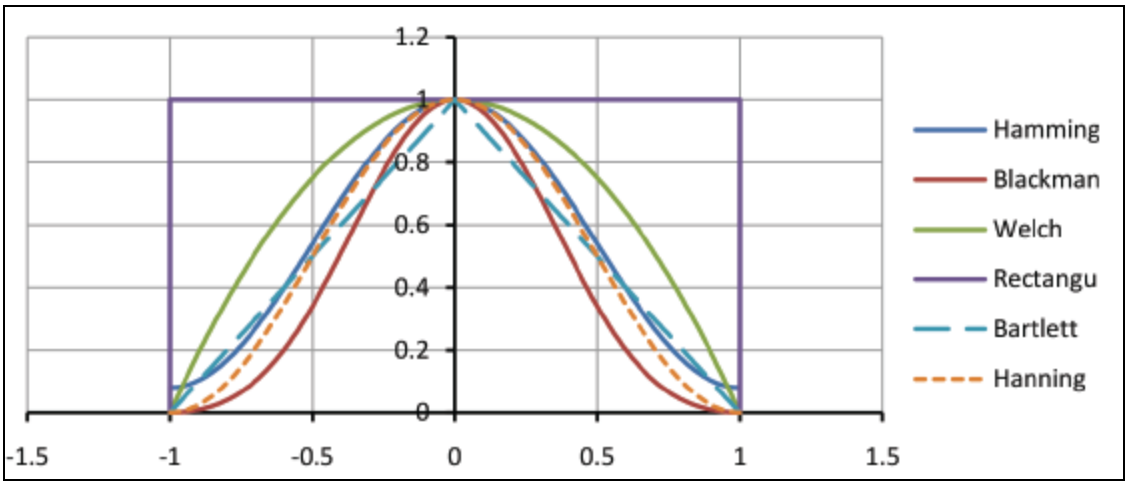


Figure 3. Window functions with a width of *w*=2

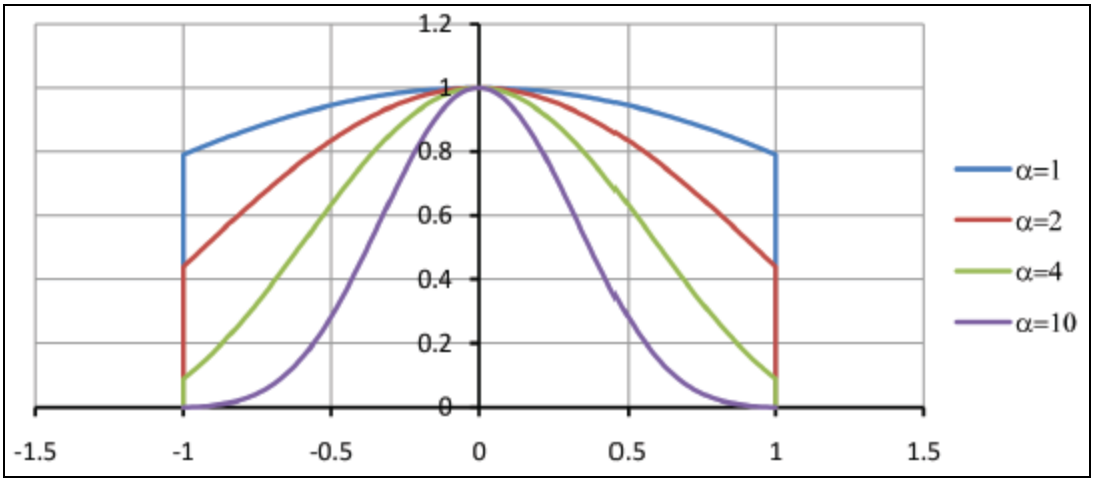


Figure 4. The Kaiser window for a width of *w*=2 and varying values

Because the spectral width *w* includes both positive and negative frequencies, it is twice the bandwidth of the sweep, *b*, which is equal to the (positive) upper frequency limit.

Ideal Step Response

It is immaterial whether we think of the window as multiplying the frequency sweep data, with the spectrum of the time-domain excitation having infinite bandwidth, or if we instead imagine we have infinite sweep data and a windowed excitation spectrum. With the latter interpretation, we can examine the effects of different windows on an ideal step without concern for what the sweep data looks like.

We will apply different windows to an ideal step function, which is approximated in HFSS and SIwave by choosing an edge and setting the rise time to 0. We continue to assume that we have a continuous spectrum, and will defer a discussion of the effects of discretization until later. The effect of the Welch window is shown in Fig. 5.

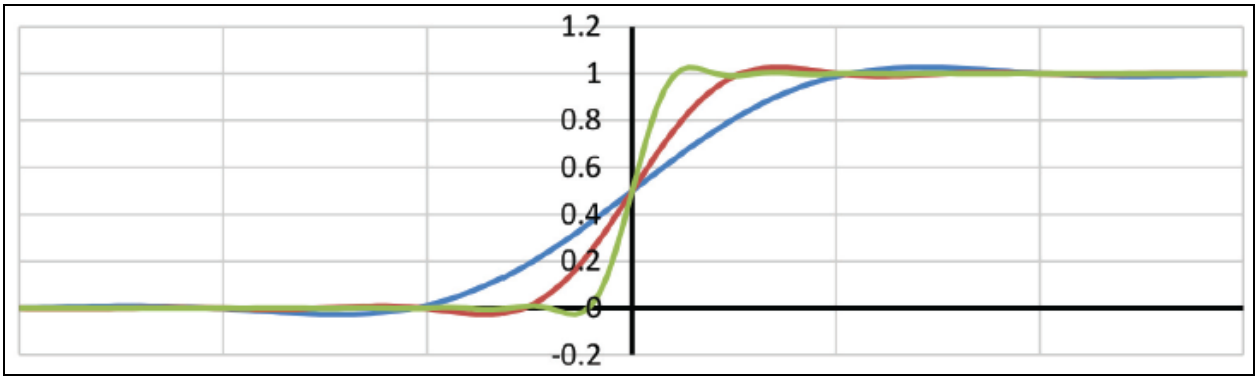


Figure 5. The effect of Welch windows of three different widths on an ideal step

Fig. 5 shows that the Welch window has substantially decreased the signal oscillation that was seen with the rectangular window. As Fig. 6 below demonstrates, the Blackman window results in almost no oscillation.

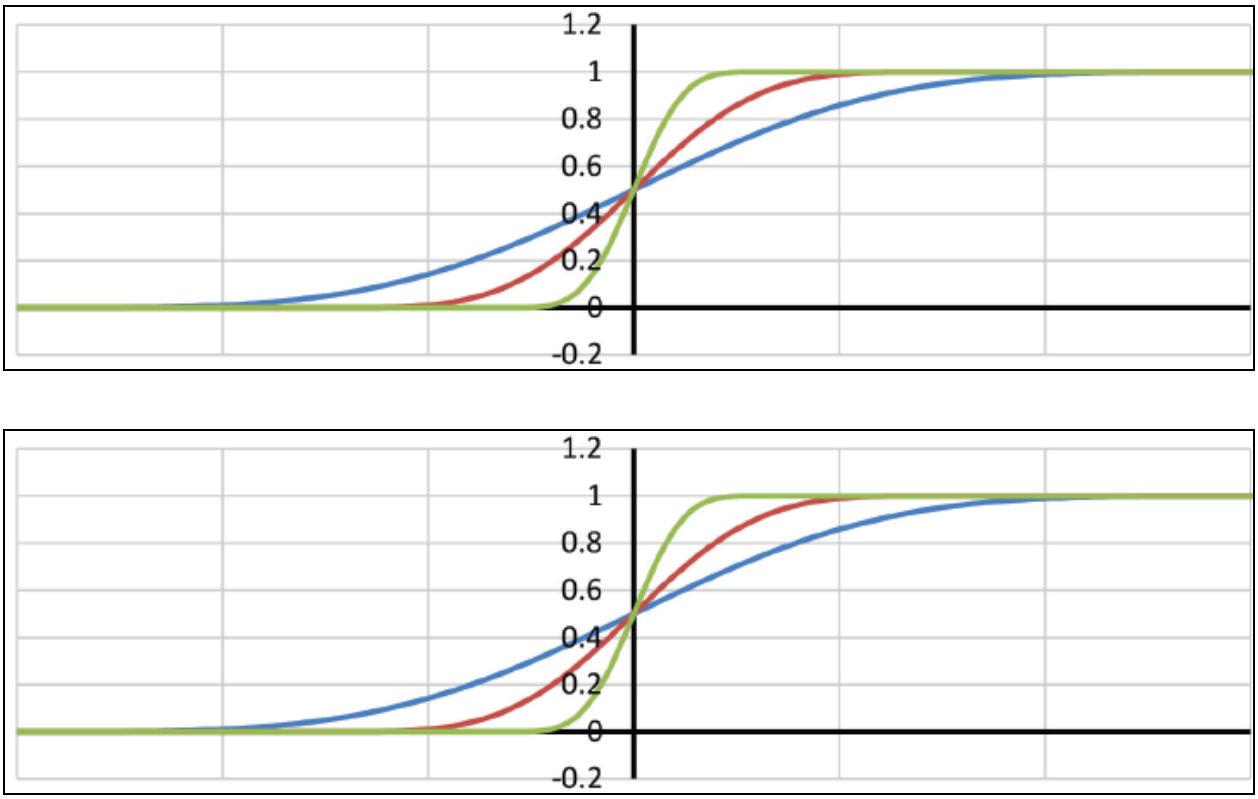
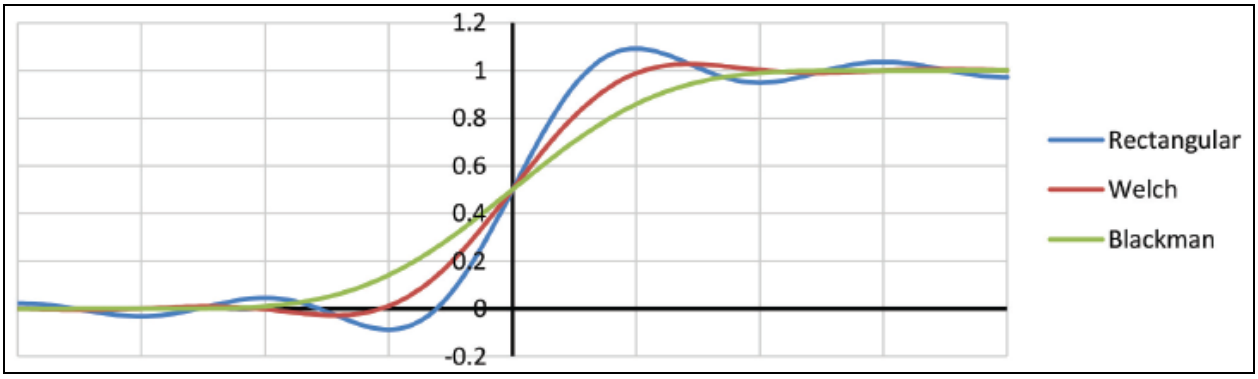


Figure 6. The effect of Blackman windows of three different widths on an ideal step.



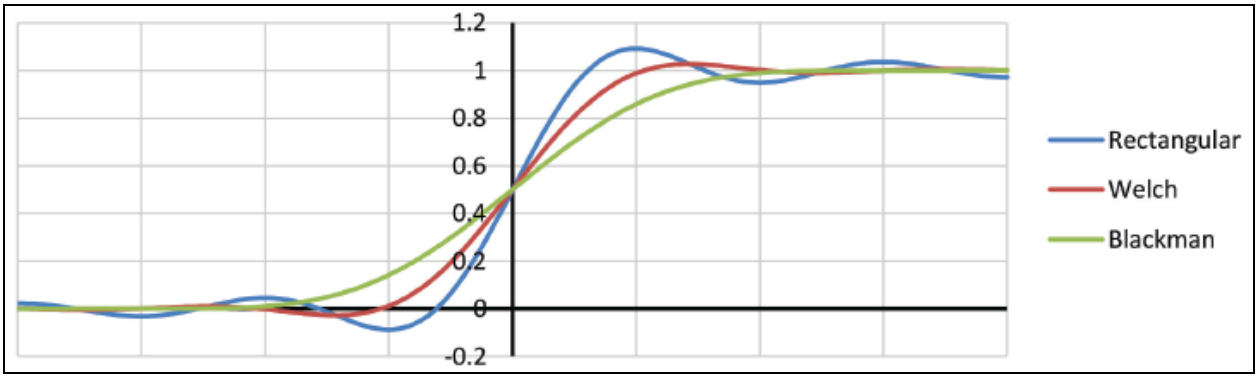


Figure 7. Step response for three different windows, each with the same bandwidth.

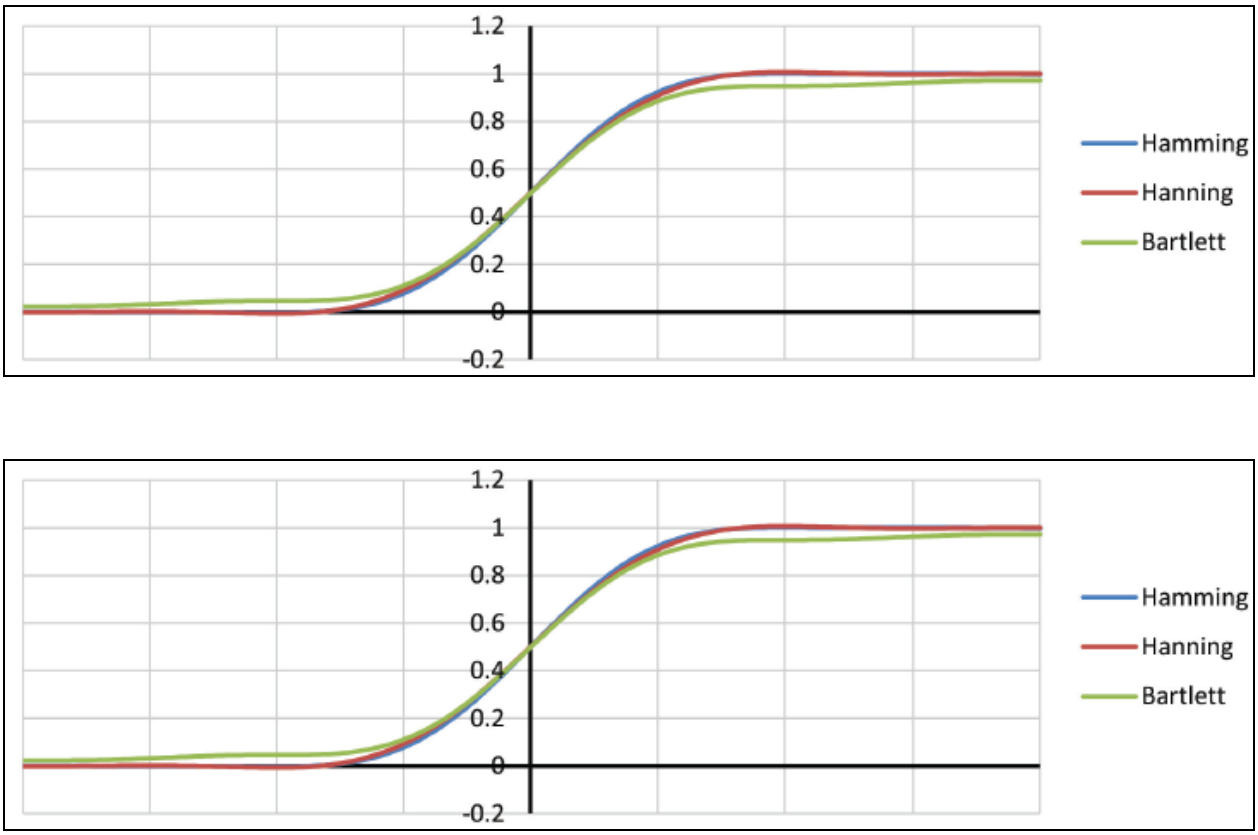


Figure 8. Hamming, Hanning, and Bartlett windows of equal bandwidth

As Fig. 8 suggests, the difference between Hamming and Hanning windows is usually quite small. The Bartlett window is generally not recommended, as it distorts the signal in the vicinity of the edge without providing any advantage over the Hamming and Hanning windows. The Kaiser window gives edges that are slower and less oscillatory with increasing α.

The rectangular, Welch, Hanning, and Blackman windows are sufficient to provide a good sampling of the edge-rate vs. oscillation tradeoff. Table 1 quantifies the characteristics of these windows on an ideal step. With the exception of the Blackman window, it is possible to derive reasonably simple expressions for the step response. In Table 1, *b* is the bandwidth or upper frequency limit of the sweep and Si(χ) refers to the sine integral function:

$$\mathrm{Si}(x) = \int_0^x \frac{\sin z}{z} dz$$

Table 1. Characteristics of selected window functions for continuous time.

Window	*Step Response* $f_s(t)$	*10-90 Edge Rate*	*Max Overshoot*
Rectangular	$\frac{1}{2} + \frac{\mathrm{Si}(2\pi bt)}{\pi}$	$\frac{0.45}{b}$	8.95%
Welch	$\frac{1}{2} + \frac{\cos(2\pi bt)}{2\pi^2 bt} - \frac{\sin(2\pi bt)}{4\pi^3 b^2 t^2} + \frac{\mathrm{Si}(2\pi bt)}{\pi}$	$\frac{0.70}{b}$	2.70%
Hanning	$\frac{2\pi + 2\mathrm{Si}(2\pi bt) - \mathrm{Si}(\pi - 2\pi bt) + \mathrm{Si}(\pi + 2\pi bt)}{4\pi}$	$\frac{0.97}{b}$	0.64%
Blackman	--	$\frac{1.19}{b}$	0.02%

Note that in the expressions for the step response, the time variable is always multiplied by the bandwidth. Changing the bandwidth scales the time response, but does not affect the shape of the edge.

Finite Edge Response

Finite edges can be simulated by providing a nonzero value for the rise time. For finite edges, the same edge rate vs. oscillation tradeoff applies. However, the spectrum of a finite edge declines with frequency at a faster rate than an ideal step. As a result, modest amounts of overshoot can be achieved even with a rectangular window. The continuous time finite edge response of a rectangular window is given by

$$f_e(t) = \frac{1}{2} - \frac{\mathrm{Si}\big(2\pi b(r-t)\big)}{\pi} + \frac{\cos(2\pi bt) - \cos\big(2\pi b(r-t)\big) + 2\pi bt\left(\mathrm{Si}\big(2\pi b(r-t)\big) + \mathrm{Si}(2\pi bt)\right)}{2\pi^2 br}$$

The value of the edge response at $t = 0$ is given by

$$f_e(0) = \frac{1}{2} + \frac{1 - \cos(2\pi br)}{2\pi^2 br} - \frac{\mathrm{Si}(2\pi br)}{\pi}$$

Along with the overshoot, f_e (O) is a useful metric for describing how closely the finite edge response approximates the ideal case, for which f_e (O) = O. The degree to which the windowed edge approximates an ideal finite edge depends only on *br*, the dimensionless product of the bandwidth and the rise time (Fig. 9).

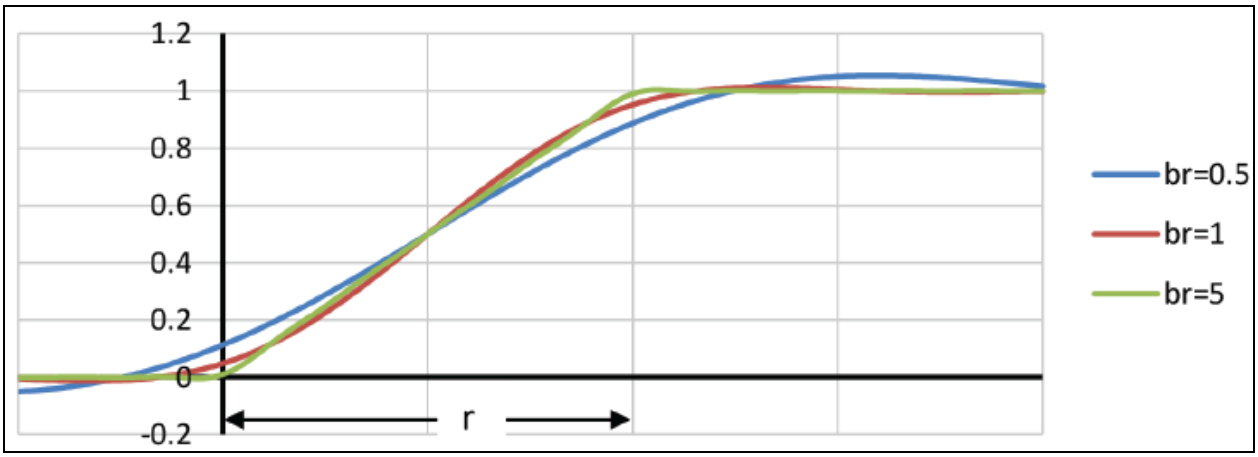


Figure 9. The effect of rectangular windows on edges with rise time *r*. The *y*-intercept and overshoot decline with increasing bandwidth *b*.

Table 2 below quantifies these relationships.

Table 2. Finite edge response for rectangular windows for continuous time

*Bandwidth*Risetime (br)*	*Edge Response at t = 0*	*Max Overshoot*
0.5	0.113	5.34%
0.75	0.056	2.41%
1	0.049	1.19%
2	0.025	0.72%
3	0.017	0.51%
5	0.010	0.32%
10	0.005	0.17%

As Fig. 9 and Table 2 show, a fairly good finite edge can be achieved with a *br* of 1, but a *br* of around 5 is needed to give a very close approximation to the ideal finite edge.

Impulse Response

The principles behind the step and edge responses also apply to the calculation of impulse responses. Rectangular windows produce the sharpest impulses, but with the greatest amount of oscillation. Hanning and Blackman windows produce impulses that are more spread out, but with less oscillation (Fig. 10).

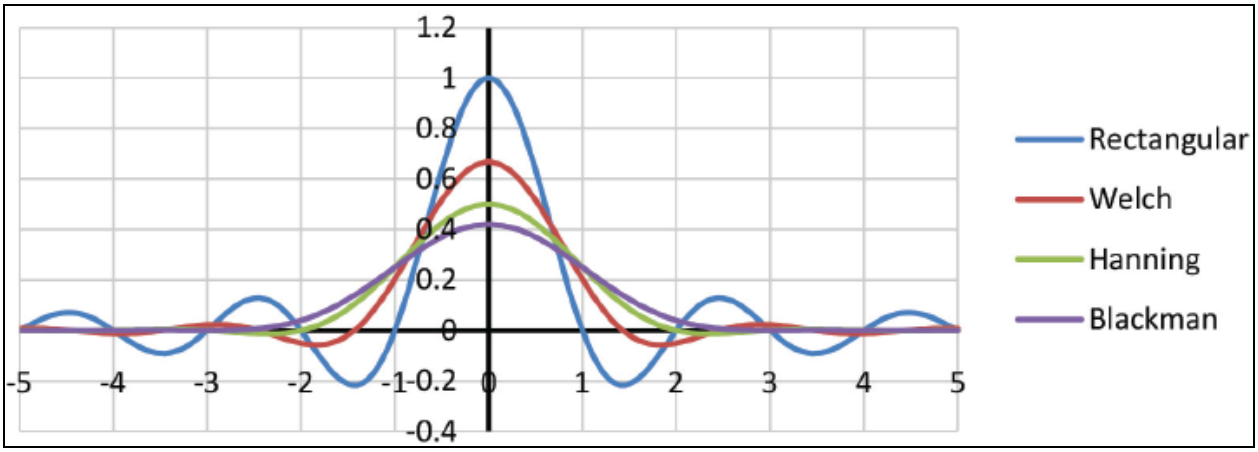


Figure 10. The impulse response for selected windows with a spectral width of 1

Discrete Time Domain Plotting

The preceding discussion treated frequency spectra as continuous functions, but in practice both the frequency and corresponding time data will be discrete. HFSS uses a discrete Fourier transform (DFT) to approximate a continuous time transform, with the frequency step size and upper limit determining the corresponding quantities in the time domain. The default time step and maximum time are given by

$$t_{step} = \frac{1}{2f_{max}} \qquad t_{max} = \frac{1}{2f_{step}}$$

Time resolution is controlled by the upper frequency in the sweep. The maximum time is controlled by the frequency resolution of the sweep. While t_{max} is fixed by the choice of frequency step and cannot be increased after the simulation, t_{step}, or the time delta, can be reduced from the default value within the **TDR Options** dialog box. Decreasing the time delta does not increase the bandwidth of the frequency data, but it does more closely approximate the band-limited continuous time spectra we have so far discussed. Although decreasing the time

delta will increase the time required to perform the DFT, the time required is rarely significant. Additionally, a smaller time delta has a significant benefit, as demonstrated in Fig. 11 below.

Fig. 11 shows the step response of a matched lossless transmission line for which the length is controlled by de-embedding the driving waveport, using rectangular window functions. The plots on the left are for a short transmission line length and those on the right correspond to a longer length. Fig. 11a shows the time response using the default values for t_{step}. There is some oscillation in the response, which is expected for a rectangular window, but the amplitude of the oscillation is different for the two length cases. This is problematic; since the line is matched and lossless, we expect that a length change will only affect the time delay of the response, not affect the shape or quality of the rising edge. The variation in the response is an undesirable artifact of the coarse time sampling. We can increase resolution by increasing the bandwidth of the sweep, but this requires additional simulation. Fig. 11b shows the same two cases, but with the time delta reduced using the TDR Options dialog box. The results in Fig. 11b agree with our intuition: the edge shape is the same for both line lengths and the only difference is the location of the edge. Setting the time delta to around 1/5 of the default value is generally sufficient, but finer time steps are needed for precise correlation to Tables 1 and 2.

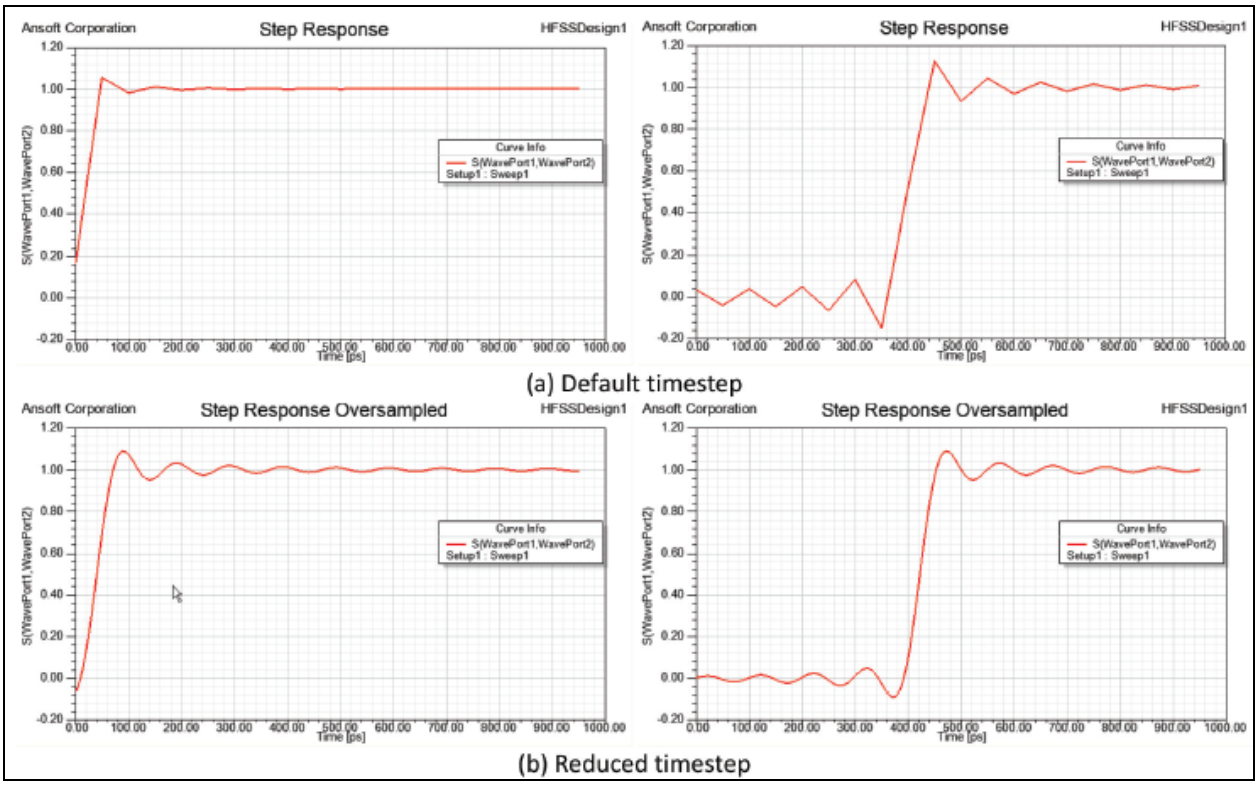


Figure 11. The time domain response of an ideal delay of two different lengths shows that a finer time sampling yields more intuitive results.

The frequency step size governs the length of the time range generated. Although a coarse frequency sampling is often sufficient to generate enough time data for a TDR plot, it is important not to set f_{step} too high in the frequency sweep. Discrete frequency spectra necessarily correspond to periodic time-domain functions, so the calculated step is actually more like a repeating series of long pulses. Fig. 12 shows the how the oscillation decays after the rising edge up to a point, but then begins increasing in anticipation of a falling edge.

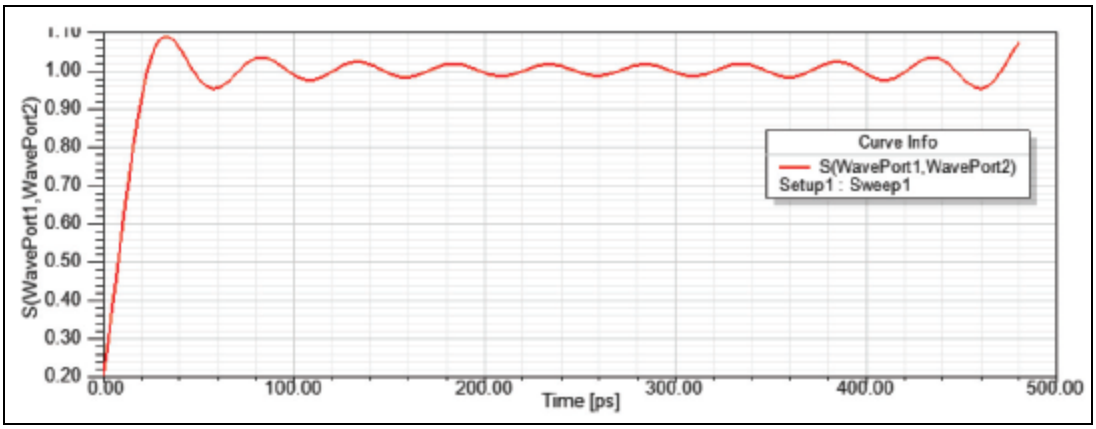


Figure 12. The oscillation caused by a rectangular window eventually starts increasing, due to the periodicity of the waveform

Setting f_{step} to a small value increases the length of the pulse, and minimizes the influence of the future falling edge. Additionally, a smaller f_{step} ensures that resonances and other sharp features in the frequency data are adequately captured. As t_{step} and f_{step} approach zero, the calculated results will converge on the continuous time descriptions given earlier.

Applications

When simulating a TDR plot, we want the fastest edge possible for the bandwidth of our simulation, subject to our preference for oscillation control. Therefore an edge with a rise time of zero is a good choice. Fig. 13 shows TDR plots of a transmission line with several impedance discontinuities. The results for a rectangular and Hanning window with a 20GHz bandwidth are compared with those for a Hanning window with a 50GHz bandwidth, which will necessarily be more accurate due to the higher bandwidth, and can be used as a reference. In all cases, the time step was set substantially lower than the default.

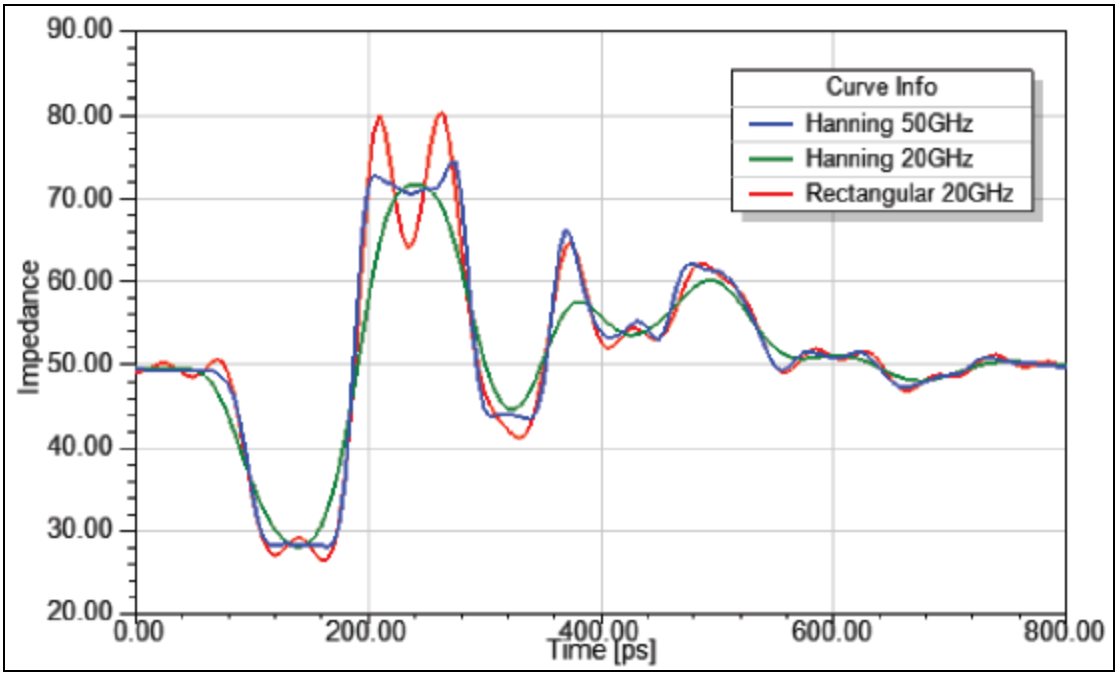
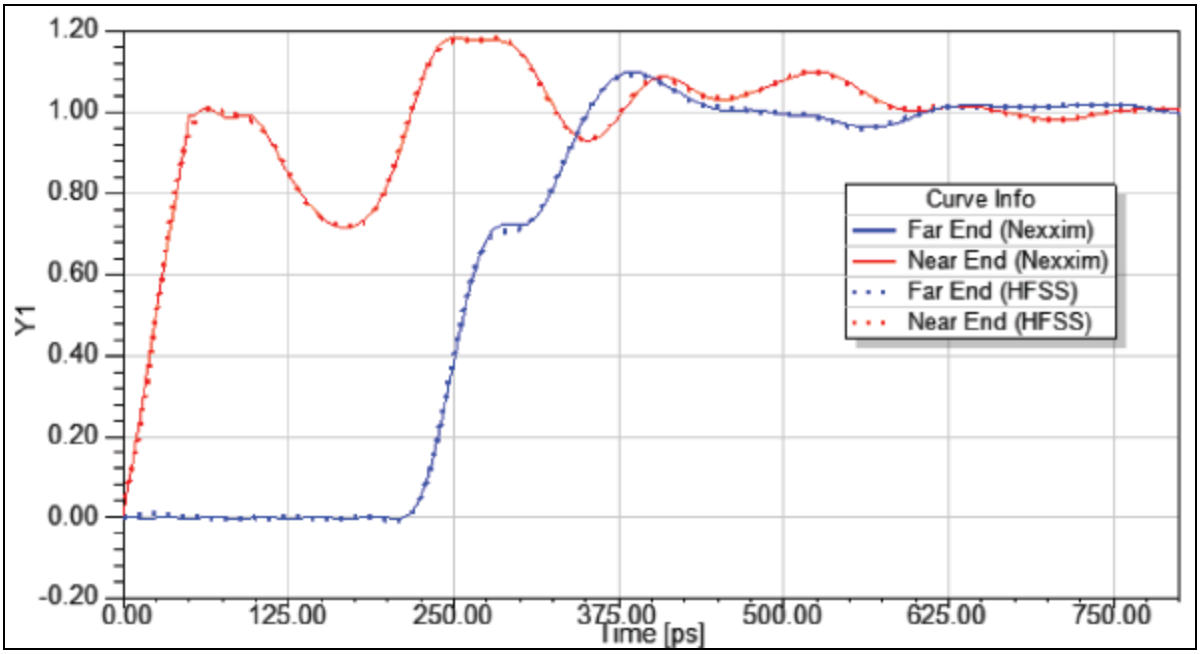


Figure 13. TDR plots for a transmission line with several impedance discontinuities

Fig. 13 shows that the rectangular window effectively captures the sharp impedance transitions, but also displays spurious oscillation. The 20GHz Hanning window does not suffer any oscillation, but gives less resolution on the sharp edges. These results are consistent with the step response characteristics of the different windows we have previously shown.

We can also use time-domain plotting to approximate how a structure would behave in a Nexxim transient simulation. When comparing the results to a transient simulation that uses a pulse or piecewise linear source, it makes sense to use a finite edge with a rectangular window. Fig. 14 compares HFSS and Nexxim results for the transmission line, using a rise time of 50ps and a rectangular window with a 20GHz bandwidth *(br* = 1).



As Fig. 14 shows, very good agreement between Nexxim and HFSS is possible when appropriate settings are used for time domain plotting.

References

Haykin, S., and M. Moher. *Introduction to Analog and Digital Communications*, 2nd ed., Wiley, Hoboken,N.J., 2007.

Kammler, D.W. *A First Course in Fourier Analysis*. Prentice-Hall, Upper Saddle River, N.J., 2000.

Lathi, B.P. *Linear Systems and Signals*, 2nd ed. Oxford University Press, New York, 2005.

Appendix: Window Function Formulas

Rectangular	$\text{Rect}\left(\frac{f}{w}\right)$ where $\text{Rect}(x) = \begin{cases} 1 & -\frac{1}{2} \le x \le \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$
Bartlett	$1 - \left\lvert\frac{2f}{w}\right\rvert$
Blackman	$0.42 + 0.5\cos\left(\frac{2\pi f}{w}\right) + 0.08\cos\left(\frac{4\pi f}{w}\right)$
Hamming	$0.54 + 0.46\cos\left(\frac{2\pi f}{w}\right)$
Hanning	$0.5\left[1 + \cos\left(\frac{2\pi f}{w}\right)\right]$
Kaiser	$\frac{1}{I_0(\alpha)} I_0\left(\alpha\sqrt{1 - 4\left(\frac{f}{w}\right)^2}\right)$ where $I_0(x)$ is a modified Bessel function of the first kind.
Welch	$1 - \left(\frac{2f}{w}\right)^2$

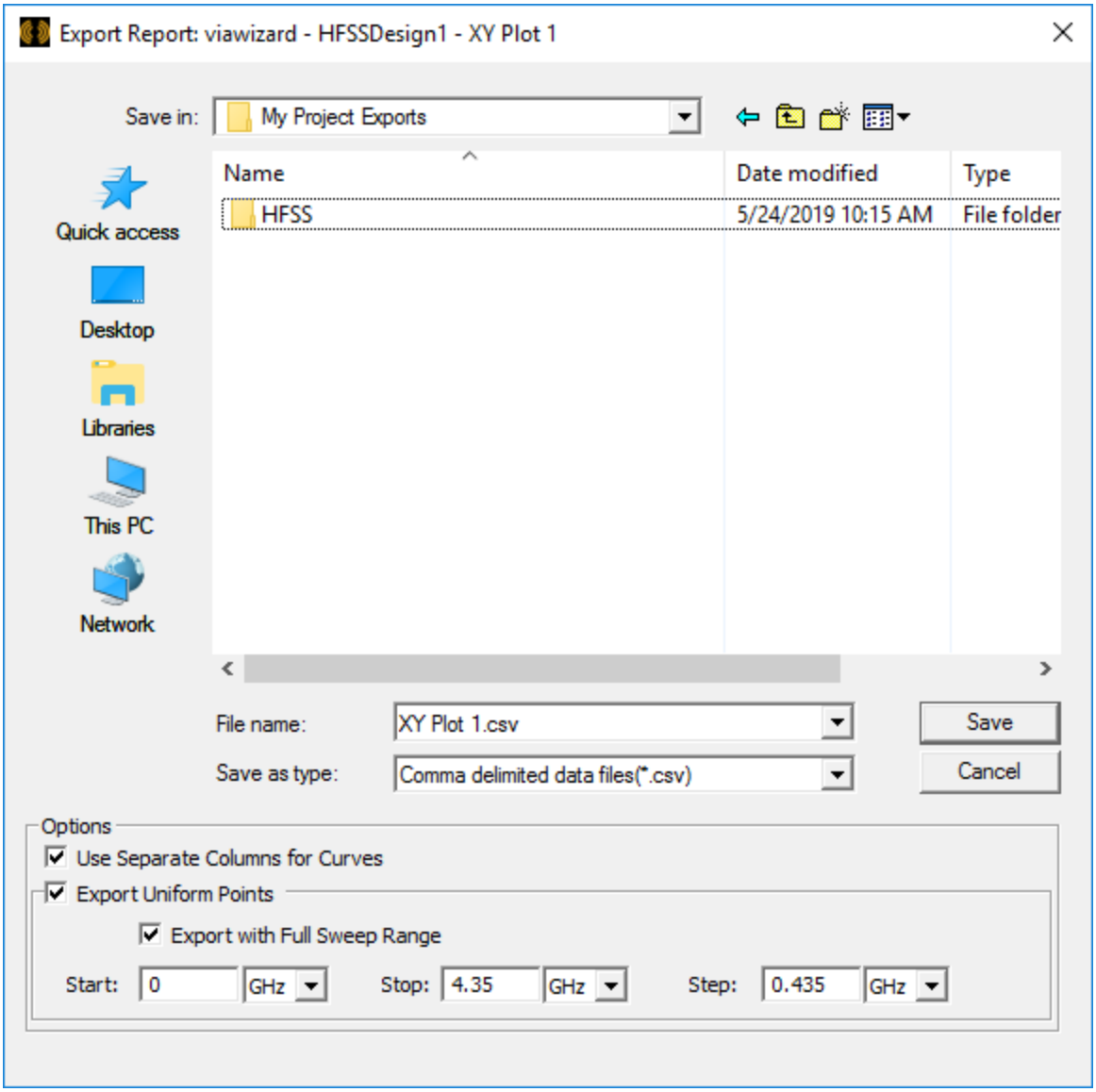
Exporting Reports

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**> **Export** or **Report3D**> **Export**.

   Alternatively, right-click the plot and select **Export**.

   The *Export Report* dialog box appears.

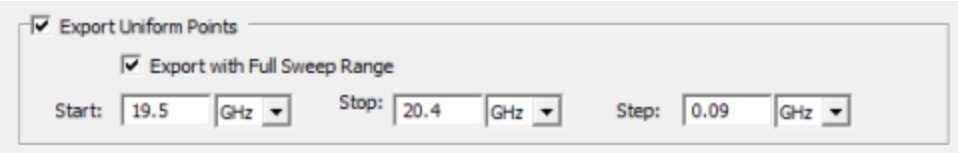


2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** field.
4. Select one of the following file formats from the **Save as type** drop-down menu:

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

Extension	Contents
.rdat	Ansoft report data file
.emf	Microsoft EMF files
.gif	GIF files
.bmp	BMP files
.wrl	VRML files
.tif, **.tiff**	TIFF files
.jpg, **.jpeg**	JPEG files

- When exporting to *.csv or .tab files for an XY Plot, Eye Diagram or Data Table, a **Separate Columns for Curves** check box will be shown. Otherwise the check box will be hidden.
- When it is checked the data is exported with one column per curve, and variable values appear in the column's header cell.
- When it is unchecked the data is exported with one column per a trace, each variable has its own column.
- When the design has many variables, uncheck the check box may keep the column header from becoming too long and hard to read.
- When trace has more curves than number of sweep values, uncheck the check box to prevent the number of columns from being larger than the number of rows.
- When a report only has one trace or all trace have same variations, this option is unchecked by default. Otherwise, this feature is checked.
- For 2D reports, you can select **Export Uniform Points** and specify a full sweep range by editing start, stop, and step values.



5. Click **Save**. The report is exported to the specified location.

Note:

For a report trace where the primary sweep is NOT the same as the x-component, exporting and then importing a report may not produce the same curve. In this circumstance, only *.rdat format imports the same curve/trace. For other formats, the import produces two separated traces.

Animated Reports

The following sections describe how you can postprocess field overlay displays to create and then view various animated reports. Any of the field overlay displays can be animated by cycling the overlays as a series of frames. Refer to the following topics for details on creating the field overlays:

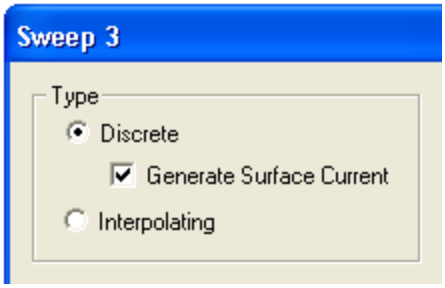
- Overlaying Surface Currents on a 3D View
- Overlaying Far Fields on a 3D View
- Overlaying Near Fields on a 3D View

There are two modes of animation: frequency-based and phase-based. Frequency-based animation creates a series of frames by calculating the field overlay at each of the swept frequencies. Phase-based animation creates a series of frames by calculating the field overlay at a fixed set of phase deviations around a selected frequency. The animation controls can also change the simulation design point.

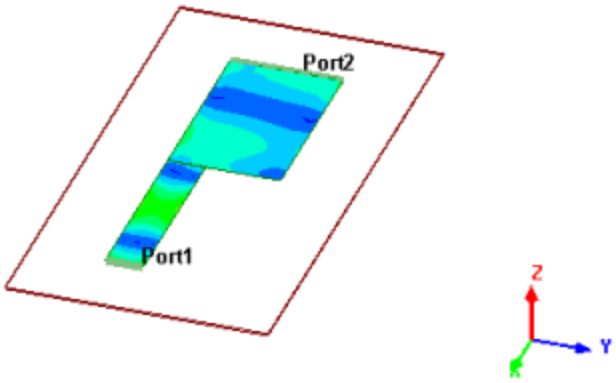
Overlaying Surface Currents on a 3D View

Surface currents calculated as the results of a Planar EM simulation can be displayed as overlays on the 3D viewer.

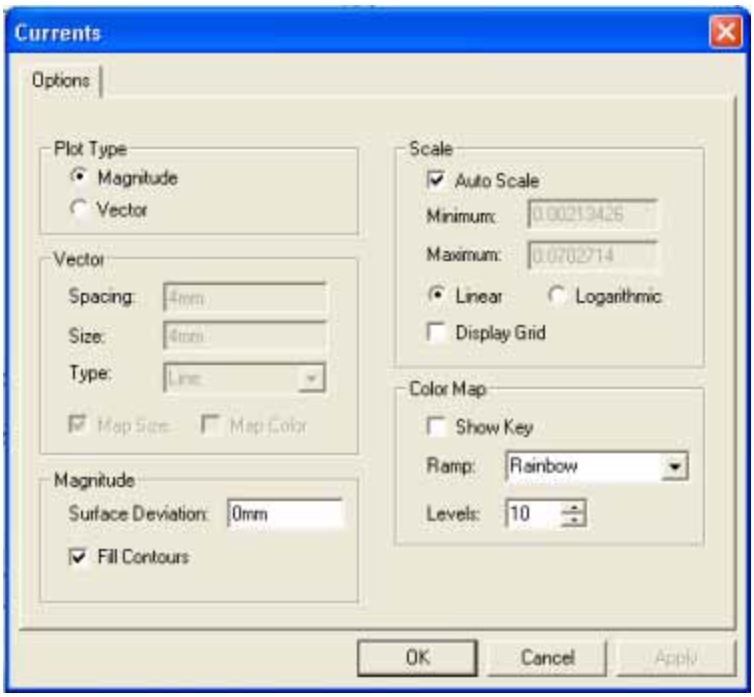
1. To ensure that the surface current information is generated, the sweep setup must specify a **Discrete** frequency sweep, and the **Generate Surface Current** option must be enabled (checked):



2. Run the Planar EM analysis with the sweep.
3. To display the surface current as an overlay, expand the **Analysis** icon in the Project window, and select **Setup***m* **> Sweep***n* **> Results > Display Currents** (*m* and *n* identify the particular solution setup and sweep setup, respectively). You can also select from a list of corresponding Setup/Sweep overlay choices which are displayed when you right-click **Field Overlays** in the **Project** tree. The 3D viewer window appears with the current values overlaid on the geometry:



4. To change the display properties of the surface current overlay, expand the **Results** icon in the Project window, and select **Setup*m*:Sweep*n*:Currents*k* > Properties** (*m*, *n*, and *k* identify the particular solution setup, sweep setup, and surface current setup, respectively). The **Currents** dialog opens:



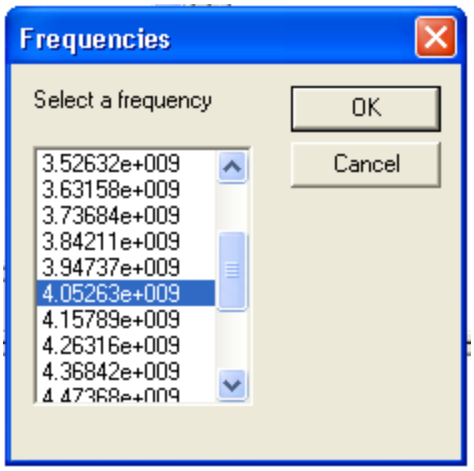
In the **Plot Type** panel, select **Magnitude** to enable the **Magnitude** panel options or select **Vector** to enable the **Vector** panel options.

In the **Scale** panel, select **Auto Scale** (the default), or deselect **Auto Scale** and enter custom **Minimum** and **Maximum** scaling values. Select **Linear** or **Logarithmic** scaling (the default is **Linear**), and toggle **Display Grid** on or off (the default is off).

In the **Color Map** panel, select the **Ramp** type (**Rainbow** is the default; other options are **HueScale**, **Magenta**, and **Temperature**), set the number of **Levels** (the default is 10 levels), and toggle the color key (**Show Key**) on and off (the default is off).

Click **Apply** to apply any changes to the display without closing the dialog box. Click **OK** to apply any changes and close the dialog box. Click **Cancel** to close the dialog without changing any options.

5. To select the frequency for the current overlay, expand the **Results** icon in the Project window, and select **Setup***m***:Sweep***n***:Currents***k* **> Frequency** (*m*, *n*, and *k* identify the particular solution setup, sweep setup, and surface current setup, respectively). The **Frequencies** dialog opens:



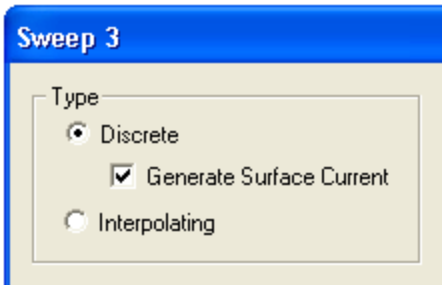
The list displays the frequencies that were swept in the analysis. When you select a frequency from the list, the overlay displays the surface current values calculated at that frequency. Click **OK** to leave the overlay at the selected frequency, or click **Cancel** to close the dialog without applying any frequency changes to the overlay.

6. To dismiss the overlay, expand the **Results** icon in the Project window, right-click **Setup***m***:Sweep***n***:Currents***k*, and select **Delete** from the drop-down menu **s** (*m*, *n*, and *k* identify the particular solution setup, sweep setup, and surface current setup, respectively).

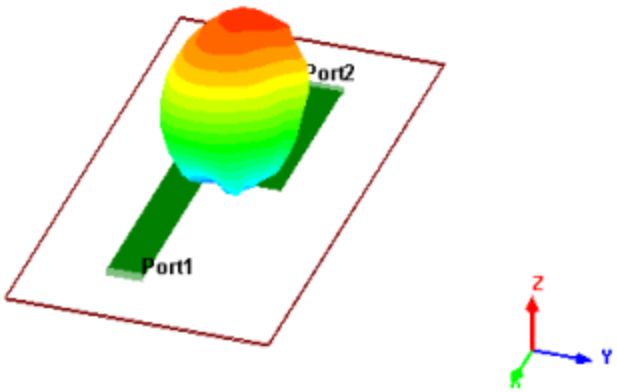
Overlaying Far Fields on a 3D View

Far fields calculated as the results of a Planar EM simulation can be displayed as overlays on the 3D viewer.

1. To ensure that the far field information can be generated, the sweep setup must specify a **Discrete** frequency sweep, and the **Generate Surface Current** option must be enabled (checked):



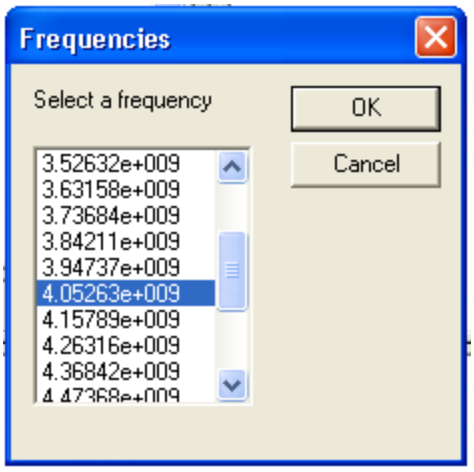
2. Run the Planar EM analysis with the sweep.
3. To display the far field as an overlay, expand the **Analysis** icon in the Project window, and select **Setup*m* > Sweep*n* > Results > Far Field** (*m* and *n* identify the particular solution setup and sweep setup, respectively). You can also select from a list of corresponding Setup/Sweep overlay choices which are displayed when you right-click **Field Overlays** in the **Project** tree. The 3D viewer window appears with the far field values overlaid on the geometry:



The display properties of the Far Field overlay cannot be changed. The Ramp type is Rainbow, and the number of levels is 20.

4. To select the frequency for the far field overlay, expand the **Results** icon in the Project window, and select **Setup*m*:Sweep*n*:Far Field*k* > Frequency**. The **Frequencies** dialog

opens:



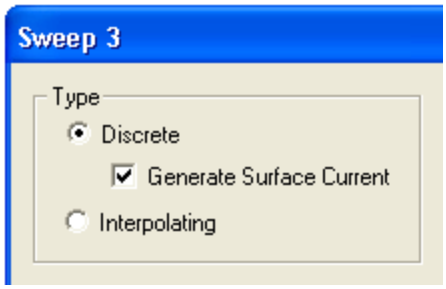
The list displays the frequencies that were swept in the analysis. When you select a frequency from the list, the overlay displays the far field values calculated at that frequency. Click **OK** to leave the overlay at the selected frequency, or click **Cancel** to close the dialog without applying any frequency changes to the overlay.

5. To dismiss the overlay, expand the **Results** icon in the Project window, right-click **Setup***m*:**Sweep***n*:**Far Field***k*, and select **Delete** from the drop-down menu (*m*, *n*, and *k* identify the particular solution setup, sweep setup, and far field setup, respectively).

Overlaying Near Fields on a 3D View

Near fields calculated as the results of a Planar EM simulation can be displayed as overlays on the 3D viewer.

1. To ensure that the near field information can be generated, the sweep setup must specify a **Discrete** frequency sweep, and the **Generate Surface Current** option must be enabled (checked):

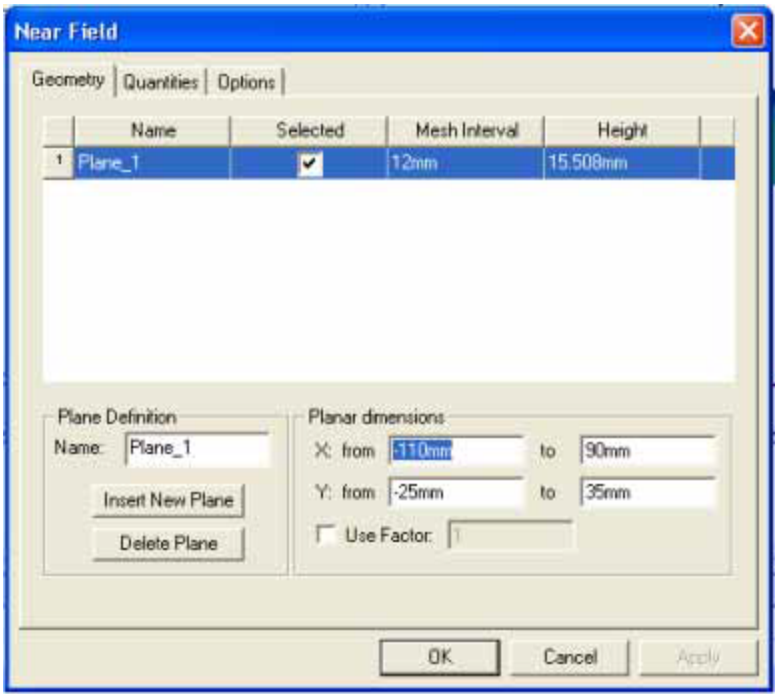


2. Run the Planar EM analysis with the sweep.

3. To display the near field overlay, expand the **Analysis** icon in the Project window, and select **Setup***m* **> Sweep***n* **> Results > Near Field** (*m* and *n* identify the particular solution setup and sweep setup, respectively). You can also select from a list of corresponding Setup/Sweep overlay choices which are displayed when you right-click **Field Overlays** in the **Project** tree.
4. The **Near Field** dialog opens. The dialog has three tabs, described below. At the bottom of each tab are three buttons:

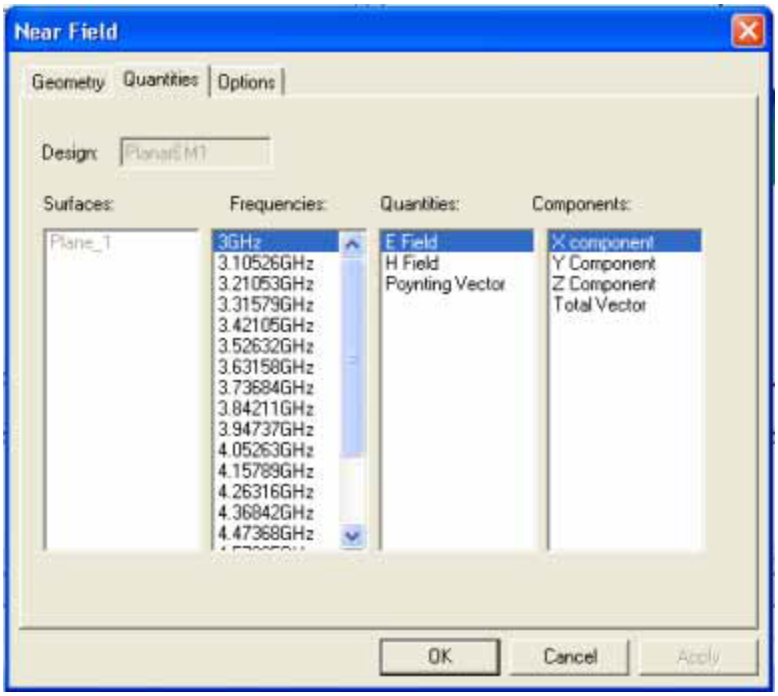
- **Apply** is activated whenever you change a value. Click **Apply** to start the display, and then to see the effect of each change. The dialog stays open.
- When no values were changed on any tab, the **OK** button starts the display. When one or more values have been changed, **OK** applies the changes. In either case, **OK** closes the dialog and adds an icon for the overlay under the **Results** icon in the Project window.
- **Cancel** is active as long as no changes have been applied. **Cancel** closes the dialog without changing any values. If the overlay is already displayed, it does not change. If **Cancel** is pressed before any overlay is displayed, the overlay is canceled.

5. The **Near Field** dialog opens with the **Geometry** tab displayed:



Use the options in the **Geometry** tab to select (or define) one or more planes for the calculation, including the dimensions to be used, and a scale factor if desired.

Use the **Quantities** tab to select the near field quantity to be calculated:



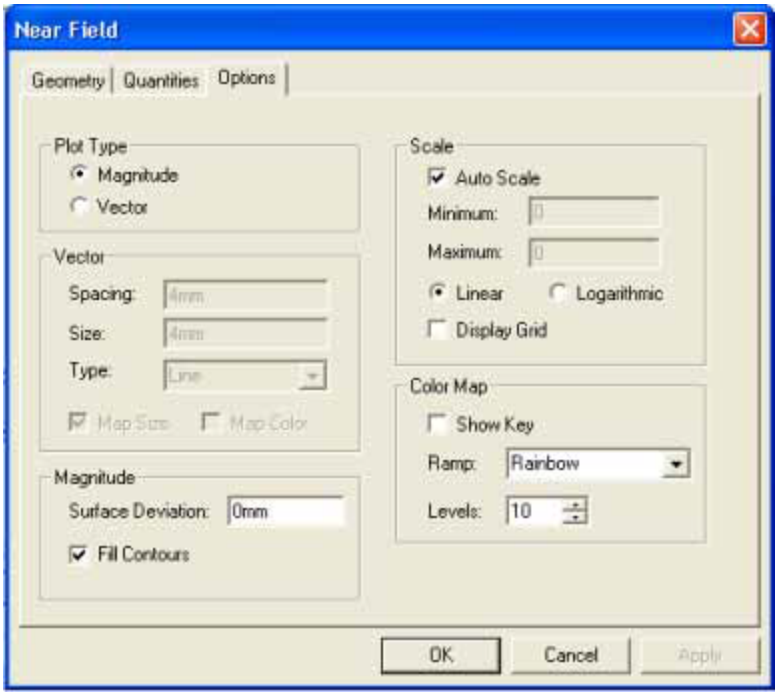
When more than one surface is involved, select a surface from the **Surfaces** list.

Select a frequency from the **Frequencies** list. The frequencies are the ones swept in the analysis.

Select a field type from the **Quantities** list. **E** is the electronic field, **H** is the magnetic field, and the **Poynting Vector** is the (E×H*) field, where H* is the complex conjugate of the H matrix.

Select a vector component from the **Components** field.

Use the **Options** tab to specify the display options for the near field overlay:

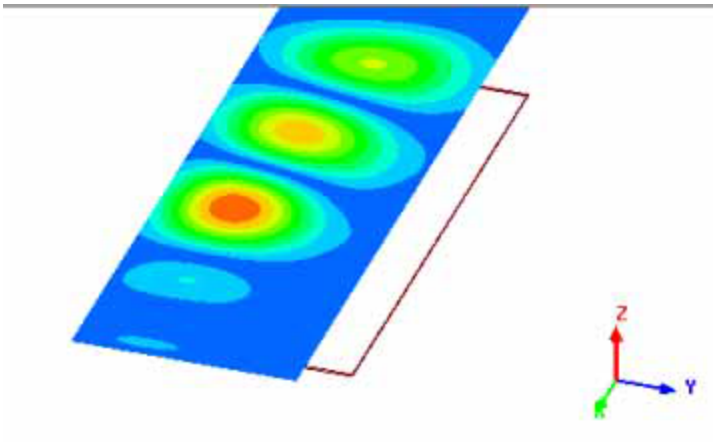


In the **Plot Type** panel, select **Magnitude** to enable the **Magnitude** panel options or select **Vector** to enable the **Vector** panel options.

In the **Scale** panel, select **Auto Scale** (the default), or deselect **Auto Scale** and enter custom **Minimum** and **Maximum** scaling values. Select **Linear** or **Logarithmic** scaling (the default is **Linear**), and toggle **Display Grid** on or off (the default is off).

In the **Color Map** panel, select the **Ramp** type (**Rainbow** is the default; other options are **HueScale**, **Magenta**, and **Temperature**), set the number of **Levels** (the default is 10 levels), and toggle the color key (**Show Key**) on and off (the default is off).

6. When you click **Apply** or **OK** in any of the **Near Field** dialog tabs, the 3D viewer window appears with the near field values overlaid on the geometry:

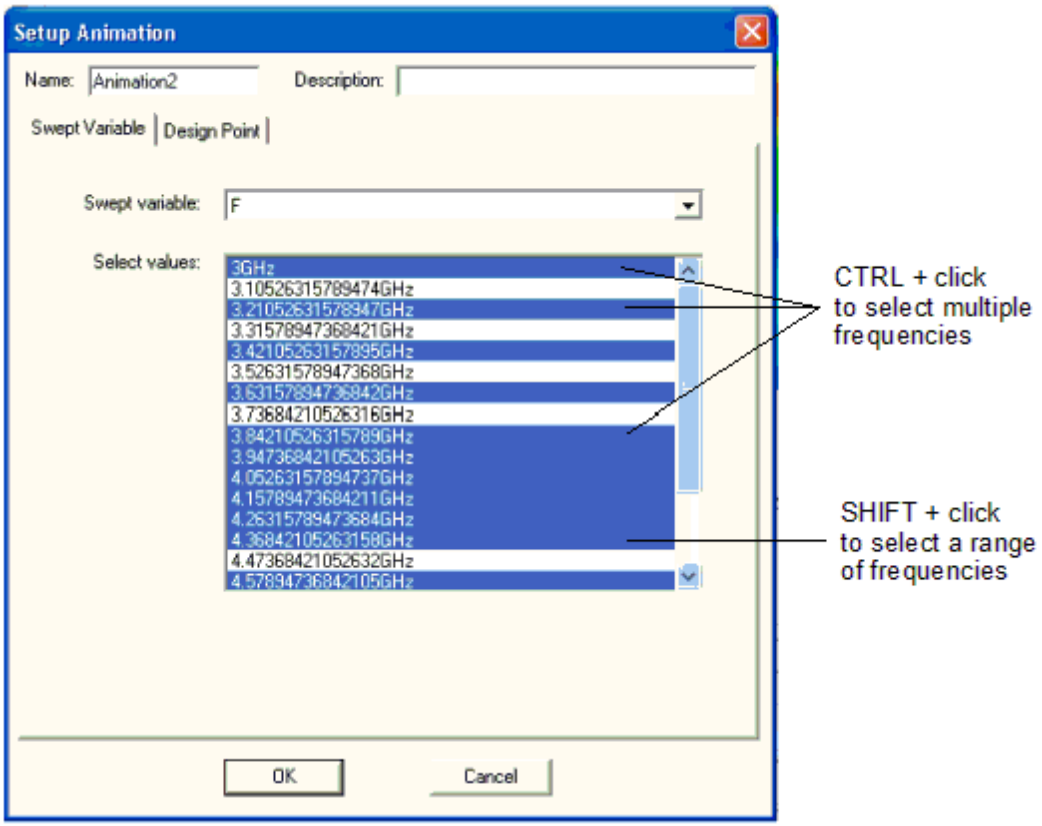


7. To dismiss the overlay, expand the **Results** icon in the Project window, right-click **Setup*m*:Sweep*n*:Near Field*k***, and select **Delete** from the drop-down menu.

Frequency Animation

1. To initiate the animation of the overlay that is currently displayed, do one of the following:
   - Select **Animate** from the **View** menu.
   - Expand the **Results** icon in the Project window, right-click the overlay entry, and select **Animate** from the menu.

2. If no animations have been defined previously, the **Setup Animation** dialog opens:



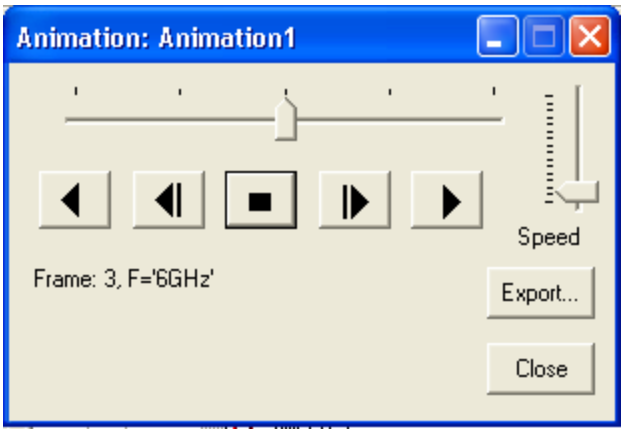
- Specify a name in the **Name** field (or accept the default, **Animation***n*, where *n* is a numeral). Optionally, enter a description.
- For a frequency animation, select **F** as the **Swept Variable**.
- By default, all the frequencies are selected (highlighted). Hold down the Ctrl key to select multiple individual frequencies, or hold down the Shift key to select a contiguous range of frequencies. [These selection modes are illustrated in the dialog example above].
- Click **OK**.

If one or more animations have been defined, selecting **Animate** from one of the menus opens the **Select Animation** dialog:

The **Select Animation** dialog provides the following operations:

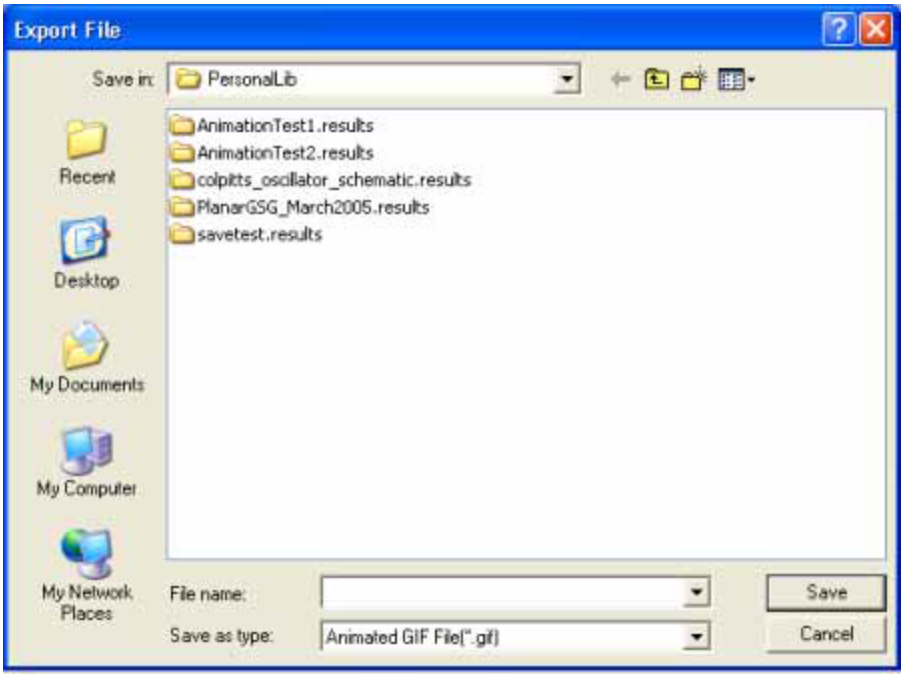
- Click to select one of the animations and click **OK** to start that animation.
- Click **New** to open the **Setup Animation** dialog described above, and close the **Select Animation** dialog.
- Select an animation and click **Delete** to delete the definition.

3. The frame data is automatically calculated. If the Progress window is displayed, you can monitor the progress of the calculation. When the frames have been calculated, the animation begins and the **Animation** control panel opens:



- Use the “VCR” buttons to play the animation. From left to right, the buttons are **Reverse**, **FastReverse**, **Stop**, **Fast Forward**, and **Forward**. The indicator at the top of the dialog shows the progress of the animation. Use the **Speed** slider to control the speed of the animation.

- To export the frame data to a file, click **Export**. The **Export File** dialog opens:



Specify the directory and file name. Use the **Save as type** menu to select the file format (Animated GIF or AVI). Click **Save** to save the data and close the dialog.

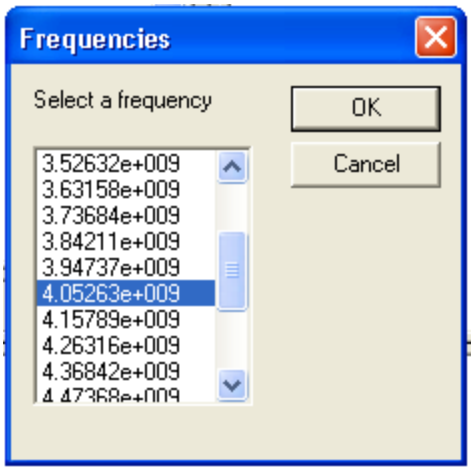
- Click **Close** on the **Animation** control panel to stop the animation and close the panel.

Phase Animation

1. To prepare for phase animation, you must select the base frequency.

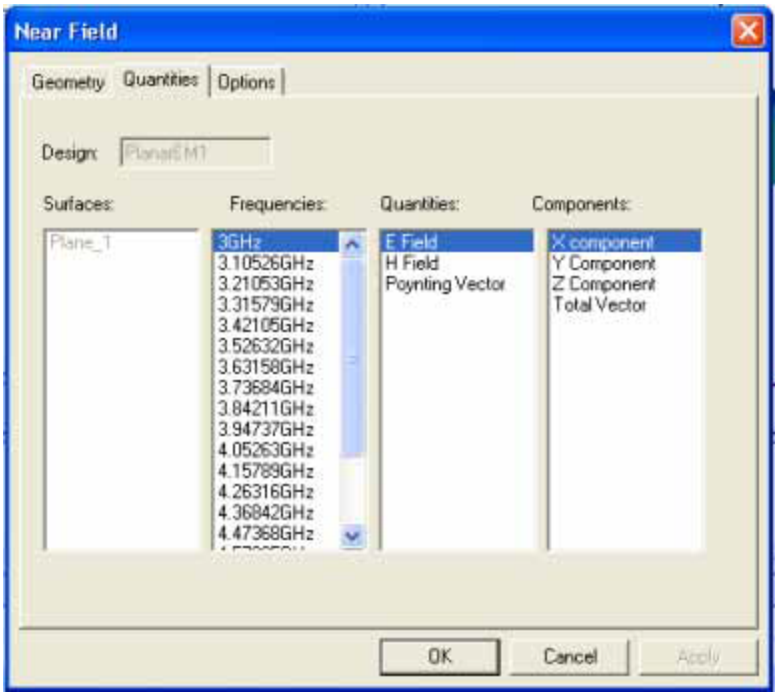
- For surface current and far field overlays, expand the **Results** icon in the Project window, right-click the overlay, and select **Frequency** from the menu. The **Frequencies** dialog

opens:



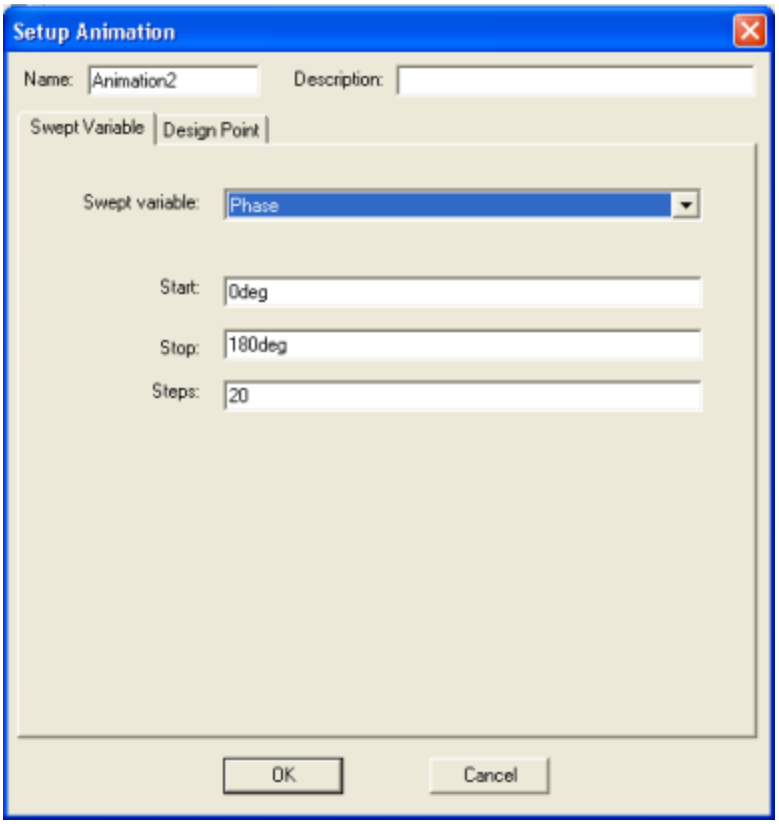
The list displays the frequencies that were swept in the analysis. Select a frequency from the list; the overlay displays the field values calculated at that frequency. Click **OK** to leave the overlay at the selected frequency.

- For near field overlays, expand the **Results** icon in the Project window, right-click the overlay, and select **Properties** from the menu. Use the **Quantities** tab to select a reference frequency for the phase animation:



Click **OK** to apply the frequency and close the dialog.

2. To initiate the animation of the overlay that is currently displayed, do one of the following:
   - Select **Animate** from the **View** menu.
   - Expand the **Results** icon in the Project window, right-click the overlay entry, and select **Animate** from the menu.
3. If no animations have been defined previously, the **Setup Animation** dialog opens:



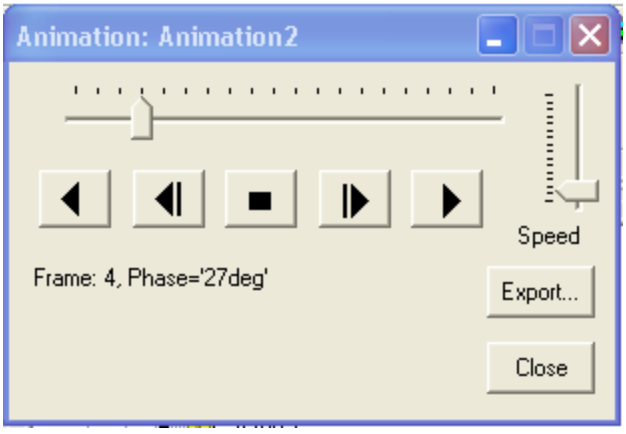
   - Specify a name in the **Name** field (or accept the default, **Animation***n*, where *n* is a numeral). Optionally, enter a description.
   - For a frequency animation, select **Phase** as the **Swept Variable**.
   - Select the Start and Stop phases in degrees.
   - Click **OK**.

If one or more animations have been defined, selecting **Animate** from one of the menus opens the **Select Animation** dialog:

The **Select Animation** dialog provides the following operations:

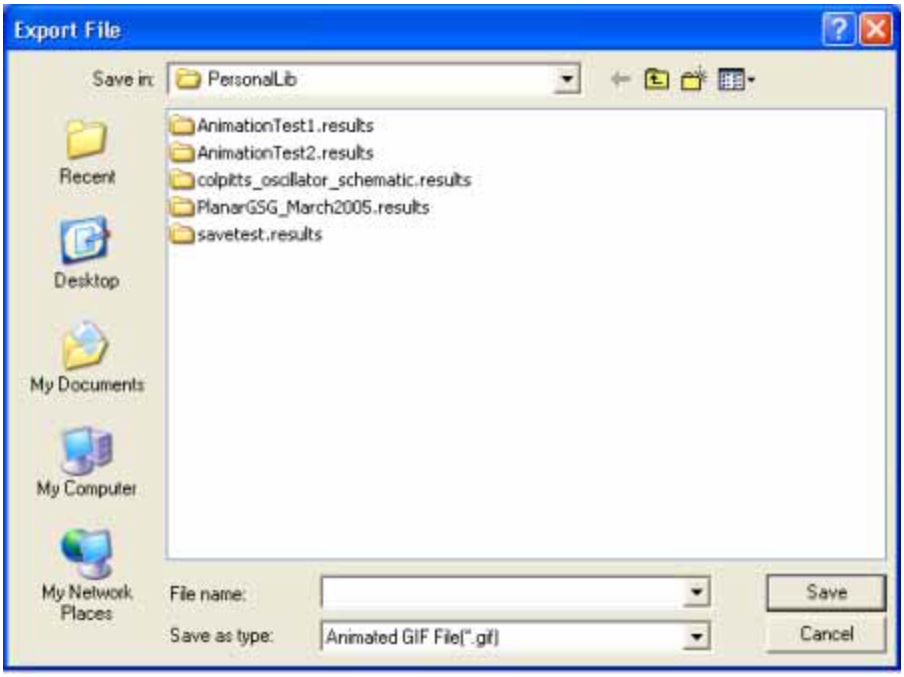
- Click to select one of the animations and click **OK** to start that animation.
- Click **New** to open the **Setup Animation** dialog described above, and close the **Select Animation** dialog.
- Select an animation and click **Delete** to delete the definition.

4. The frame data is calculated automatically. If the Progress window is displayed, you can monitor the progress of the calculation. When the frames have been calculated, the animation begins and the **Animation** control panel opens:



- Use the “VCR” buttons to play the animation. From left to right, the buttons are **Reverse**, **FastReverse**, **Stop**, **Fast Forward**, and **Forward**. The indicator at the top of the dialog shows the progress of the animation. Use the **Speed** slider to control the speed of the animation.

- To export the frame data to a file, click **Export**. The **Export File** dialog opens:

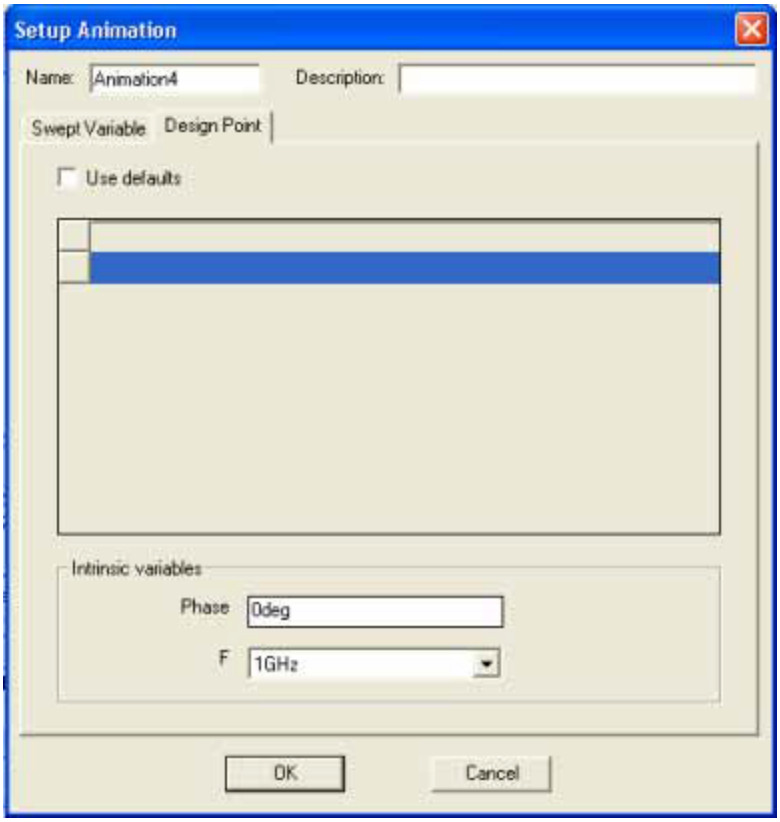


Specify the directory and file name. Use the **Save as type** menu to select the file format (Animated GIF or AVI). Click **Save** to save the data and close the dialog.

- Click **Close** on the **Animation** control panel to stop the animation and close the panel.

Changing the Design Point

Calculating frames for an animation is equivalent to re-simulating the planar design. You can specify design point parameters for the animation calculations that are different from the ones used in the original simulation. On the **Animation Setup** dialog box, select the **Design Point** tab and deselect the **Use defaults** option. The following fields are displayed:

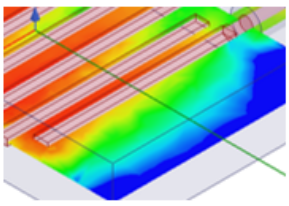


Make any desired changes, and then click **OK** to apply the changes and close the dialog box. Clicking **Cancel** closes the dialog without making any changes.

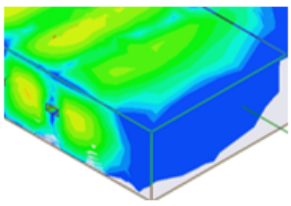
Field Overlays in Q3D Extractor and 2D Extractor

Field overlays are representations of basic or derived field quantities on surfaces or objects for the current design variation.

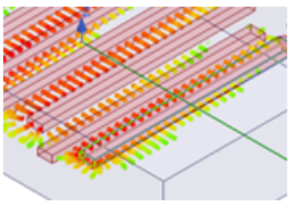
See the examples below:



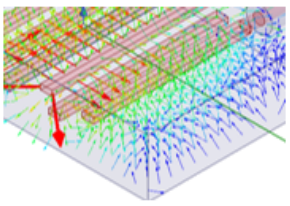
Mag E on Surface



Mag E on Solid



Vector E Streamline



Vector E

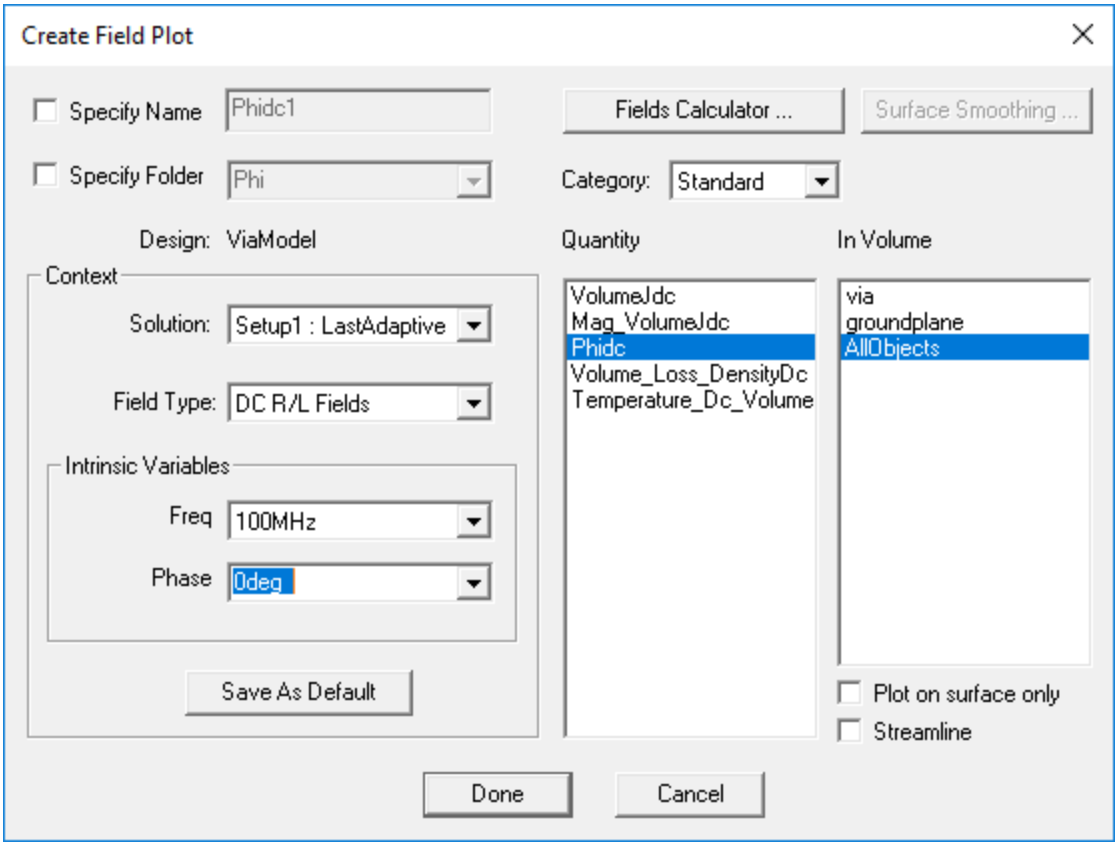
Adding Field Overlays

The general process for plotting a field overlay is:

1. Select a geometry (point, line, plane, cutplane, surface, object).
2. In the **Project Manager**, right-click **Field Overlays**.
3. Select [**Solver**] **Fields** > [**Desired Field**].

> **Important:**
>
> - You must have selected **Save Fields** in the solution setup in order to enable the **Plot Fields** selections.
> - The available options depend on the design, the selected geometry, and solutions.

The **Create Field Plot** window appears.

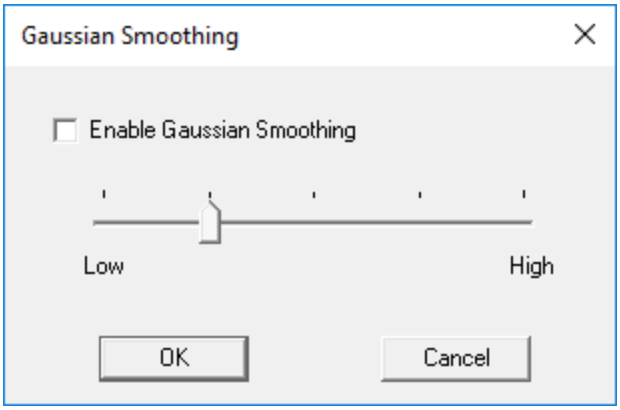


This window contains the following options:

- **Specify Name** – choose a name for the field overlay, or leave the default.
- **Specify Folder** – select the folder under which the overlay will appear in the Project Manager. All field plots under a folder share the same color key.
- **Category** – from this drop-down menu, select either **Standard** to use standard quantities or **Calculator** and click **Fields Calculator** to specify calculated quantities.
- **Context** – select the **Solution**, **Field Type**, and **Intrinsic Variables** to alter the available quantities in the **Quantity** list.
- **Quantity** – list of either Standard quantities or Calculator quantities, depending on the **Category** selection.

> **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 return 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.
> - 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. The list will not contain scalar values when a line is selected as the geometry.

- **In Volume** – 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.
- **Plot on surface only** (*Q3D Extractor only*) – select to plot on the surface only. Selecting this option also enables the **Surface Smoothing** button, from which you can enable Gaussian smoothing.

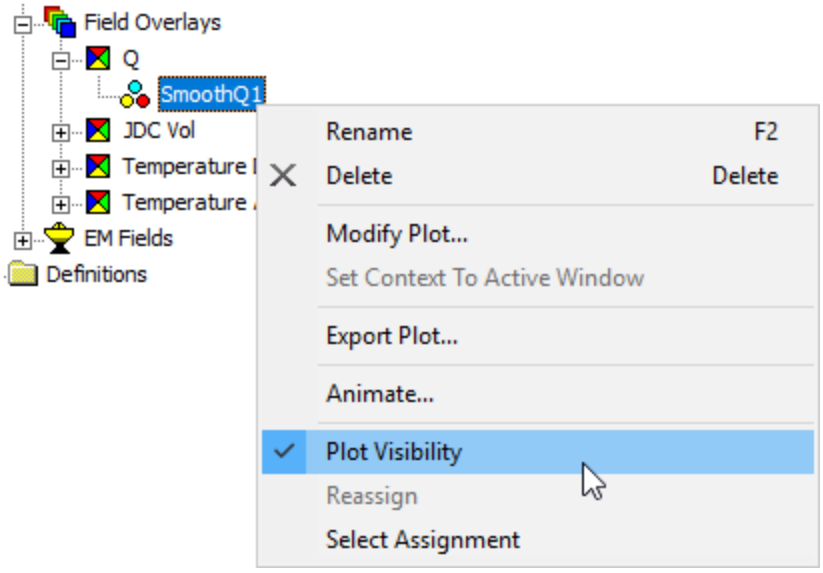


- **Plot on edge only** (*2D Extractor only*) – select to plot on the edge only. Selecting this option *disables* the **Surface Smoothing** button.
- **Streamline** – select to display the overlay as streamlines. These are often used to indicate magnetic flux lines, etc.

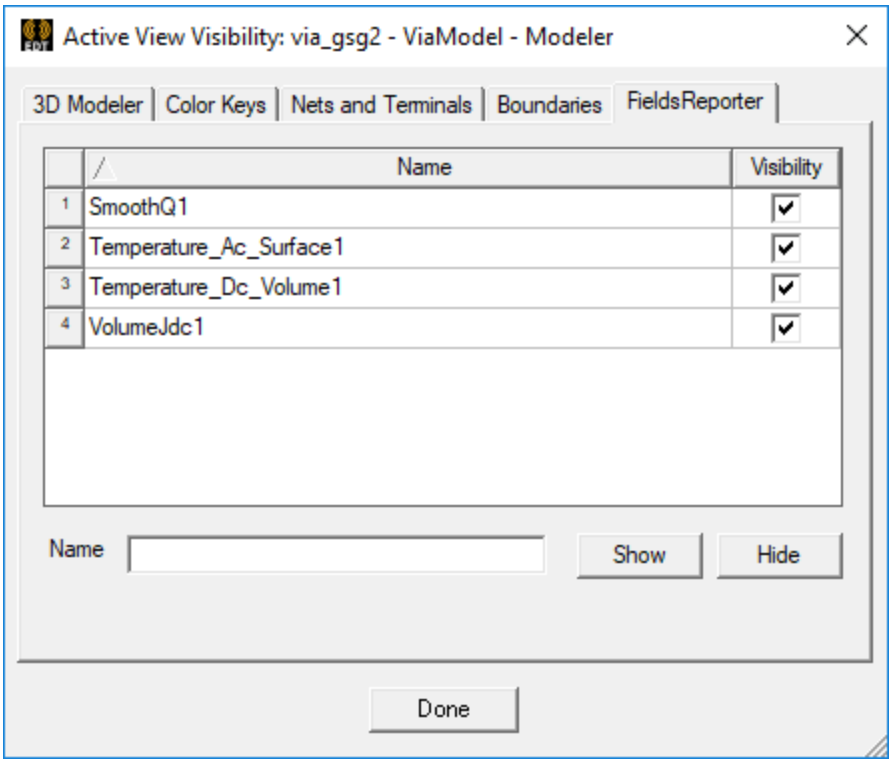
4. Select the desired options and click **Done**.
5. The field overlay appears under **Field Overlays** in the **Project Manager**.

Viewing Field Overlays

Once you have added a field overlay, control its visibility by right-clicking it and selecting or deselecting **Plot Visibility**.



Alternately, navigate to **View** > **Visibility** > **Active View Visibility** to open the **Active View Visibility** dialog box. The **Fields Reporter** tab lists all field overlays in the design and you can select and deselect their **Visibility**.



Modifying Field Overlays

To modify a field overlay, right-click it in the Project Manager and select **Modify Plot**.

The **Modify Field Plot** window appears, with the same options as the **Create Field Plot** window.

Make the desired changes and click **Apply**.

List of Field Quantities in Q3D Extractor

The following field quantities are available in Q3D Extractor, depending on the solution.

CG Fields

Q	
SmoothQ	Smoothed surface charge density for CG problems in Q3D. A signed quantity measured in Coulombs/meter^2.

ABS_Q	Absolute value of SmoothQ.
E	
Vector_E	The electric field. This is a complex vector quantity.
Mag_E	The magnitude of E.
Other	
EM Fields	See: Plotting EM Fields.

DC R/L Fields

Phidc	Voltage (electrostatic potential) from the DC solution, referenced to the sink terminal of the net.
VolumeJdc	DC current density vector in the conductor, measured in Amps/meter^2.
Mag_VolumeJdc	Magnitude of VolumeJdc.
ComplexMag_VolumeJdc	Complex magnitude of VolumJdc containing real and imaginary components.
Volume_Loss_DensityDc	The resistive (ohmic) power loss density from the DC solution, and is defined as: $\frac{1}{\sigma}\left\|f_{\mathrm{dc}}\right\|^2$ Where σ = conductivity of the metal.
Temperature_Dc_Volume	Temperature distribution mapped from a thermal solver such as Ansys Mechanical.
Harmonic_Loss_Density	Harmonic loss density. See: Electrothermal Flow for Power Electronics.

DC R/L PEC Fields

SurfaceJdc	DC surface current density vector on the surface of a PEC conductor, measured in Amps/meter.
Mag_SurfaceJdc	Magnitude of SurfaceJdc.
ComplexMag_SurfaceJdc	Complex magnitude of SurfaceJdc including real and imaginary components.

DC R/L Thin Conductor Fields

JdcThinCond	DC surface current density vector in a thin conductor, measured in Amps/meter.
Mag_JdcThinCond	Magnitude of JdcThinCond.
ComplexMag_ JdcThinCond	Complex magnitude of JdcThinCond including real and imaginary components.
PhidcThinCond	Electrostatic potential in a thin conductor.

AC R/L Fields

J	
SurfaceJac	Surface current density vector from AC RL solution, measured in Amps/meter.
Mag_SurfaceJac	Magnitude of SurfaceJac vector.
ComplexMag_SurfaceJac	Complex magnitude of SurfaceJac including real and imaginary components.
H	
Mag_H	Magnitude of H.
ComplexMag_H	Complex magnitude of H including real and imaginary components.
Vector_H	The magnetic field. This is a complex vector quantity.
Other	
Surface_Loss_DensityAC	Resistive power loss density from the AC RL solution, defined as: $0.5\sqrt{\frac{\mu_0 \pi f}{\sigma}}\left\|\mathbf{J_{ac}}\right\|^2$
Temperature_AC_Surface	Surface temperature distribution mapped from a thermal solver such as Ansys Mechanical.

List of Field Quantities in 2D Extractor

The following field quantities are available in 2D Extractor, depending on the solution.

CG Fields

Phi	
Mag_Phi	The magnitude of Phi.

PhiAtPhase	The real part of Phi at a particular value of phase.
E	
VectorE	The electric field. This is a complex vector quantity.
Mag_E	The magnitude of E.
J	
Mag_Jcg	The magnitude of Jcg.
VectorJcg	The conduction current density vector. This is the material conductivity multiplied by the electric field vector E.
Other	
EnergyCG	The energy density of the electric fields.

R/L Fields

A	
VectorA	The magnetic vector potential. This is a complex vector quantity.
Flux_Lines	A plot showing contours of constant magnitude of A.
B	
VectorB	The magnetic flux density vector. This is a complex vector quantity.
Mag_B	The magnitude of B.
H	
VectorH	The magnetic field vector. This is a complex vector quantity.
Mag_H	The magnitude of H.
J	
VectorJrl	The conduction current density.
Jrl	The magnitude of VectorJrl.
Other	
Emloss	In RL solution, emloss combines the ohmic loss and dielectric loss. Emloss is represented by: $\frac{1}{2}\omega Im\left(\tilde{B} \bullet H\right) + \frac{1}{2} Re\left(\tilde{J} \bullet E\right)$

EdgeLossDensity	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." It is represented by: $Re\left(Z_{\text{surf}}\right)\left\|n \times H\right\|^2$
EnergyRL	The energy density of the magnetic fields, as represented by: $\frac{1}{4} Re\left(\widetilde{H} \bullet B\right)$
Temperature_RL	Temperature distribution imported from a thermal solver such as Ansys Mechanical.

Setting Field Overlay Attributes

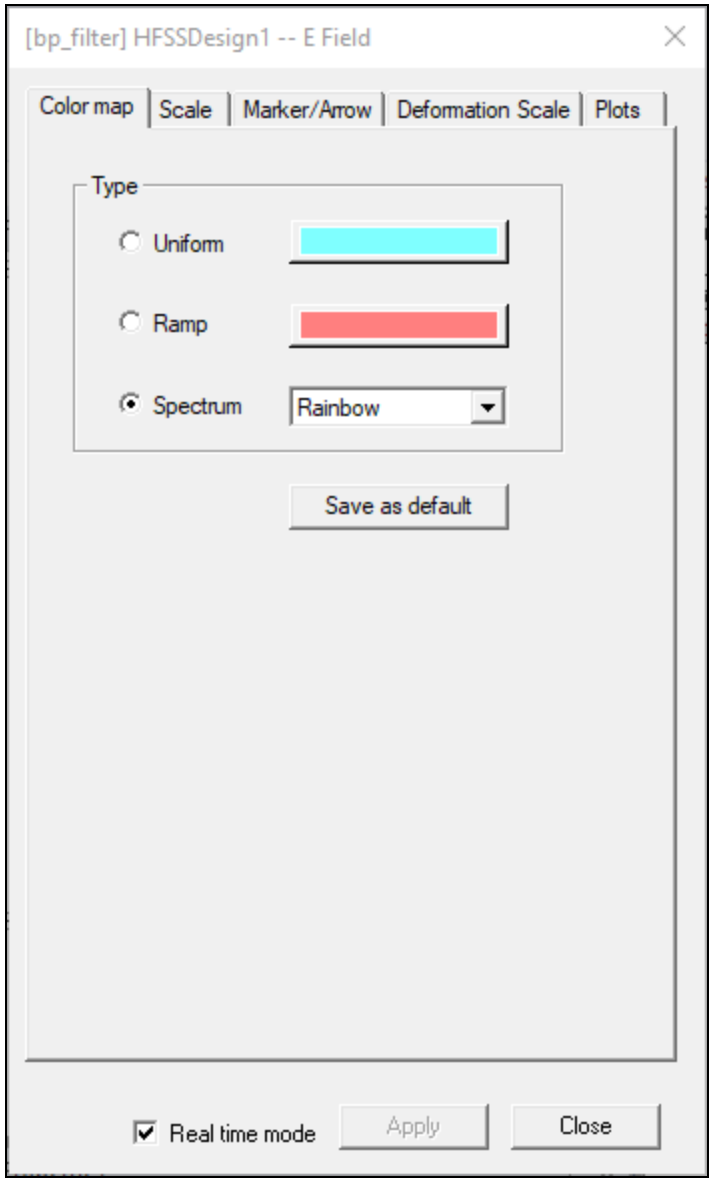
After creating a mesh or field overlay on a surface or volume, you can modify its appearance by changing the settings in the **Plot Attributes** dialog box. You will modify the settings for a plot folder and all plots in that folder will use the same attributes.

1. Click **Q3D Extractor** > **Fields** > **Modify Plot Attributes**. Alternatively, right-click **Field Overlays** in the Project Manager and select **Modify Attributes** from the shortcut menu.

   The **Select Plot Folder** dialog box appears.

2. In the **Select Plot Folder** dialog box, select the plot type that you want to modify and click **OK**. You can also right-click the specific plot in the Project Manager and choose **Modify Attributes** from the shortcut menu.

   A dialog box with attribute settings for the selected plot type (whether a field result or a mesh plot) appears.



3. For field result overlays (not mesh plots), the available plot attributes under each tab in the dialog box are listed in the following table:
    - **Color map:**

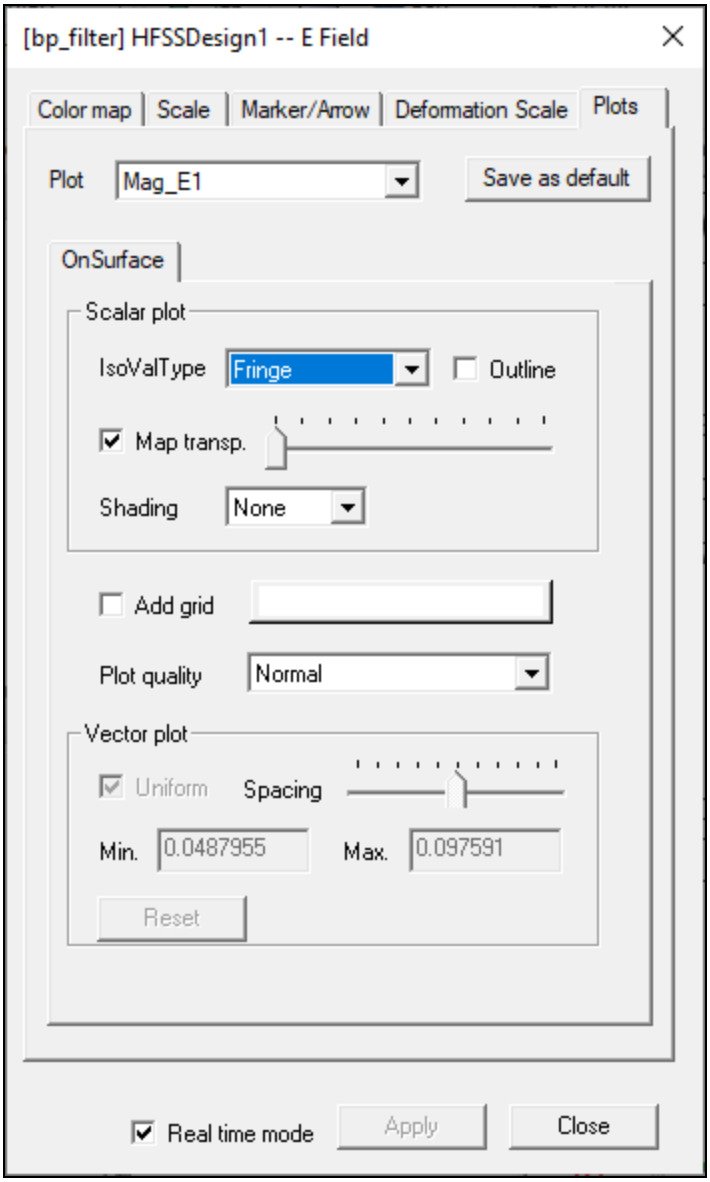
      The number of colors used and how they are displayed. The field data must be available for the color key to appear.

    - **Scale tab:**

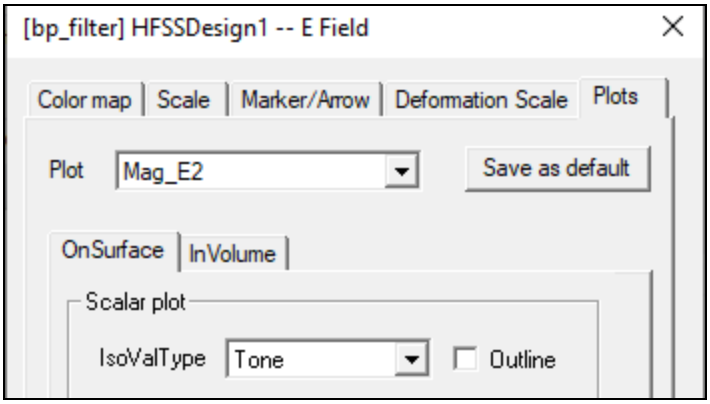
The scale of field quantities, including the number of divisions in the scale, whether to use dB as the units, whether to use a linear or log scale, auto schall options, and plot number format.

- **Marker/Arrow**
  - The appearance of points (for scalar point plots).
  - The appearance of arrows (for vector plots).
  - Magnitude filtering (for vector plots). That is, you uncheck Map size, and specify a Min and Max Magnitude, or use a slider to set the Min threshold.
- **Deformation Scale:**

  This is for use with plots that include Stress feedback from Ansys Workbench Integration.
- **Plots (if not vector or streamline):**
  - The **Plot** selected. If multiple plots are available, you can select from a drop down menu. Plots can be OnSurface or InVolume. If both types are available, the dialog includes tabs for each.

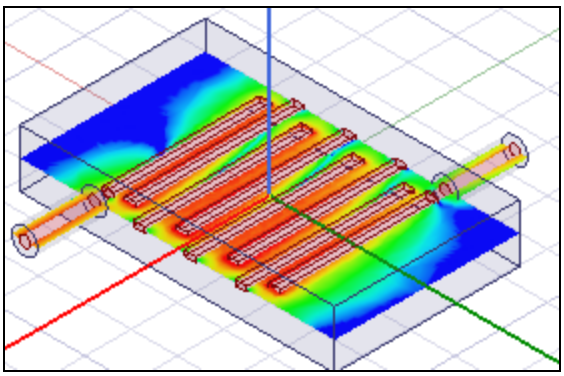


- Plots can be **OnSurface** or **InVolume**. If both types are available, the dialog includes tabs for each.

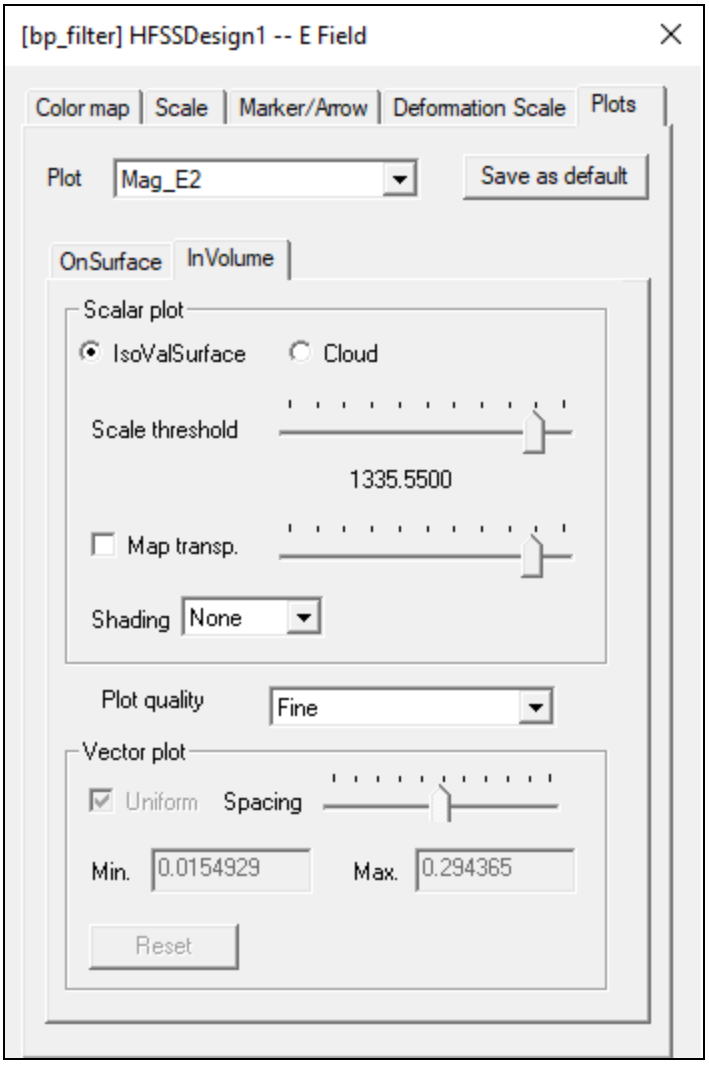


- The type of isovalue display (for scalar plots.) For *Line, Fringe* and *Tone* **IsoValType**, *Outine* is enabled. For *Gourad*, it is disabled. The *IsoValue* display selection also affects *InVolume*, which also has other options discussed below.
- Whether to use Shading (for scalar plots), if lighting 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**.

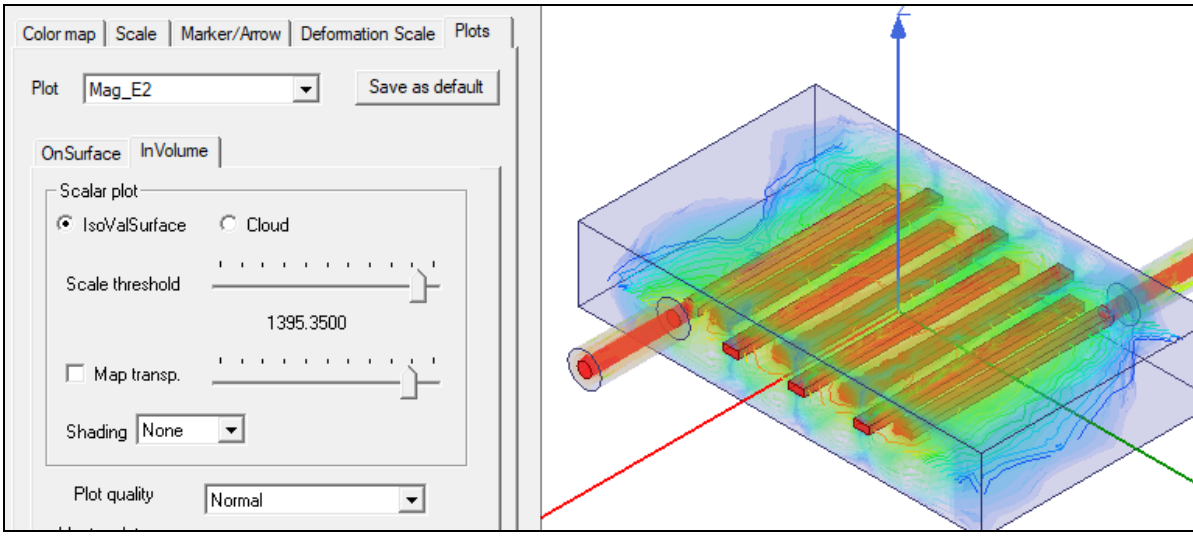
  This affects the use of memory for animating plots. For large plots with more frames to animate, use **Coarse** or **Normal** to reduce memory requirements and improve performance. For smaller plots with few frames, if higher resolution is required, use **Fine** or **Very Fine**.



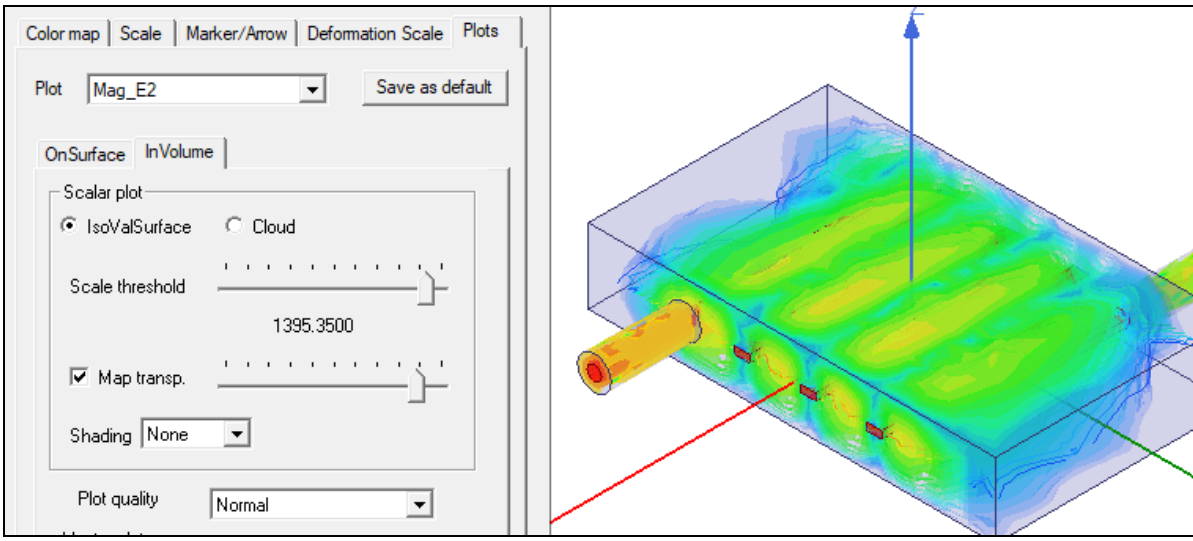
If an InVolume plot is available, and you select the tab, additional display options are available to allow visibility of internal layers and objects, particularly those with high field values. For *IsoValSurface* for an *InVolume* Scalar plot, the choices are for **Scale threshold** and **Map transp**.



- The "**Scale threshold**" slider bar allows you to set a field value threshold so that iso surfaces with field value lower than the given threshold will show as translucent with given transparency specified by the slider bar below it.

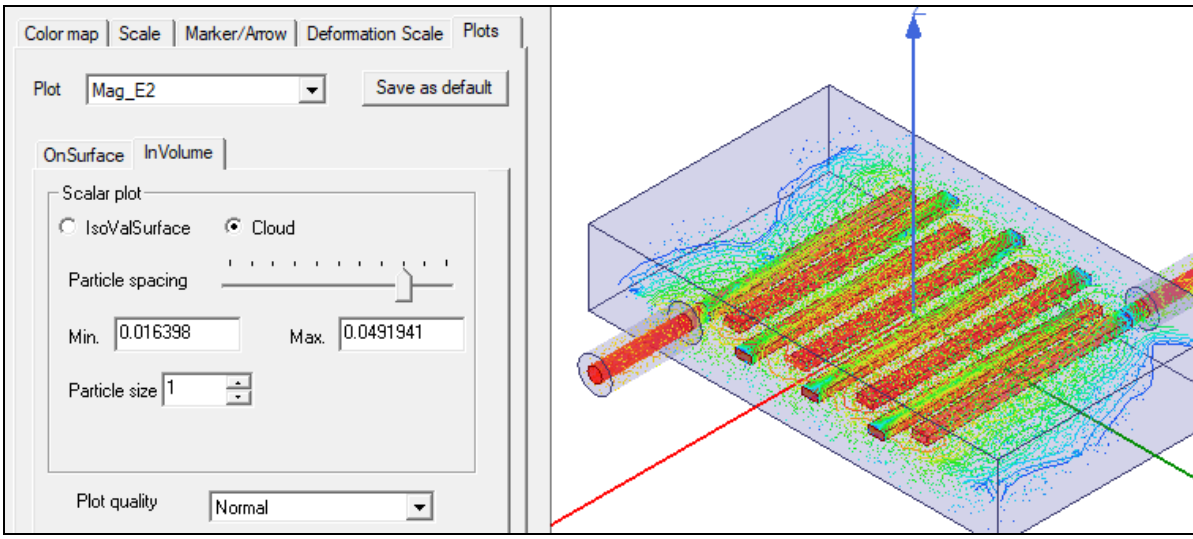


- Iso surfaces with field value higher than the threshold will show as opaque. The range of the "**Scale threshold**" is the same as the color key range from *Min* to *Max*.
- When "**Map transp.**" is checked, translucent iso surfaces field values map to [0.0, transparency] so lower field value map to higher transparency while higher field value to lower transparency. If it is unchecked, all translucent iso surfaces will have the same given transparency from the slider bar.



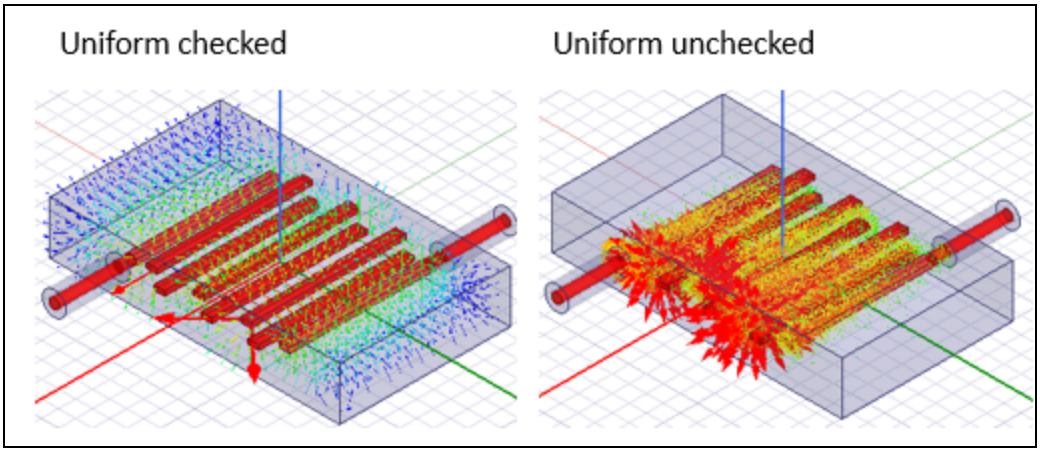
> **Note:**
>
> When the **Map transp.** slider bar is at the left most position, transparency is 0. This feature is essentially disabled, and the "**Scale threshold**" slider bar will be disabled. Also, high fidelity transparency should be enabled for best results.

- For *Cloud* plots, field values are represented by points that illustrate the spatial distribution of the solution. You can use the **Particle spacing** slider bar to adjust the plot. The higher the solution value, the greater the cloud density. You can also specify **Min** and **Max** values (numeric values based on the active model length unit) and adjust **Particle size**.



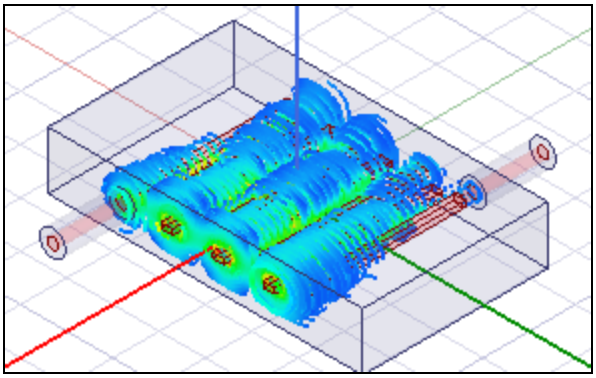
- **Plots (for vector plots):**
  - The **Plot** selected.
  - The **Min** and **Max** spacing of arrows (for vector plots). These values are based on the model's active length unit and are only relevant when the **Uniform** option is selected.
  - **Plot quality**, as *Normal, Coarse, Fine,* or *Very Fine*.
  - **Uniform** vector spacing, based on the *Spacing* slider position and the *Min* and *Max* values.

    When Uniform is *not* selected, a vector is rendered at every node (both corner and mid-side nodes of the elements).



The **Reset** button resets a good initial spacing (and thus Min, Max spacing range) without the need to recreate the field plot.

- **Plots (if streamline is checked):**
  - The **Plot** selected
  - The **Line style** 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 (in *Magnitude filtering* settings).



a. Under each tab, click **Save as default** if you want the tab's settings to apply to field overlay plots created after this point.
b. Select **Real time mode** if you want the changes to take effect immediately in the view window.
c. If this option is cleared, click **Apply** when you want to see the changes.

Setting Field Overlay 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 Manager.
2. Click **Q3D Extractor** > **Fields** > **Set Plot Defaults**.

   The **Set Plot Defaults** dialog box appears.
3. Select the solution to plot from the **Solution** drop-down menu.
4. Select the plot folder in which new plots will be stored from the **Quantity type** drop-down menu. Choose one of the following options:

New Folder	Each new plot will be stored in a separate folder in the project tree.
Automatic	Each new plot will be stored in a folder determined by Q3D Extractor as the most appropriate based on the plotted field quantity. For example, all surface magnitude E plots will be 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 will use the same color key. The **Auto** scale setting will be 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**.

Modifying the Field Plot Scale

To change how field quantities are scaled on the field overlay plot:

1. Click **Q3D Extractor** > **Fields** > **Modify Plot Attributes** or, in the Project Manager, right-click Field Overlays and select **Modify Plots** from the shortcut menu (or use the "**m**" hotkey).

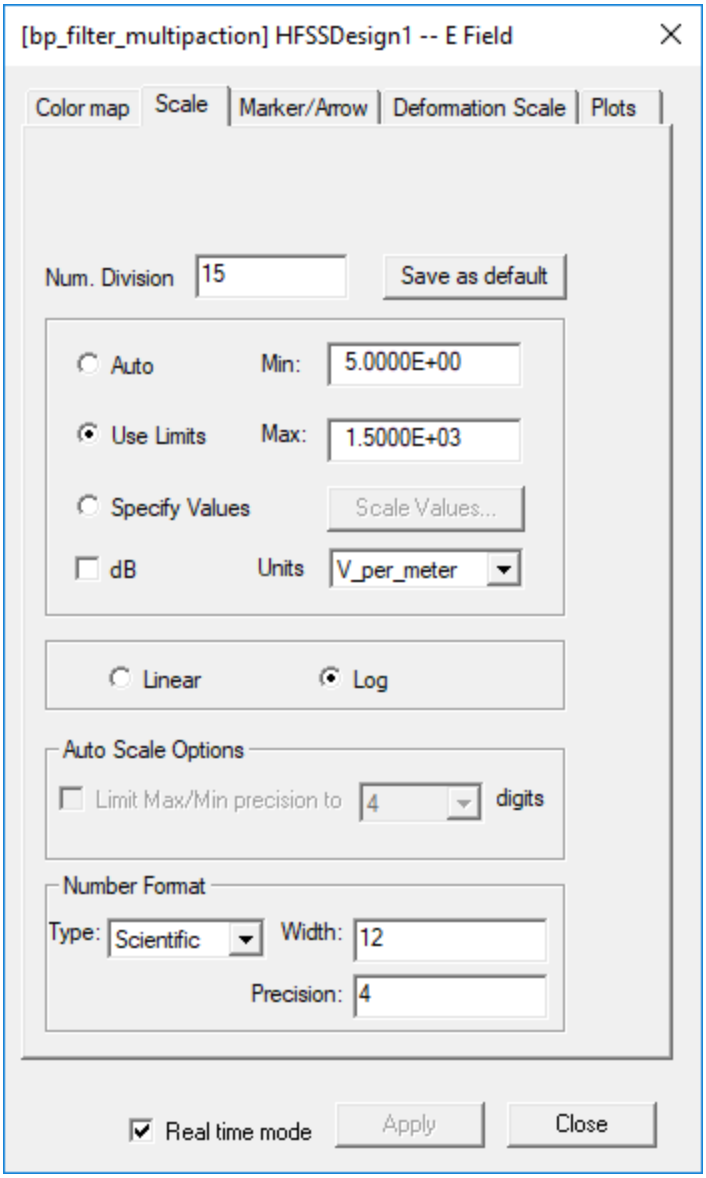
   The **Select Plot Folder** dialog box appears.

2. Select the plot folder you want to modify and 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 **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.

   Optionally, when dB is checked, dB scale is used for the plot. It disables the Units field, Linear and Log and have "Linear" selected.

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 **Use Limits**, enter the lowest field value to be plotted in the **Min.** text box and the highest field value to be plotted in the **Max.** text box.

   If you selected **Auto** or **Use Limits** without having dB checked, the **Auto Scale Options** are enabled. You should only changed 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)

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 **OK**. To close the dialog box without making changes, click **Cancel**.

8. If you selected **Auto** or use **Limits** or **dB**, 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.)

9. Specify the Number Format for the plot as Auto, Scientific Notation, or Decimal. You can also specify Width and Precision for the plot.
10. 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.

11. Optionally, you can use the **Save As Default** button to save the following to the 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 the default for new plots. For scalar-in-volume plots, iso-surface (rather than cloud) is the default display

12. Click **Close** to dismiss the dialog box.

Modifying Field Plot Colors

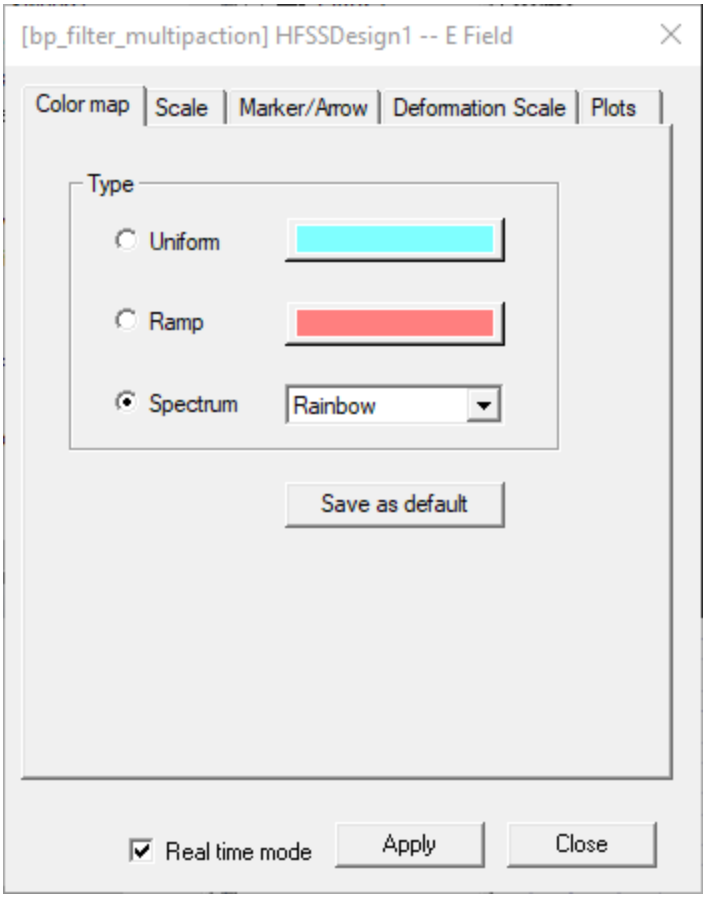
1. Click **Q3D Extractor** > **Fields**> **Modify Plot Attributes** or, in the Project Manager, right-click **Field Overlays** and select **Modify Plots** from the short-cut menu (or use the "m" hotkey).

   The **Select Plot Folder** dialog box appears.

2. Select the plot folder you want to modify and click **OK**.

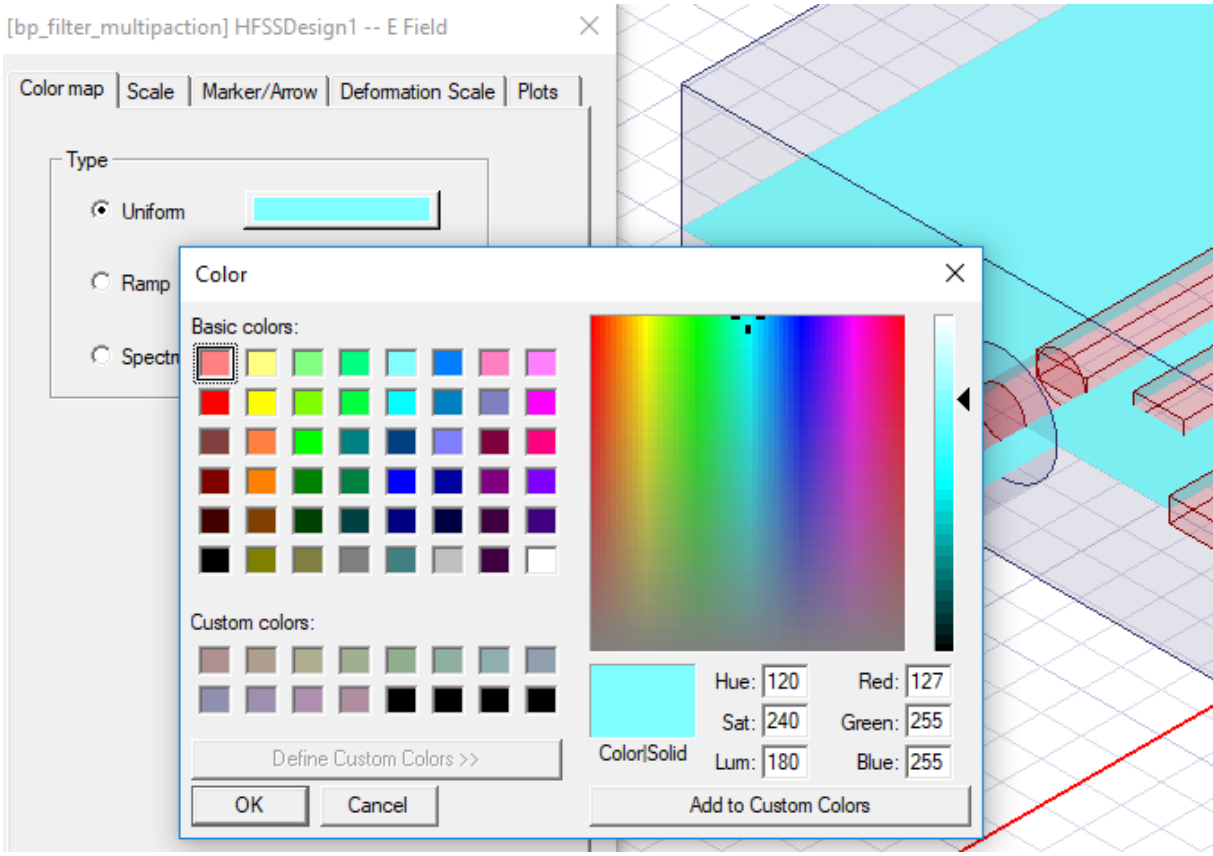
   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.

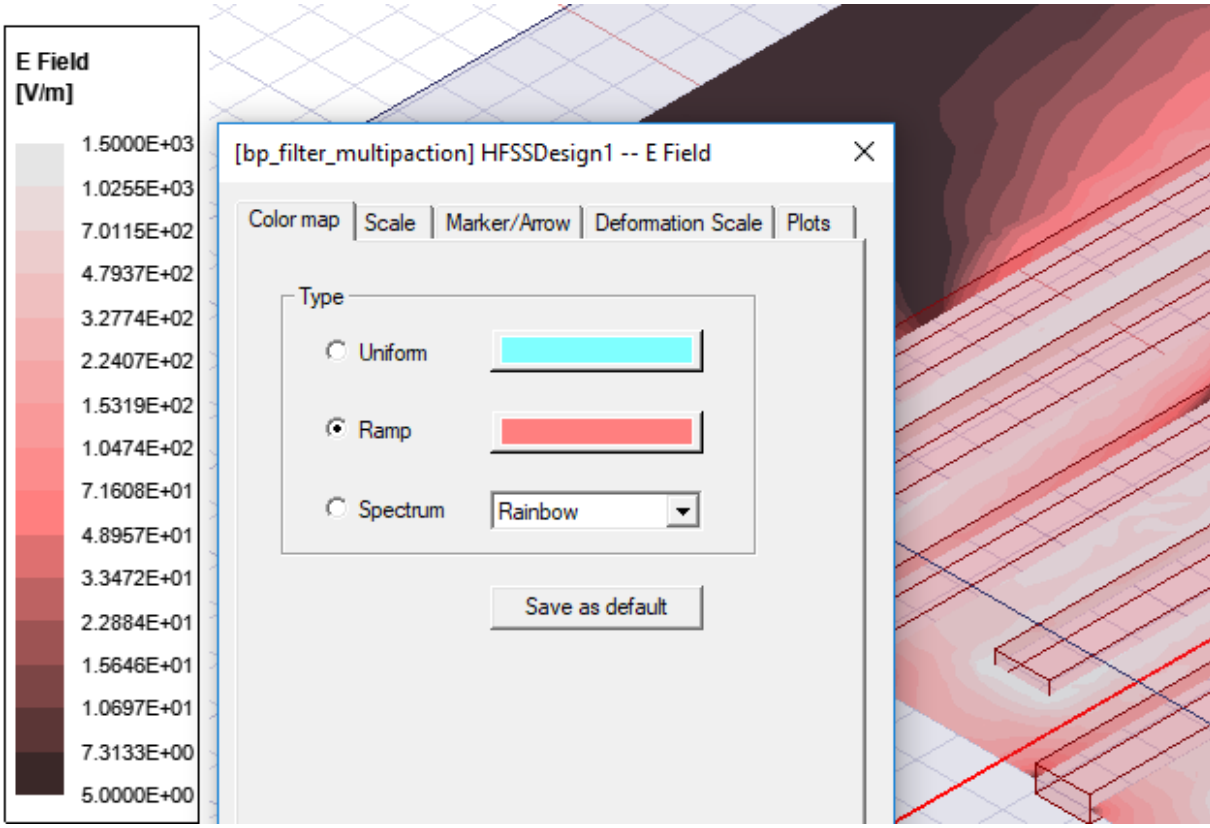


3. Click the **Color Map** tab.
4. Select one of the following color types:

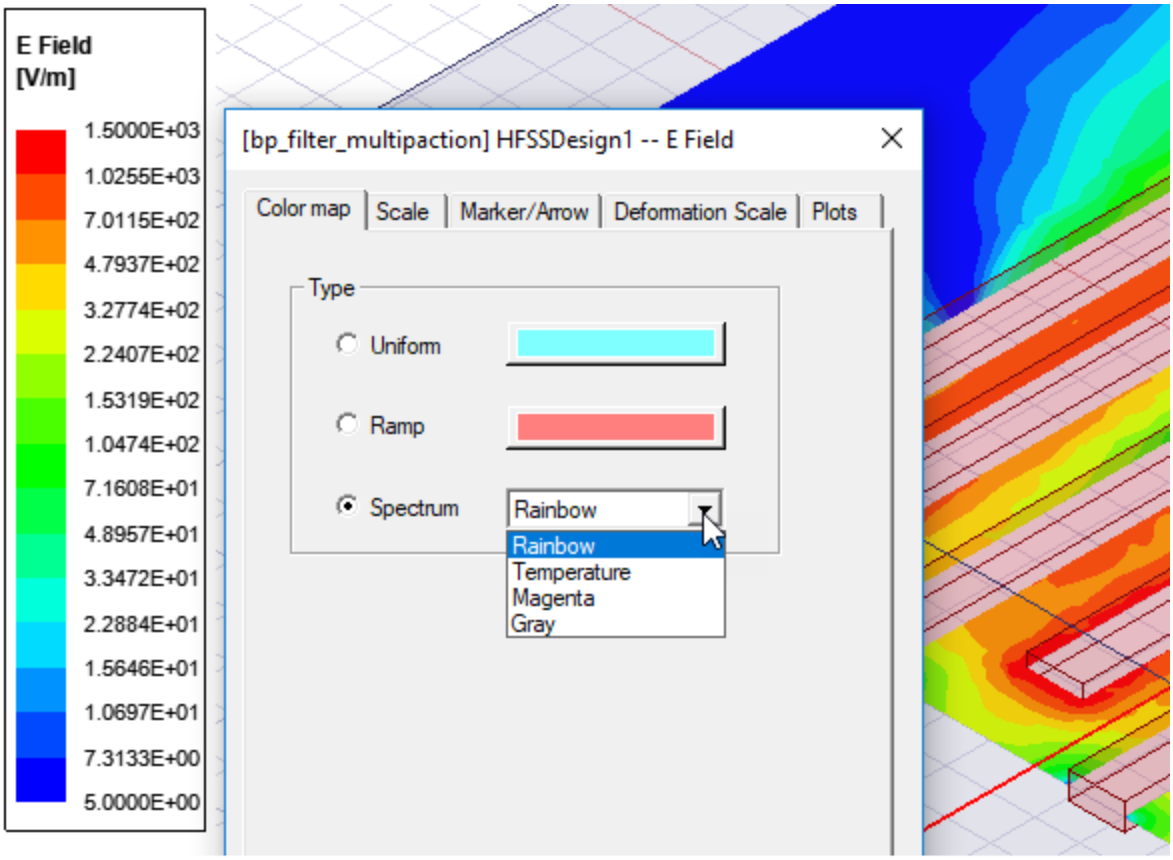
Uniform Field quantities are plotted in a single color. Click the button to choose the plot color from the **Color** button palette.



Ramp Field quantities are plotted in shades of a single color. Choose 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. Choose a color spectrum from the drop-down menu. The values are Rainbow, Temperature, Magenta, and Grey. 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.

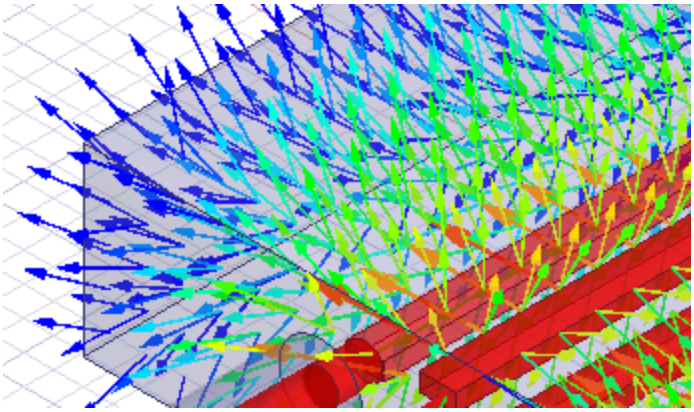
   You can choose **Save as Default**, if you want to use the current settings.

   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.

Modifying Vector Field Plot Arrows

To change the appearance of a vector field plot's arrows:



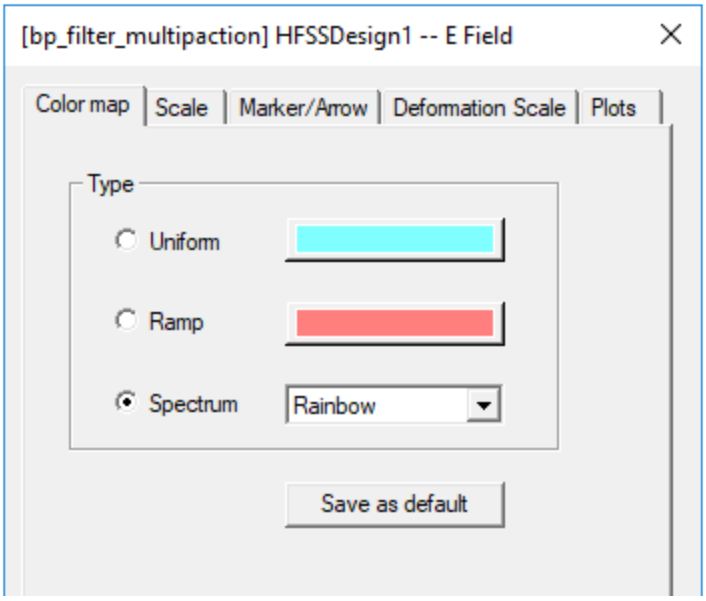
1. Click **Q3D Extractor**> **Fields**> **Modify Plot Attributes** or, in the Project Manager, right-click on **Field Overlays** and select **Modify Plot Attributes**. Alternatively, under **Field Overlays** in the Project Manager, right-click a subfolder (overlay type) heading and select **Modify Attributes**.

   If you select Modify Plot Attributes, the **Select Plot Folder** dialog box appears. If you select Modify Attributes, the appropriate attributes dialog box appears, bypassing the **Select Plot Folder** dialog box. In this case, skip to step 3.

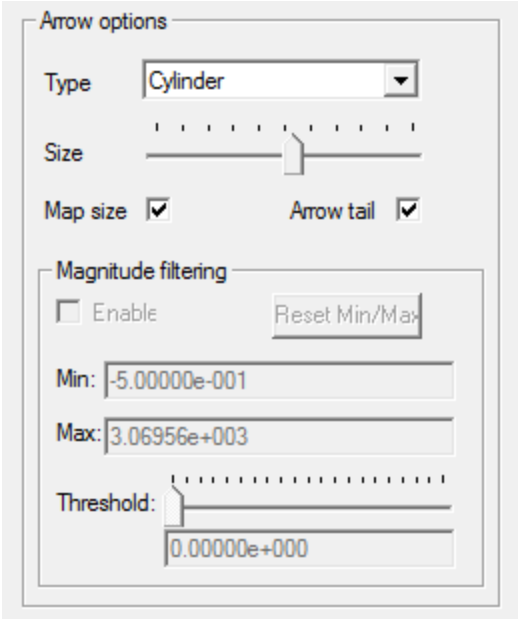
2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

   A dialog box with attribute settings for the selected folder appears.



3. Click the **Marker/Arrow** tab.



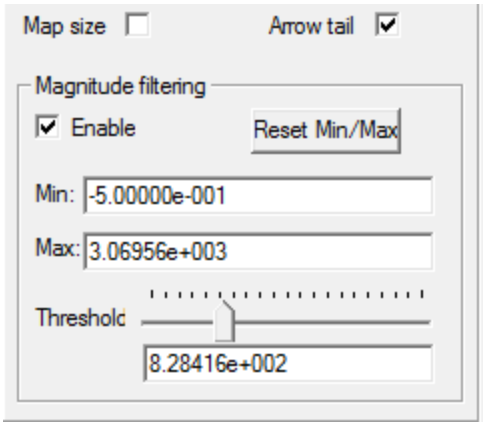
4. Under **Arrow Options**, select one of the following arrow types:

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 **Arrow tail** to include tails on all arrows.
7. Check **Map size** to scale the size of the arrows to the magnitude of the field quantity being plotted.

   If you uncheck **Map size**, the Magnitude filtering is enabled. You can specify the Min and Max magnitude filtering, or use a Threshold slider to set a threshold for plotting vectors. Vectors under the threshold or below the Min are not plotted.



   The **Reset Min/Max** button lets you reset min/max to the current intrinsic values.

8. Click the **Plots** tab.
9. Q3D Extractor plots arrows on a grid that is superimposed on the surface or object you selected for the plot. 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.)
   - Select **Uniform** if you want the arrows to be spaced equally.
10. 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.

11. Click **Close** to dismiss the dialog box.

Modifying Scalar Field Plot Isovalues

1. Click **Q3D Extractor**> **Fields**> **Modify Plot Attributes**.

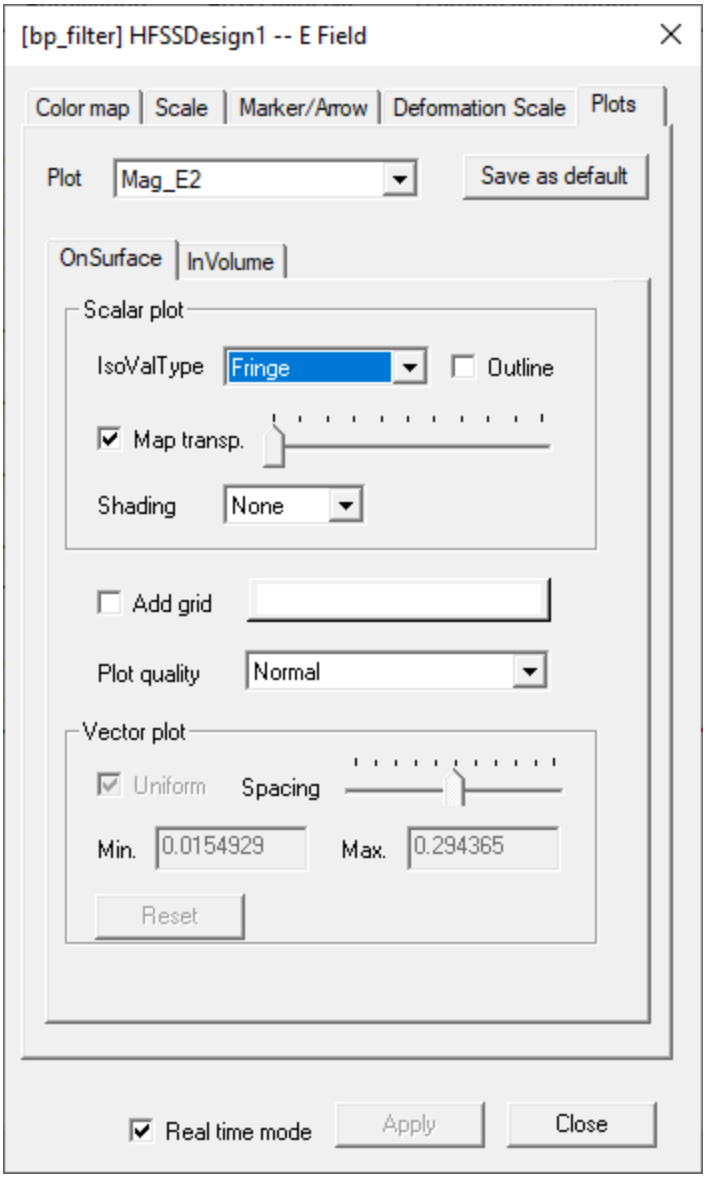
   The **Select Plot Folder** window appears.

2. Select the plot folder you want to modify, and then click **OK**.

   All plots in the selected folder will be modified.

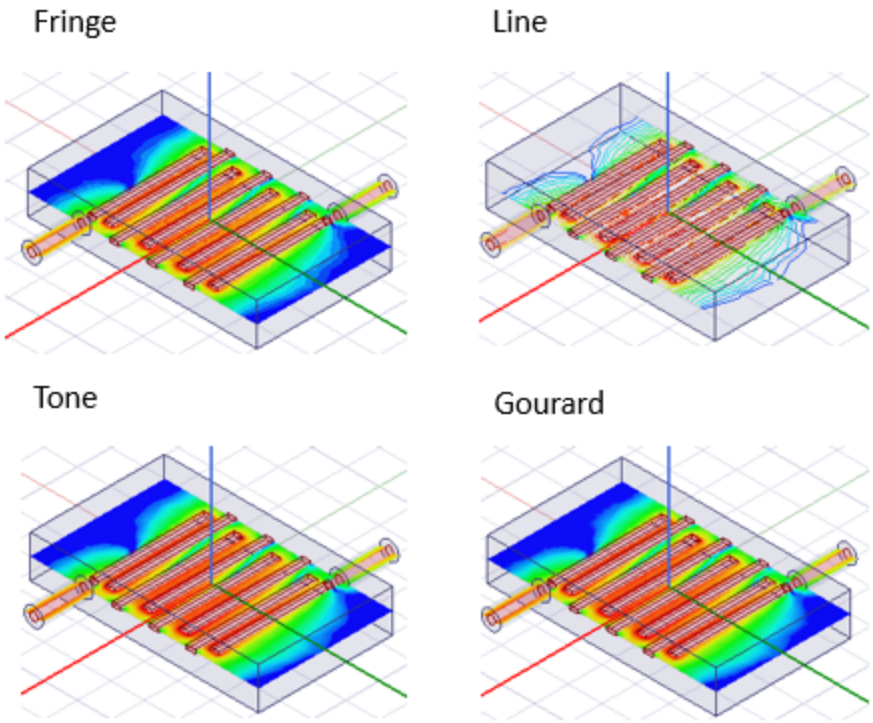
   A dialog box with attribute settings for the selected folder appears.

3. Click the **Plots** tab. If you have more than one plot, a drop down lets you select. The OnSurface tab appears for surface plots and InVolume for volume plots.



4. If the plot is a scalar surface plot, do the following:
   a. Select one of the following isosurface display types in the **IsoValType** drop-down menu:

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.



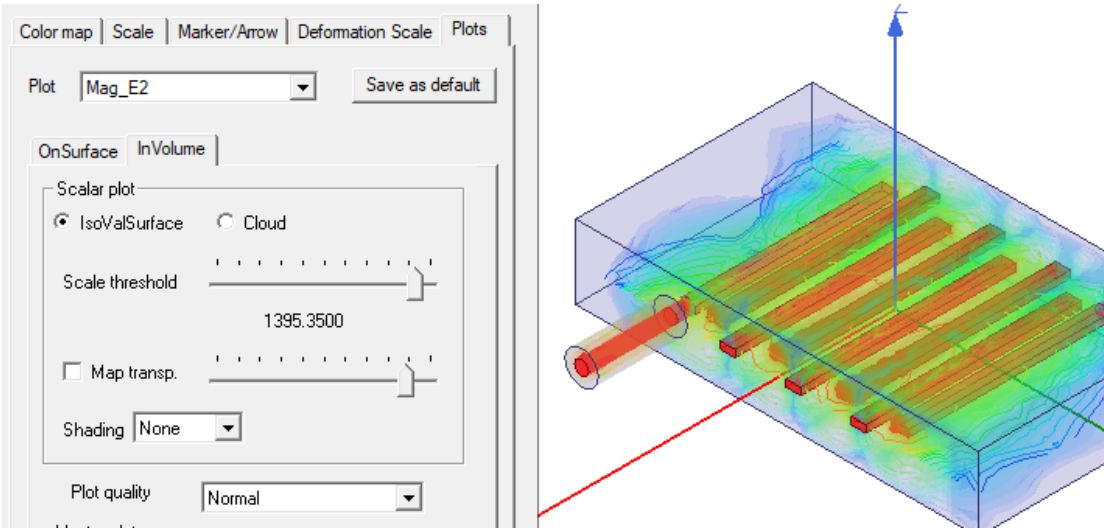
b. Optionally, if you selected **Fringe** or **Tone**, select **Outline** to add a border line between isovalues.

c. For shading on OnSurface plots, if you are using Lighting, you can adjust the Shading as None, Flat, or Smooth.

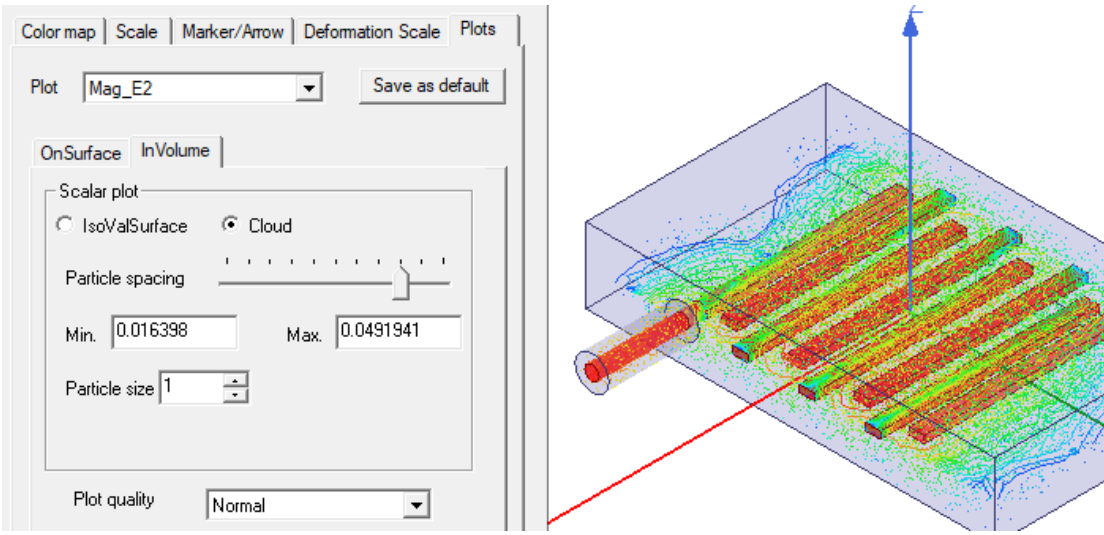
5. If the plot is a scalar volume plot, do the following:

a. Select one of the following display types:

IsoValSurface	Color is drawn on the isovalues. This choice includes “Scale threshold” slider barthat allows you to set a field value threshold so that iso surfaces with field value lower than the given threshold will show as translucent with given transparency specified by the slider bar below it. When "Map transp." is checked, translucent iso surfaces field values map to [0.0, transparency] so lower field value map to higher transparency while higher field value to lower transparency. If it is unchecked, all translucent iso surfaces will have the same given transparency from the slider bar.



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. You can adjust particle spacing, Min. and Max, values, and Particle size.



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 **Close** to dismiss the window.

Setting the Mesh Visibility on Field Plots

To display or hide the mesh on field plots, or change the mesh's color:

1. Click **Q3D Extractor**> **Fields**> **Modify Plot Attributes** .

   The **Select Plot Folder** dialog box appears.

2. Select the plot folder you want to modify and 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. 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 **Close** to dismiss the window.

Color Key Settings

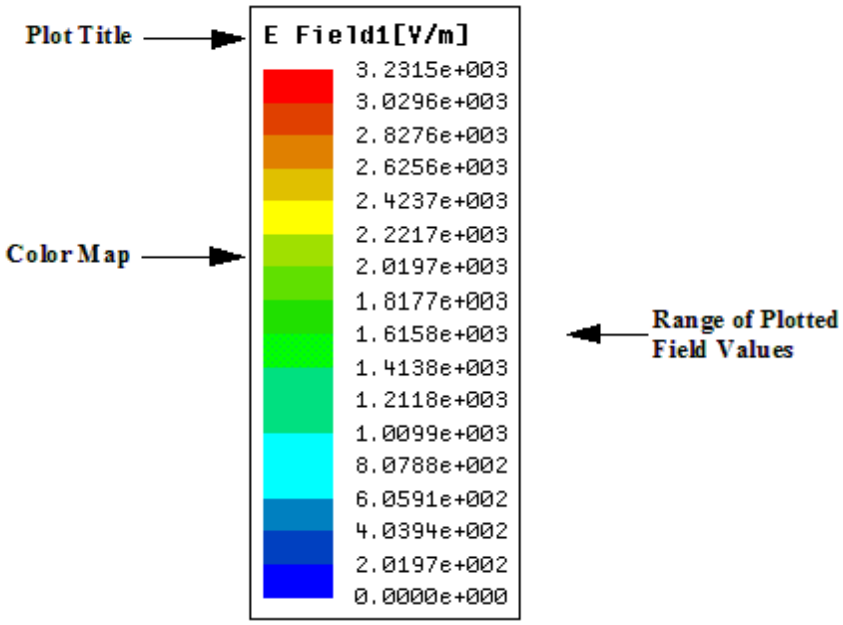
The color key (shown below) displays the range of plotted field values for a field overlay plot. It displays the colors that correspond to the range of field values on the plot. You can click and drag the color key to move it. In the case of no field data, the color key does not appear.

1. Click **View**> **Visibility**> **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 in which you want to hide the color key from view.
4. Click **Done** to dismiss the dialog box.

Alternatively, to hide the color key, right-click on the color key in the active view window and choose **Hide** from the shortcut menu.



Color keys will only be visible for the plots that are selected in the **Active View Visibility** dialog box. If a hidden color key is shared by two or more overlay plots (such as a Heat Flux key, which is shared by heat flux magnitude and heat flux vector plots), then the color key will be hidden in all plots that share it. The **Active View Visibility** dialog box only lists a single incidence of a particular key, regardless of how many overlay plots share it. Use this dialog box to restore visibility of hidden color keys.

Overlaying 2D Radiation Field Plots on Models

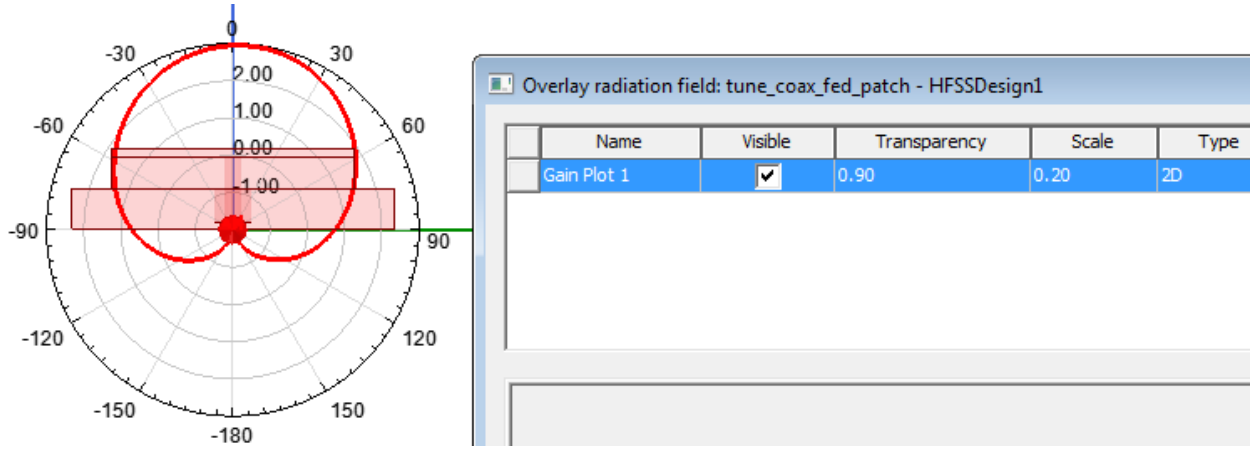
You can overlay Radiation Pattern plots (with restrictions) on the geometry in the Modeler window. The eligibility depends on whether the plot is defined on a single plane. For a Radiation Field plot, you right-click **Results** in the Project tree, and click **Results>Create Far Fields Report>Radiation Pattern**.

There are two cases of eligible radiation pattern plots for overlay, defining a single plane in a 3D Polar Plot.

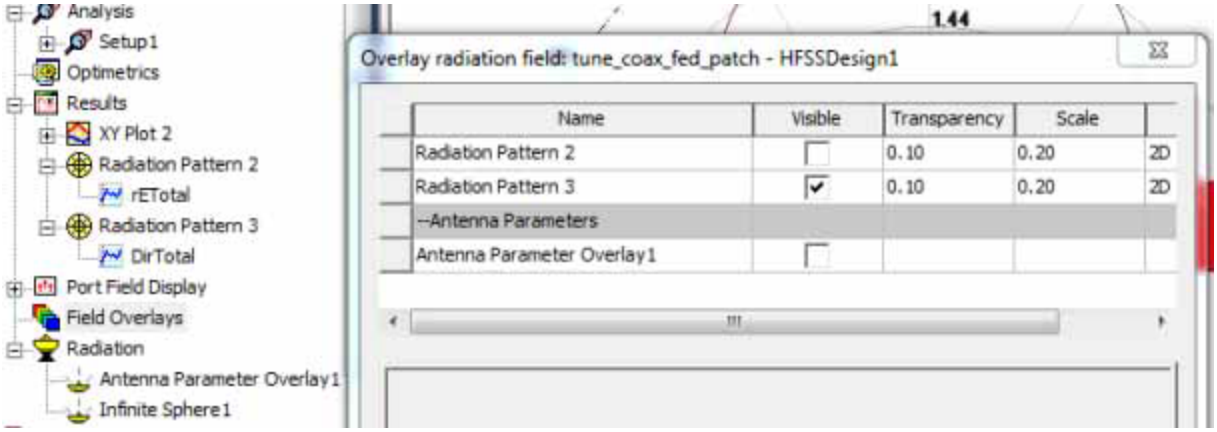
- Primary sweep on "Theta" and with one value for "Phi" in "Families" of curves, typically 0 or 90.
- Primary sweep on "Phi" and with one value for "Theta" in "Families" of curves and "Theta" is either 90deg or -90deg.

Note that the plot can contain multiple traces within this plane - for example traces for multiple frequencies of a discrete sweep. The Radiation Pattern plot will overlay on the 3D view, in the

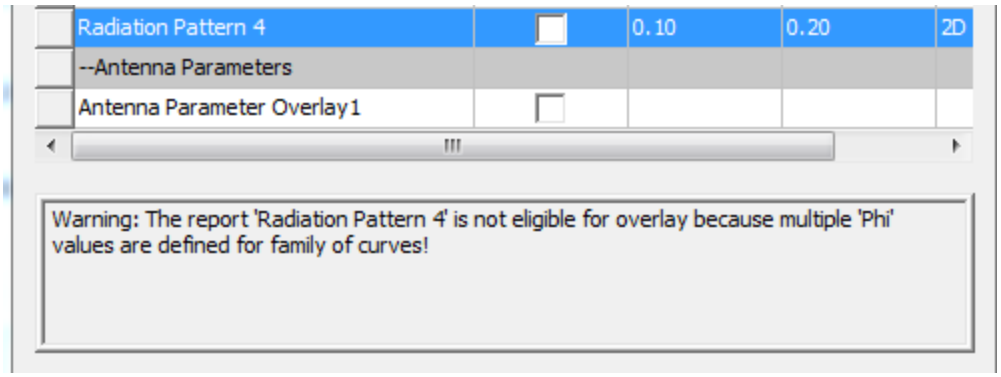
appropriate rotated plane of the far field CS, and include all traces and the axes/grid. There is no need to show the title or legend.



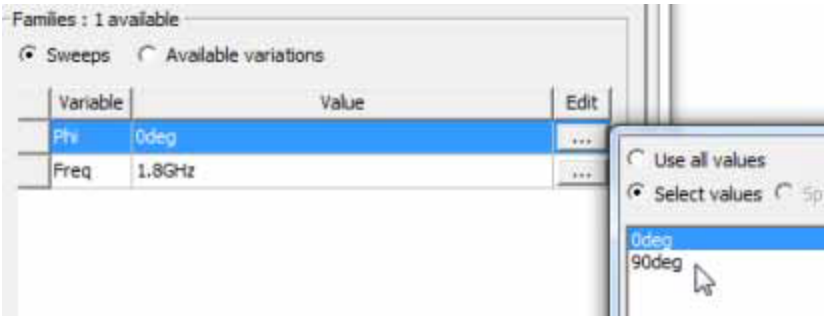
Radiation Pattern plots are added to the **Overlay radiation field** dialog that displays when you right-click **Field Overlays** in the Project Tree and select **Plot Fields>Radiation Field...** from the shortcut menu. The dialog lets you toggle visibility for the Model window and set plot parameters for Transparency, Scale, and Type (2D or 3D).



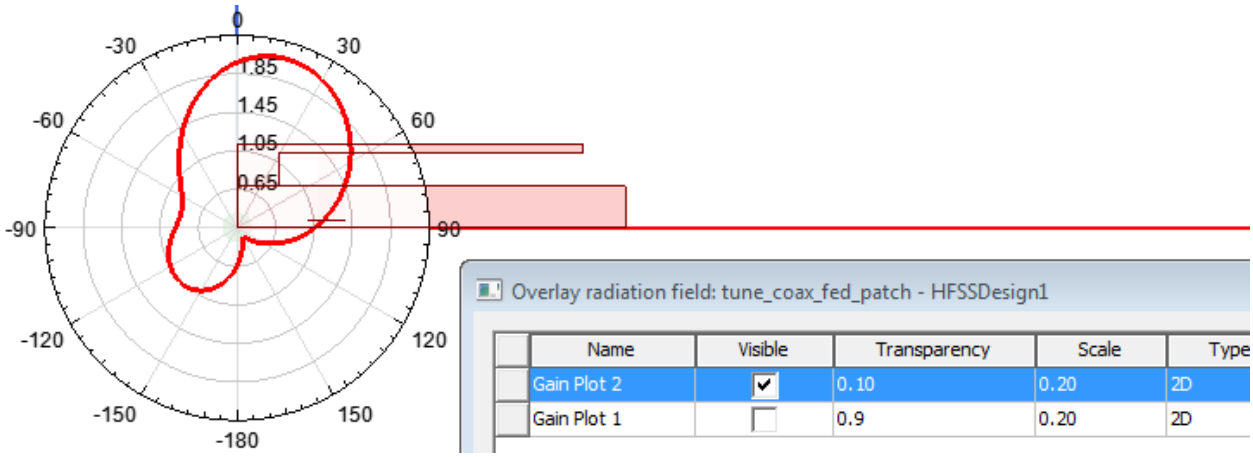
There are several cases mentioned above where an overlay will not be valid for display. This includes any radiation plot that has multiple planes. The **Overlay radiation field** dialog always lists all possibly overlays for selection, whether they are valid or not. However, if you check the Visibility column an overlay that is not valid, and then click **Apply**, the text field in the dialog gives a warning describing why the overlay is not valid, and then unchecks that overlay. Similarly, if a valid overlay is already being shown, and then you edit the plot settings or antenna parameters such that the overlay becomes invalid, the visibility column becomes unchecked and the overlay stops displaying.



You can then use **Modify Report** to change the plot based on the warning to make it valid for display. Typically, you can select the **Families** Tab, and select the desired Phi value.



When the plot is valid, **Apply** then shows the plot in the modeler window.

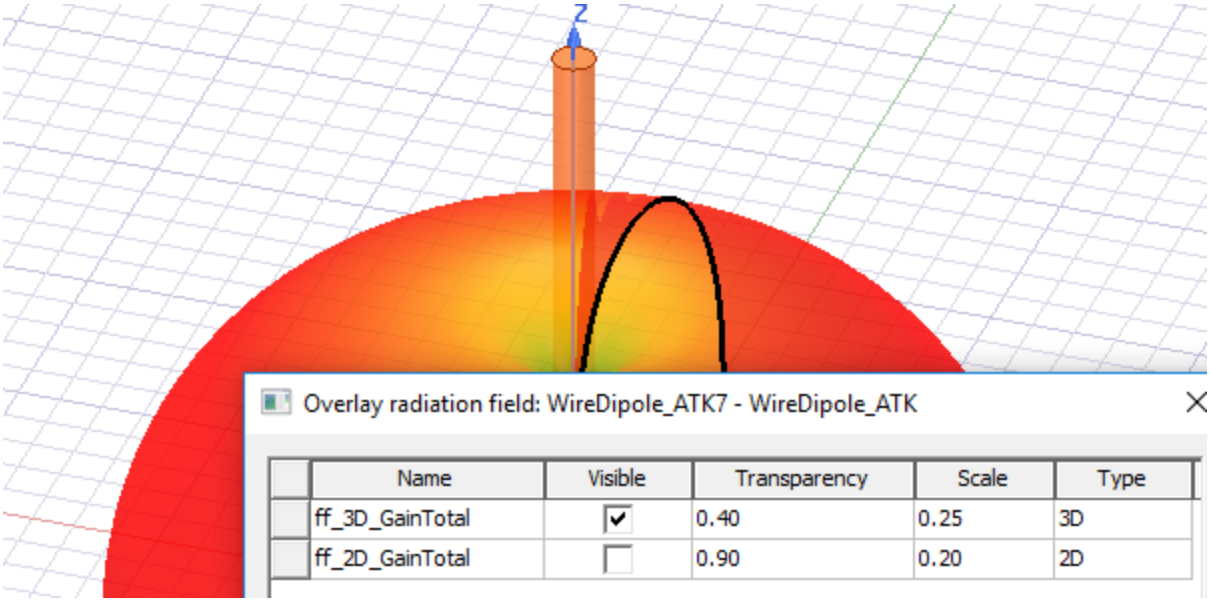


If you have Saved Antenna Parameters for Overlay, you also use this dialog to control whether to display a table of antenna parameters. See Antenna Parameter: Save for Overlay.

Also see Creating 2D Radiation Field Plot Overlay Animations

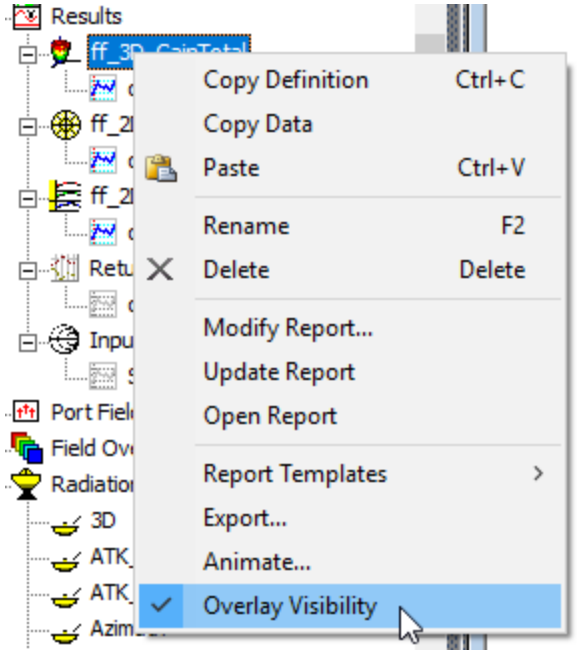
Overlaying 3D Polar Plots of Near or Far Fields on Models

You can overlay existing 3D Polar Plots of near or far fields on the 3D model window by using the **Overlay Radiation** field dialog invoked by using the [*solver*]**>Fields>Plot Fields>Radiation Field...** command, or by right-clicking on Field Overlays in the **Project** tree and selecting **Field Overlays>Radiation Field...**.



You can edit the visibility, transparency and scale of the polar plot by using the dialog.

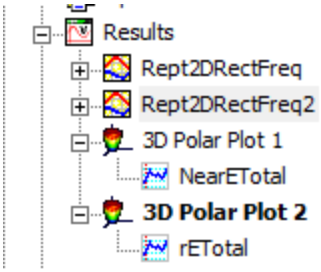
You also toggle the overlay of an existing plot by right-clicking on the plot in the **Project** tree and clicking **Overlay Visibility** on the shortcut menu.



Using the Overlay Radiation Field Dialog

For a 3D polar plot to be eligible for overlay, it must have its primary and secondary sweep from variables Phi and Theta or IWavePhi and IWave Theta in that order. If the plot is unsuitable, the **Overlay** commands are disabled.

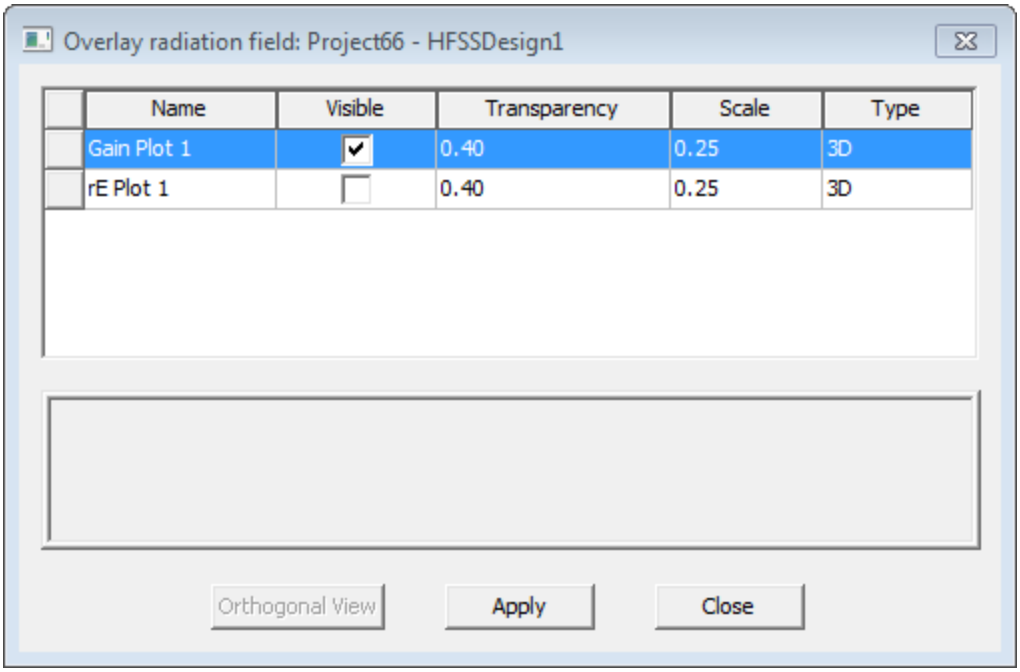
1. Create one or more suitable 3D Polar Plots of near or far fields for your model.



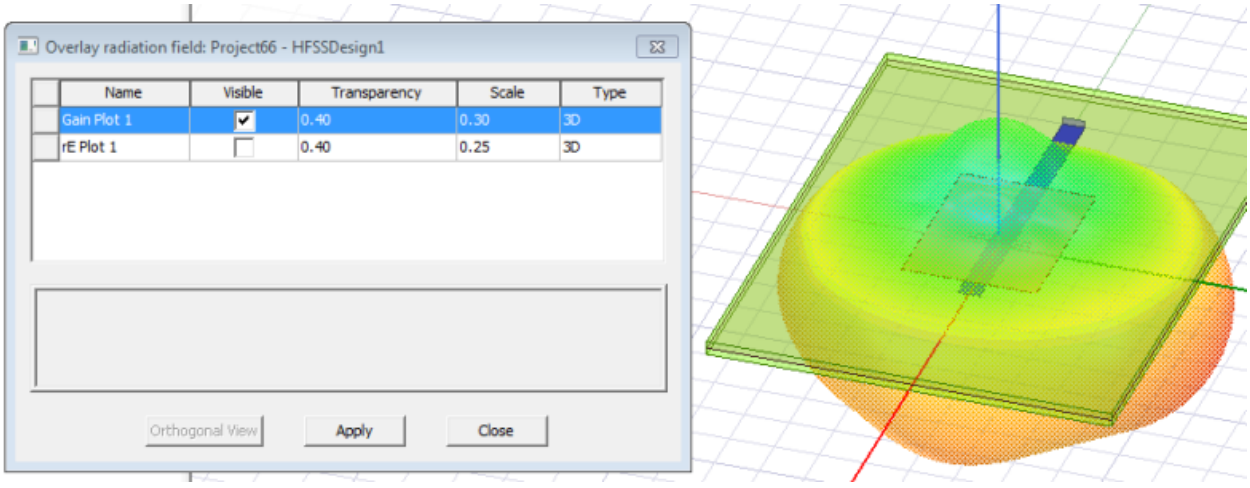
This enables the **Radiation Field...** command on the **Plot Fields** submenu.

2. Click the [*solver*]**>Fields>Plot Fields>**[*field*] command, or by right-clicking on Field Overlays in the Project tree and selecting **Plot Fields>Radiation Field...**

This displays a dialog listing any existing 3D polar plots of near or far fields.

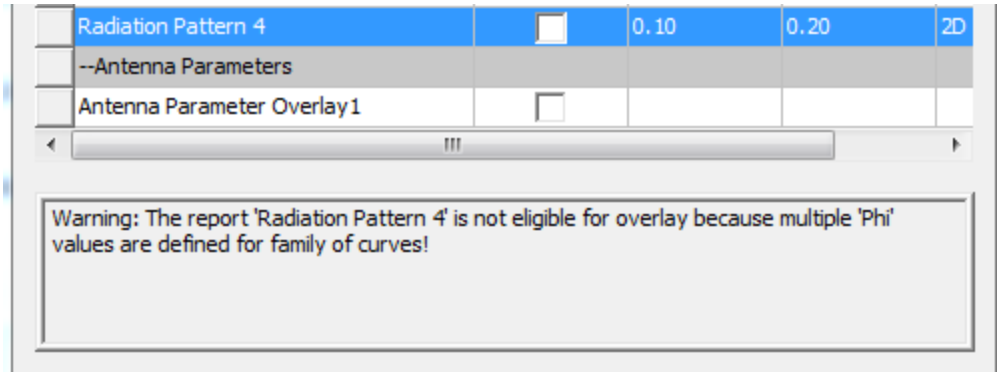


3. Check the Visible box and click **Apply** to cause that plot to appear in the model window.



You can also edit the Transparency and Scale in the dialog box. Other properties of the 3D plot are controlled in its properties window.

If a 3D polar plot is unsuitable for overlay, the message field provides a description of the issue.

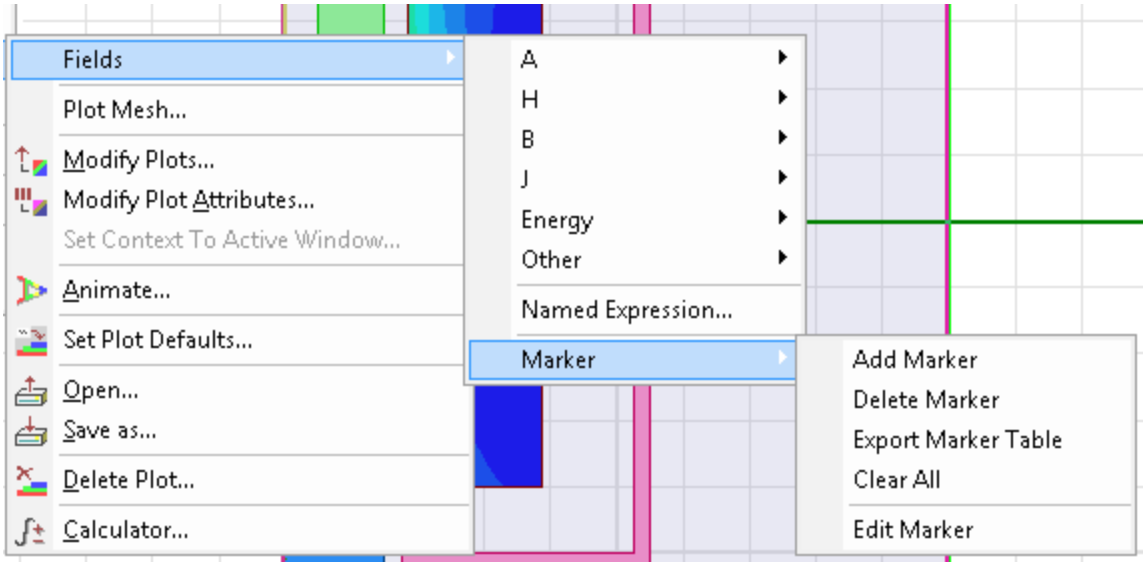


If you have Saved Antenna Parameters for Overlay, you also use this dialog to control whether to display a table of antenna parameters. See Antenna Parameter: Save for Overlay.

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



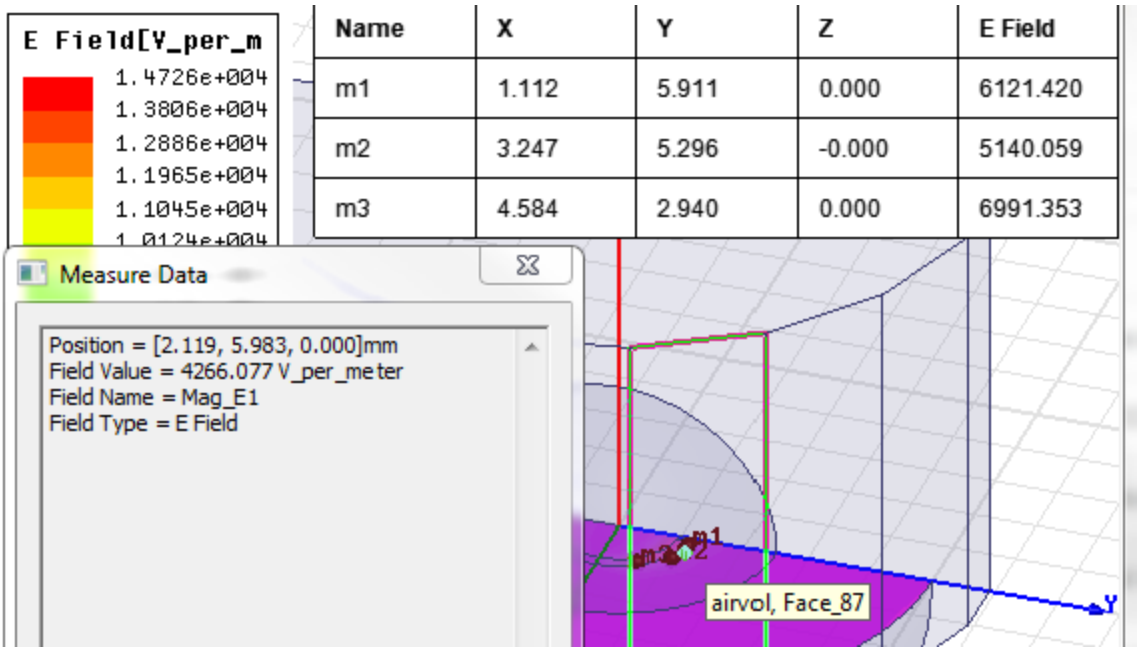
Adding a Field Plot Marker

To add one or more field plot markers to a scalar field:

1. On the main menu click **Q3D Extractor** and then select **Fields** > **Plot Fields** > **Marker** > **Add 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** > **Add Marker**.

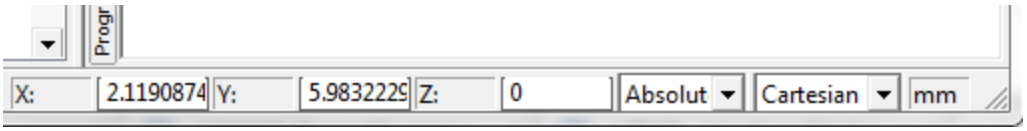
   The *Measure Data* dialog box opens, and a round dot appears at the tip of the cursor.

2. Drag the dot over the spot on the field overlay plot where you want to add a marker. The Measure Data dialog box shows detailed information for the spot currently under the dot.



Name	X	Y	Z	E Field
m1	1.112	5.911	0.000	6121.420
m2	3.247	5.296	-0.000	5140.059
m3	4.584	2.940	0.000	6991.353

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.

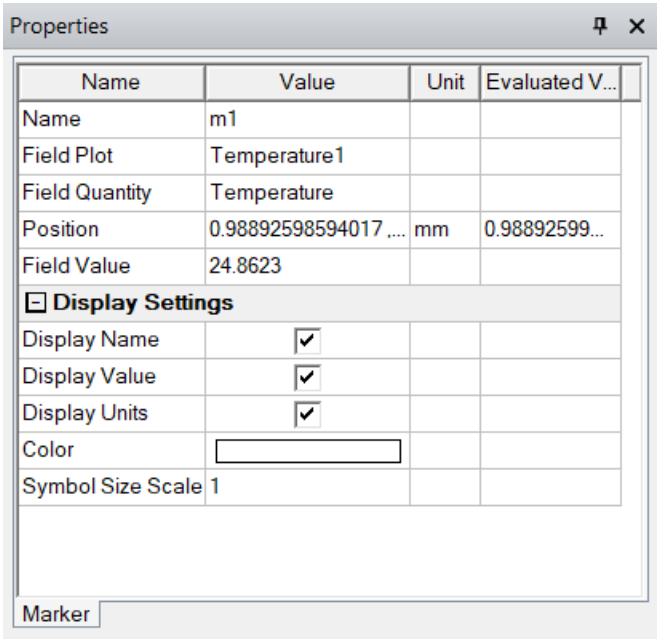
4. Repeat as desired to add additional markers.
5. Press the **Esc** key when finished adding markers.

Editing Field Plot Markers

To edit field plot markers:

1. In the marker table, click on the marker you wish to edit. The marker in the 3D Modeler window and its row in the table are highlighted. The properties of the marker are displayed in the **Properties** window.

> **Note:**
>
> Press **Ctrl** and select multiple marker rows in the table to edit marker properties simultaneously.



2. You can edit the following marker properties:
   - **Name**: Specify the name that appears next to the marker in the 3D Modeler window.
   - **Display Name**: Toggle to show or hide the marker's name.
   - **Display Value**: Toggle to show or hide the marker's value.
   - **Display Units**: Toggle to show or hide the marker's units.
   - **Color**: Click the color to change the marker's color in the **Color** dialog box.
   - **Symbol Size Scale**: Enter a different number to adjust the size of the marker in the 3D Modeler window.

3. Optionally, when a marker is selected, press the **Delete** key to delete the marker and its corresponding entry in the marker table.

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 **Q3D Extractor** and then select **Fields> Plot 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 **Plot 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 **Q3D Extractor** and then select **Fields> Plot 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 **Plot 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

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 **Q3D Extractor** and then select **Fields> Plot Fields> Marker> Clear All**.
- Alternatively, right-click anywhere in the Modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Plot Fields> Marker> Clear All**.

Deleting all the markers also removes the marker table.

Mapping Scalar Field Plot Transparency to Field Values

1. Click **Q3D Extractor** > **Fields**> **Modify Plot Attributes**.

   The *Select Plot Folder* dialog box 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. 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 **Close** to dismiss the dialog box.

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 **Q3D Extractor**> **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 **Marker/Arrow** tab in the plot attributes window.

4. Under **Marker options**, select one of the marker types to represent the field quantity at the point:
   - **Sphere**
   - **Box**
   - **Tetrahedron**
   - **Octahedron**
5. Use the **Size** slider to increase or decrease 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 **Close**.

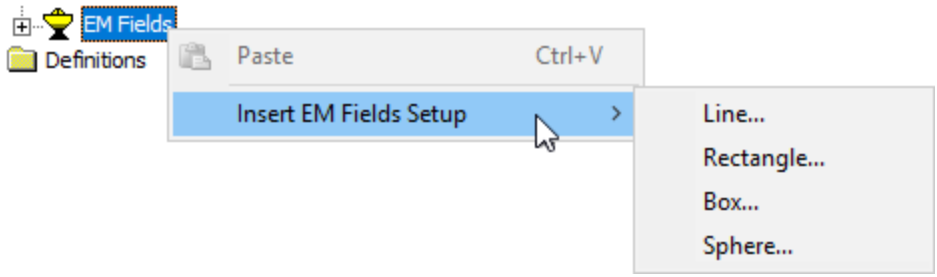
EM Fields in Q3D Extractor

Q3D Extractor can visualize EM fields on non-model lines, rectangles, boxes, and spheres. Once you have set up a field, you can view EM Fields reports or plot the field overlay.

> **Important:**
>
> In the **Solve Setup** window, you *must* select the **Save Fields** option. See: Specifying Solution Settings in Q3D Extractor.

To set up EM fields, access the EM Fields setup window one of two ways:

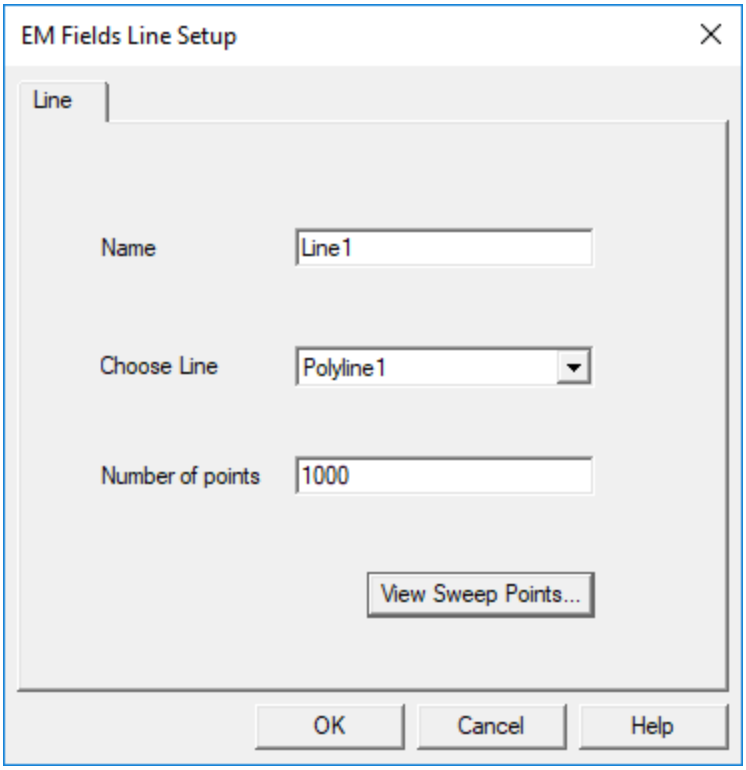
- In the Project Manager, right-click **EM Fields** > **Insert EM Fields Setup** > [**Field Type**].



- Select **Q3D Extractor** > **EM Fields** > **Insert EM Fields Setup** > [**Field Type**].

> **Note:**
>
> To display the **Insert Infinite Sphere Setup** menu item and enable Far Field calculations, you must enable the beta option.

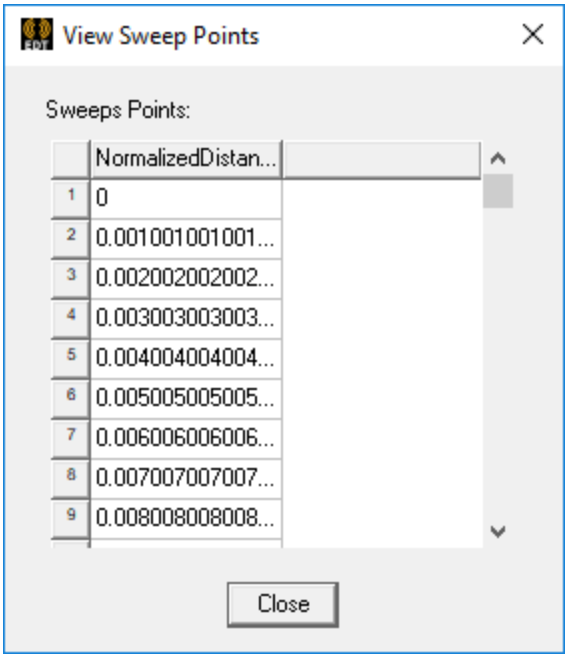
EM Fields Line Setup



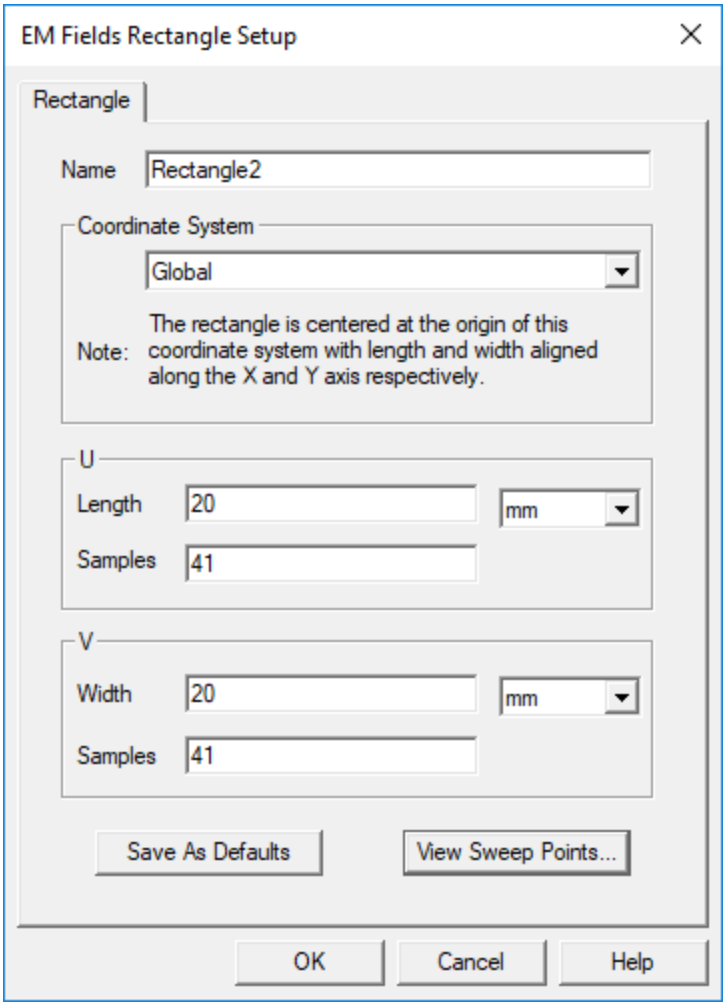
In the **EM Fields Line Setup** window:

- Enter a **Name** for the field.
- Use the **Choose Line** drop-down menu to select a valid polyline.
- In the **Number of points** field, enter the number of sweep points.

  If desired, you can click **View Sweep Points** to preview a list of points based on your entry.



EM Fields Rectangle Setup



In the **EM Fields Rectangle Setup** window:

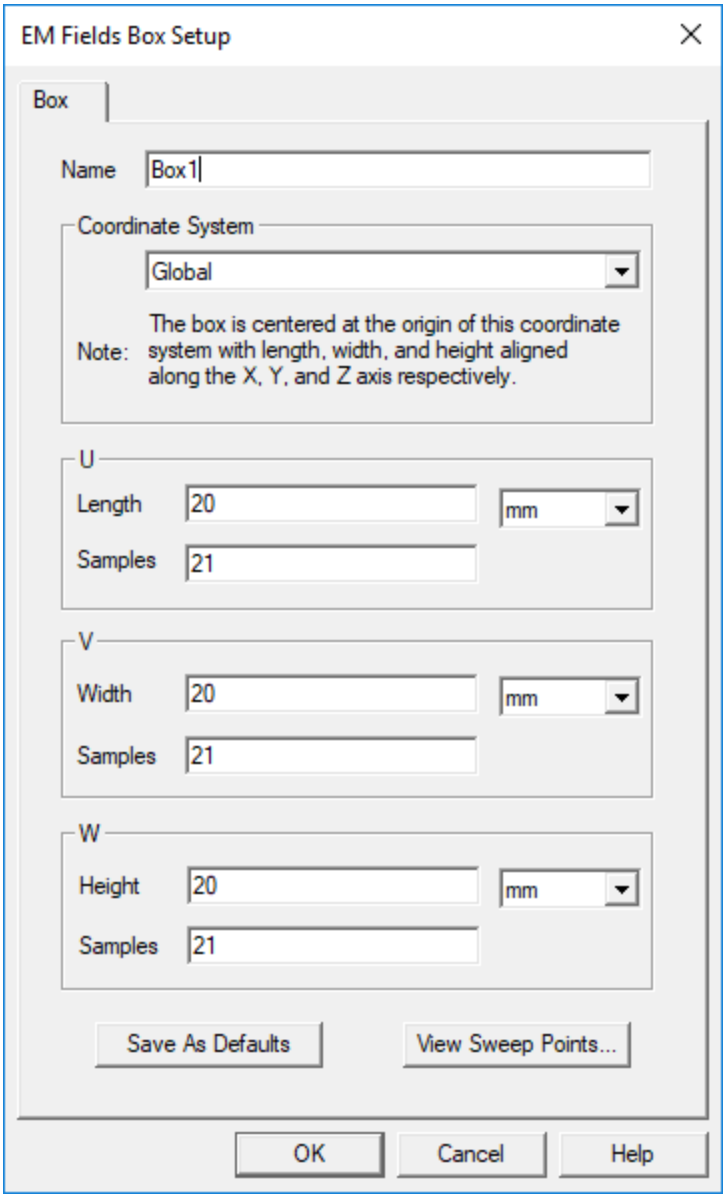
- Enter a **Name** for the field.
- In the **Coordinate System** area, use the drop-down menu to select either the **Global** coordinate system or another coordinate system in the design.

  The rectangle is centered at the coordinate system's origin, with length and width aligned along the X and Y axis, respectively.

- In the **U** area, enter the rectangular field's **Length**. Use the drop-down menu to select a unit of measure. Enter the number of **Samples**.
- In the **V** area, enter the rectangular field's **Width**. Use the drop-down menu to select a unit of measure. Enter the number of **Samples**.

  If desired, you can click **View Sweep Points** to preview a list of points based on your entries.

EM Fields Box Setup



In the **EM Fields Box Setup** window:

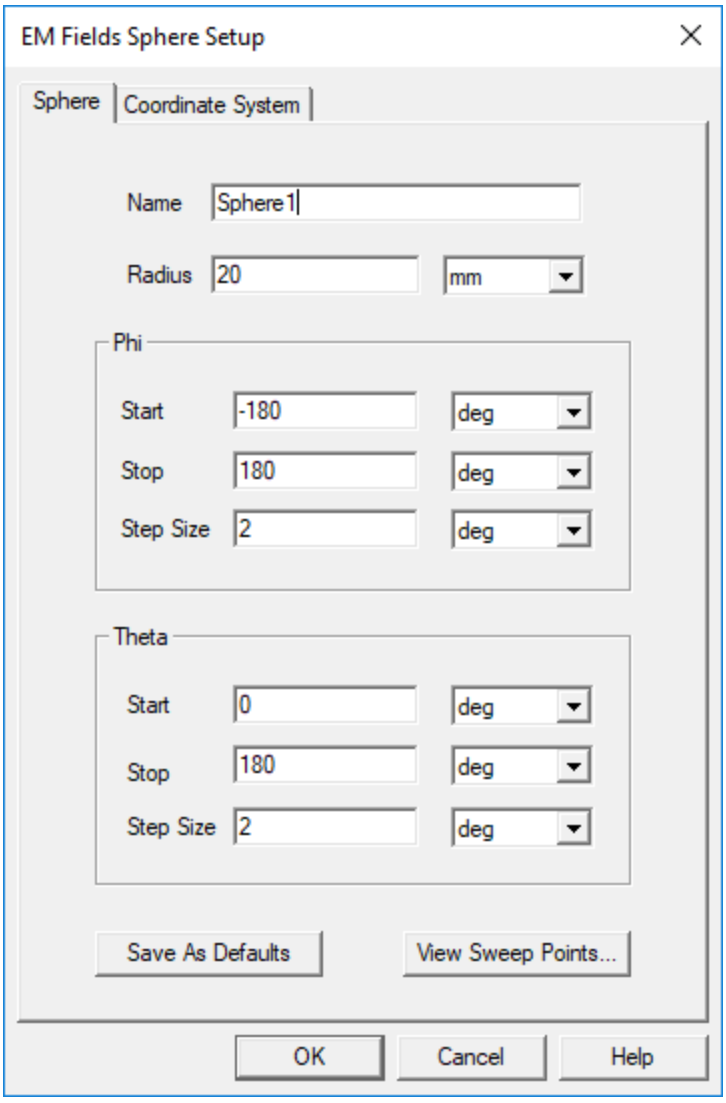
- Enter a **Name** for the field.
- In the **Coordinate System** area, use the drop-down menu to select either the **Global** coordinate system or another coordinate system in the design.

  The box is centered at the coordinate system's origin, with length, width, and height aligned along the X, Y, and Z axis, respectively.

- In the **U** area, enter the rectangular field's **Length**. Use the drop-down menu to select a unit of measure. Enter the number of **Samples**.
- In the **V** area, enter the rectangular field's **Width**. Use the drop-down menu to select a unit of measure. Enter the number of **Samples**.
- In the **W** area, enter the rectangular field's **Height**. Use the drop-down menu to select a unit of measure. Enter the number of **Samples**.

  If desired, you can click **View Sweep Points** to preview a list of points based on your entries.

EM Fields Sphere Setup



In the **EM Fields Sphere Setup** window:

- Enter a **Name** for the field.
- Enter a **Radius** for the field, and select a unit of measure from the drop-down menu.
- In the **Phi** area, enter **Start**, **Stop**, and **Step Size** values. Select units of measure from the drop-down menus.
- In the **Theta** area, enter **Start**, **Stop**, and **Step Size** values. Select units of measure from the drop-down menus.

  If desired, you can click **View Sweep Points** to preview a list of points based on your entries.

- Click the **Coordinate System** tab to select either **Use global coordinate system** or **Use local coordinate system**.

Viewing EM Fields Reports in Q3D Extractor

In order to view **EM Fields** reports, you must first set up EM fields and run a CG solution.

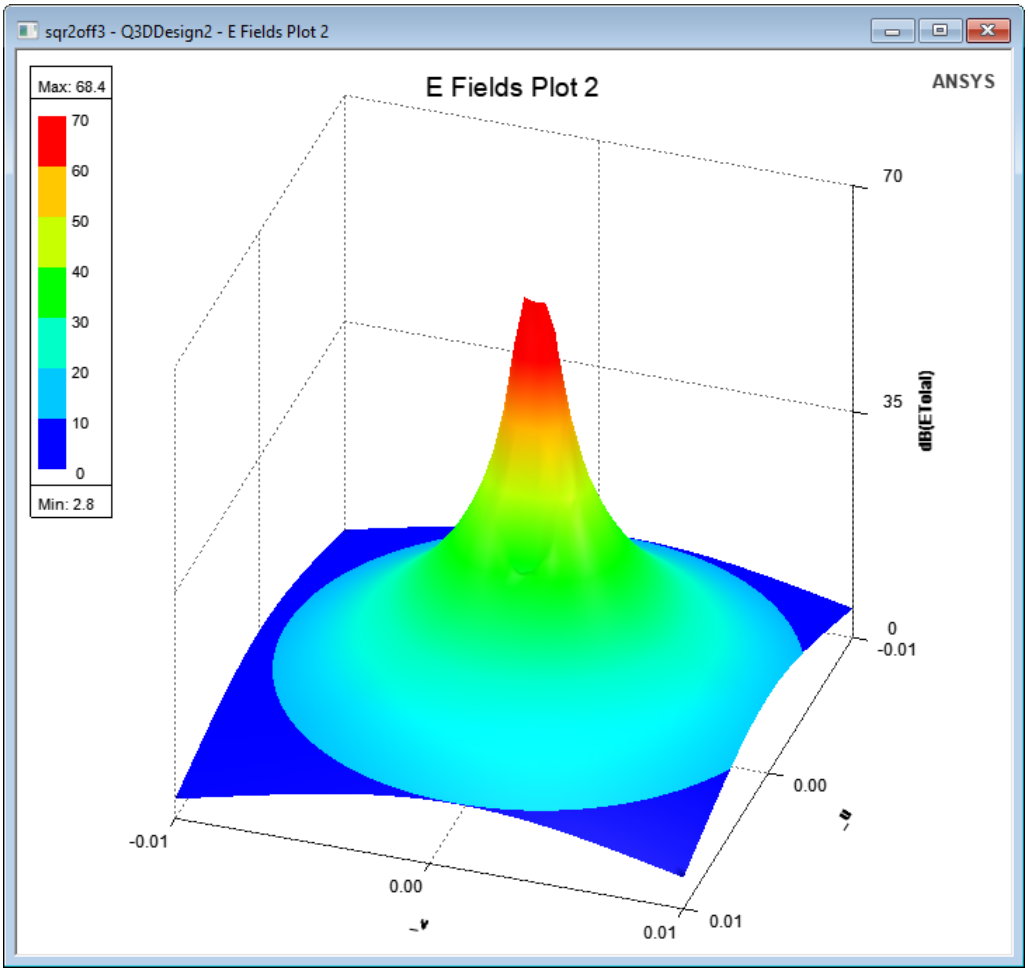
> **Important:**
>
> In the **Solve Setup** window, you *must* select the **Save Fields** option. See: Specifying Solution Settings in Q3D Extractor.

Once you have performed those prerequisites, add an EM Fields report:

1. Access the **Report** window one of two ways:
   - In the **Project Manager**, right-click **Results** > **Create Static EM Fields Report** > [**Report Type**].
   - On the **Results** tab, click **Static EM Fields Report** > [**Report Type**].
2. In the **Report** window:
   - From the **Category** list, select **E Fields**.
   - Select the other settings as applicable. See: Creating a New Report.
3. Click **New Report**.

   The EM Fields report appears under **Results** in the **Project Manager**.

Double-click the report to view it, or add an EM Fields overlay.



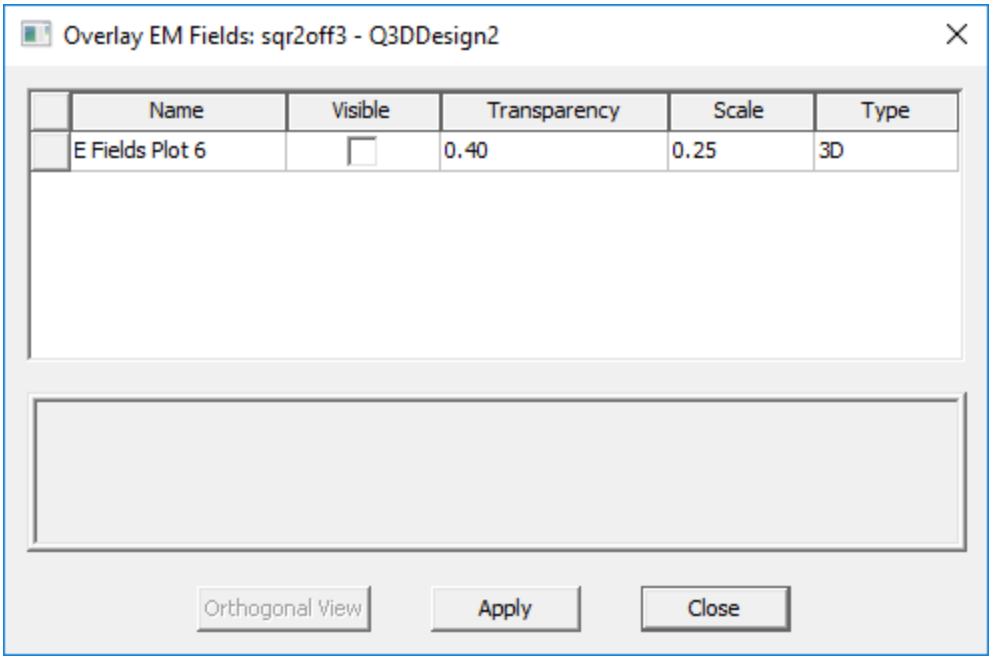
Plotting EM Fields Overlays in Q3D Extractor

In order to view **EM Fields** overlays, you must first create an EM Fields report.

Once you have added an EM Fields report, you can overlay it on your design:

1. In the **Project Manager**, right-click **Field Overlays** > **CG Fields** > **EM Fields**.

   The **Overlay EM Fields** window appears.



2. In the **Visible** column, select the check box next to the EM Fields report you wish to overlay.

Far Fields in Q3D Extractor

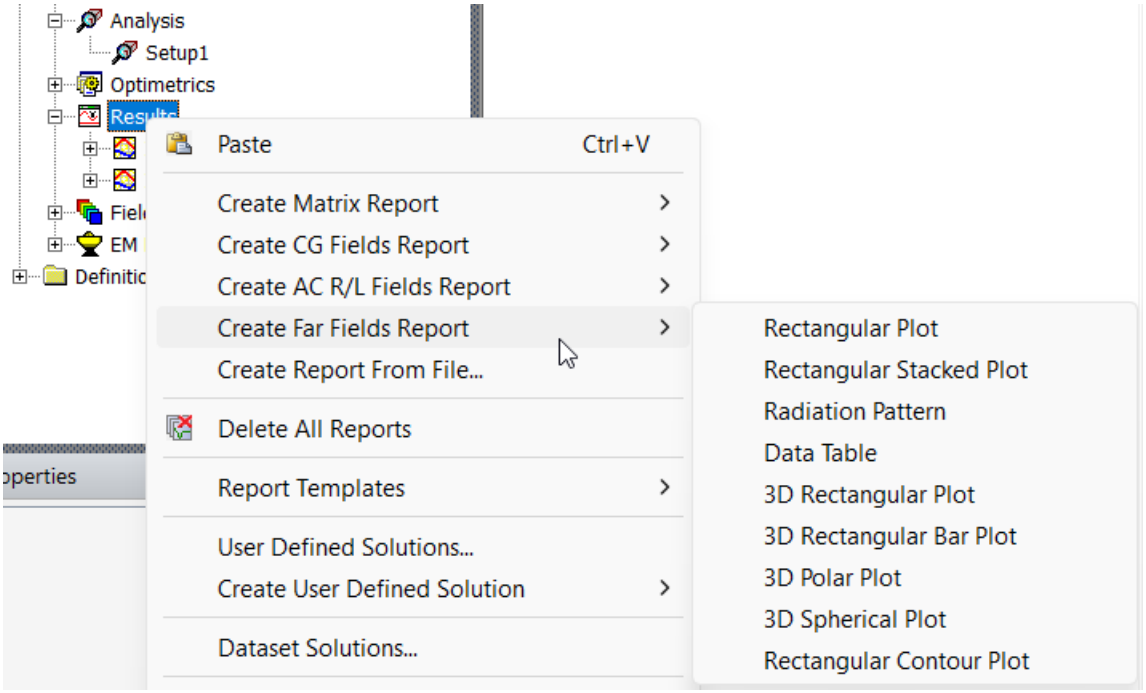
In order for Q3D Extractor to plot Far Fields, the beta option must first be enabled:

1. Click **Tools** > **Options** > **General Options**.
2. On the **General** tab, click **Beta Options**.
3. Enable **Q3D Far Field Calculation**.

Once the beta option is enabled, perform the following steps:

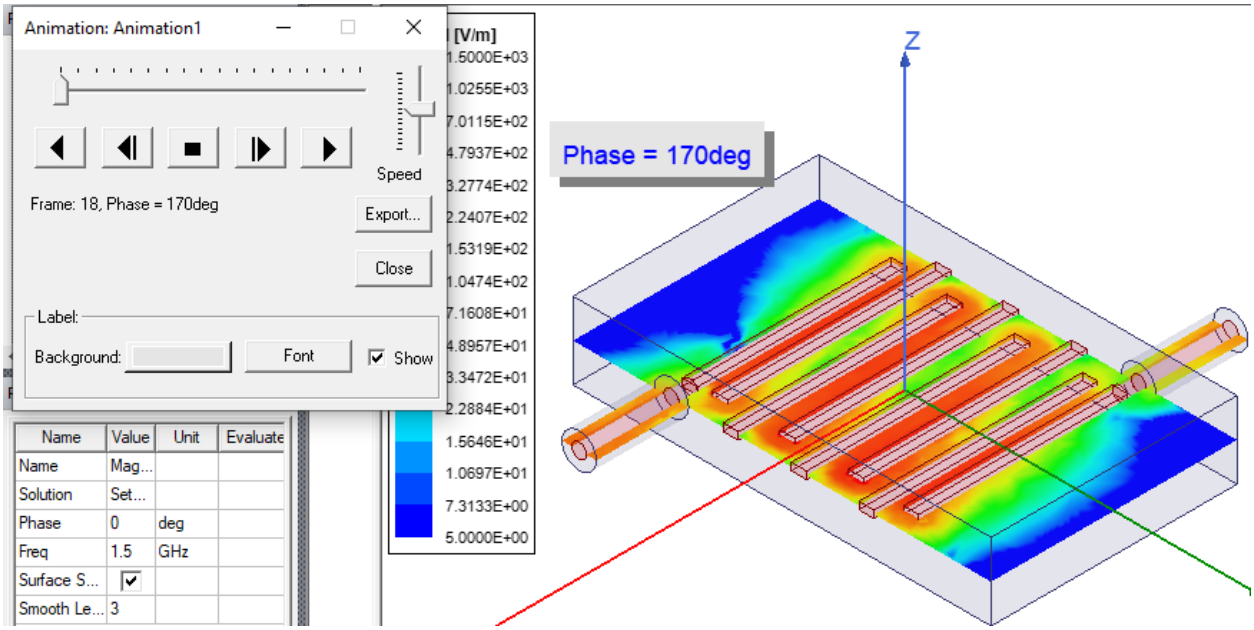
1. In the **Solve Setup window**, enable both **Save Fields** and **Compute Dynamic Fields**.
2. Run the simulation.
3. Use the **EM Fields menu** to **Insert Infinite Sphere Setup**.

Far Fields reports are available from the **Results** menu:

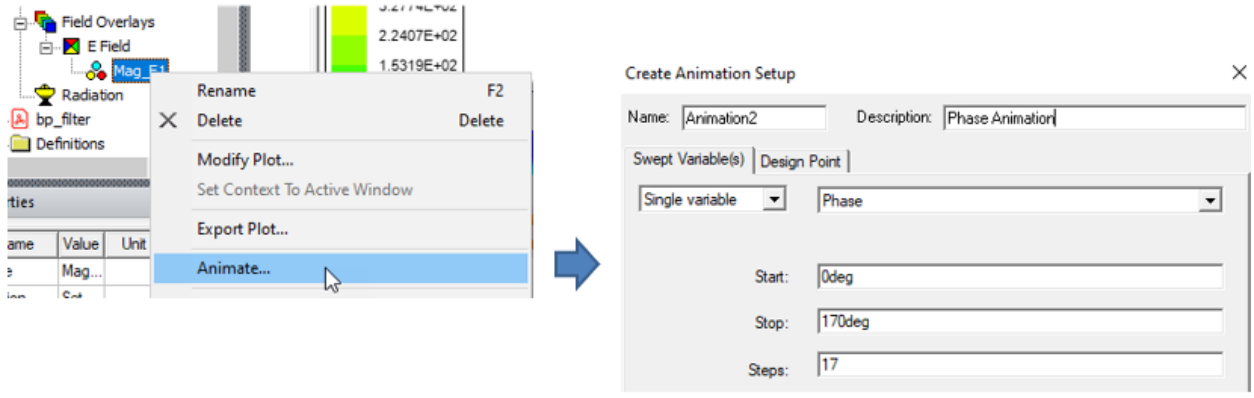


Creating Animations

An animated plot is a series of frames that displays a field, frequency, parametric value, mesh, virtual ray trace or geometry at varying values.



To create an animated plot, you specify the values of the plot that you want to include, just as an animator takes snapshots of individual drawings that make up a cartoon. Each value is a frame in the animation. You specify how many frames to include in the animation.



The following sections describe how to create different kinds of animations.

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.

Note:

Graphics System requirements for Optimal Performance

In order to obtain optimal performance improvements on fields overlay plot, a workstation-class 3D-capable graphics card with at least 512 MB of memory that supports OpenGL version 2.0 or higher is needed.

On Windows, the default OpenGL version support is v1.1, so you might need to update graphics driver to the latest version;

If you access the application through Windows Remote Desktop which only supports Generic GDI (functionally equivalent to OpenGL v1.0), the performance improvement will also not be available;

To view OpenGL version/extensions supported by your card, the OpenGL Extension Viewer tool is accessible via softpedia.com.

If animation is slow, especially for complex models, for some older graphics cards, you can improve performance by accessing **NVIDIA Control Panel**> **3D Settings**> **Manage 3D Settings – Global Settings** tab and choosing the **Workstation App – Dynamic Streaming** option from the **Global Presets** drop-down menu.

You can export the animation to animated Graphics Interchange Format (GIF), to Audio Video Interleave (AVI) format, or to WebM (.webm) format.

Creating Design of Experiments Animations

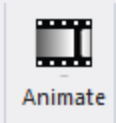
Animations can be based on Design of Experiments setups. Following is the general procedure for creating an animation of a geometry model using a Design of Experiments setup.

Prerequisites

- For animation of geometry solving Design of Experiments setup is not required.

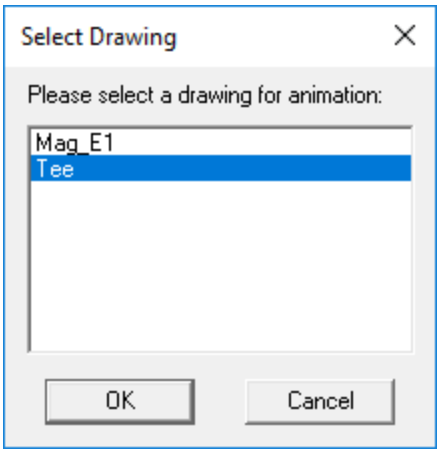
Procedure:

1. Right-click in the **Modeler** window, and click **View**> **Animate...**, right click on the Design of Experiments setup in the Project tree and click **Animate...** in the short-cut menu, or select the **View** tab of the ribbon, and click the **Animate** icon.

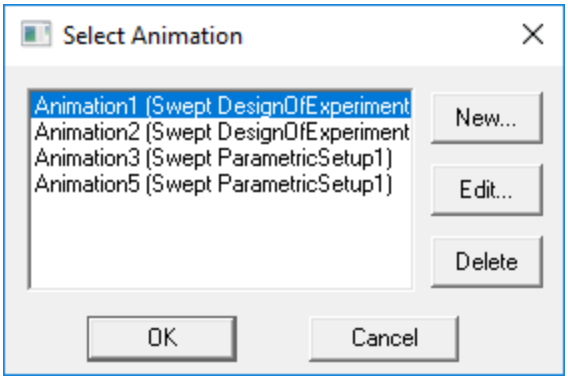


   If multiple geometries can be varied in the design, the **Select Drawing** dialog box appears, proceed to step 2. If only one geometry is assigned variable(s), proceed to step 3.

2. In the **Select Drawing** dialog box select the drawing for animation. The drawing should be appropriate for geometry animation:

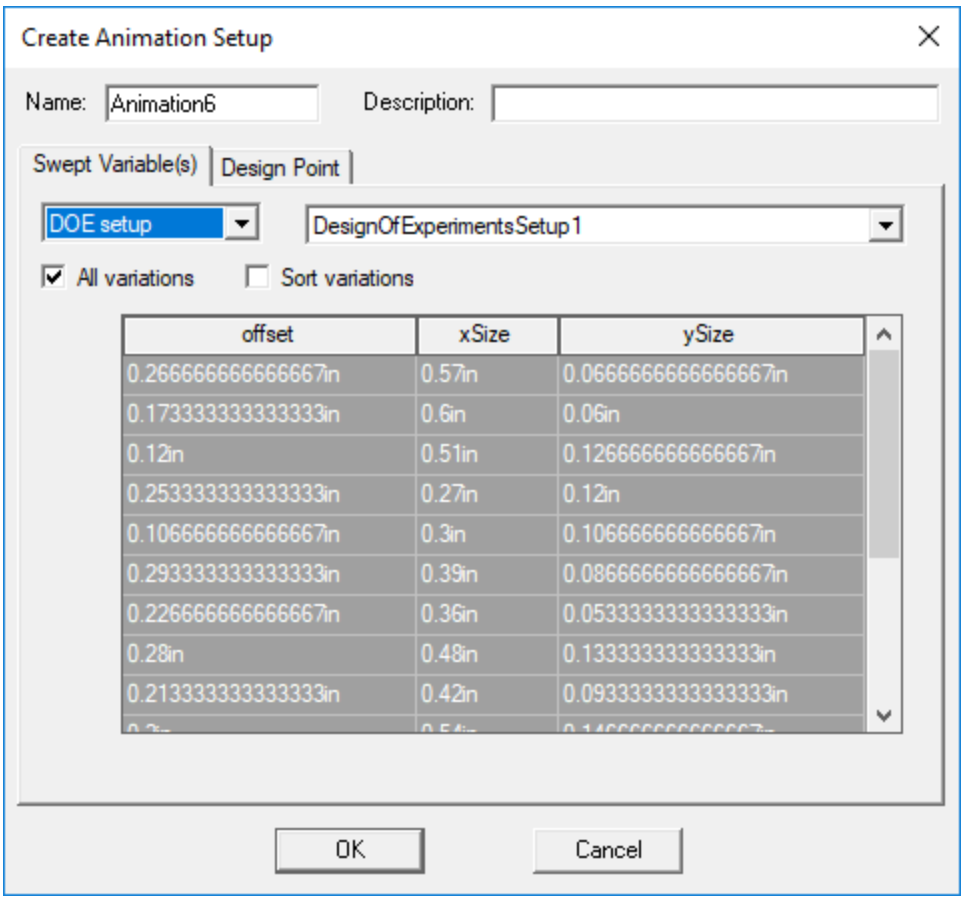


> **Note:**
>
> If previous animations have been created for this project, the **Select Animation** dialog will appear. You may choose an animation setup from the list if one is associated with the geometry variable of interest and the animation will start.



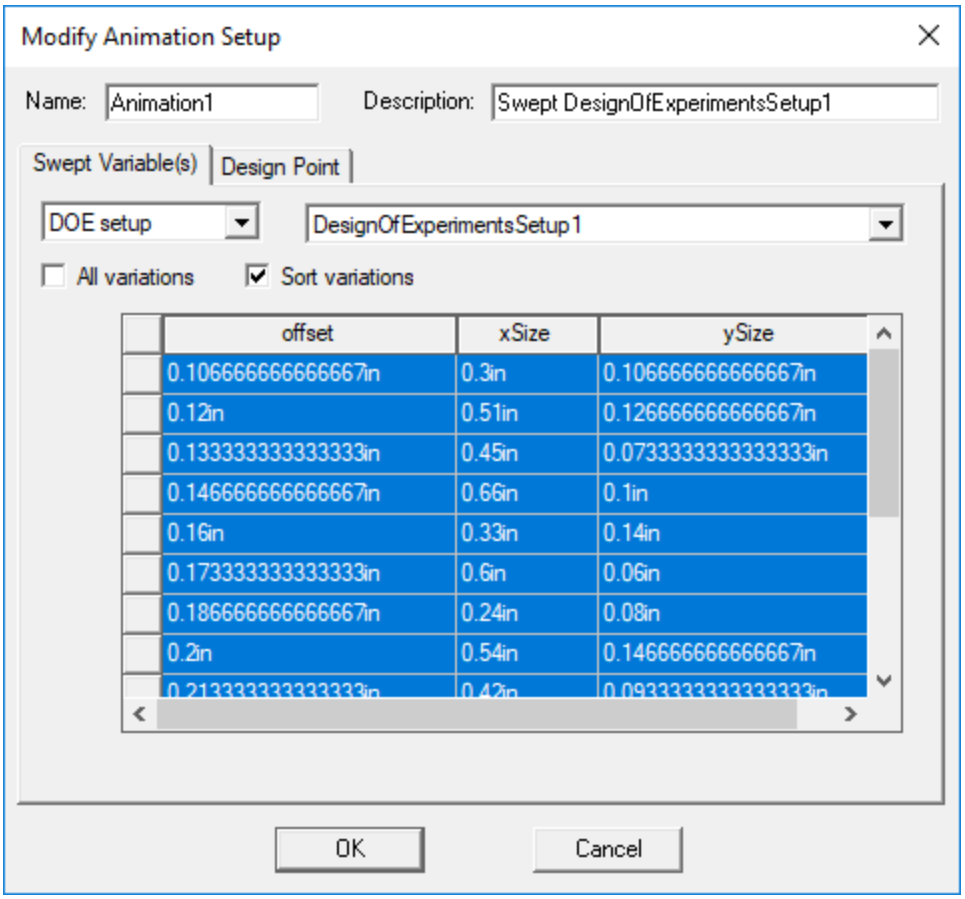
If no existing animation setup is acceptable, select **New** and continue at Step 3 below.

The **Create Animation Setup** dialog box appears.

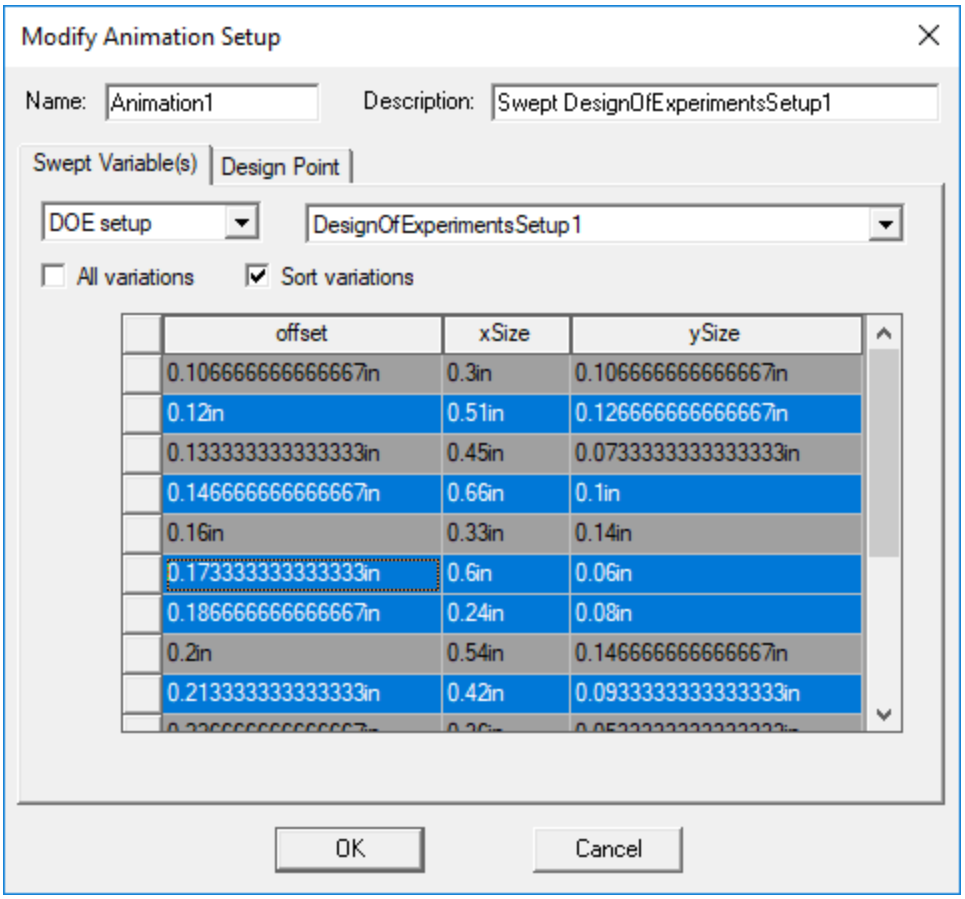


3. If your have created one or more Design of Experiments setups for one or more geometric variables, you can select, **DOE setup** to filter existing DesignOfExperimentsSetups, and the setup to use, and the variable values to animate. With All variations checked, the

variable table is not selectable. If you uncheck All variations, you can use select from the available setup values. All variations are selected [highlighted] when you uncheck All variations.



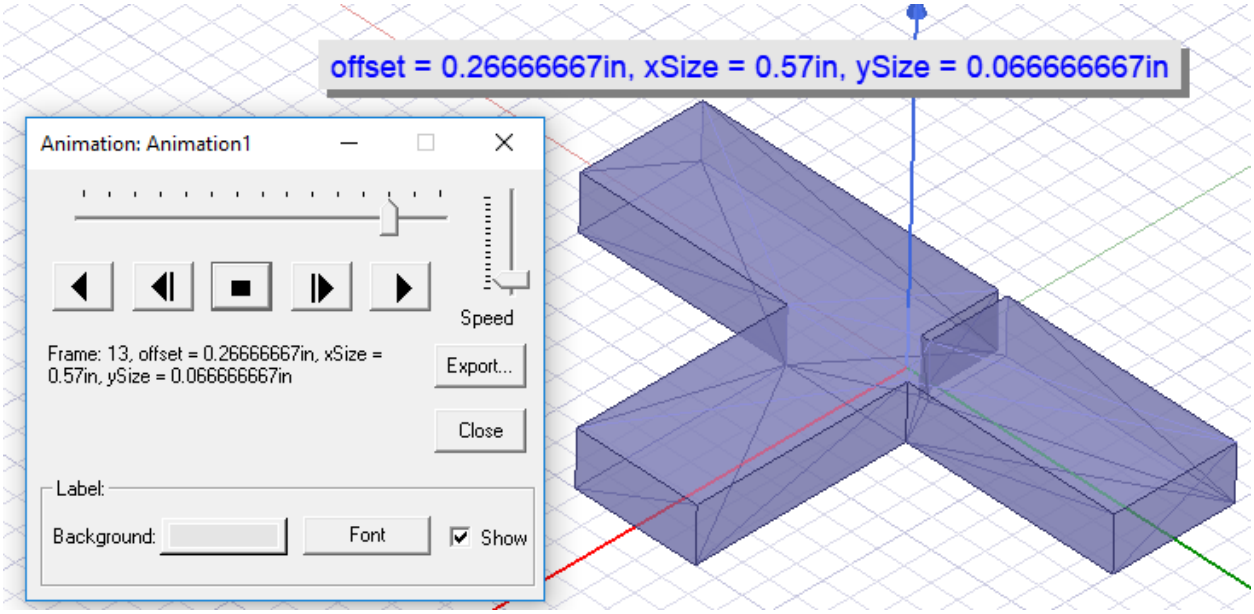
If you check Sort variations, the values are sorted. You can click to select the variations you want.



4. Click **OK**.

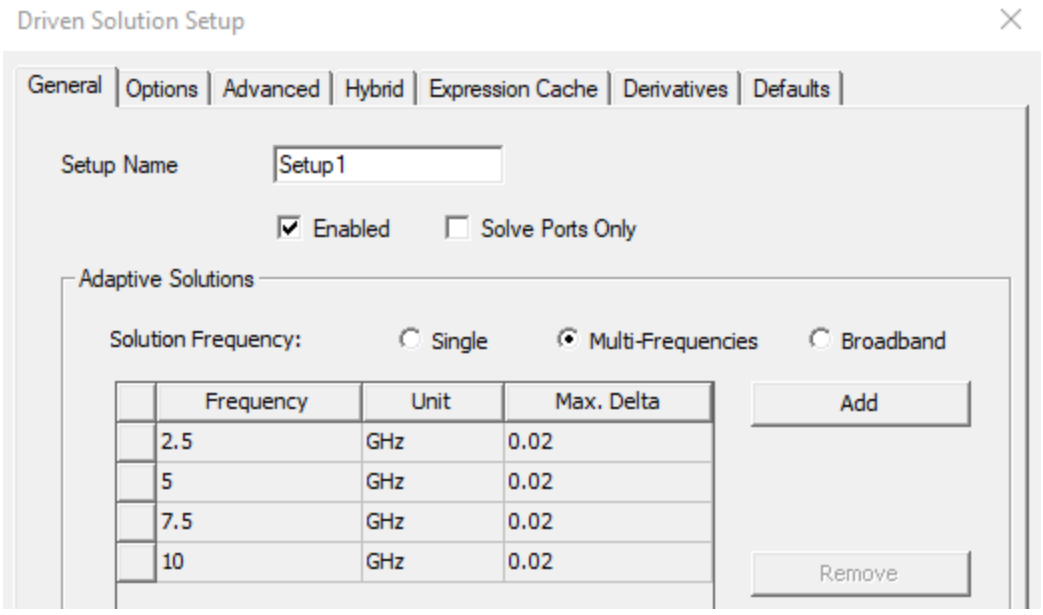
   The animation begins in the view window. It will display one frame for each variable value.

   The **Animation** control panel lets you to stop, restart, and control the speed and sequence of the frames.



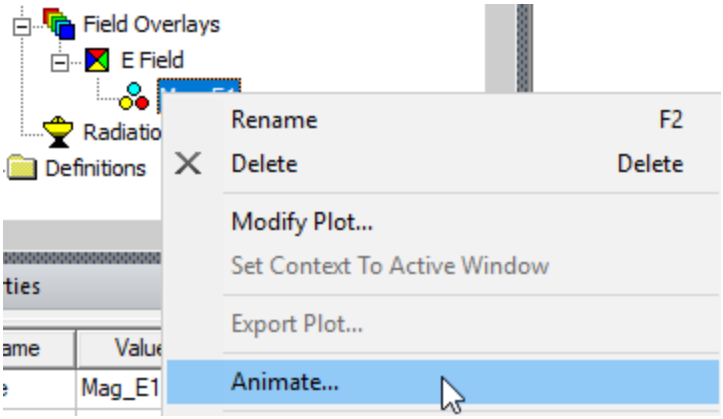
Creating Frequency Animations

1. In order to create Frequency animations, you must define the frequencies of interest in the Solution setup, using the Multi-Frequencies option.

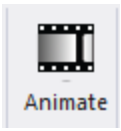


2. After running the analysis, Create a field overlay plot to animate.
3. Do one of the following:

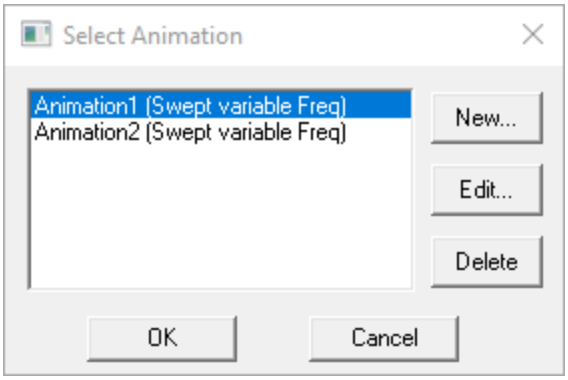
- Click **HFSS>Fields>Animate**  .
- Right-click the field overlay plot of interest and click **Animate...** from the shortcut menu.



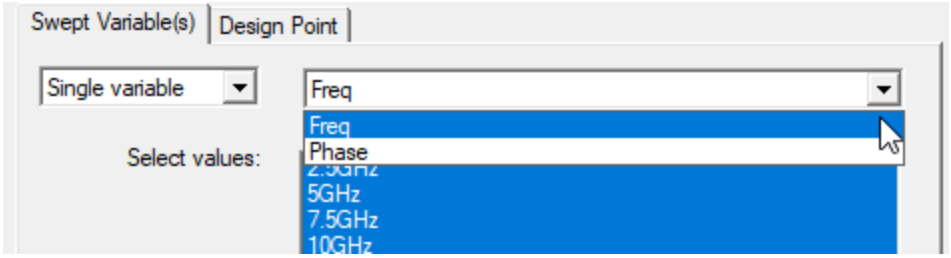
- Select the field overlay plot of interest, select the **View** tab of the ribbon, and click the Animate icon.



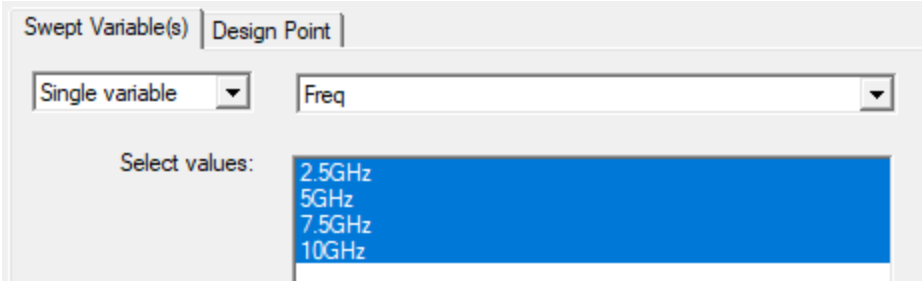
4. If you already created an animation, the **Select Animation** dialog box appears



Selecting an existing animation from that list starts it when you click **OK**. Click **Edit...** to open the **Modify Animation Setup** dialog. To create a new animation, click **New** and the **Create Animation Setup** dialog box appears.

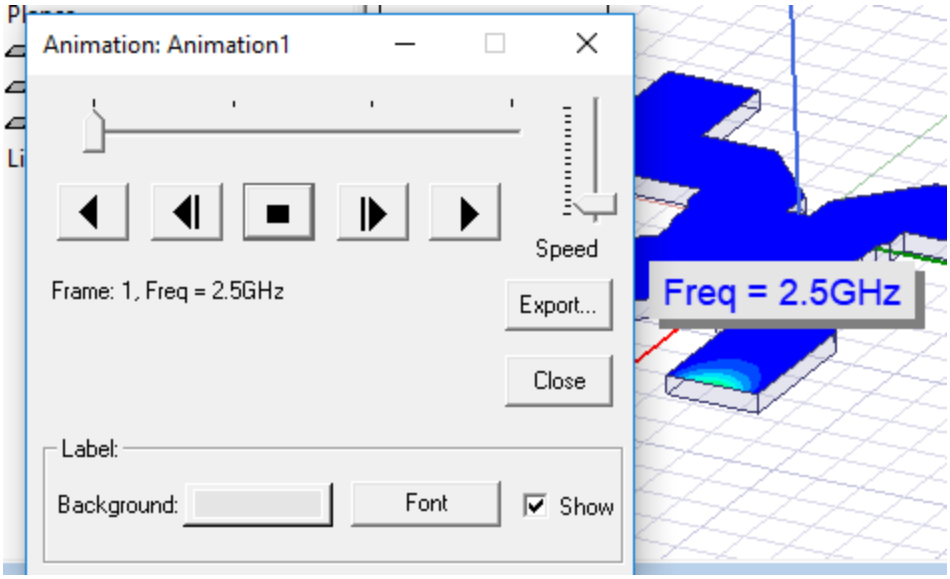


5. Under the **Swept Variable** tab, select **Frequency** from the **Swept Variable** list.
6. Select the frequency values you want to include in the animation from the **Select values** list.



   Use the **Shift** key to select a series of values, and the **Ctrl** key to select values that are not in sequence.
7. If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.
8. Click the **Design Point** tab.
9. Deselect the **Use defaults** check box. In the table, select the row corresponding to the variable setting of interest.
10. Click **OK** and the animation begins in the view window. It will display one frame for each frequency value you selected. 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.



Creating Geometry Animations

You can create geometry animations to evaluate the effect of varying geometry variables on the model. You can also create more complex animations by creating time variables associated with object locations. This section describes the general procedure for geometry animation, and shows a geometry animation with a field overlay, and a second animation for an HFSS Antenna Example project that includes multiple objects moving with respect to a time variable.

Prerequisites

You must define at least one variable associated with the geometry before you create a geometry animation.

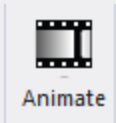
If you want to overlay a Field Solution on a geometry animation you must have solved fields for an Optimetrics setup using the same variable.

Procedure:

Following is the general procedure for creating an animation that varies a part of the model geometry.

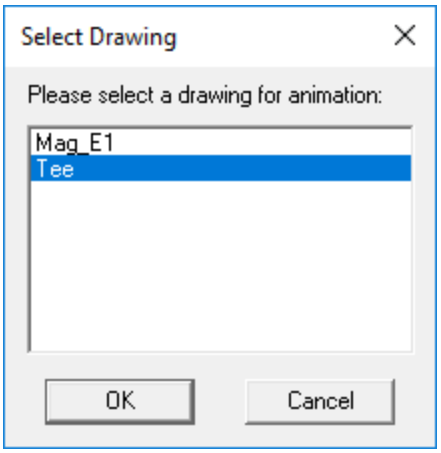
1. Right-click in the *Modeler* window, and click to **View** > **Animate...** or

   Select the **View** tab of the ribbon, and click the **Animate** icon.

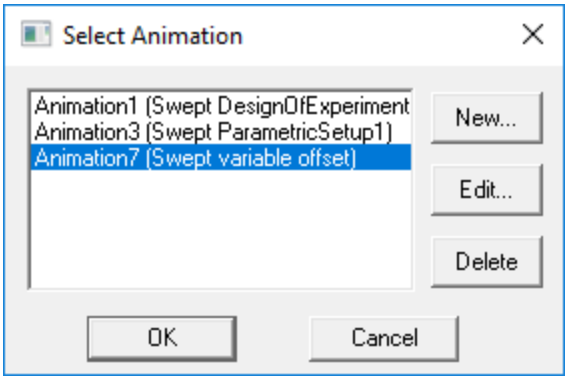


If multiple geometries can be varied in the design, the *Select Drawing* dialog box appears; proceed to step 2. If only one geometry is variable, proceed to step 3.

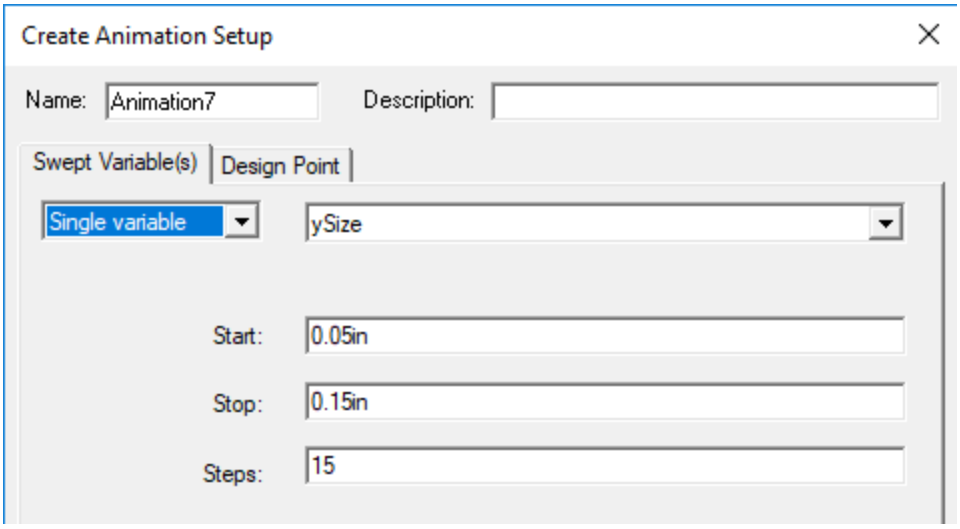
2. In the *Select Drawing* dialog box, select the object that you want to animate.



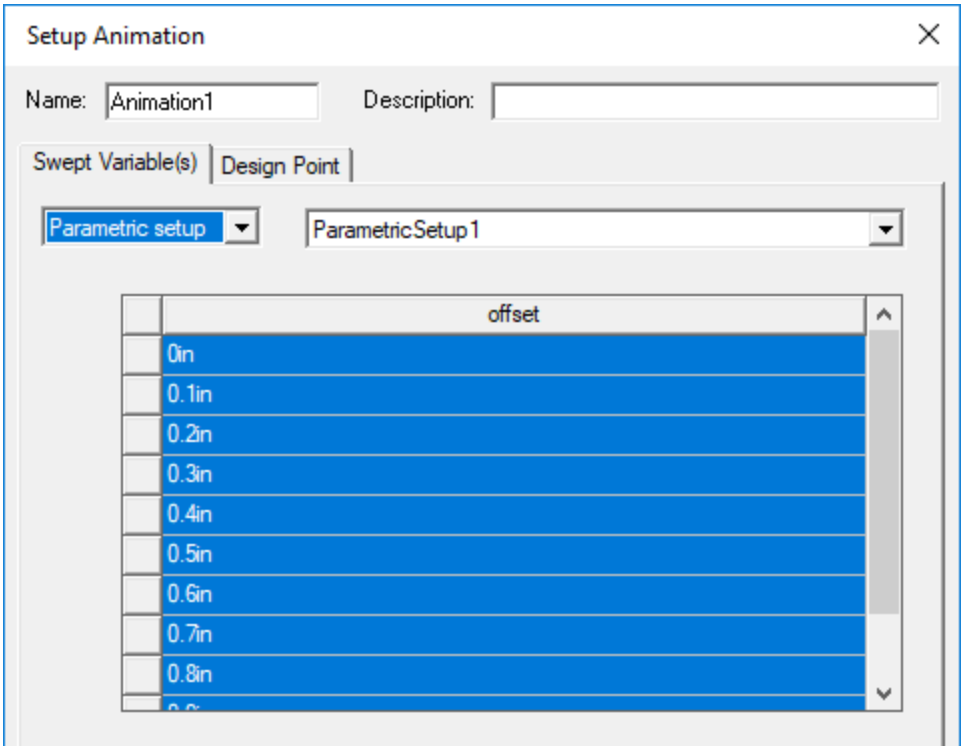
If previous animations have been created for this project, the *Select Animation* dialog box will appear. Selecting an existing animation from that list starts it when you click **OK**. Click **Edit...** to open the *Modify Animation Setup* dialog box. You may choose an animation setup from the list 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.



The *Create Animation Setup* dialog box appears:



3. For single variable:
   a. Select the geometry variable that you want to animate from the **Swept Variable** list.
   b. Specify the Start, Stop, and Steps values to include in the animation.
4. If your have created one or more Parametric Sweeps for one or more geometric variable, you can select, **Parametric setup**, the setup to use, and the variable values to animate.



5. If you have created a **Design of Experiments** setup, you can select to define an animation based on that setup and values.
6. If the design has multiple project or intrinsic variables, you use the **Design Point** tab to set the values of the non-animated variables.
7. Click **OK**.

   The animation begins in the view window. It will display one frame for each variable value.

Creating Mesh Animations

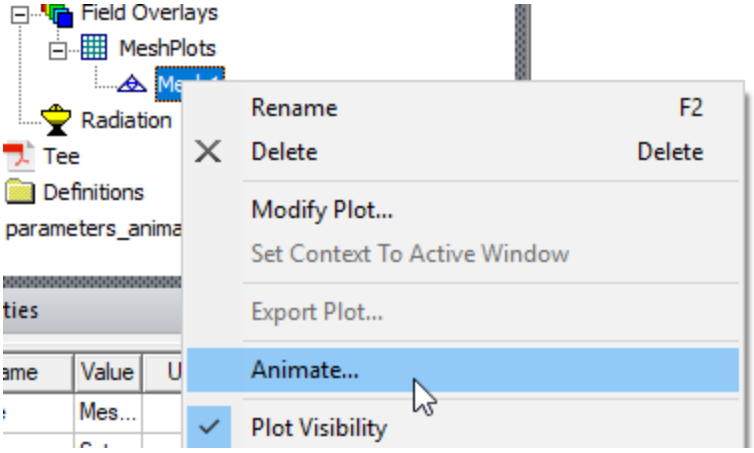
You can create mesh animations to evaluate the effect of single variables or parametric sweep variables on the model. Following is the general procedure for creating a mesh animation that varies a part of the model geometry.

Prerequisites:

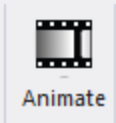
- You must define at least one variable associated with the geometry before you create a mesh animation.
- You must create a Mesh plot.

Procedure:

1. Select the Mesh plot in the Project Manager, right-click in the *Modeler* window, and click **View> Animate...**, or right-click the Mesh plot, and click **Animate...**



   Select the **View** tab of the ribbon, and click the **Animate** icon.

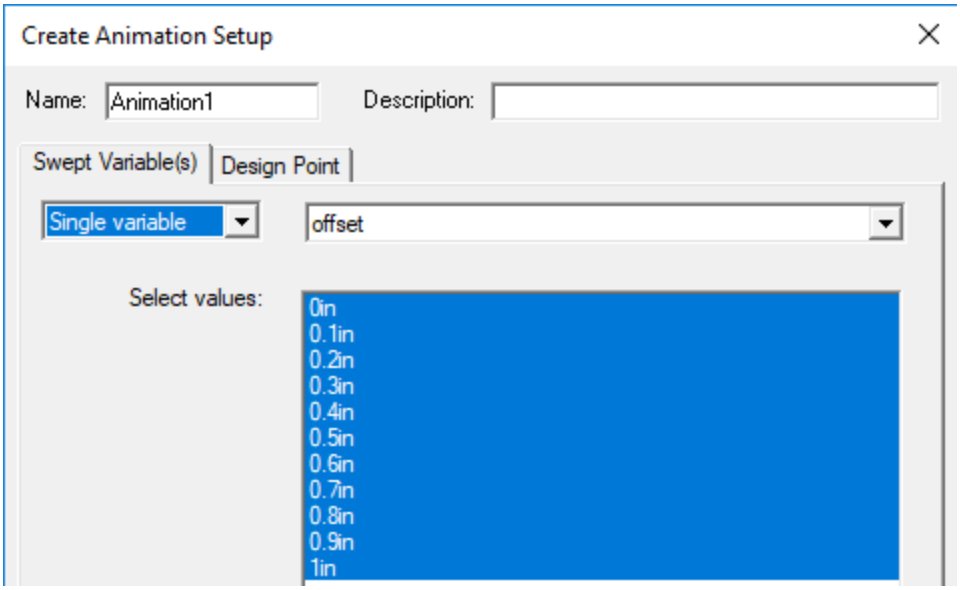


If multiple geometries can be varied in the design, the *Select Drawing* dialog box appears; proceed to step 2. If only one geometry is variable, proceed to step 3.

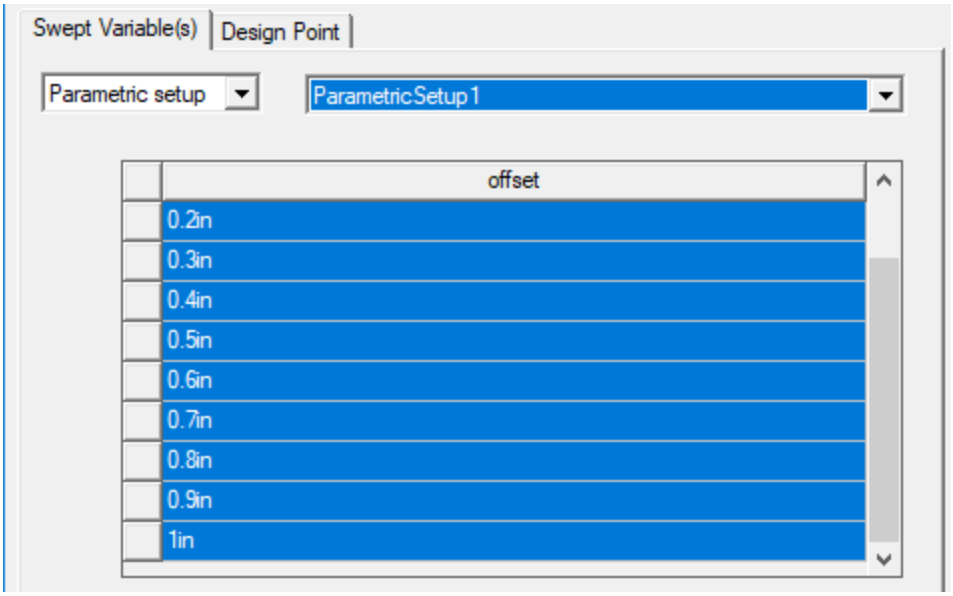
2. In the *Select Drawing* dialog box, select the object that you want to animate.

> **Note:**
>
> If previous animations have been created for this project, the *Select Animation* dialog box will appear. Selecting an existing animation from that list starts it when you click **OK**. Click **Edit...** to open the *Modify Animation Setup* dialog box. If no existing animation setup is acceptable, select **New** and continue below.

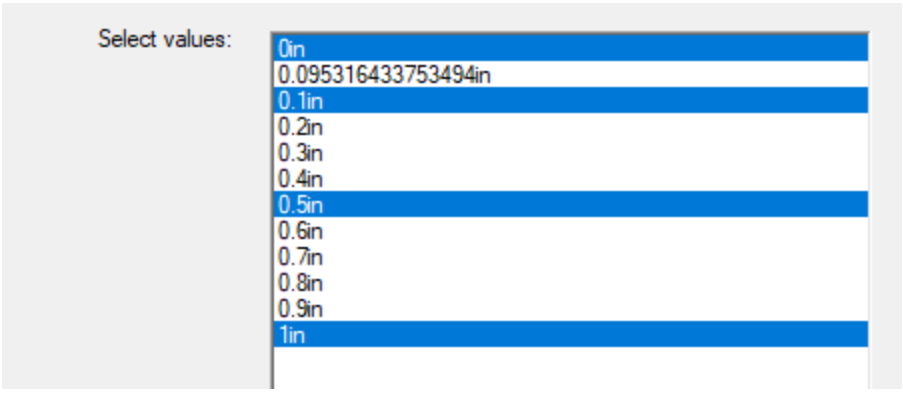
The *Create Animation Setup* dialog box appears.



3. In the *Create Animation Setup* dialog box:
   a. Select the Single variable, or Parametric setup, or DOE setup if available. If multiple single variables are available, you can select from a drop-down menu. If one or more Parametric setups exist, you can choose from the drop-down menu. The values for the selected parametric setup are displayed.



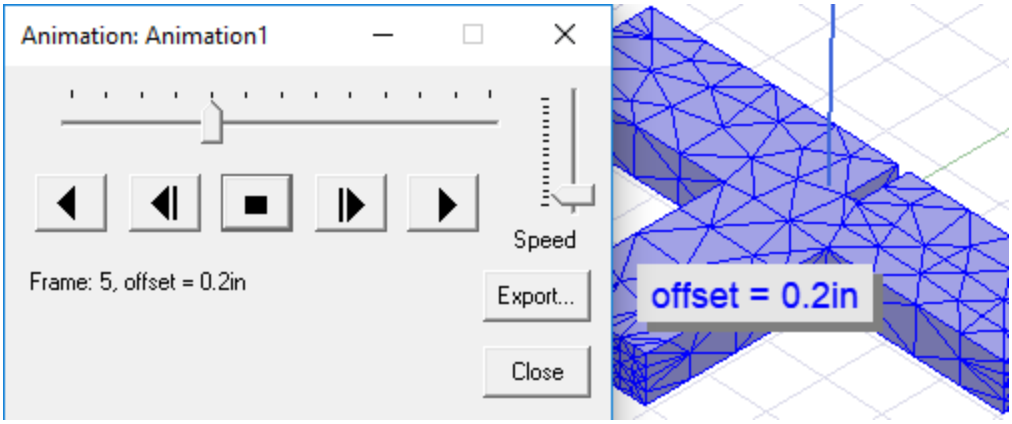
b. If desired, specify the values to include in the animation. Selected values are highlighted.



c. If the design has multiple project or intrinsic variables, you use the **Design Point** tab to set the values of the non-animated variables.

4. Click **OK**.

The animation begins in the view window. It will display one frame for each variable value.



The *Animation* control panel lets you to stop, restart, and control the speed and sequence of the frames.

Creating Parametric Animations

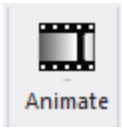
Parametric animations can be based on parametric sweeps. Following is the general procedure for creating an animation of a plot based a parametric sweep.

Prerequisites

- For animation of a field plot/overlay plot over a geometric variable, you must define and solve at least one optimetric sweep with same variable values and Save Fields selected before you can create a parametric animation of a field solution. See Creating Geometry Animations.
- For animation of geometry and VRT plots, solving parametric setup is not required.
- Animations can be based on time variables and objects whose location is linked to the time variable. See the animated example below.

Procedure:

1. Right-click in the **Modeler** window, and click **View**> **Animate...**, or select the **View** tab of the ribbon, and click the **Animate** icon.

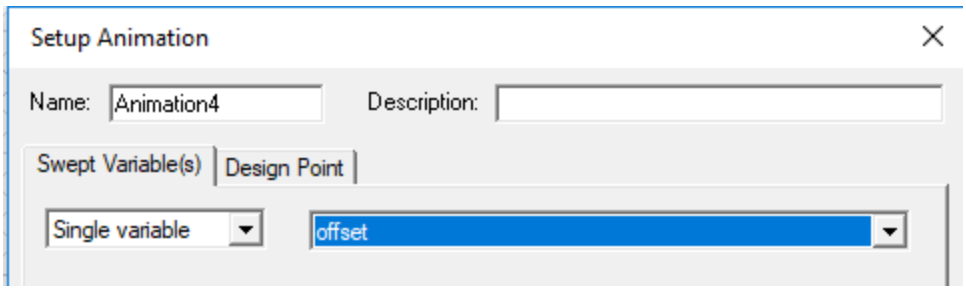


If multiple geometries can be varied in the design, the *Select Drawing* dialog box appears; proceed to step 2. If only one geometry is variable, proceed to step 3.

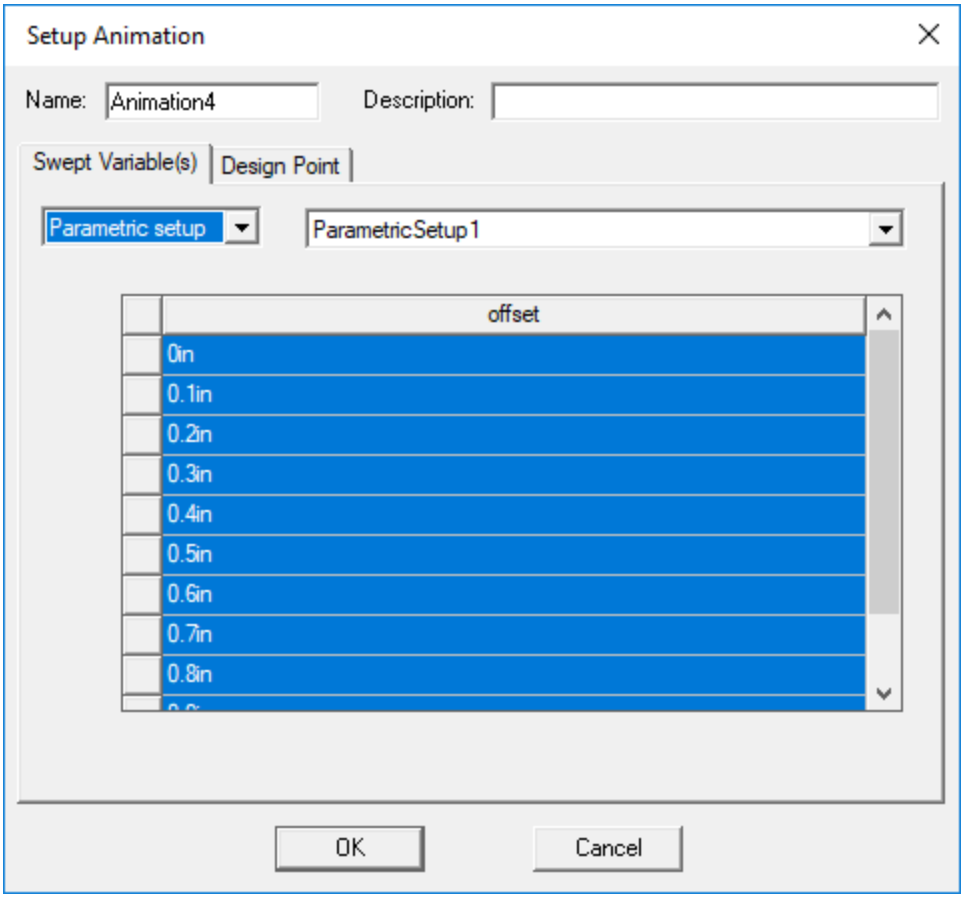
2. In the *Select Drawing* dialog box:
   a. Select the geometry variable to vary in the animation.
   b. Select the object you want to animate.

> **Note:**
>
> If previous animations have been created for this project, the *Select Animation* dialog box 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.

The *Setup Animation* dialog box appears.



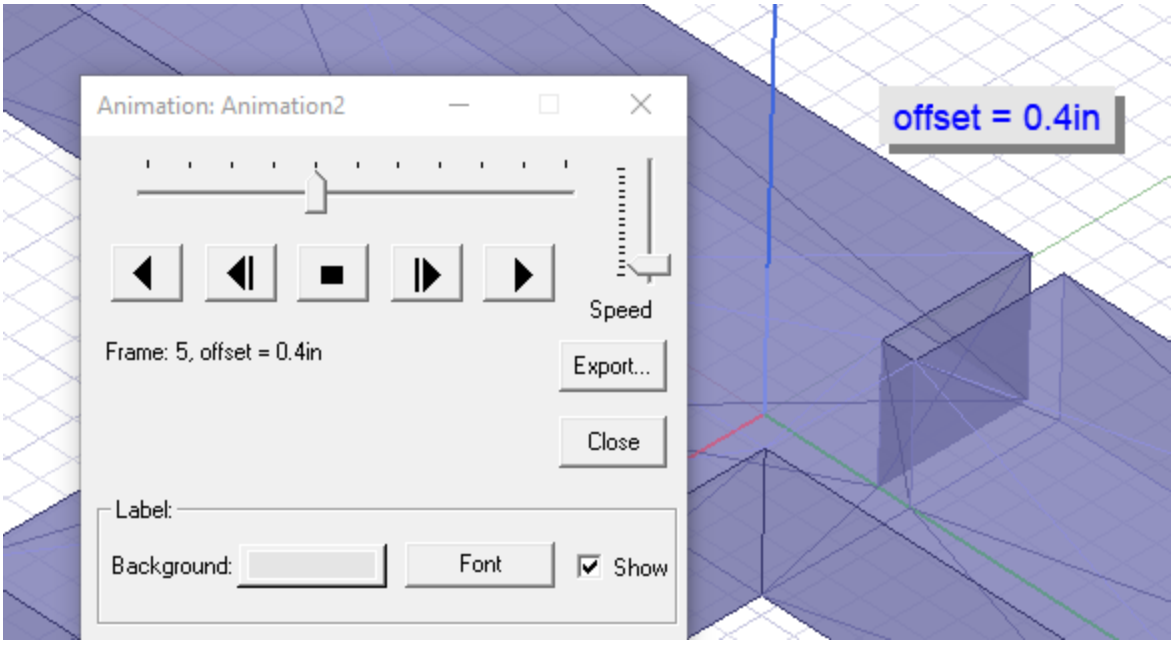
3. If your have created one or more Parametric Sweeps for one or more geometric variables, you can select **Parametric setup**, the setup to use, and the variable values to animate.



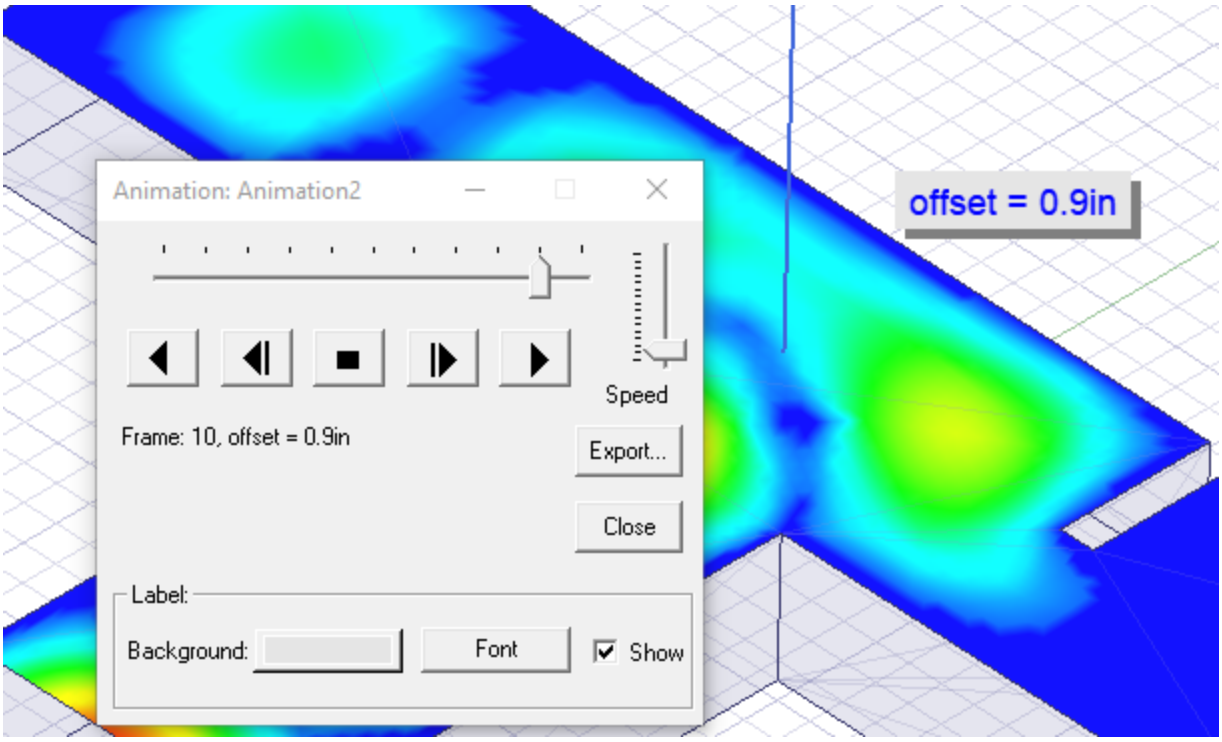
4. Click **OK**.

   The animation begins in the view window. It will display one frame for each variable value.

   The **Animation** control panel lets you to stop, restart, and control the speed and sequence of the frames.



While a parametric animation is running, you can also turn on other field plots.



Creating Phase Animations

You can create phase animations of field plots of frequency sweeps.

Prerequisites

Before creating a phase animation, you must solve a frequency sweep for the project.

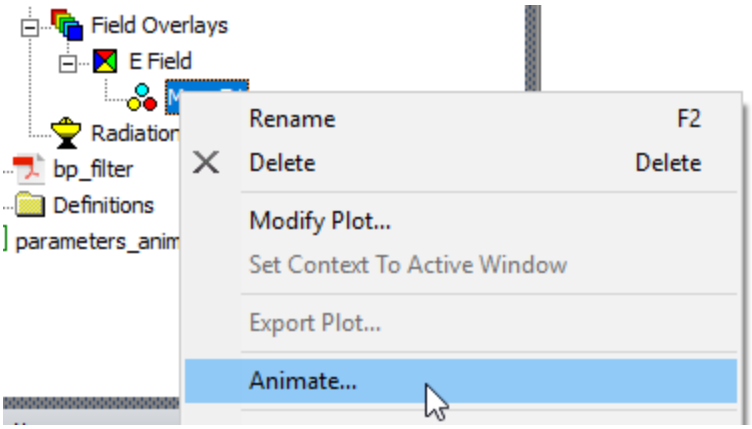
Create a field plot.

Procedure:

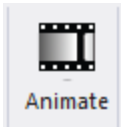
To animate a plot with respect to the phase of the plotted field:

1. Create a field overlay plot to animate.
2. Select the plot in the **Project** tree, and click **HFSS>Fields>Animate...**, or right-click in the **Modeler** window, and click **View>Animate...**.

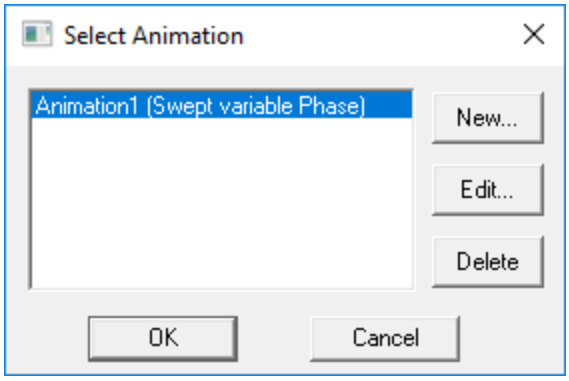
Or right-click the field overlay plot of interest and click **Animate...** from the shortcut menu,



Or select the field overlay plot of interest, select the **View** tab of the ribbon, and click the **Animate** icon.

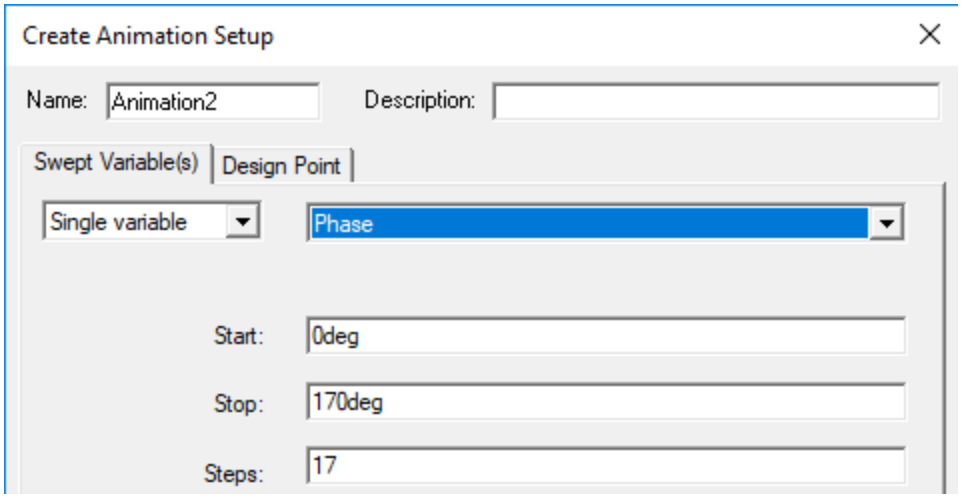


If you already created an animation, the **Select Animation** dialog box appears. Selecting an existing animation from that list starts it.



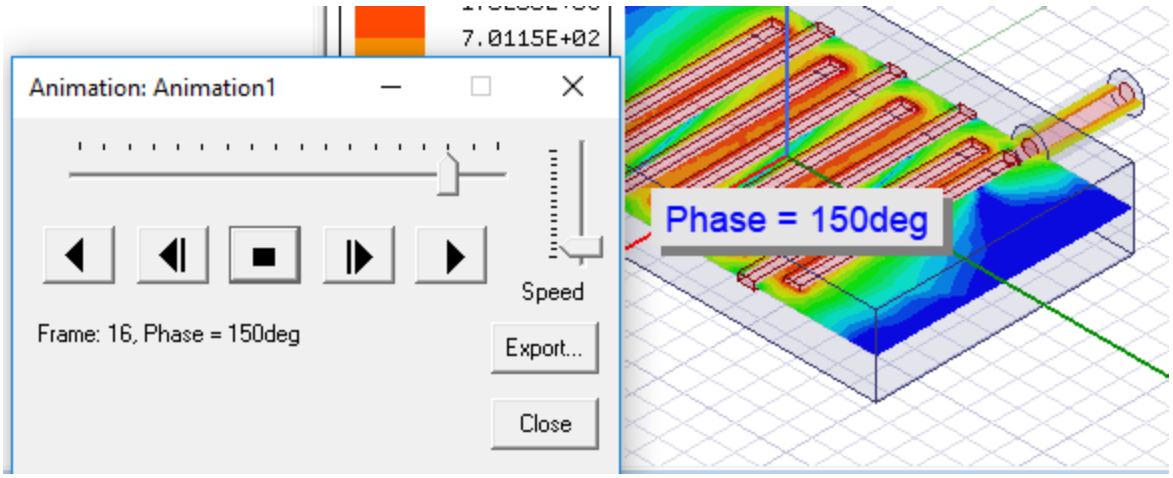
Selecting an existing animation from that list starts it when you click **OK**. Click **Edit...** to open the **Modify Animation Setup** dialog.To create a new animation, click **New**.

The **Create Animation Setup** dialog box appears.



3. Under the **Swept Variable** tab, select **Single Variable** and accept Phase.
4. Specify the phase values you want to include in the animation, as Start, Stop, and Steps to include (e.g., if the **Start** value is **10**, the **Stop** value is **160**, and the number of steps is **10**, the animation will display the plot at 10 phase values between 10 and 160. The start value will be the first frame displayed, resulting in a total of 11 frames in the animation.). If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.
5. Click the **Design Point** tab.
6. Deselect the **Use defaults** check box.
7. In the table, select the row corresponding to the variable setting of interest.

8. Click **OK** and the animation begins in the view window. 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.



Creating Geometry Animations in 2D Extractor

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**.

The **Setup Animation** dialog box appears.

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 geometric project and design variables. Select the geometry variable that you want to animate from the **Swept variable** drop-down menu.
5. Specify the values of the variable that you want to include in the animation:
   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 **Step Size** 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.

6. Click **OK**.

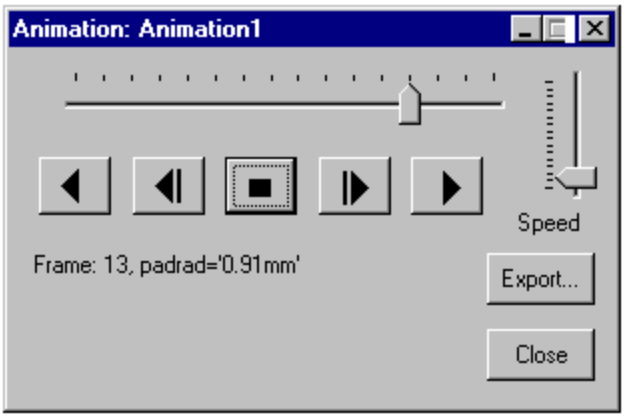
The animation begins in the view window. It will display one frame for each variable value.

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.

Controlling the Animation Display in 2D Extractor

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.



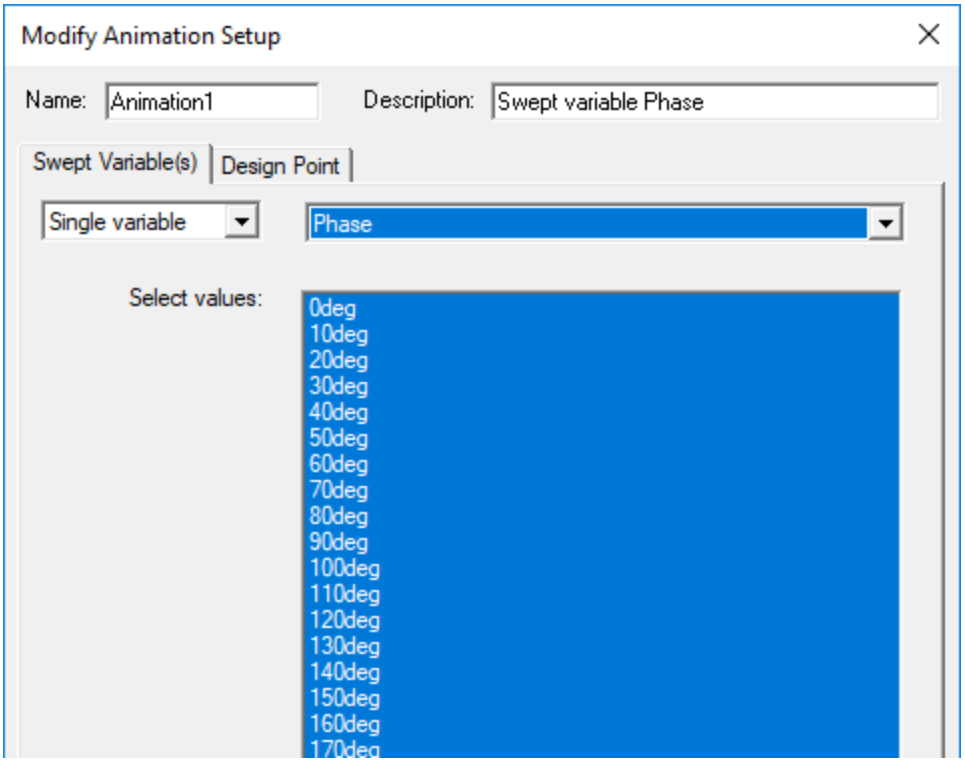
Animation slider	Each dot on the slider represents a frame in the animation. Drag the slider to the right to display the next frame in the animated plot. Drag the slider to the left to display the previous frame.
	Plays the plot 's animation sequence backwards.
	Steps backward through the animated plot 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.
Close	Closes the animation window.

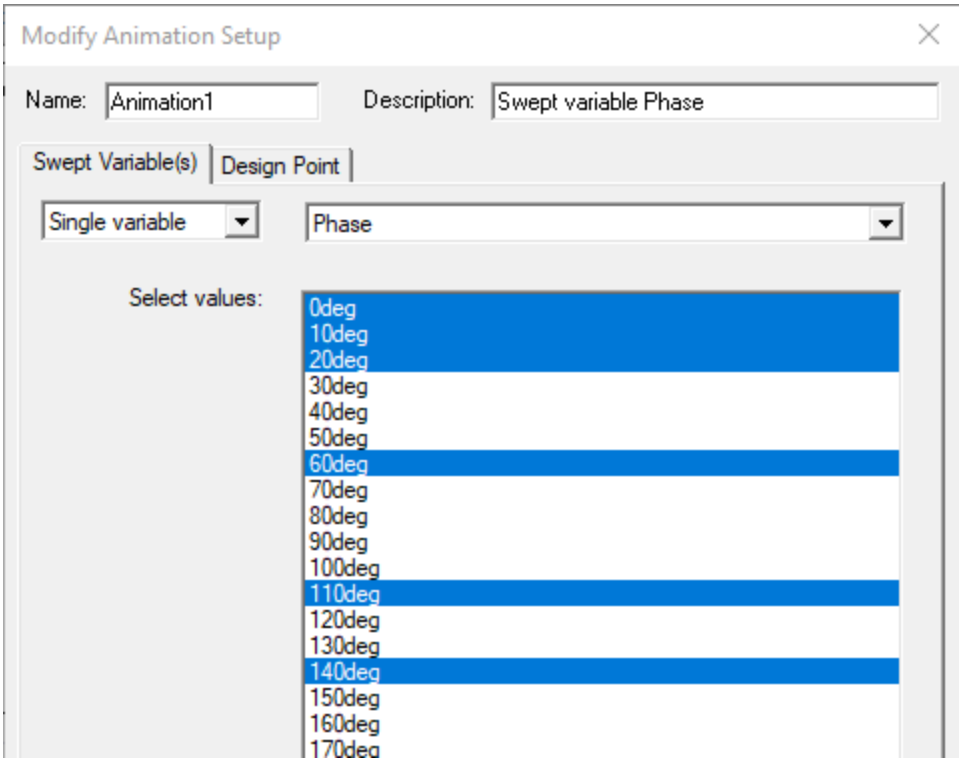
Modifying an Animation Setup

The **Edit** button on the *Select Animation* dialog box opens the *Modify Animation Setup* dialog box.

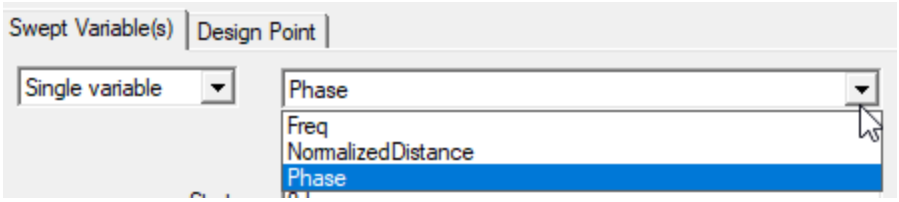
It opens showing the selections for variables and the selected values.



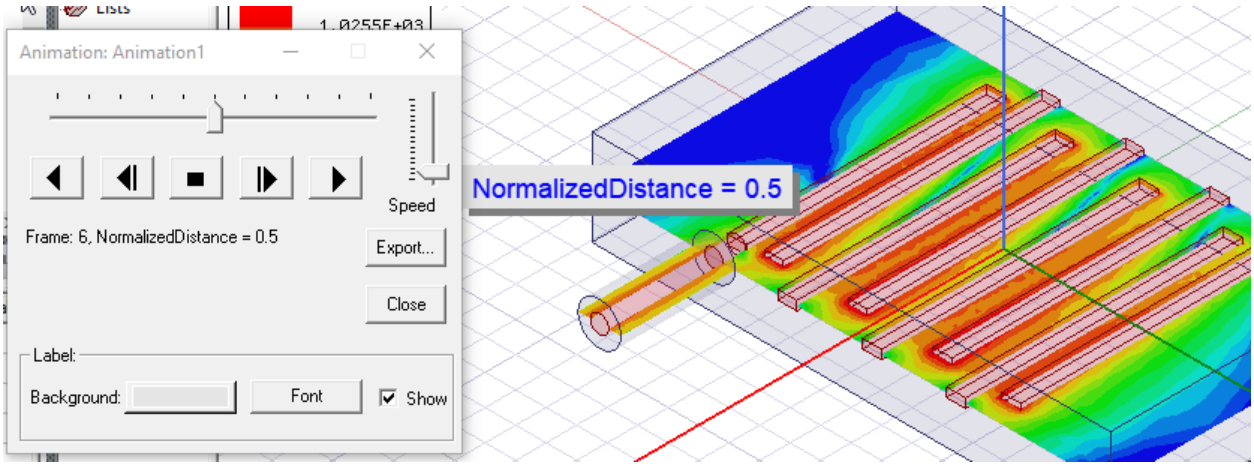
You can de-select or select any values.



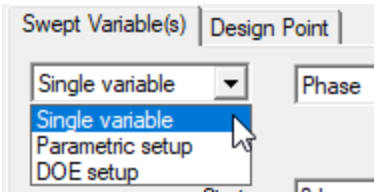
If other variables have been defined, you can select from the drop down list.



If the other variables have or can be set to appropriate values, you can modify the animation.



If Parametric Setups or DOE Setups are available, you can select these.

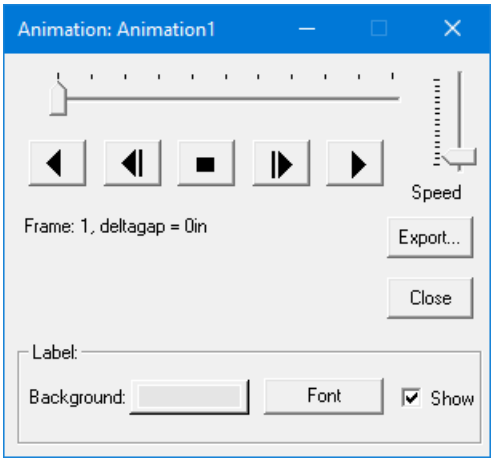


Creating Animations

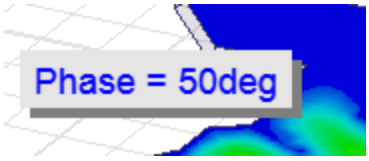
Controlling the Animation's Display

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 image below to learn its function.

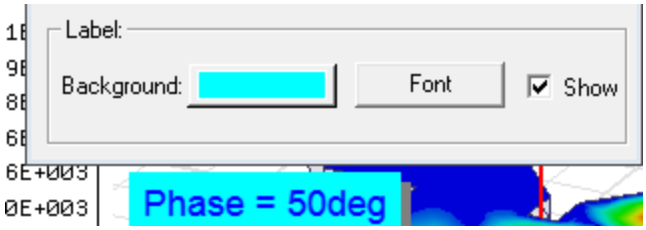


Animation slider	Each dot on the slider represents a frame in the animation. Drag the slider to the right to display the next frame in the animated plot. Drag the slider to the left to display the previous frame in the animation.
	Plays the plot's animation sequence backwards.
	Steps backward through 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 check box	If you select the Show check box, a label showing the swept variable value appears in the animation. You can select the label with the mouse and drag it to a location.



Background button.

Click the background button to open a color pallet dialog that lets you set the background for the swept variable label.



Font

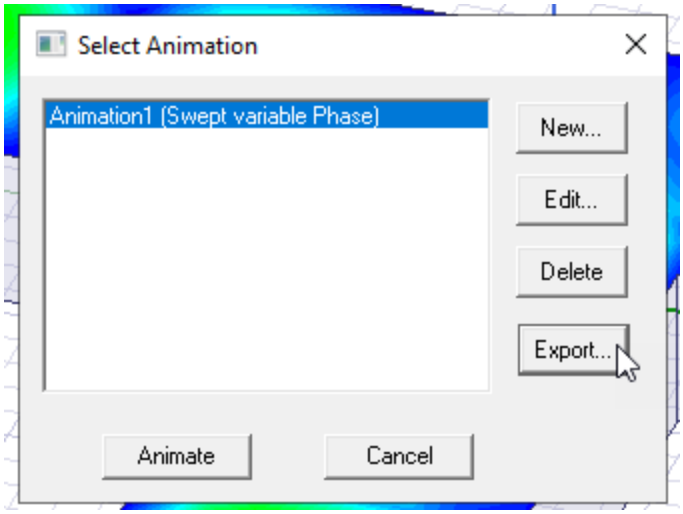
The font button opens a font selection dialog that you can use to set the Font, Font Style, and Size for the label. The Default is Arial Narrow 14pt.

Exporting Animations

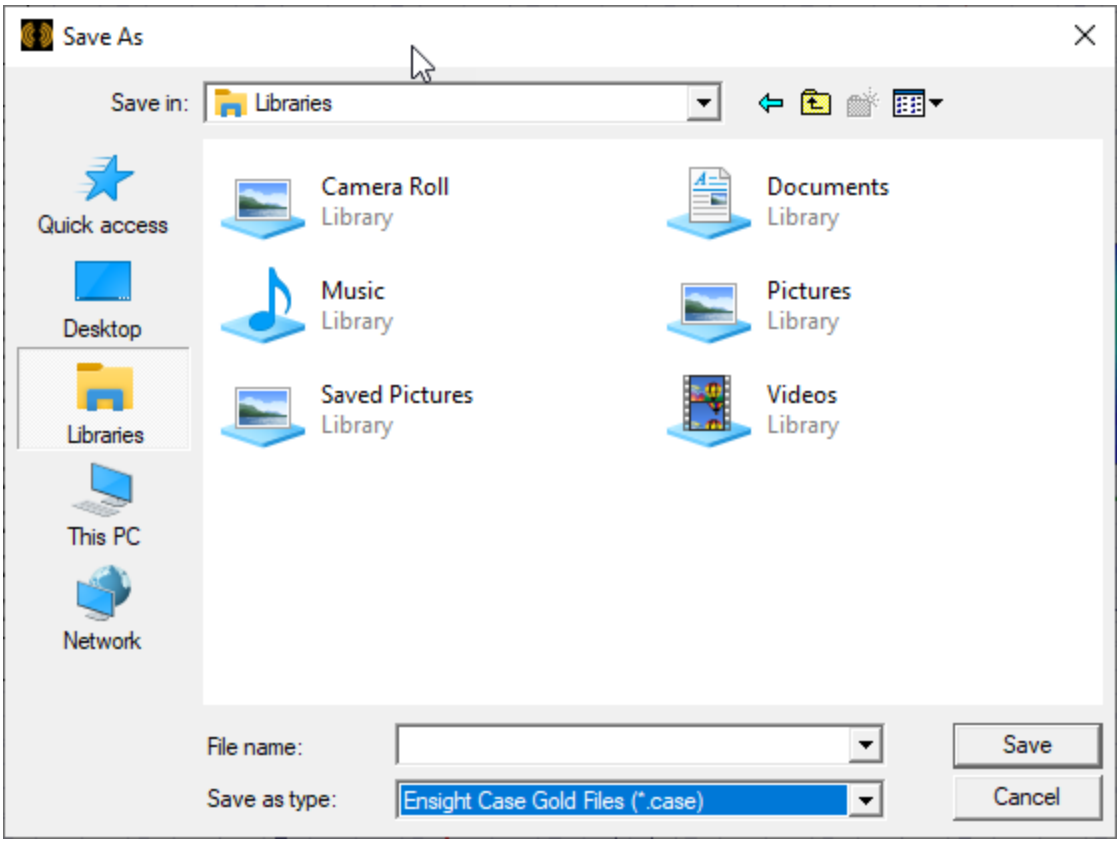
You can export animations from the **Animation** control dialog, or, for Ensight format, from the **Select Animation** dialog.

Exporting Ensight from the Select Animation Dialog

1. Create the animation you want to export.
2. In the **Select Animation** dialog, select **Export...**

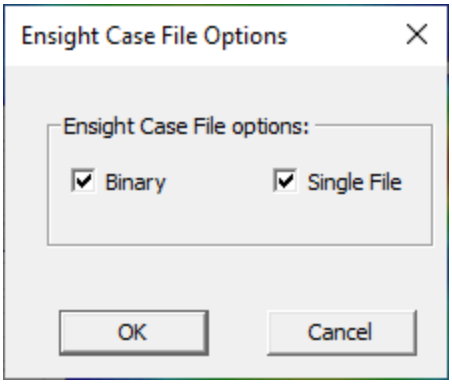


3. This opens the Browser window.



Navigate to the desired location and provide a file name. The format will be Ensight .case files.

4. When you select **Save**, the **Ensight Case File Options** dialog opens.

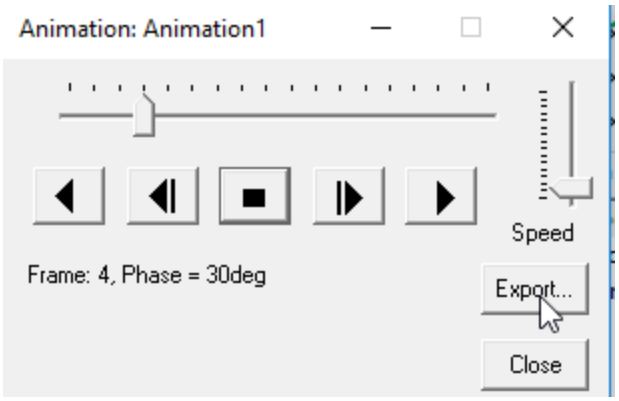


The binary option permits a more compact and efficient animation creation process in Ensight. The Single File options leverages the Ensight transient dataset capability to export all frames of field/geometry data to a single case file. On **OK**, every visible and

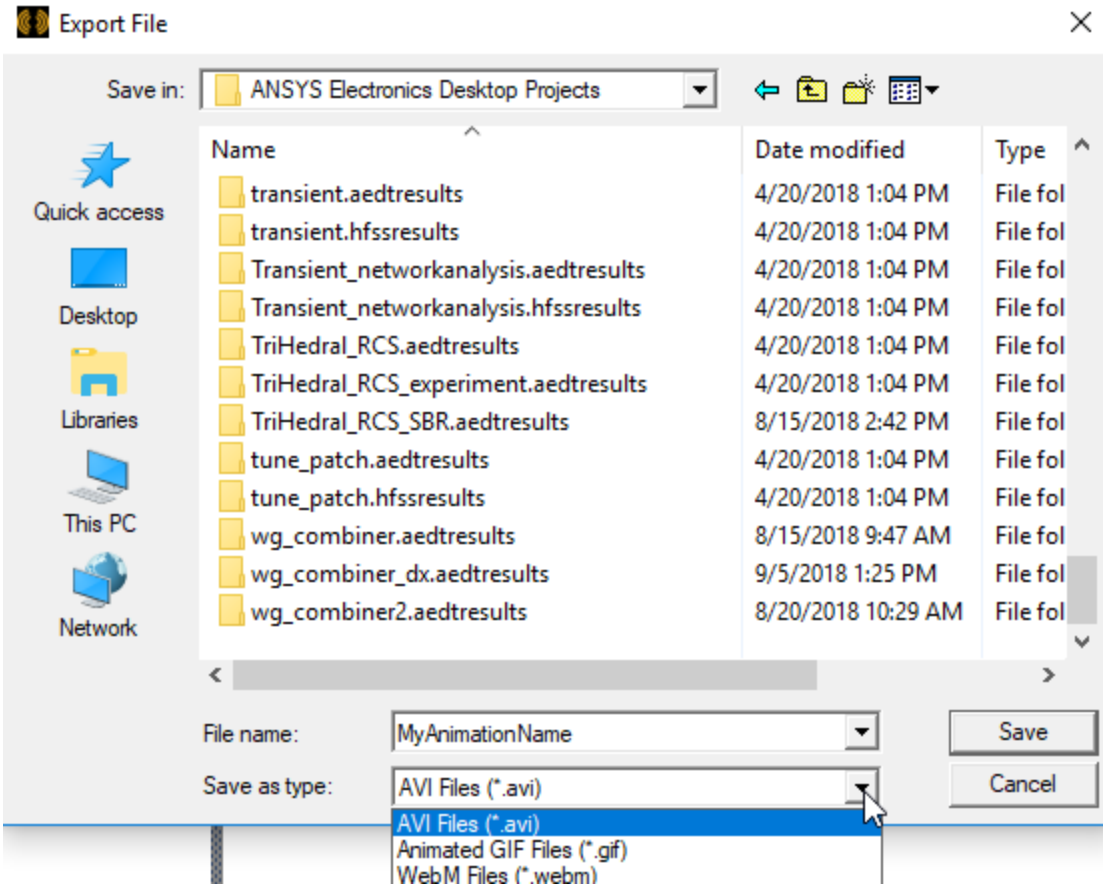
animatable field plot should generate a case file with name in the format of "UserTypedName_fieldPlotName.case"

Exporting from the Animation Control Dialog

1. Create the animation you want to export.
2. In the play panel, click **Export...**.

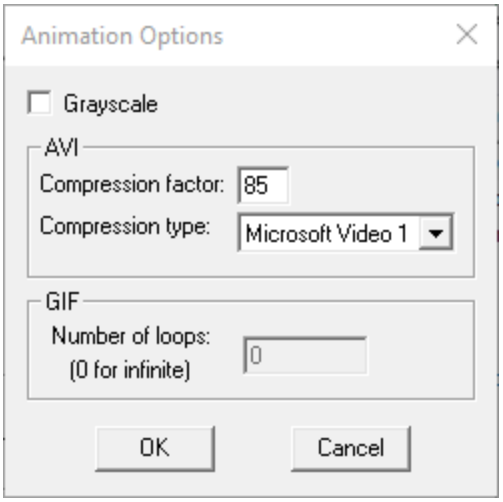


The *Export File* dialog box appears.



3. Specify the directory to **Save in**, the **File name**, and use the **Save as type** drop-down menu to select **Animated GIF File (.gif), 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.

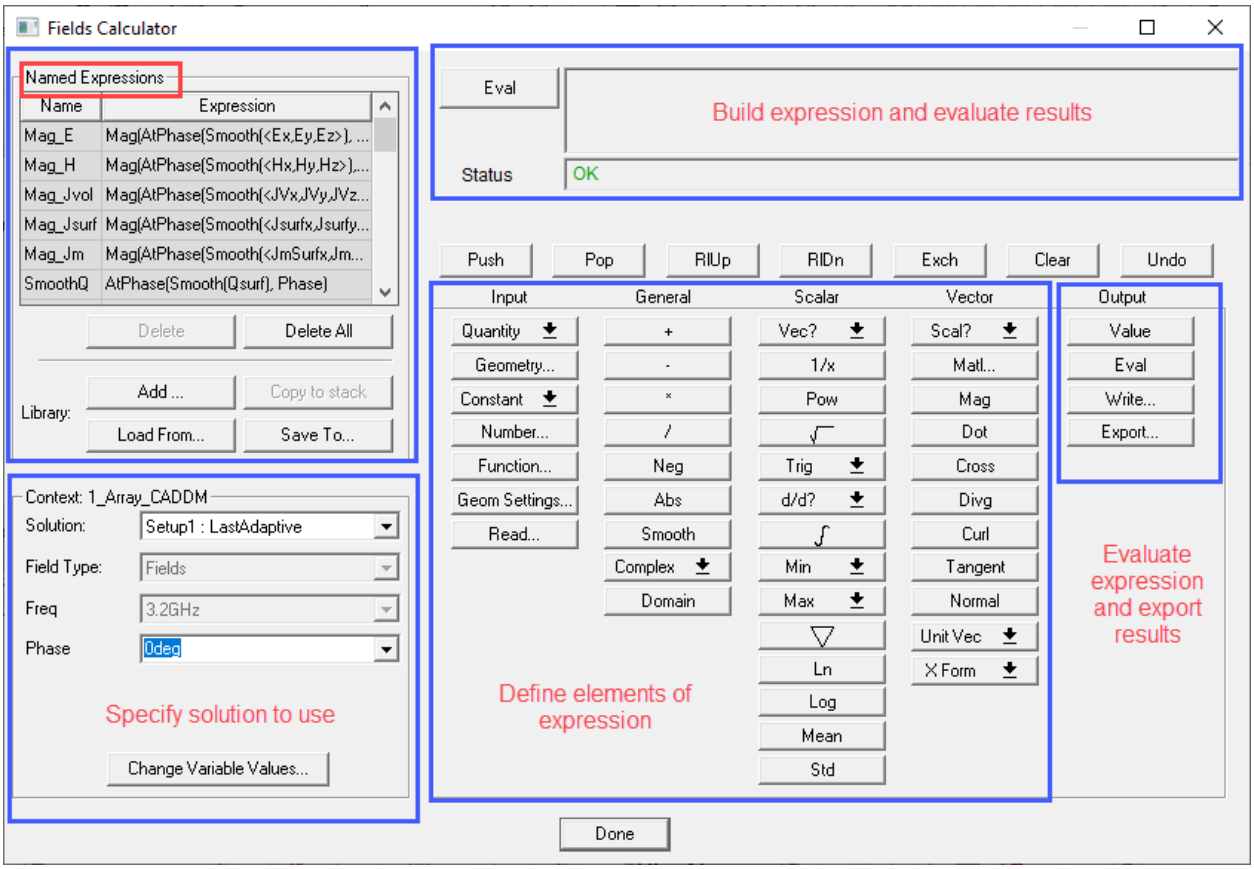
5. For AVI format export, specify the **Compression factor** (the default is 85) and one of the following **Compression types**:
   - INTEL Indeo
   - Cinepak
   - Microsoft Video 1
   - None
6. For GIF format export, specify the number of loops. The default "0" denotes infinite loops.
7. Click **OK** to close the *Animation Options* dialog box.

   The animation is exported to the file format you specified.

Using the Fields Calculator

The Fields Calculator is a very powerful tool for post processing. It is available for HFSS, Maxwell, Q3D, Icepak, and Mechanical solutions. While standard post -processed results (such as S-parameters, Y or Z matrix, animated field plots, and near and far field patterns) serve most simulation requirements, the Fields Calculator enables you to perform further computations using basic field quantities. The calculator computes derived quantities from the general electric, thermal, or displacement field solution; writes field quantities to files; locates maximum and minimum field values; and performs other operations on the field solution. Using this calculator, you can perform mathematical operations on all saved field data in the modeled geometry at a single frequency. The resulting quantities can be plotted, tabulated, or exported.

The Fields Calculator includes predefined expressions appropriate for each solver and lets you create and save additional named expressions.



At the top-left corner of the calculator is a list of **Named Expressions**, which are standard or user-defined field quantities that are accessible from outside of the calculator. They can be added, copied to stack, saved to, or loaded from a library file using the buttons right beneath the list.

At the lower-left corner of the calculator is the **Solution Context** section, in which you can select the desired solutions, field types, frequency, and phase for the current session.

The top right of the calculator contains the **Data Stack**, in which calculator entries are held in stack registers. The data type in the Data Stack is denoted by its prefix abbreviation.

Immediately beneath the stack is the row of **Stack Command** buttons that define some basic operations for the data in the Data Stack.

The bottom half of the calculator holds the columns containing the actual calculator buttons, organized into columns, classifying them by the type of operation and the type of data upon which the operation can be performed. These columns are headed Input, General, Scalar,

Vector, and Output. At the very bottom of the calculator is the button to exit, **Done**. The Fields Calculator interface is discussed in detail in subsequent sections.

The calculator does not perform the computations until a value is needed or is forced for a result. 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.

An algebraic expression is either a `namedExpression`, a `constant`, a `name`, an `operator`, two `expressions` separated by `binaryOperator`, or an `expression` enclosed in parenthesis.

An `operator` is an *opName* followed by `arguments`.

An `argument` is either empty parenthesis, an `expression` enclosed by parenthesis, or two `expressions` separated by a comma and enclosed by parenthesis.

A `binaryOperator` is one of the following literals: '+', '-', '*', '/'

A `name` is either a project, design, or intrinsic variable (scalar or vector), or a geometry.

A *constant* can be one of the following:

- scalar literal; for example, "1.5e-10"
- complex number; for example, "3.5-1.0i" or "<(3.5, -1.0)>"
- 3D vector; for example, "<1.0, 2.0, 3.0>"
- complex 3D vector; for example, "<1.0-2.0i, 2.0-3.0i, 3.0-4.0i>" or "<(1.0, -2.0), (2.0, -3.0) ,(3.0, -4.0)>"

The grammar for an algebraic expression is summarized below:

```
expression : namedExpression
   | constant
   | name
   | operator
   | expression binaryOperator expression
   | '(' expression ')'
operator : opName arguments
binaryOperator : + | - | * | /
arguments : '(' <empty> ')'
   | '(' expression ')'
   | '(' expression ',' expression ')'
```

Scripting Support

All calculator operations are fully scriptable. You can save the commands used in a Fields Calculator session by first clicking the **Tools** > **Record Script to File** menu and replay the same commands by clicking the **Tools** > **Run Script** menu in a later session.

Preliminary Considerations for Using the Fields Calculator

The following sections provides some consideration notes regarding use of the Post-Processor Field Calculator. Most of the statements below are fairly generalized, and may not apply to all 3D solver projects. When in doubt about the applicability of a particular consideration for a particular project, please feel free to contact your local HF Applications Engineer for further assistance.

Field Convergence and Accuracy

HFSS, Q3D, and Maxwell use finite element method (FEM) field solvers, which arrive upon a solution via adaptive meshing convergence. There are different algorithms available for determining where in each given model mesh adaptation is performed, but convergence is always evaluated by comparison of S-parameters (for driven solutions), changes in overall scattering energy (for incident wave problems) or resonant frequencies (for eigenmode solutions) from pass to pass. Since these quantities represent the results of the model as a whole, they tend to converge more rapidly than the field values. Each point in the modeled space can be said to have converged to some value. As a result, specific field quantities at each mesh point are likely to be less accurate than the overall S-parameter or Eigen frequency result of a project solution.

In order to obtain high accuracy results from calculations on field data, we recommend that you take extra precautions to assure that the model's field data is dependable. This might include:

- Running the project to a tighter than usual convergence value.
- Seeding or manually refining the mesh in the areas to be used for calculations.
- Running parametric variations to isolate sensitivity to modeling parameters (such as adaptation frequency or circular cross-section facetization).
- Specifying expressions for output convergence.

As long as the accuracy of specific field data points to be used has been assured, the results of the Field Calculator operations should provide valuable information for your electromagnetic design tasks.

Fast Sweep and Dispersive Models

If an HFSS solution includes a Fast Frequency Sweep, the Fields Post-Processor can be tuned to display field data at any point in the frequency band swept. The specific frequency selected for viewing need not even be a precise data point at which the S-parameters were calculated. While

field calculator operations may be performed at any frequency to which the Fields Post-Processor is set, fast sweep solution field data (away from the center frequency of the sweep) may not be as accurate for lossy and dispersive media, within the interior of solid-meshed finite conductors, etc. For higher accuracy under these conditions, field calculator operations should be performed on a full matrix solution completed at the desired frequency.

Inputs/Excitations

Use **HFSS** > **Fields** > **Edit Sources** to set the field excitation appropriate to the calculation to be performed. In some cases (e.g., FSS calculations) picking the right field solution set (incident, scattered, or total) is also paramount to obtaining the intended result.

Any field calculation which has not yet been completed (such that the calculator stack still shows some form of “text” string rather than a simple numerical value) is merely a placeholder. Altering the field data loaded in the Post-Processor (by altering port excitations, changing frequency, or picking a different solution set using **HFSS** > **Fields** > **Edit Sources**) will result in subsequent evaluation of the placeholder to the newly loaded data. To preserve a placeholder's association to an existing data set before altering the excitation to a different data set, you should export the register stack by using the **Write** button. You can bring the correctly associated quantity back into the stack by using **Read** after you have changed the field data set selection.

Units

All units in Driven HFSS field solutions are expressed in the MKS system, regardless of drawing units. Therefore E-mag is always in V/m, H-mag in A/m, etc. The exception is that when plotting along a geometry (e.g., along a line) the dimension along the X axis of the graph shows the position along the line in the drawing units, while the vertical (field quantity) axis will be in the MKS system.

Eigenmode Solutions

Field values in Eigenmode solutions are normalized to a peak value of 1.0, since there is no real excitation to which to scale the internal field results. If desired, you can scale the peak value to a user-selected number using the **HFSS > Fields > Edit Sources** menu.

Named Expressions in Q3D Extractor

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. Please see Field quantities in Q3D Extractor and Field quantities in 2D Extractor for more information.

Opening the Fields Calculator

To open the Fields Calculator, from the Desktop Menu bar do one of the following:

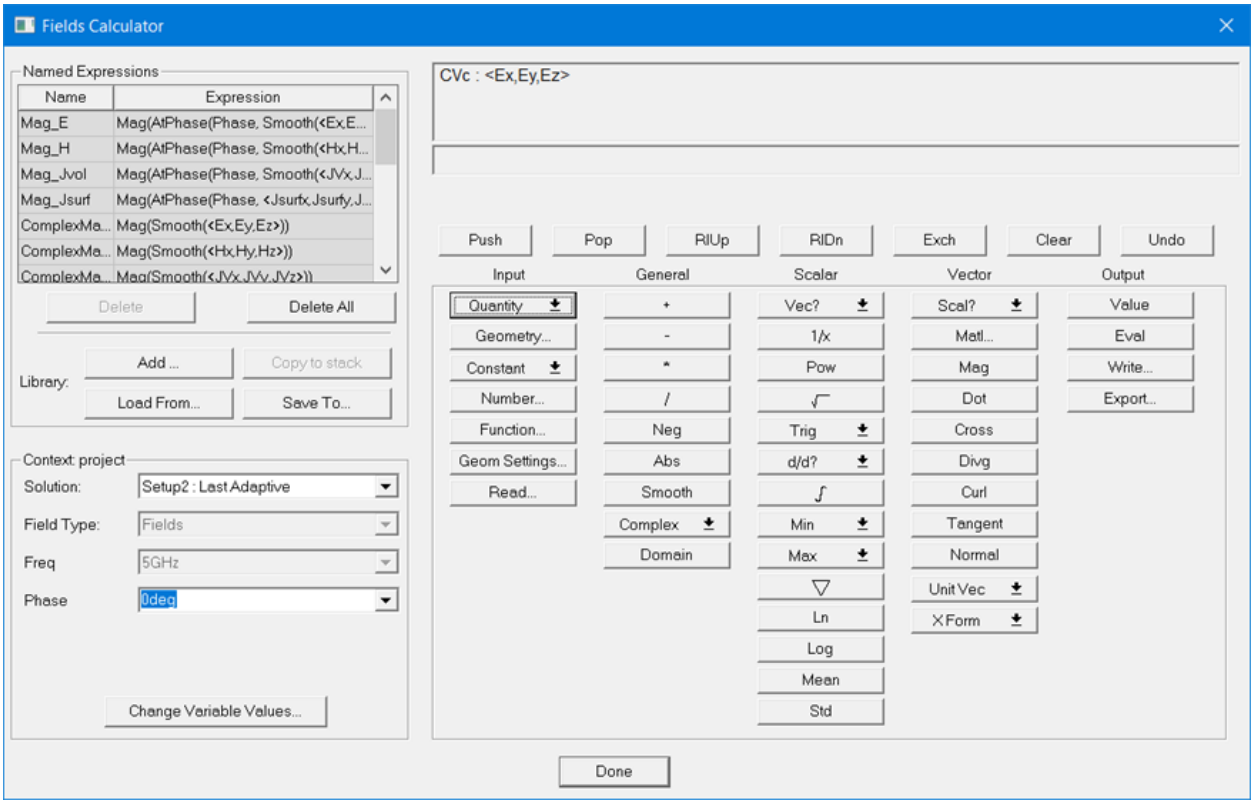
- Click **Q3D Extractor> Fields>** **Calculator**

  or

- Right-click **Field Overlays** in the Project Manager and choose **Calculator** from the shortcut menu.

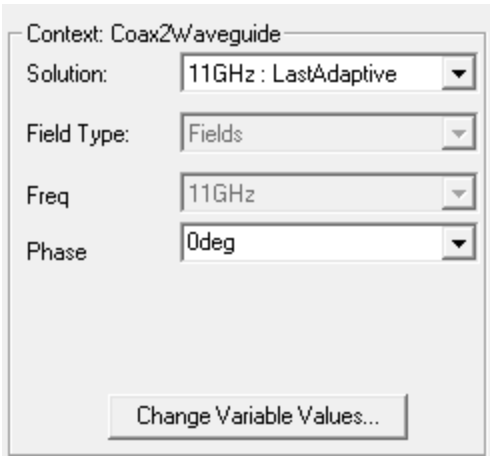
The *Fields Calculator* window appears.

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**, **Freq**, **Phase**, **IWavePhi** and **IWaveTheta**. The **IWavePhi** and **IWaveTheta** are available only for incident wave projects in which the wave is defined with spherical coordinates.

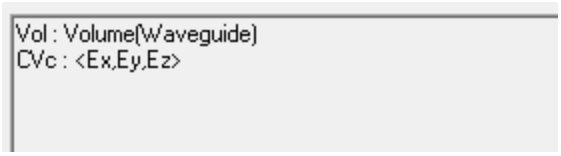


The Field Type here is not related to the edit sources. This is a general term among Ansys EM products (HFSS, Maxwell, Q3D, Icepak, and Mechanical). Some products have more than one field type for different solution types. If only one Field Type is available the drop-down menu is unavailable. For Hybrid design in HFSS R18, even if the design contains an IE Region, you can select J and Q input quantities. You no longer need to select between Fields and IE Surface Fields.

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 box to accept the selection.

The Calculator Stack

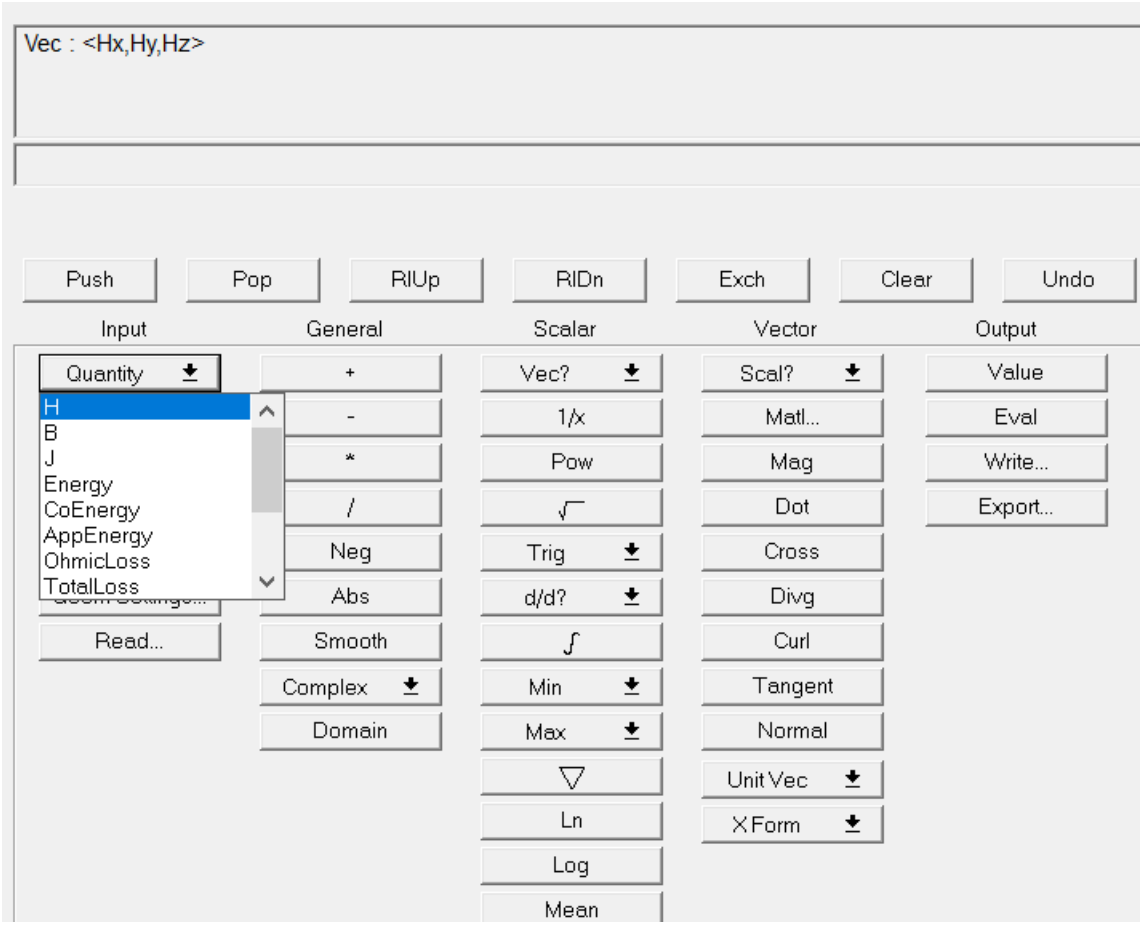
The calculator is made up of a stack of registers. Registers are displayed in the register display area at the center of the calculator window.



Each register can hold

- field quantities such as the H-field or E-field.
- 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.
- exported to a file, allowing you to superimpose saved solutions.

Stack Commands

Stack Commands influence the entries in the calculator stack and their position. These commands are for manipulating stack contents only, such as copy/paste, delete, or rearrange. The names for these commands match standard stack manipulation conventions. Many are self explanatory, and some can be used in pairs.

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.

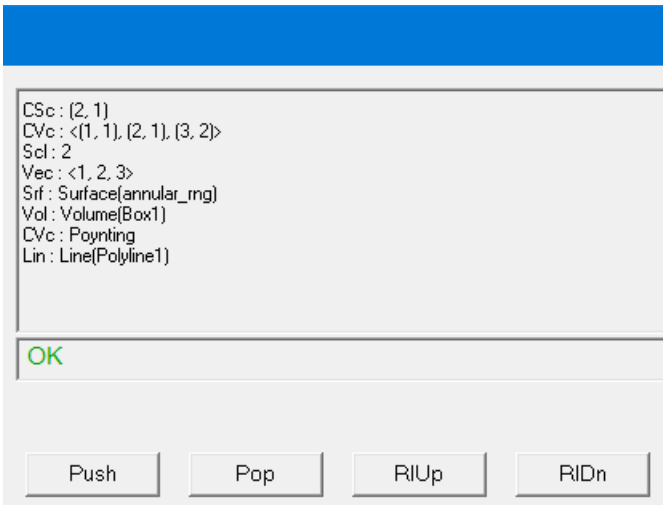
> **Note:**
>
> You cannot undo a simple operation such as loading a field quantity, constant, function, or geometry into the calculator. Instead, use the **Pop** or **Clear** commands to delete these items from the calculator stack.

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.

> **Note:** For Maxwell and HFSS users, make sure the Calculator is in **Show Stack** mode.

The calculator is capable of performing operations on a number of different data types. In many instances, a calculation requires certain type(s) of data to be present in the correct order in the stack register. Many operations result in a different data type than the inputs. In order to show you the type of data contained in each stack entry, calculator denotes its data type by a prefix indicator as shown in Stack Contents showing Data Type Indicators (at left), below.



The following table describes and defines each indicator, and provides guidance regarding operations which can convert data from one type to another.

Vec	Vector (non-complex) 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. Vectors are always evaluated in the coordinate system of the model. To convert a vector quantity to a scalar, use the Scal? drop-down menu from the Vector column. Suboptions ScalarX, ScalarY, and ScalarZ will take the appropriate scalar component of the vector data. Optionally, you can also Dot the vector with another vector to obtain the appropriate scalar result, or use the Tangent (return the tangential scalar component of) or Normal (return the normal scalar component of) operations to relate the vector quantity to a geometric data (Lin, Srf) stack entry. Convert to a Complex quantity using the CmplxR and CmplxI operations described in Scl, above.
Scl	Scalar quantities, which have a magnitude only. This is a simple numerical value. To convert a scalar to a vector quantity, use the

	Vec? drop-down menu in the Scalar column. The choices VecX, VecY, and VecZ convert the scalar data to vector data aligned with the X, Y, or Z unit vectors, respectively. You can also multiply the scalar quantity by a desired vector direction entered manually (Num drop-down in the Input column) or obtained using the Unit Vec button from the Vector column. To convert a scalar to a complex quantity, use either CmplxR (assign the scalar value as the real component of a complex quantity) or CmplxI (assign the scalar value as the imaginary component of a complex quantity), both found under the Cmplx drop-down in the General Column.
Cvc	Complex vector quantities. This is a numerical value with real and imaginary components. Convert to a vector quantity using the same techniques described for Scl, above. Convert to a scalar using Real (take the real component), Imag (take the imaginary component), CmplxMag, (take the magnitude of the complex number) or CmplxPhase (take the phase of the complex number), all within the Cmplx dropdown in the General column.
Csc	Complex scalar quantities. This is a numerical value with real and imaginary components. Convert to a vector quantity using the same techniques described for Scl, above. Convert to a scalar using Real (take the real component), Imag (take the imaginary component), CmplxMag, (take the magnitude of the complex number) or CmplxPhase (take the phase of the complex number), all within the Cmplx drop-down in the General column.
Pnt	Points.
Lin	Lines. Lines may be straight, curved, or “polylines” in three dimensional space. Lines may also be open (have two endpoints) or closed (ending vertex same as starting vertex).
Srf	Surfaces. Surfaces need not be planar, and may actually comprise a list of object faces (faces list) as well as planar slices through the entire model space (cutplanes).
Vol	Volumes. Volumes may include sets of discontinuous object volumes created as an Object.
SclLin	Scalar value on a line.
VecLine	Vector value on a line.
SclSrf	Scalar value on a surface.
VecSrf	Vector value on a surface.
List	These indicators may exist alone, representing geometric data only, or in combination with one of the categories above, indicating a type of data applied to the geometric entity in question. For example, the notation SclSrf identifies a stack entry containing Scalar data on a

	Surface geometry set. To select only the portion of a given data entry which exists along, on, or within a given geometry quantity, use the Value button in the Output column of the calculator. Other operations (e.g., integration, the Normal button) operate when a data quantity is in the second stack register and a geometric quantity is in the top stack register. Full descriptions of the register requirements for each individual command is available in the online help.

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.

Users must be cautious concerning what type of data the Fields Calculator is manipulating and whether or not it is compatible with the desired operation. For example, the integral operation is often misused. Note that the integral sign is in the *Scalar* column, implying that, to integrate complex number/quantities, the user will have to integrate the real and imaginary components separately. In other words, performing integration on complex number/quantities must be achieved by parts.

Enlarging the Register Display Area

If there are too many registers to fit into the display area, do one of the following:

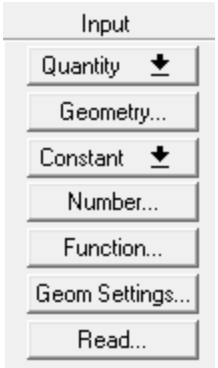
- Use the scroll bars to view the hidden registers.
- Enlarge the calculator window by dragging window's borders.

Units of Measure

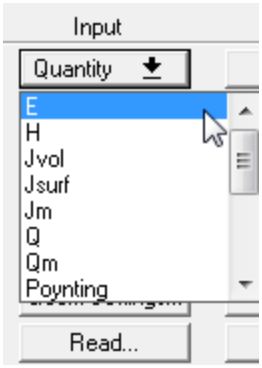
Unless you are prompted specifically for the unit of measure, all measurements should be assumed to be in SI base units, not model units.

Input Commands

The *Input* Column contains all the calculator functions which place new values into the stack, such as field data, constant, geometry data, coordinate system selection, or user-entered vector or complex numbers.



Field data (e.g., E-field, H-field, Poynting vector) for the current project solution is input from the **Quantity** drop-down menu selection.



The quantities specifically available from the calculator are dependent on the solution type and include the E-field, H-field, J-vol (volume current), Poynting vector, Displacement, Temperature, Heat Flux, and Surface and Volume Loss Densities. All quantities are Peak Phasors, and not RMS quantities, with the phase information captured in the real and imaginary components and the field orientation captured in the vector components. Although the Poynting vector is automatically calculated by the interface as 0.5, it will appear in the calculator stack as a Complex Vector quantity. The imaginary portion should however be zero or ignored.

Use the following commands to load data onto the top of the calculator stack:

Quantity	Basic field quantities, such as E and H, and simple derived quantities such as volume current.
Geometry	Geometries such as planes, points, polylines, 3D lines, face lists, 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	User-defined or intrinsic variables
Geom Settings	Number of equally spaced points used to integrate fields and other quantities on a line.
Read	previously saved calculator registers containing field quantities.
Output Vars	This button appears only for eigenmode problems. Freq is the only value there. Evaluation for Freq returns a complex value.

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.

Quantity Command

The **Quantity** command loads a specified field quantity into the top register of the calculator. Phasors in the calculator are peak phasors. The **Poynting** command in the calculator therefore implements the Poynting vector for peak phasors. Calculations which compute either average or instantaneous time domain quantities must adhere to the peak phasor conventions. Transient solution types display the transient quantities (with "_t" appended).

The available quantities are:

E	The electric field, E
E_t	Transient electric field, E.
H	The magnetic field, H
H_t	Transient magnetic field, H.
Jvol	The volume current density, J_{vol}
Jsurf	The surface current density, J_{surf}
Jsurf_t	Transient surface current density.
Poynting	The Poynting vector, defined as 0.5E x H*
Poynting_t	Transient Poynting vector.
LocalSAR	The local Specific Absorption Rate
AverageSAR	Peak spatial average SAR, dependent on the Average SAR Method chosen in the Specific Absorption Rate Setting.
Certification SAR	IEEE standard Specific Absorption Rate certification number. (See Modifying SAR Settings and Calculating the SAR.) To calculate certification SAR on a specific object (rather than the whole model) proceed as follows:

1. In the Calculator Input area, click the **Quantity** button and select **Certification SAR**.

   Certification SAR is displayed in the calculator stack.

2. In the Calculator, click the **Geometry** button.

   The *Geometry* dialog box displays with the *Volume* radio button selected and the available geometries listed.

3. Select the Geometry of interest.

   This enables the OK button.

4. Click the **OK** button.
5. This adds the selected Volume geometry to the calculator stack.
6. In the Calculator Output area, press the **Value** button.

   This prepares the calculation for the selected quantity and volume.

7. Press the **Eval** button to evaluate.

   Both the value and location will be shown on the calculator stack.

SurfaceLossDensity

This contains the surface impedance (if any) loss at every node in every triangle. This is calculated as:

$$Loss = 0.5 \times Real\left((\vec{E} \times \overrightarrow{normal}) \cdot \overrightarrow{(\vec{E} \times \overrightarrow{normal})} \div \overrightarrow{Impedance}\right)$$

To export a REG file containing the surface loss density, place the SurfaceLossDensity in the top register and use the **Write...** command, selecting Reg format.

VolumeLossDensity

The volume loss density *p* is calculated as:

$$p_v = \frac{1}{2} Re(E \cdot \tilde{J} + j\omega B \cdot \tilde{H}) = \frac{1}{2} Re(E \cdot \tilde{J} - curlE \cdot \tilde{H})$$

where *E* is the electric field, $\tilde{J}$ is the conjugate of the volumetric current density, *B* is the magnetic flux density, and $\tilde{H}$ is the conjugate of the magnetic field.

To export a Reg file containing the volume loss density, place the VolumeLossDensity into the top register, and use the

	Write....command, selecting the Reg format.
VolumeLossDensity_t	Transient volume loss density, calculated as: 0.5 * *Re*(*E* * *J*)
SurfaceForceDensity	Surface Forces exist when one side is conductor, but the other is not, or finite conductivity and layered impedance boundary. This is mainly for the purpose of mapping surface force density in HFSS to Workbench Mechanical.
Temp	Temperature.
Displacement	This is value is for use with Workbench when exploring stress feedback.

Geometry Command

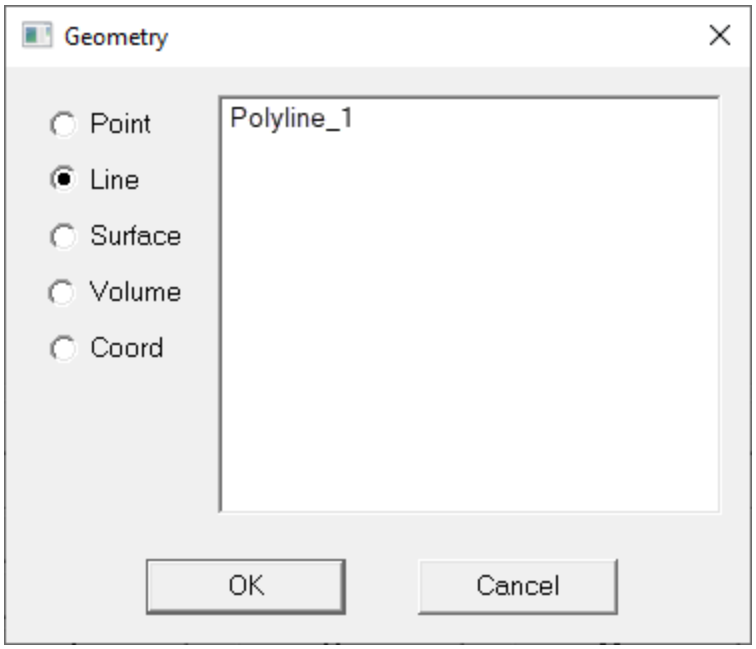
The Fields Calculator **Geometry** command opens a dialog box for selecting a geometry to load into the top register of the calculator.

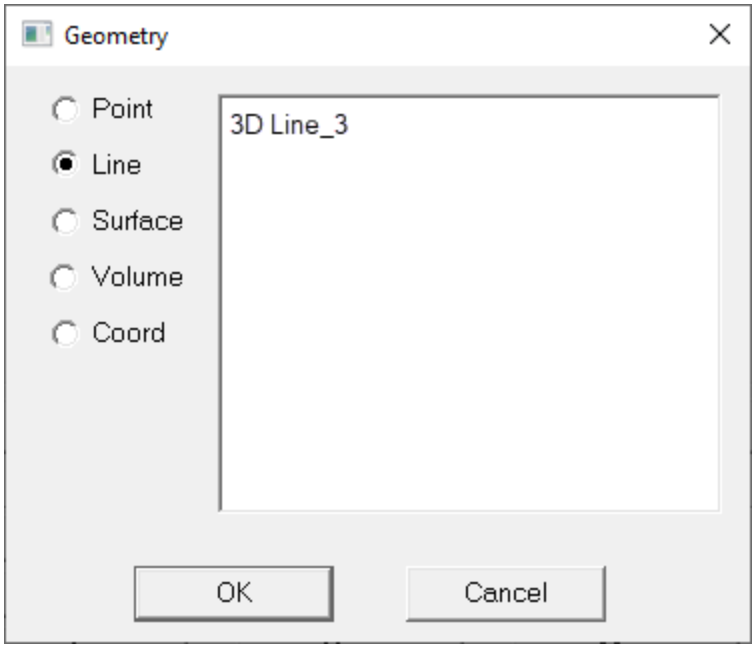
Do this to

- 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. Drawn points are listed in the History Tree, and in the calculator's *Geometry* dialog box when **Point** is selected.
Line	See Drawing a Line Segment or Drawing a 3D Line Segment. Drawn lines are listed in the **History Tree**, and in the calculator's *Geometry* dialog box when **Line** is selected. To set the number of points on a line, see Geom Settings.

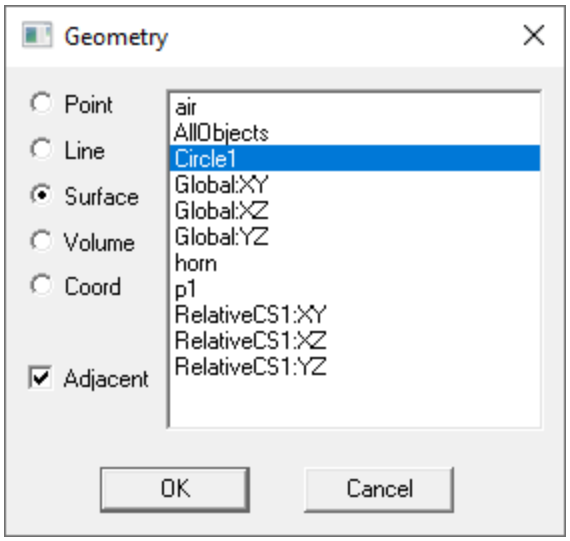




Surface

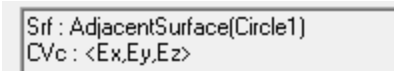
Creatable **Sheet** objects and Face Lists, (e.g., of radiation boundaries), are listed in the History Tree and in the calculator's **Geometry** dialog box when **Surface** is selected.

Due to the ambiguity of the normal vector of a sheet, the result may require a multiplication by (1) or (-1). If a Sheet object is selected from the Face list, the dialog also lists an **Adjacent** command that controls the side of the sheet on which the fields are calculated.



This check box is only available when users select a sheet object for surface geometry

- Allow operation on the adjacent side surface geometry just like regular surface geometry.
- All calculation on this geometry will be done using data from the adjacent side of the surface.
- Support scripting, saving and loading expression with adjacent side surface geometry.



Volume — 3D objects, Regions, and Object Lists (of 3D objects including AllObjects) are available in the calculator's *Geometry* dialog box when **Volume** is selected.

Coord — Coordinate systems are available in the calculator's *Geometry* dialog box when **Coord** is selected.

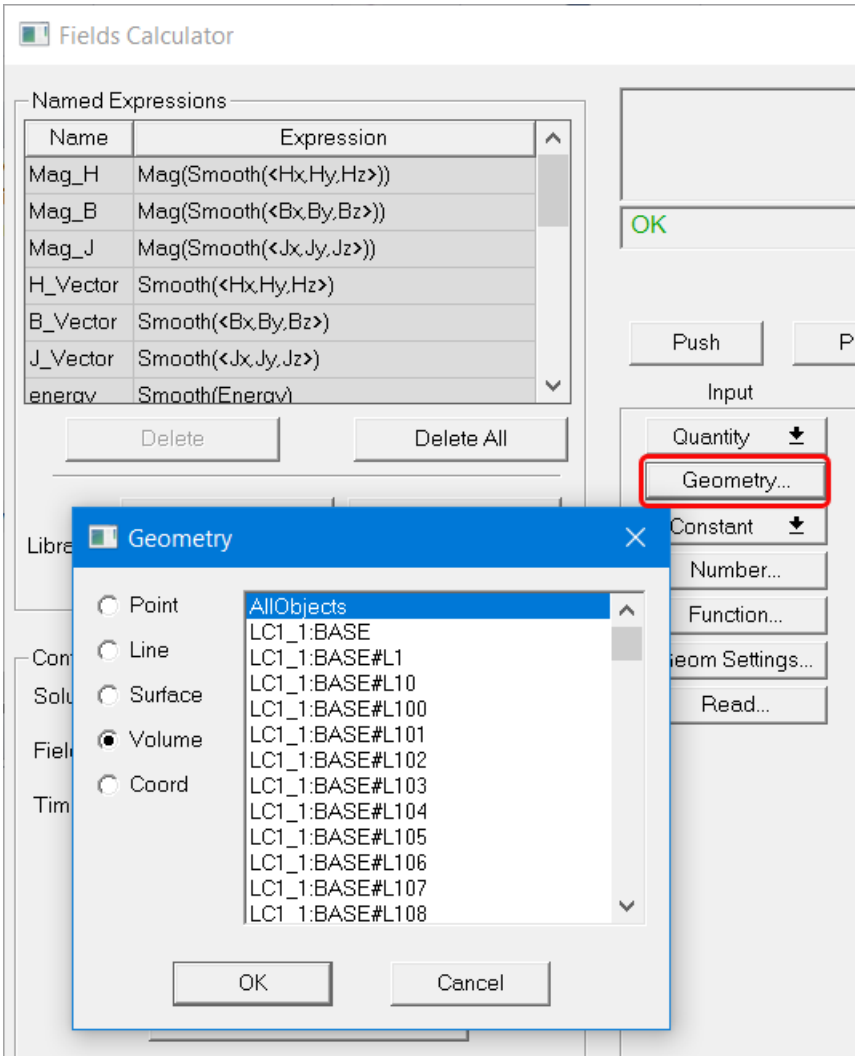
To load a geometry into the calculator:

1. In the Fields Calculator, click **Geometry**.

   The **Geometry** dialog box appears.

2. Select a geometry type.

   A list of all applicable geometries appears. If a design includes Layout Components, the Geometry list shows the available component geometries as well.



In the modeler window, these Layout Component objects are drawn as visualizations without actual geometry. However, they can be used to calculate field quantities as shown below.

These geometry operators can be used to perform computations just like one constructed

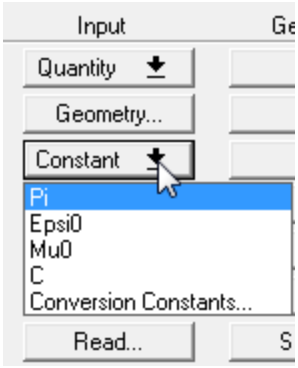
from regular 3D geometry. These additional objects are generated only after reading a mesh, so they won't show up if there is no solution.

3. Click the geometry.
4. Click **OK** to load the geometry.

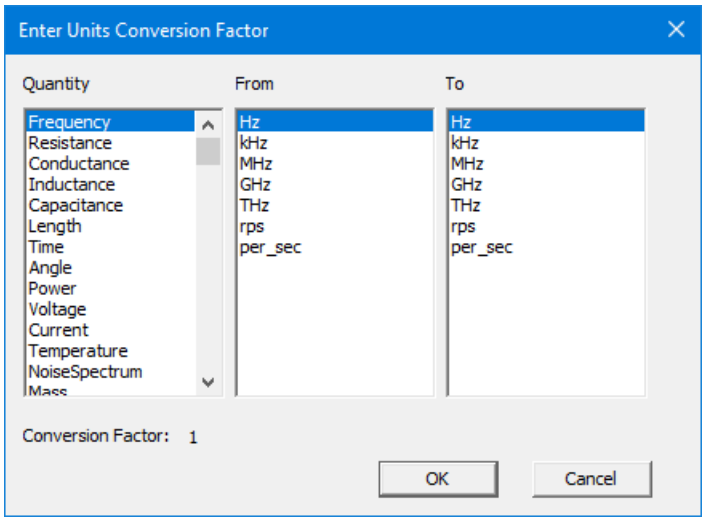
Note: Consider a box (Box2) that is completely enclosed in a bigger box (Box1), so that no faces of Box2 are touching any faces of Box1. Box2 is actually implicitly subtracted from Box1 as is done in our solvers. So Box1 is used as if Box2 were already subtracted from Box1. Volume(Box1) is Box1 excluding Box2, and Surface (Box1) contains faces from both Box1 and Box2.

Constant Command

The **Constant** command loads one of the following four predefined constants or a selected conversion constant into the top register of the calculator:



Pi	π
Epsi0	The permittivity of free space, $\varepsilon_0 = 8.85418782 \times 10^{-12}$ C^2/Nm^2
Mu0	The permeability of free space, $\mu_0 = 4\pi \times 10^{-7}$ Wb/Am
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 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 **Conversion Factor**.

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** text 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, 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.

Function Command

The **Function** command enters a predefined scalar or vector function into the top register of the calculator:

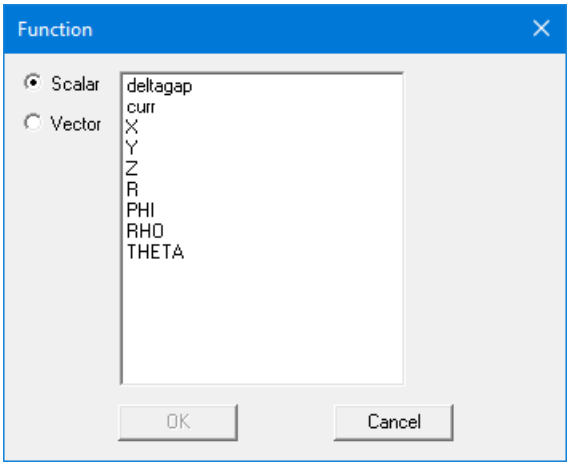
Scalar

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.

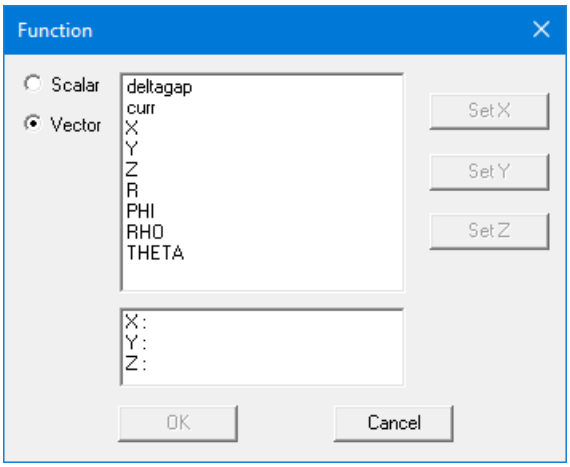
Vector

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,** and **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.

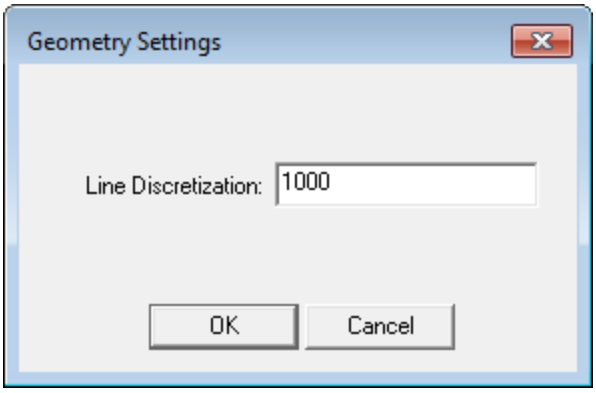
Geom Settings Command

Clicking the **Geom Settings** button opens the *Geometry Settings* dialog box. This 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** text box and click **OK**.

Read Command

This **Read** 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.

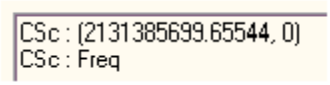
To read in a register:

1. Click **Read**.
2. Use the file browser to specify the register's file name and directory path. A .reg extension is automatically assumed for register files.
3. Click **OK**.

   The contents of the file are copied to the top register in the stack.

Output Vars [Input for Eigenmode problems]

This Output Vars button appears in the **Inputs** column of the Fields calculator only for Eigenmode problems. Freq is the only value listed. After you push Freq to the stack, you can click **Eval** to return a complex value.



General Commands

The General Column contains calculator operations which can be performed on many different data types (e.g., vector, scalar, complex). With the exception of the Complex menu, all are distinct functions. Most are self-explanatory, with the exception of Smooth which performs some data “smoothing” or statistical averaging on the top stack entry.

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).

> **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.

*** (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 by taking a weighted average from all of a node's neighboring elements. The weights are based on angles, so elements with larger angles provide larger contributions. 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 includes a drop-down menu that 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 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	Lets you specify the phase angle, θ, at which an 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.
CmplxPeak	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.

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 Q3D Extractor 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).

Takes the mean of the quantity in the top register.

Scalar Commands

The Scalar Column contains calculator operations which can only be performed on scalar stack entries.

Drop-down menus in this column include Vec? (convert scalar to vector), Trig (trigonometric, containing sin, cos, etc. functions), d/d? (derivative with respect to...), Max and Min (self-explanatory), Gradient, ln (natural logarithm base e), log (logarithm base 10), mean, and Std (standard deviation). Note that the calculator's Integrate function is located in the Scalar column. The implication is that integration can only be performed on scalar quantities. To perform integration upon complex quantities, you must perform the integration separately on real and imaginary subcomponents.

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	Performs a selected trigonometric operation on the value in the top register of the calculator stack (Sin, Cos, Tan, ASin, ACos, ATan). Angles are in radians. Additionally, the ATan2 operation is available, which requires two arguments. This operator calculates the arc tangent of the following ratio: The second register in the stack divided by the top register.

	Example: If the top register value is 1.5, and the second register value is 2, the ATan2 function returns ATan(2/1.5) = 0.5880 radians. All other Trig functions operate on a single argument (top register only).
d/d?	Takes the partial derivative of the quantity in the top register.
∫? (Integral)	Takes the integral of a scalar quantity over a volume, surface, or line. To perform integration upon complex quantities, you must perform the integration separately on real and imaginary subcomponents.
Min	Computes the minimum of a scalar field quantity on a line, surface, or volume relative to your choice of Value or Position.
Max	Computes the maximum of a scalar field quantity on a line, surface, or volume relative to your choice of Value or Position.
∇ (Gradient)	Takes the gradient of the scalar quantity in the top register.
ln	Takes the natural logarithm (base e) of the scalar quantity in the top register.
log	Takes the logarithm (base 10) of the scalar quantity in the top register.
Mean	Takes the mean of the quantity in the top register.
Std	Takes the standard deviation of the quantity in the top register.

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.

1/x (Inverse) Command

Takes the inverse of the scalar quantity in the top register.

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.

(Square Root) Command

Takes the square root of the quantity in the top register.

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.

(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 any geometry selection onto the calculator stack. Q3D Extractor 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. Choose the ∫ command to integrate the scalar quantity over the geometry.

To find the numerical results of an integration, use the **Eval** command.

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. • 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.

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. • 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 find the real part of the quantity using the **Cmplx/Real** or **Cmplx/AtPhase** commands.
2. Load a point, line, or volume into the calculator using one of the **Geometry** commands.

3. Do one of the following:
    - Choose **Max/Value** to compute the maximum field value on the geometry.
    - Choose **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.

∇ (Gradient) Command

Takes the gradient of the scalar quantity in the top register.

Ln Command

Takes the natural logarithm (base e) of the scalar quantity in the top register.

Log Command

Takes the logarithm (base 10) of the scalar quantity in the top register.

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.

Matl Command

Operates on the vector field quantity in the top register of the Fields Calculator based on a material property. At each tetrahedron, the field quantity is operated on by the value of the selected material property — taking the different material attributes of each object into account.

To operate on a vector quantity by a material property:

1. Click **Matl**.

    The *Material Operation* window appears.

2. Select a material property. Available properties are

Permittivity (epsi)	The permittivity, ε.
Permeability (mu)	The permeability, μ.
Conductivity	The conductivity, σ.
Omega (w)	The angular frequency, ω. The angular frequency is equal to $2\pi f$, where f is the frequency at which the solution was generated.
MassDensity	This is 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 box.

3. For Permittivity, Permeability, Conductivity or Omega, select an operation — **Multiply** or **Divide**. Selecting MassDensity disables these operations. The MassDensity scalar can be used like any other named expression.
4. Choose **OK** to operate on 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.

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 non-negative scalar. In previous software versions, this command returned a complex scalar.

Dot Command

Takes the dot product of the vector quantities in the top two registers.

Cross Command

Takes the cross product of the vector quantities in the top two registers.

Divg Command

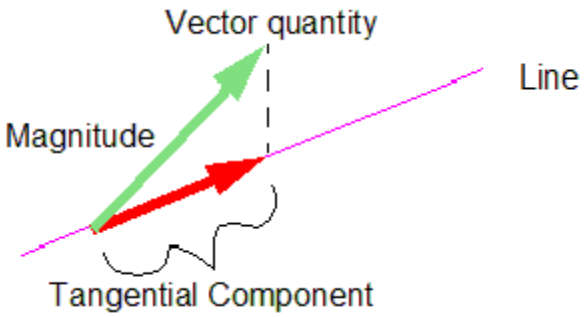
Takes the divergence of the vector quantity in the top register.

Curl Command

Takes the curl of the vector quantity in the top register.

Tangent Command

Computes the tangential component of a vector quantity along a line.



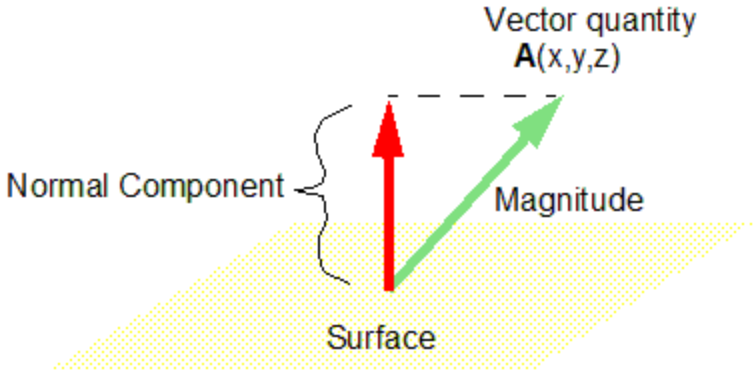
To take the tangent of a vector:

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**.

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:

$$Normal = A\left(x,y,z\right) \cdot \hat{n}$$



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 Q3D Extractor, the fields calculator uses the surface normal of the face of the 3D object that is coincident with the sheet.

Unit Vec Command

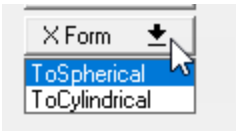
Computes the normal or tangent unit vector. The unit vector is a "wild card" entry. The context is specified at the time of plotting, integrating, or report generation.

Select from the following:

Tangent	Computes the unit vector tangent to the line specified at the time of plotting, integrating, or report generation based on the context.
Normal	Computes the unit vector normal to the surface specified at the time of plotting, integrating, or report generation based on the context.
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.

X Form

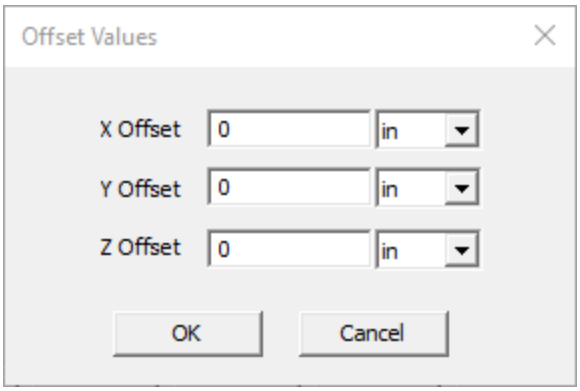
Fields are defined in the Cartesian coordinate system. In many applications it is much more convenient to use a cylindrical or spherical coordinate system – for example, when the object is of cylindrical or spherical shape. The **XForm** command allows the user to transform vectors into either the cylindrical or spherical coordinate system. Selecting **XForm** displays a selection menu for the coordinate system (**ToCylindrical** or **ToSpherical**):



ToSpherical – Transforms the vector in the top register of the calculator stack into the spherical coordinate system.

ToCylindrical – Transforms the vector in the top register of the calculator stack into the cylindrical coordinate system.

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>)

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.

Std Command

Takes the standard deviation of the quantity in the top register.

Vector Commands

The Vector Column contains calculator operations which can only be performed on vector stack entries. Drop-down menus in this column are Scal? (convert vector to scalar) and Unit Vec (create unit vector). Standard vector algebra operations (Dot, Cross, etc.) are also present.

Use these commands to perform operations on vector quantities.

Scal?	Replaces the vector in the top register with a scalar quantity whose value is a component of the vector, your choice of ScalarX, Scalar Y, or ScalarZ.
Matl	Multiplies or divides the vector field quantity in the top register by a material property, or if you select MassDensity as the material property, produces a scalar that operates like a named variable.
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. The drop-down lets you select from Tangent, Normal, or CoordSys[X, or Y, or Z]
X Form	Transforms to your choice of Cylindrical or Spherical.

Output Commands

The Output Column contains those calculator operations that result in final data outputs from calculations. The Eval button obtains final numerical results from the last stack placeholder (such as integrations).

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 on a geometry.
Eval command	Numerically evaluates and displays the results of calculator operations from the last stack placeholder, such as integrations.
Write command	Saves the contents of the top register to a disk file for future re-use.
Export command	Saves field quantities in a format that can be read by other modeling or post-processing software packages.

To plot/tabulate the extracted data using create report or create field overlays, you should use a named expression.

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.

The **Value** command applies the value of the next-to-top stack entry (a field quantity) on the geometry entry at the top stack entry. The field quantity entry may be scalars, real or complex vectors. The geometry entry may be of any type (point, surface, line, or volume).

For example, if the top stack entry in the Field Calculator is a surface and the second stack register is for the real part of the E Vector ("Real(<Ex, Ey, Ez>)"), the resulting quantity will be an expression for the real part of the E vector on the surface (a VecSrf quantity). Again, the result of this command is still a placeholder expression, not the final numerical values.

Strictly speaking, the Value command provides access to the field quantity, not data outputting. It simply provides a handle to the (numerical) values of a field quantity on a given geometry, and thus the outcome of the command is not numerical values themselves. If the numerical value of a field quantity is desired, you will need to follow it up with one of the three other output commands, or define a named expression for the quantity and plot it with the Reporter.

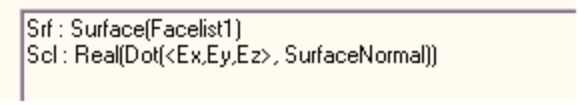
To find the value of a field quantity at a point:

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**.

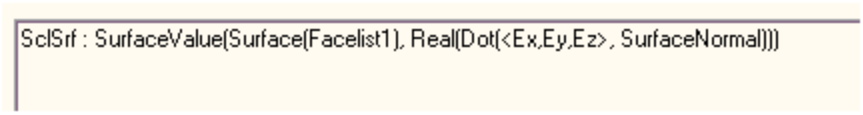
To view the numerical results of this operation, use the **Eval** command.

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.

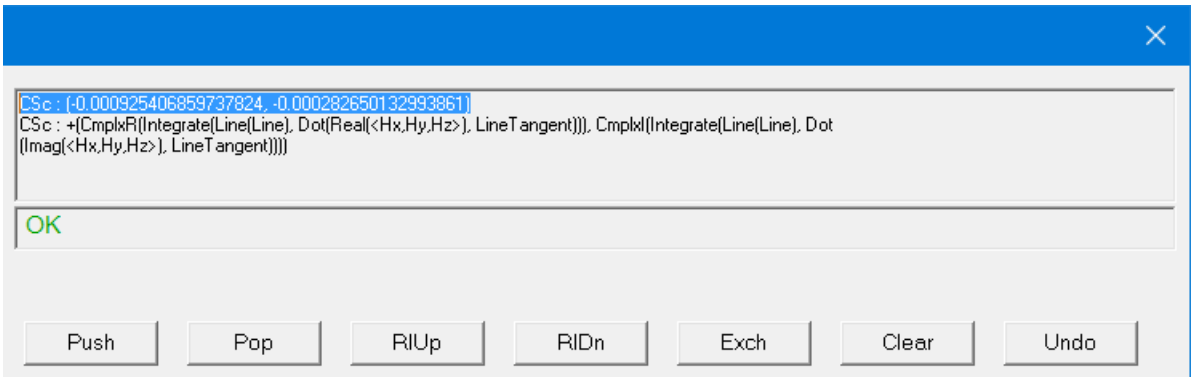
Now you can perform suitable operations such as Write, Integrate, etc. For PointValue you can also do **Eval**.

Eval Command

The **Eval** command finalizes computations and converts the text-string indicating the computation being performed to an actual numerical value. 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. For this command to work properly, the stack entry must be reduced to a single-value entry which can be a scalar, complex, or vector (real or complex). 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. Units of the numerical value are not provided.

For a calculator expression that evaluates to a single value, the **Eval** command offers a quicker way to obtaining the final numerical result than defining a named expression and then trying to plot/tabulate it in the reporter. But for many expressions that don't evaluate to single values, such as the wave impedance along a line mentioned earlier, **Eval** command does not apply. As such, use of named expression and the Report Editor will be the only option.

Evaluate the expression for voltage along a line shows the definition of an expression for the voltage between two conductors by integrating the electric field along a line using the recipe given for Calculating the Current along a Wire or Trace. Then, by clicking the Evaluate command in the Output column, we obtain the numerical value for the voltage as shown in the top of the stack. It can then be either written down or copied/pasted to other programs.



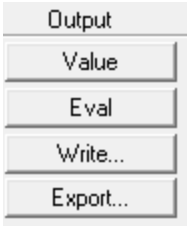
$$I = \oint H \cdot dl$$

Because **H** and **I** are complex quantities, you will 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 **Cmplx/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 ∫ 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 **Cmplx/Imag** command) to obtain the imaginary part of **I**.

Write Command

If the calculator operations performed have obtained a stack entry that is intended for use in still other calculator operations, the stack entry can be saved for this purpose by using the **Write** button in the calculator Output column.



Note that in this case no name will be requested for the expression; only a save filename location will be required. This function will not work for field values derived upon a specific geometric quantity (those containing either Lin, Srf, or Vol in the stack data type indicator) as the calculator cannot know that these geometric quantities exist in identical forms in other postprocessing sessions.

When you try to read a file written with the **Write** command by clicking the **Read** button in the Input column, you must ensure that the mesh used for generating the current solution is the same as the one with which the saved field data is generated. Since the two results must be generated on the same mesh, not only the geometries of the designs on which the **Write/Read** commands were applied must be the same, they must also have gone through the same adaptive meshing process. In practice, this limits the use of Write/Read command to solutions for different frequency sweeps of the same analysis (adaptive) setup of the same design.

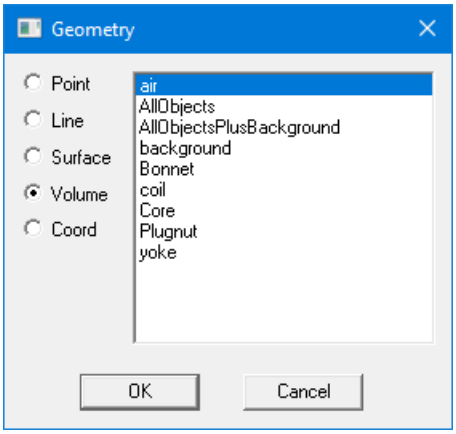
This command saves the contents of the top register to a disk file. Use this command to:

- Save registers for use during a later postprocessing session.
- Save a field quantity for use when postprocessing 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 data with a constrained quantity (such as jsurf), you see a dialog box that gives a choice of constraining geometries. For example:



3. Select the geometry of interest, and select **OK**.

   This displays a file browser.

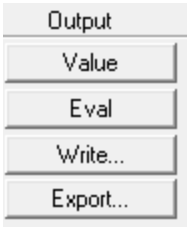
4. Use the file browser to specify the register's file name and directory path. A .reg extension is automatically assigned to register files and a .fld extension is assigned to field files. You can choose to save both .reg and .fld files, or either one.
5. Click **OK**.

   The contents of the register are saved to the file you specified.

Export Command

To output a field quantity or calculation result for use by some third-party post-processors, use the **Export** button in the calculator Output column. You can map the field quantity to either a customized grid of points specified by a points file or a three-dimensional Cartesian, Cylindrical, or Spherical grid specified interactively through a dialog box. In the latter case, you must specify the dimensions and spacing of the grid in the coordinate system with units. For Cylindrical and Spherical coordinate systems, you can also specify an offset from the origin.

The **Export** command opens the **Export Solution** dialog box, 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 Cartesian, cylindrical, or spherical coordinates, with units that you specify. The initial units are taken from the model.

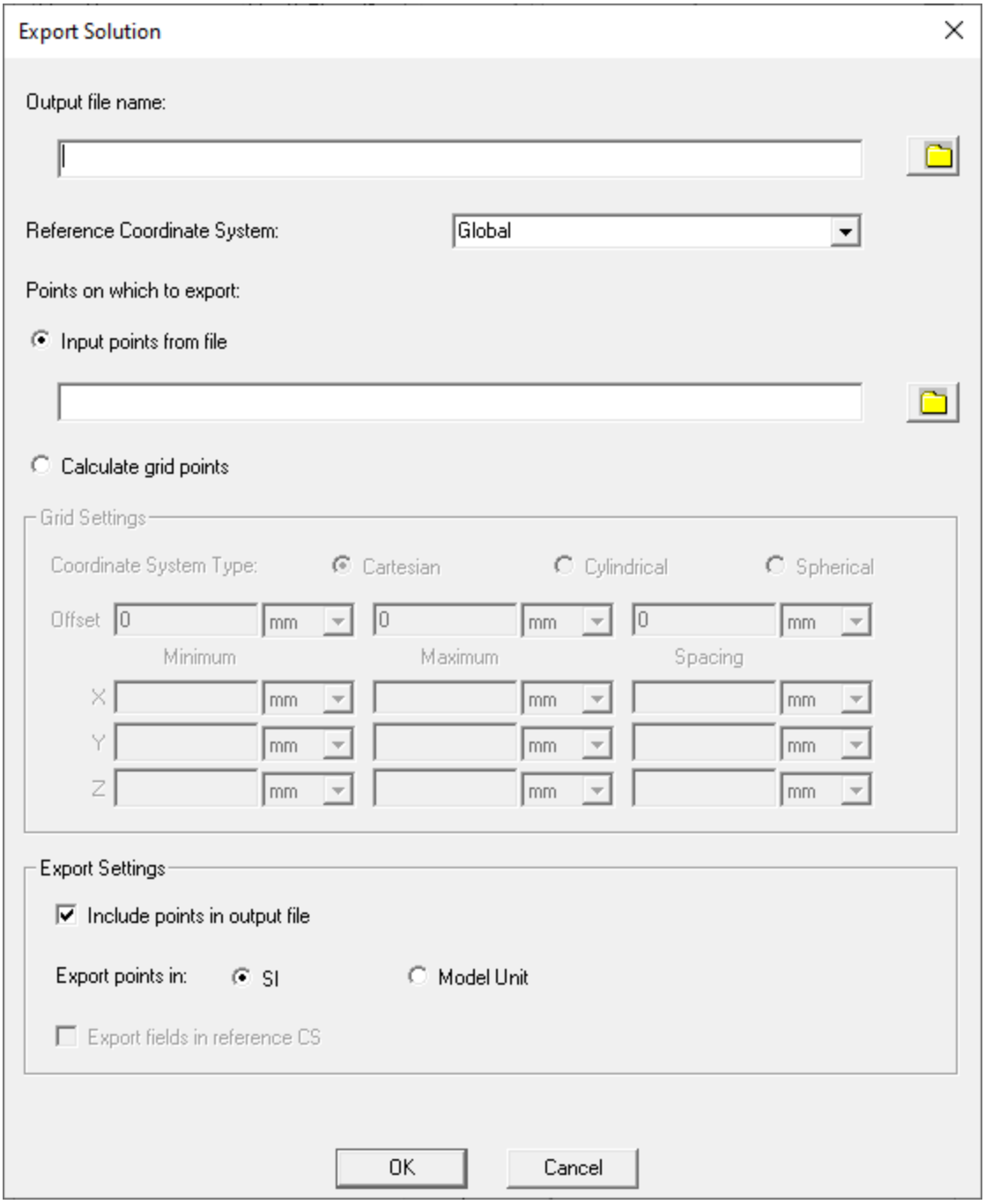
> **Note:**
>
> Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

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 box.



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. The **.fld** extension is automatically assigned to this file.
5. If the design contains multiple coordinate systems, a drop-down menu lets you select the coordinate system to use. This can be helpful in evaluating the fields output when relative

coordinate systems used and unitized in the design.

6. Click either the **Input grid points from file** radio button if you have a created a .pts file containing the grid points, or click the **Calculate grid points** radio 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:

     The **.pts** file should contain the units to use for the export as shown in this file stub:

     Unit=mm

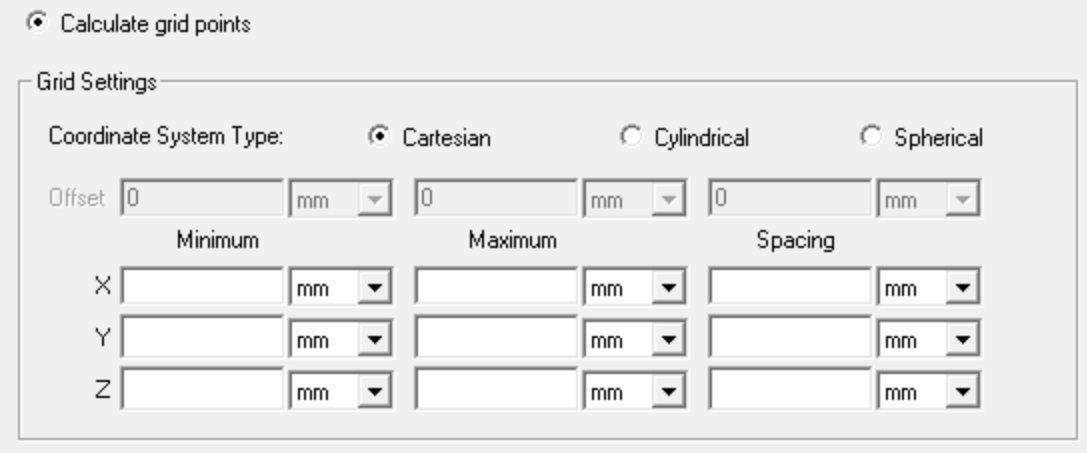
     -5.5 -5.5 -5.21475

     -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.

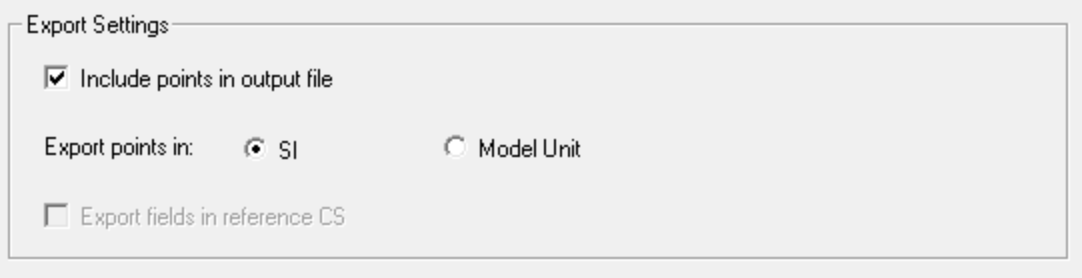
Cylindrical: for each dimension Rho, Phi, and Z, enter the Minimum, Maximum, and grid point Spacing. You can also specify an origin of Offset.

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 will be Cartesian. The default offset will be all zeroes. 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.

7. For larger files, for the Export Settings, 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.
8. You can also specify SI or Model Units. Model Units can be helpful to evaluate the fields output when relative coordinate systems are unitized in the design.



9. If field to be exported has type vector or complex vector, the 'Export Field in reference CS' checkbox is enabled and you can specify that field to be exported in reference CS. It can be helpful to evaluate the fields output when rotated coordinate systems are unitized in the design
10. Click **OK** to export the file.

    The field quantity is mapped to the grid and saved to the file you specified (**.fld** extension).

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 **Q3D Extractor**, and then click **Results** > **Create Report**.

   The *Create Report* dialog box appears.

2. In the **Target Design** drop-down menu, click the design containing the solution data you want to plot.
3. Select **Fields** from the **Report Type** drop-down menu.
4. In the **Display Type** drop-down menu, select the type of report you want to create.
5. Click **OK**.

   The *Traces* dialog box appears.

6. In the **Solution** drop-down menu, 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. 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 **Q3D Extractor** or **2D Extractor** and then click **Results> Create Report**.

   The *Create Report* dialog box appears.

2. In the **Target Design** drop-down menu, click the design containing the solution data you want to plot.
3. Select **Fields** from the **Report Type** drop-down menu.
4. In the **Display Type** drop-down menu, select the type of report you want to create.

5. Click **OK**.

   The *Traces* dialog box appears.

6. In the **Solution** drop-down menu, click the solution containing the data you want to plot.
7. Select the geometry you want to plot from the **Geometry** drop-down menu.
8. Add one or more traces to include in the report.

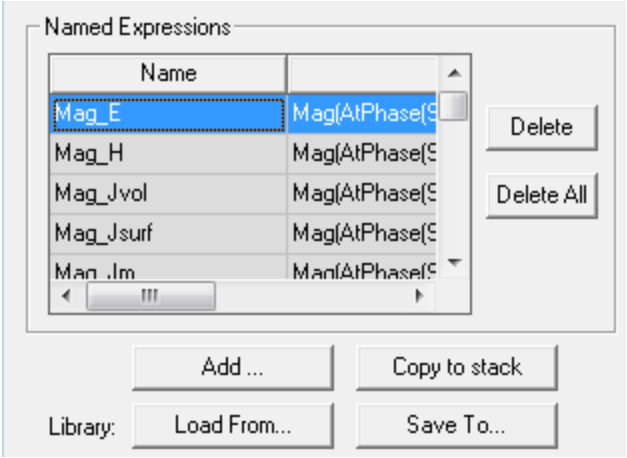
   > **Note:**
   >
   > If you select a polyline for the **Geometry** and a **Category** that is a named expression based on the line, then the **Quantity** list only lists corresponding expressions. The list will not contain scalar values when a line is selected as the geometry.

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, 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. Once you have created a report, additional options become available on the **Results** submenu.

Calculating Derived Field Quantities Using Named Expressions

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 then clicking the Add button. This lets you add to the Named expression library.



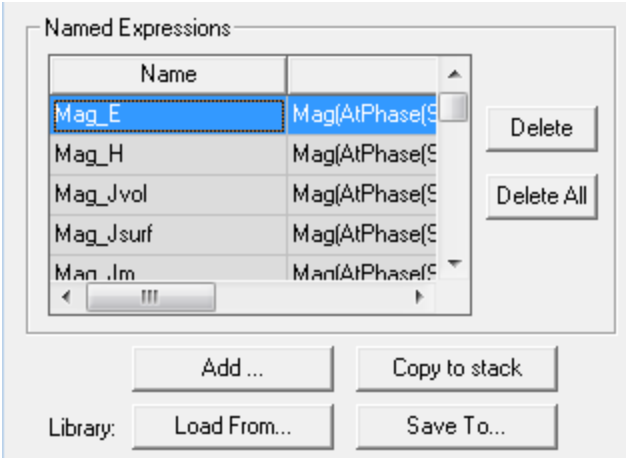
Click on a named expression to select it. When a named expression has been selected, the **Copy to Stack** button is activated. Click **Copy to Stack** to push the expression on the top of the stack.

When a Q3D Extractor design is open and a Solution Setup has been performed, the following predefined named expressions are available.

Expression Name	Expression Definition
Abs_Q	CmplxMag(Smooth(Q))
ComplexMag_E	Mag(Smooth(<Ex,Ey,Ez>))
ComplexMag_H	Mag(Smooth(<Hx,Hy,Hz>))
ComplexMag_JdcThinCond	Mag(Smooth (<dcfSurfJx,dcfSurfJy,dcfSurfJz>))
ComplexMag_SurfaceJac	Mag(Smooth(<ACRL_SurfaceJacx,ACRL_ SurfaceJacy,ACRL_SurfaceJacz>))
ComplexMag_SurfaceJdc	Mag(Smooth(<Jx,Jy,Jz>))
ComplexMag_VolumeJdc	Mag(Smooth(<dcvJx,dcvJy,dcvJz>))
Harmonic_Loss_Density	Smooth(HarmonicLoss)
JdcThinCond	AtPhase(Phase, Smooth (<dcfSurfJx,dcfSurfJy,dcfSurfJz>))
Mag_E	Mag(AtPhase(Phase, Smooth(<Ex,Ey,Ez>)))
Mag_H	Mag(AtPhase(Phase, Smooth (<Hx,Hy,Hz>)))
Mag_JdcThinCond	Mag(AtPhase(Phase, Smooth (<dcfSurfJx,dcfSurfJy,dcfSurfJz>)))
Mag_SurfaceJac	Mag(AtPhase(Phase, Smooth (<SurfaceJacx,SurfaceJacy,SurfaceJacz>)))
Mag_SurfaceJdc	Mag(AtPhase(Phase, Smooth(<Jx,Jy,Jz>)))
Mag_VolumeJdc	Mag(AtPhase(Phase, Smooth (<dcvJx,dcvyJ,dcvJz>)))
Phidc	AtPhase(Phase, Smooth(dcvPhi))
PhidcThinCond	AtPhase(Phase, Smooth(dcfPhi))
SmoothQ	AtPhase(Phase, Smooth(Q))
Surface_Loss_DensityAc	Smooth(SurfaceLossDensityAc)
SurfaceJac	AtPhase(Phase,Smooth (<SurfaceJacx,SurfaceJacy,SurfaceJacz>))
SurfaceJdc	AtPhase(Phase, Smooth(<Jx,Jy,Jz>))
Temperature_Ac_Surface	Smooth(TemperatureAcSurf)
Temperature_Dc_Volume	Smooth(TemperatureDcVol)
Vector_H	AtPhase(Phase, Smooth(<Hx,Hy,Hz>))
Vector_E	AtPhase(Phase, Smooth(<Ex,Ey,Ez>))
Volume_Loss_DensityDc	Smooth(dcvLossDensity)
VolumeJdc	AtPhase(Phase, Smooth (<dcvJz,dcvJy,dcvJz>))

Named Expression Library

The named expression library in the Fields Calculator provides a way to conveniently calculate frequently used quantities. Some standard field quantities have been predefined and are always accessible. These field quantities have a gray background color and cannot be deleted. You can assign a name to the top entry of Data Stack and add it in the Named Expression list. The library comes with several predefined expressions. Once these new expressions are added/loaded, they are available through Q3D Extractor's standard post-process capabilities. For example, they can be copied back to the Data Stack in calculator by **Copy to Stack** for further operations, or be plotted using Report Editor and Field Overlay.



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 **Save To** on the Fields Calculator.

   The *Select Expressions for Saving* dialog box displays.

2. If any new named expressions exist, you can select one or more to save to a file.
3. Specify 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.

Exiting the Fields Calculator

Click **Done** to exit the Fields Calculator.

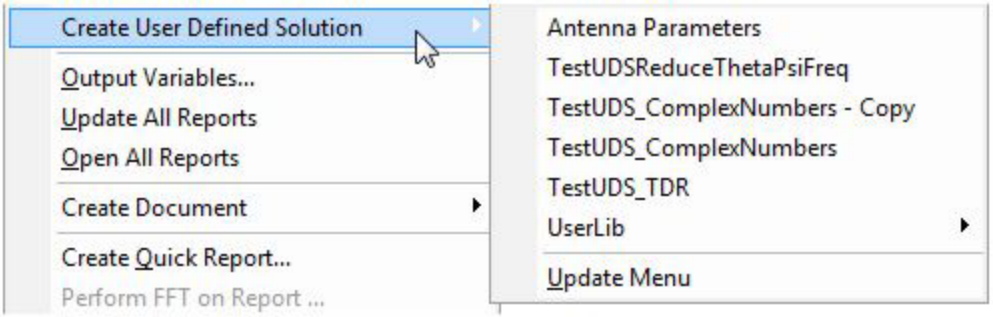
User Defined Outputs (UDOs)

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 **syslib**, **userlib**, or **Personallib** with any directory

structure needed for organization. (The **Lib** directory name is special and its purpose will be explained later on in the document.)

The UDO scripts that are placed 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.

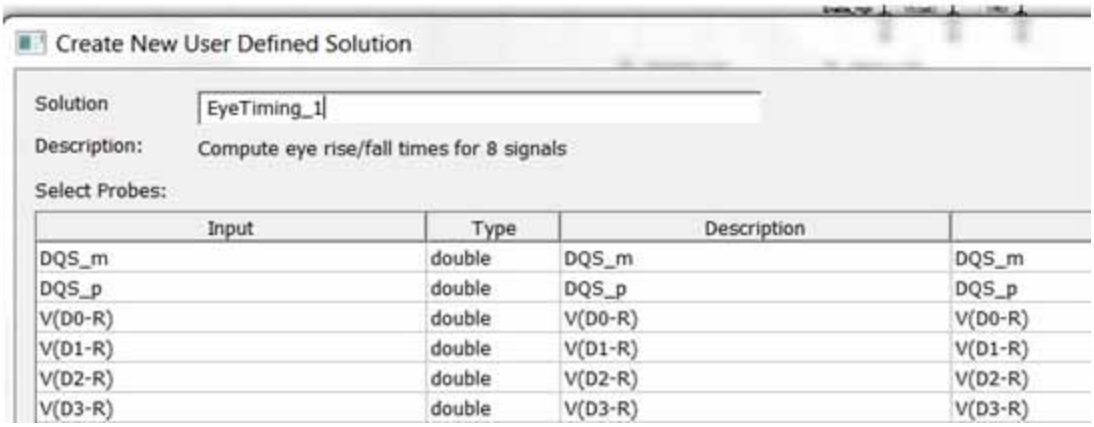
Use **Results** > **Create User Defined Solution** > **Update Menu** to refresh the menu to include the new UDO scripts that might have been copied to syslib, userlib, or Personallib; or exclude them if they have been deleted, after the launch of desktop. Once the user-defined-solution is created, the solution and the calculations defined by UDO become available in Reporter as any other quantities in a new "User Defined" report type.



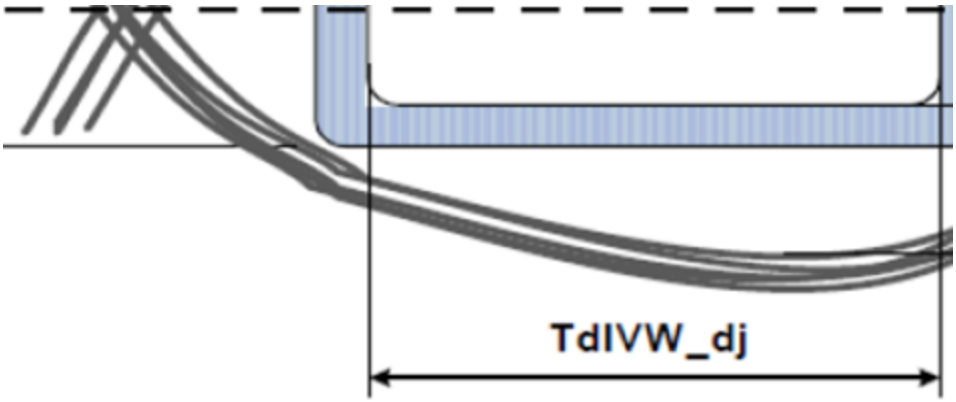
Named Probes and Properties in User Defined Outputs

UDOs allow processing data across traces, solutions, and report types. A UDO specifies the named probes and properties for which user selects or 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 or 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 EyeTiming_1.



You can then access this solution in the Reporter.



In another example, you could create a user-defined solution called **UDS Distance Trace Arithmetic Result1**:

Then access this solution in the Create User Defined Solution menu:

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. *When those calculations are subsequently plotted by user, reporter will use the cached results instead of invoking the computation on UDO*.

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 a function of *n* intrinsic variable, while another probe in same UDO can have data that is function of *m* intrinsic variables, with *n* and *m* are potentially different. UDOs allow you to reduce any number of these intrinsic variables. In the above example, UDO calculations can be a function of any number of intrinsic variables including not being function of any intrinsic variable at all. UDO calculations can also be a function of 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.*

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 end user. This allows UDOs to access the data for probes without requiring the end user to enter each individual probe. For example, a UDO can access data for a huge S matrix for 100 port design without having the end user enter the probe information for each of those 10,000 quantities. Each dynamic probe is associated with a named probe that is entered by user. Information about solution, context, and intrinsic variables is used from selected probes; however multiple dynamic probes can be associated with the same user selected probe. The dynamic probes are acquired 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.

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
- Data Types Used in UDO Python Scripts
- 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

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

Import Statements

The base class to use 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 *
```

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**.

The UDOExtension abstract class declares the optional Validate abstract method that may be implemented in the UDOExtension class or one of its base classes.

Example:

```
def BaseClassUDO ((IUDOPluginExtension):
 #base class implementation
 …
 def UDOExtension ((BaseClassUDO):
 #UDO class implementation …
```

Validate

This optional method is used to validate the user choices. The values of the properties entered, the probes etc. can be checked for suitability. This function, while a part of the IUDOExtensionabstract class, has a meaningful default implementation and is therefore optional. However, it can be overridden to take advantage of advanced functionality.

UI Access	NA		
Parameters	Name <errorStringList>	Type List<string>	Description [out] C# list of Python strings. Should be set only if validation failed;

			ignored if validation is successful. One error string should be set per each validation error.
	<udsProbParams>	List<UDSProbeParams>	[in] C# list of UDSProbeParams objects.
	<propList>	IPropertyList object	[in] list of properties
	< userSelectionForDynamicProbes >	List<UDSProbeParams>	[in] C# list of UDSProbeParams objects.
Return Value	Boolean. True on validation success and false on failure. The default implementation always returns true.		

Python Syntax	Validate (<*errorList*>, <*probeList*>, <*propertiesList*>, <*dynamicProbes*>)

Python Example	`def Validate(self, errorStringList, probeList, propList, dynamicProbes):`
	`  if probeList == None or probeList.Count == 0:`
	`  errorStringList.Add("Empty probe list")`
	`  return False`
	`  return True`

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
- GetQuantityInfo
- GetInputUDSParams
- GetDynamicProbes
- Compute

GetUDSName

Return a string that is used as a prefix for all solution instances created using this UDO.

UI Access	NA
Parameters	None
Return Value	String

Python Syntax	GetUDSName

Python Example	`def GetUDSName(self):`
	`  return "MinMaxAvg"`
	`udsName = UDOPluginEntention.GetUDSName`

GetUDSDescription

Returns a description for the UDO, its purpose, etc. This is used in multiple UDO related dialogs in the application to describe the UDO.

UI Access	NA
Parameters	None
Return Value	String

Python Syntax	GetUDSDescription
Python Example	`def GetUDSDescription(self):` `  return "Sample UDO for dimension reducing quantities`

GetUDSSweepNames

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.

UI Access	NA
Parameters	None
Return Value	List of strings. If the UDO outputs have no sweeps, return the empty list [].

Python Syntax	GetUDSSweepNames

Python Example	`# Returns list of sweeps names`
	`# We have no sweeps as we reduce them.`
	`def GetUDSSweepNames(self):`
	`    return []`

GetCategoryNames

The outputs that the UDO solution provides or generates can be classified into multiple categories, like how the application is displayed in the report creation dialog box.

UI Access	The output will be listed in the categories box in the dialog when creating reports from the UDO generated solution data.
Parameters	None
Return Value	List of strings

Python Syntax	GetCategoryNames
Python Example	`def GetCategoryNames(self):` `    return ["UDOOutputs"]`

GetQuantityNames

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:**
>
> The quantity names must be unique across the categories; that is, no two categories can have quantities with the same name.

UI Access	NA		
Parameters	Name <*categoryName*>	Type String	Description [in] Name of category.
Return Value	List of strings		

Python Syntax	GetQuantityNames(<*categoryName*>)

Python 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 []`

GetQuantityInfo

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.

UI Access	NA		
Parameters	Name <*quantityName*>	Type String	Description Name of quantity
Return Value	Object of type QuantityInfo.		

Python Syntax	GetQuantityInfo(<*quantityName*>)

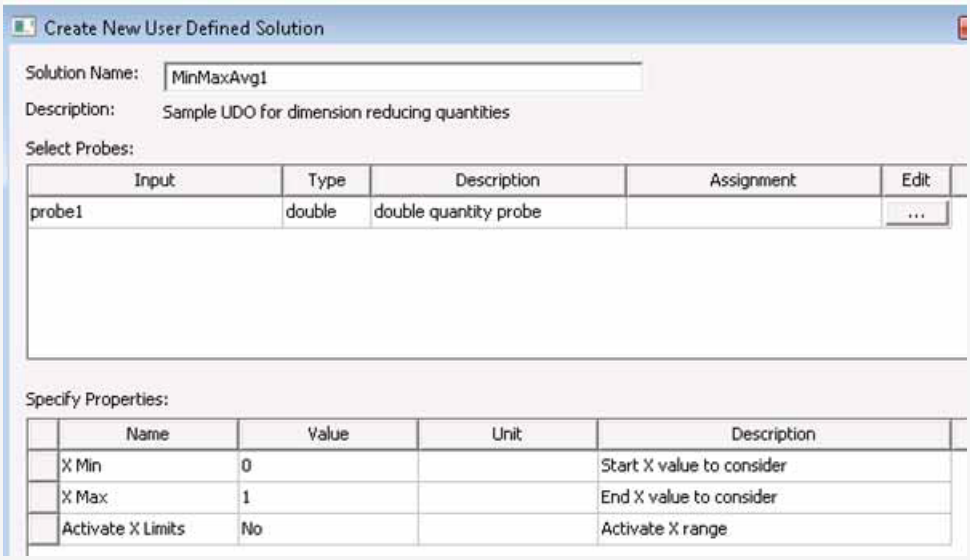
Python Example	`# Returns an instance of QuantityInfo for the qtyName supplied or None if such a`
	`# quantity could not be found`
	`def GetQuantityInfo(self, qtyName):`
	`    # All the quantities we have are simple doubles`
	`    # we can leave them unitless`
	`    return QuantityInfo(Constants.kDoubleParamStr)`

GetInputUDSParams

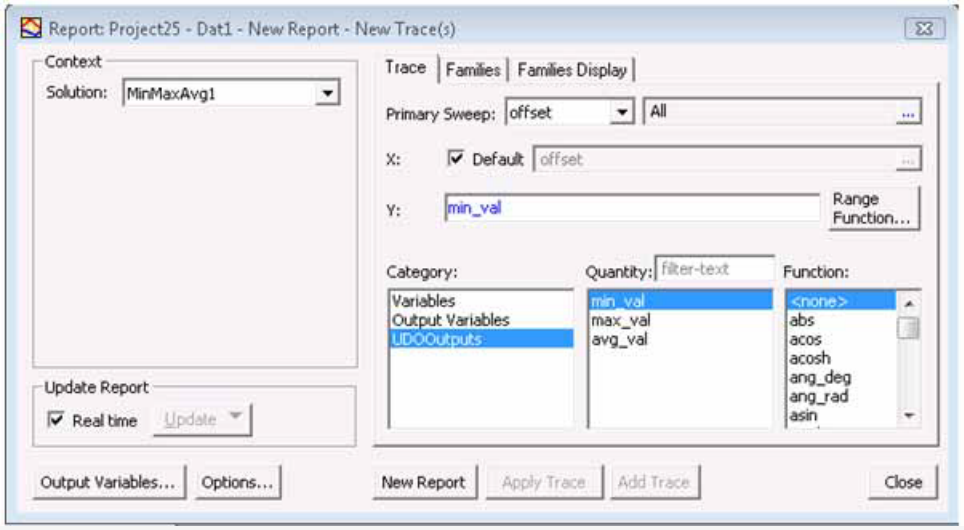
This is the main definition method of the UDO. The supplied arguments populate details of the parameters to which the UDO user will specify value, specify the probe names and their types as well as the dynamic probe selections.

UI Access

The **GetInputUDSParams** function results in the following dialog when you click **Results**>**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 the **Create New User Defined Solution** window.



When a report is created from the UDO dialog box, the category and quantity names specified by the UDO are used (as seen below).



Parameters

Name	Type	Description
<udsParams>	List< UDSProbeParams >	Parameters for the probe. The UDO script must add one instance of UDSProbeParams for each probe definition it will display. When creating the UDO solution, the UDO user must assign a matching quantity to each probe. When user define a UDS in the GUI, the user must select a design-solution-quantity for each UDSProbeParam .
<propList>	IPropertyList	The propList object is used to add properties that should be displayed to the user for data collection. These properties with their user-supplied values are returned to the UDO script by the Compute method.
<	List<	When user defines a

	userSelectionForDynamicProbes >	UDSProbeParams >	UDS in the GUI, the user ONLY needs to specify the Design-Solution. This quantity will be assigned dynamically by matching the component-expression defined in each UDSProbeParam.
Return Value	Boolean. If true, the event was handled successfully. If false, it was not.		

Python Syntax	GetInputUDSParams(<*udsParams*>, <*propList*>, <*userSelectionForDynamicProbes*>)

Python 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`

GetDynamicProbes

This is the primary mechanism by which the UDO script obtains the probe data (as double precision values) for its compute process.

UI Access	NA		
Parameters	Name <*dynamicProbes*>	Type List< UDSDynamicProbes >	Description [out] List of dynamic probes
Return Value	Boolean. If true, the method was successful. If false, it was not.		

Python Syntax	GetDynamicProbes(<*dynamicProbes*>)

Python Example	`# Returns list of UDSParams and list of dynamic properties`
	`# output UDSDynamicProbeCollection probes`
	`  def GetDynamicProbes(self, probes):`
	`  pass`

Compute

This is the main computation method which generates the data for the quantities that make up the UDO solution.

UI Access	The data is received from the UI using IUDSInputData. It is processed and the result data is sent to the UI using IUDSOutputData.		
Parameters	Name <*inData*> <*outData*> <*propList*> <*progressMonitor*>	Type IUDSInputData IUDSOutputData IPropertyList IProgressMonitor	Description Used to get the input probe data. Used to set the UDO solution quantity and sweep data. Used to get the user entered values for each of the properties defined during the GetInputUDSParams call. This can be used to set progress for long running calculations, check for user initiated abort etc.
Return Value	Boolean. If true, the method was successful. If false, it was not.		

Python Syntax	Compute (<*inData*>, <*outData*>, <*propList*>, <*progressMonitor*>)

Python 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 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
```


Data Types Used in UDO Python Scripts

There are several types that you must use while authoring a Python script. Some of them are used to pass data from the UI to a Python script and to provide interface for working with this data. Some are used to pass data from a Python script to the UI.

To pass data from a Python script to the UI, the objects must be created in the Python script. Then they can be set as a function's return values or set to the output parameters using their API.

Constants

- **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

Abstract Classes

- IProgressMonitor
- IUDSInput
- IUDSOutput

IUDSInputData

The purpose of this class is to get data (probe and sweep) from Desktop. The IUDSInputData abstract class declares the following abstract methods:

- GetDoubleProbeData
- GetSweepsDataForProbe
- GetComplexProbeData
- GetSweepNamesForProbe
- GetRequiredQuantities

- GetVariableValues
- GetInterpolationOrdersData

Examples in this section are just to show proper syntax of the function calls. For actual usage of the class, see the Compute function's example.

GetDoubleProbeData

This is the primary mechanism by which the UDO script obtains the probe data (as double precision values) for its compute process.

UI Access	NA		
Parameters	Name <*probeName*>	Type String	Description Name of the probe for which data is requested. This is one of the many probes supplied during a call to the UDO's GetInputUDSParams method. Each probeName must be unique within an UDO.
Return Value	Double array of data for the specified probe if the probe exists or null if the probe is unknown.		

Python Syntax	GetDoubleProbeData (<*probeName*>)
Python Example	`# doubleData is a list of floats` `doubleData = inData.GetDoubleProbeData("probe1")`

GetSweepsDataForProbe

All probe data that is supplied is associated with one or more sweep (an intrinsic quantity like Time, Frequency, Theta, Phi, etc. that is swept) quantities.

UI Access	NA		
Parameters	Name <*probeName*> <*sweepName*>	Type String String	Description Name of the probe for which data is requested. This is one of the many probes supplied during a call to the UDO's GetInputUDSParams method. Each probeName must be unique within an UDO. Name of the sweep.

Return Value	Double array of data for the specified probe and sweep.

Python Syntax	GetSweepsDataForProbe(<*probeName*>, <*sweepName*>)
Python Example	`# sweepData is C# Array of doubles (floats in python)` `sweepData = inData.GetSweepsDataForProbe ("FarFieldsProbe","Freq"])`

GetComplexProbeData

The primary mechanism by which the UDO retrieves data for its input probes (if it expects complex data for the probe).

UI Access	NA		
Parameters	Name <*probeName*>	Type String	Description Name of the probe for which data is requested. This is one of the many probes supplied during a call to the UDO's GetInputUDSParams method. Each probeName must be unique within an UDO.
Return Value	Double array (float in Python) of data for the specified probe. Each pair of floats represents one complex number. The first value is for real part and the second value it the imaginary part. For instance, array [10.0, 0, 5.1, 2.1] represents 2 complex numbers: (10.0, 0) and (5.1, 2.1).		

Python Syntax	GetComplexProbeData(<*probeName*>)

Python 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]))`

GetSweepNamesForProbe

Retrieves a 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).

UI Access	NA		
Parameters	Name <*probeName*>	Type String	Description Name of the probe for which data is requested. This is one of the many probes supplied during a call to the UDO's GetInputUDSParams method. Each probeName must be unique within an UDO.
Return Value	IList<string> - list of sweep names for the current probe name.		

Python Syntax	GetSweepNamesForProbe(<*probeName*>)
Python Example	`# sweepNames is C# Array of strings` `sweepNames = inData.GetSweepNamesForProbe("FarFieldsProbe")`

GetRequiredQuantities

A given UDO can specify that it provides one or more computed quantities. The user might choose to create a report from only a few among the various available UDO outputs. This function returns the list of UDO output quantities that the user requested. Only these need be computed in the UDO's Compute method.

UI Access	NA
Parameters	None
Return Value	IList<string>

Python Syntax	GetRequiredQuantities()

Python Example	`# quantities is C# Array of strings` `quantities = inData.GetRequiredQuantities()`

GetVariableValues

This allows the UDO to obtain the names and values of all the design variables for which the UDO quantities are being requested.

UI Access	NA
Parameters	None
Return Value	IDictionary<string,string> of key-value pairs for variables. Both key and value are strings.

Python Syntax	GetVariableValues()

Python 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`

GetInterpolationOrdersData

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.

UI Access	NA		
Parameters	Name <*probeName*>	Type String	Description Name of the probe for which data is requested. This is one of the many probes supplied during a call to the UDO's GetInputUDSParams method. Each probeName must be unique within an UDO.
Return Value	Byte array of interpolation order for the specified probe. These are to be		

	treated as 8-bit signed integers, that is, their values range from 0-127.

Python Syntax	GetInterpolationOrdersData(<*probeName*>)

Python Example	`# interData is C# Array of bytes (integers in Python)`
	`interData = inData.GetInterpolationOrdersData(kProbeNames[0])`
	`for interValue in theDict:`
	`  # interValue and order are integers`
	`  order = interValue`

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. The IUDSOutputData abstract class declares the following abstract methods:

- SetSweepsData
- SetDoubleQuantityData
- SetComplexQuantityData

Examples in this section are just to show proper syntax is function calls. For actual usage of the class, see the Compute function example.

SetSweepsData

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.

UI Access	NA		
Parameters	Name	Type	Description
	<*sweepName* >	String	Name of the sweep
	<*sweepData*>	List<floats>	Sweep data for the specified sweep.
Return Value	Boolean. If true, the event was handled successfully. If false, it was not.		

Python Syntax	SetSweepsData (<*sweepName*>,<*sweepData*>)

Python Example	`sweepList = [12.3, 14.5, 16.7]` `outData.SetSweepsData("Freq", sweepList)`

SetDoubleQuantityData

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.

UI Access	NA		
Parameters	Name <*qtyName*> <*qtyData*>	Type String List<Floats>	Description Name of the quantity. Quantity data for the specified quantity.
Return Value	Boolean. If true, the event was handled successfully. If false, it was not.		

Python Syntax	SetDoubleQuantityData (<*qtyName*>,<*qtyData*>)
Python Example	`doubleList = [12.3, 14.5, 16.7]` `outData.SetDoubleQuantityData("V1PlusV2", doubleList)`

SetComplexQuantityData

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.

UI Access	NA		
Parameters	Name <*qtyName* > <*qtyData* >	Type String List<floats>	Description Name of the quantity. Quantity data for the specified quantity. Complex numbers are passed as pairs of floats in a list.
Return Value	Boolean. If true, the method was successful. If false, it was not.		

Python Syntax	SetComplexQuantityData (<*qtyName*>,<*qtyData*>)

Python 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)

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 three 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

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

Constants Class

The constants used in a 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
```

UDSProbeParams Class

This class defines which data quantity that a UDO will be pulled from the design to compute the UDO output (results). The objects of this class must be created in a Python script with the 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,
"", "",)
```

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 )
```

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")
```

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It displays a task's calculated progress in the UI and checks if the user has requested to abort the computation. Use this object for computations that take a long time to complete or if the UI might freeze during 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

This class provides the following functionality and abort interaction:

- SetTaskName (string taskName)
- SetSubTaskName (string subTaskName)
- BeginTask (string name)
- SetTaskProgressPercentage(int progressPercent)
- CheckForAbort()
- 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)
```

> **Note:**
>
> There should be the same number of calls to BeginTask as to EndTask.

SetTaskName

Sets the name of the task whose progress is being monitored.

UI Access	NA		
Parameters	Name <*TaskName*>	Type String	Description Name of the task
Return Value	None		

Python Syntax	SetTaskName (<*TaskName*>)
Python Example	`progMon.SetTaskName("Compute UI")`

SetSubTaskName

Sets the name of the subtask whose progress is being monitored.

UI Access	NA		
Parameters	Name <*TaskName*>	Type String	Description Name of the task
Return Value	None		

Python Syntax	SetSubTaskName (<*TaskName*>)
Python Example	`progMon.SetSubTaskName("Compute UI segments")`

BeginTask

Sets the name of the task whose progress will be monitored. Call this method only once for each task and subtask.

UI Access	NA		
Parameters	Name <*Name*>	Type String	Description Name of the task
Return Value	None		

Python Syntax	BeginTask (<*Name*>)
Python Example	`progMon.BeginTask("Process DQS")`

SetTaskProgressPercentage

Sets the progress percentage of a task to a specific value.

UI Access	NA		
Parameters	Name <Percent>	Type Integer	Description Percent of progress
Return Value	None		

Python Syntax	SetTaskProgressPercentage (<*Percent*>)
Python Example	`progMon.SetTaskProgressPercentage(33)`

CheckForAbort

Checks if a user has aborted the task. Call this method if the quantities are computationally expensive. This method can be called multiple times.

UI Access	Abort button on the progress dialog.
Parameters	None
Return Value	Boolean flag that if true indicated that the user requested an abort and you should call EndTask. If false, allow the task to continue.

Python Syntax	CheckForAbort
Python Example	`should_abort = progMon.CheckForAbort`

EndTask

Stops a task from running. Call this method only once for each task and subtask. In a script, there should be the same number of BeginTask calls as **EndTask** calls.

UI Access	NA		
Parameters	Name <PassFail>	Type Boolean	Description Flag that indicates whether to continue the main task or not. If true, the task will continue to process the next subtask. If false, the main task stops and an error icon appears in the UI.
Return Value	None		

Python Syntax	EndTask (<*PassFail*>)
Python Example	`progMon.EndTask(True)`

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)
```

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
- AddInfoMessage
- AddWarningMessage

Example script:

```
############################################################
# 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
```

AddErrorMessage

Call this method to convey an error condition to the user.

UI Access	NA		
Parameters	Name <message>	Type String	Description Error message
Return Value	None		

Python Syntax	AddErrorMessage (<*message*>)
Python Example	`AddErrorMessage(self.GetName() + "Unexpected sweep count 0 or > 1 in Compute"))`

AddInfoMessage

Call this method to convey an informational message to the user. This is the call to use when outputting messages for debugging purposes.

UI Access	NA		
Parameters	Name <message>	Type String	Description Warning message
Return Value	None		

Python Syntax	AddInfoMessage (<*message*>)
Python Example	`AddInfoMessage("Enter starting X value.")`

AddWarningMessage

Call this method to convey a warning message, typically used for conditions that are not ideal but can be tolerated by the script.

UI Access	NA		
Parameters	Name <message>	Type String	Description Warning message
Return Value	None		

Python Syntax	AddWarningMessage (<*message*>)
Python Example	`AddWarningMessage("Unexpected sweep count in Compute")`

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

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 through the top level *UserDefinedOutputs* directory can be imported).

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.

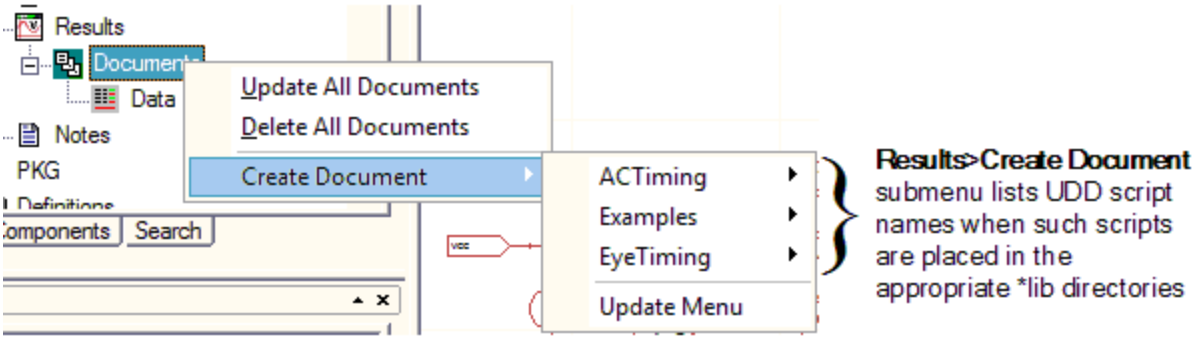
1. Build your .NET assembly for .NET 2.0 runtime.
2. Drop the built assembly in any **Lib** directory upstream of the UDO script location: that is, if you have your UDO script in *C:\Users\x\PersonalLib\UserDefinedOutputs\a\b\c\myudo.py* and have a .NET assembly called *com.Acme.UDOLib* You can keep the .NET assembly under:
    - UserDefinedOutputs\Lib
    - UserDefinedOutputs \a\Lib
    - UserDefinedOutputs \a\b\Lib
    - UserDefinedOutputs\a\b\c\Lib
3. Add the following lines 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")
```

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 box that lets you specify trace and solution inputs. After you confirm your input selections, XML, HTML, and PDF documents are generated. A web browser window opens to display the generated HTML file. The created document appears in the Project Manager, under Results in the Documents folder.



The general UDD process flow is as follows:

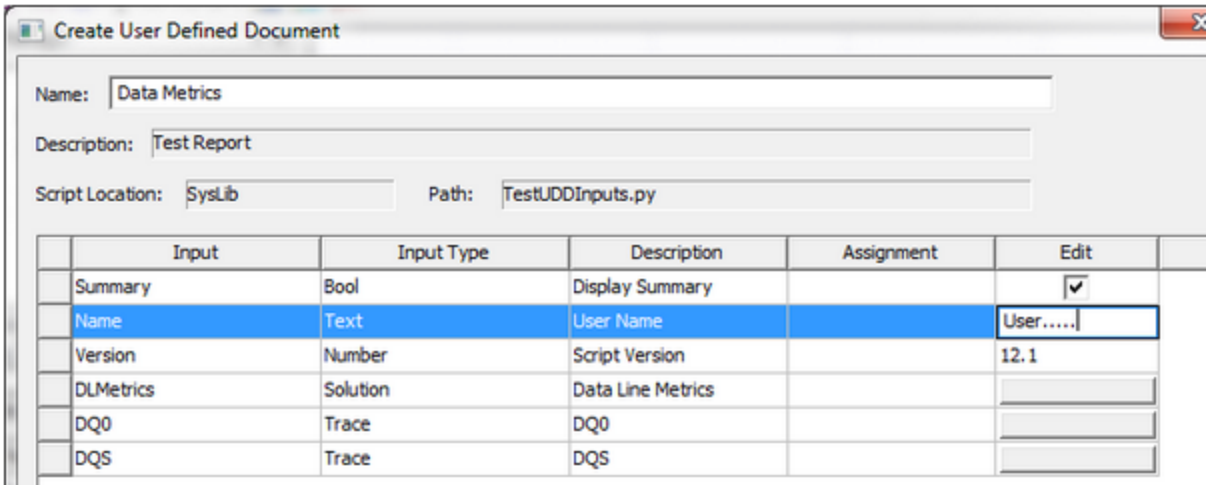
The UDD Python scripts must be placed in the **UserDefinedDocuments** directory under either of **syslib**, **userlib**, or **Personallib** with any subdirectory structure needed. The Lib directory can contain Python scripts that have common code that other scripts can use.

Use **Q3D Extractor** > **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 are available through the **Q3D Extractor** > **Results** > **Create Document** menu.

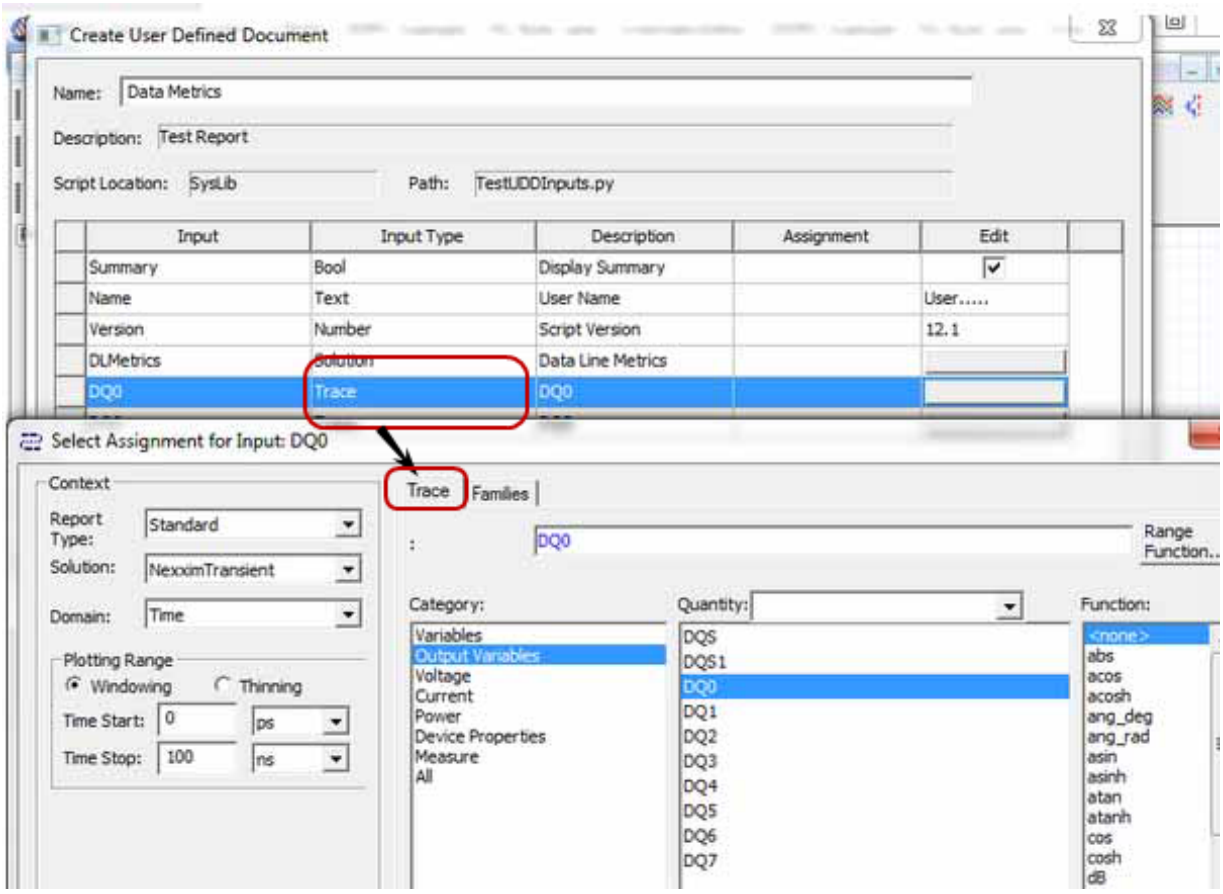
Create User-Defined Document Inputs

User-defined documents allow data from traces, solutions, and report types as inputs. A UDD can specify the named inputs, which you select or enter the values of in the *Create User Defined Document* dialog box that displays when you run **Q3D Extractor** > **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.

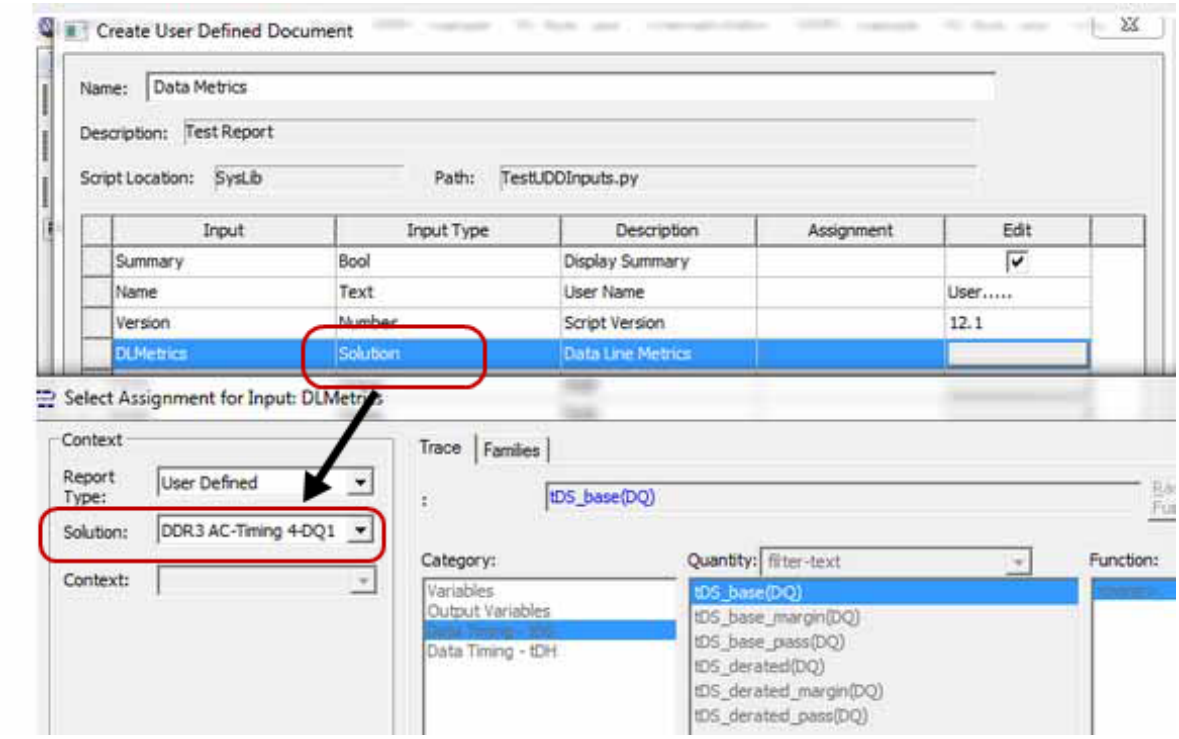
At the time of selection, you can choose from the *Reporter* dialog box the report type (Standard, Eye Diagram, User Defined), solution name, context, and the quantity for which you want the trace data.



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 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.

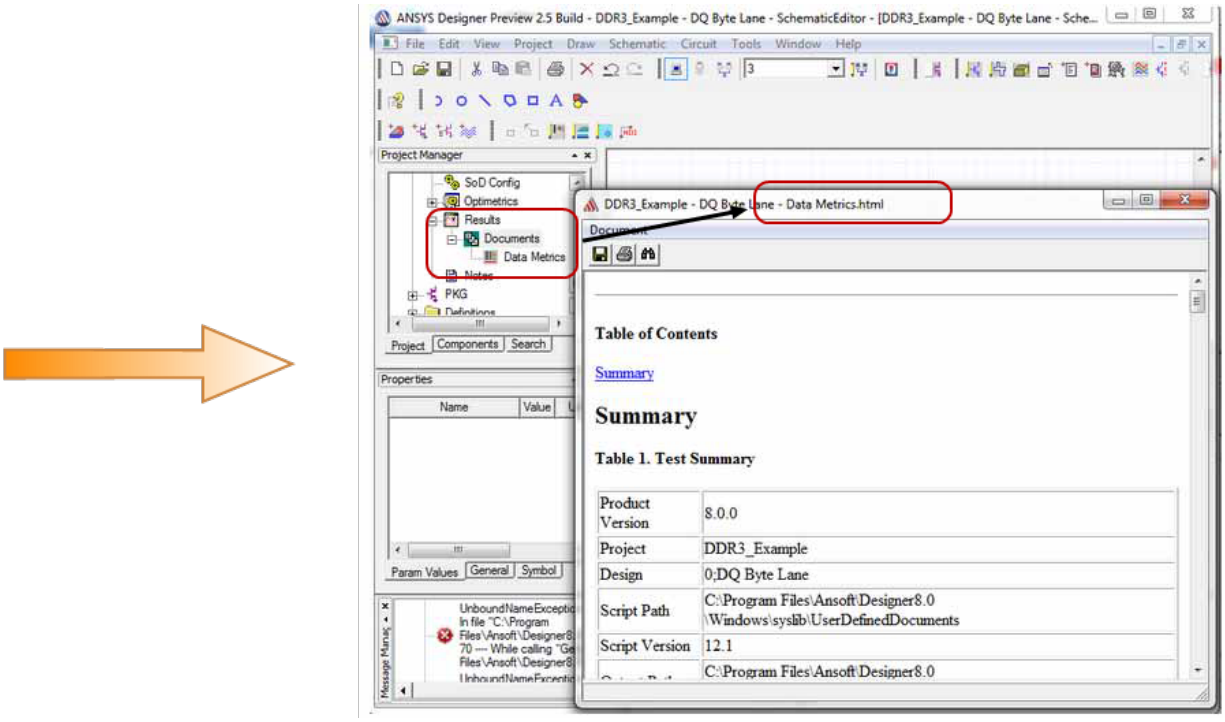
> **Note:** The Category/Quantity/Function portion of the dialog is disabled for user input.



UDD Document Creation and Display

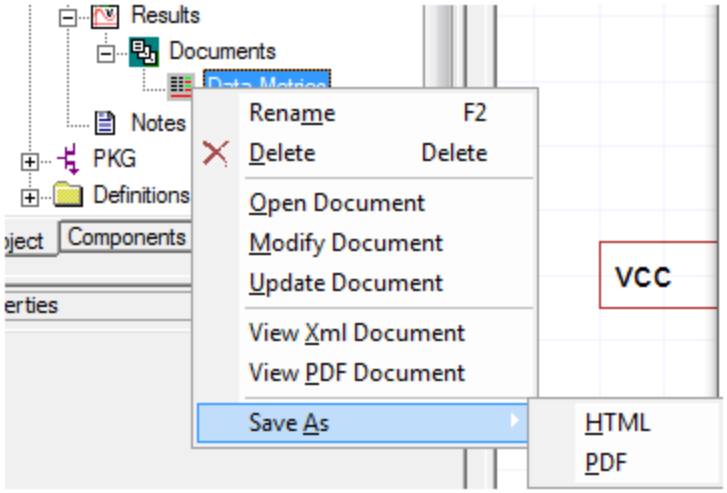
After all the input selections for a UDD are confirmed, based on the script, XML, HTML, and PDF documents are generated based on the inputs provided by the user. (The XML, HTML, and PDF generation is based on specific calls in the Python script, which are explained in a following section.) A web browser window also opens to display the generated HTMLfile.

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.



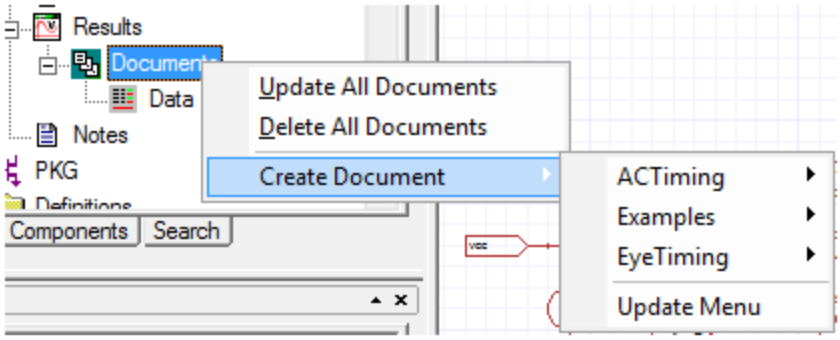
Managing Documents Listed in the Project Window Under Results

Right-click 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 box, 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's Context Menu

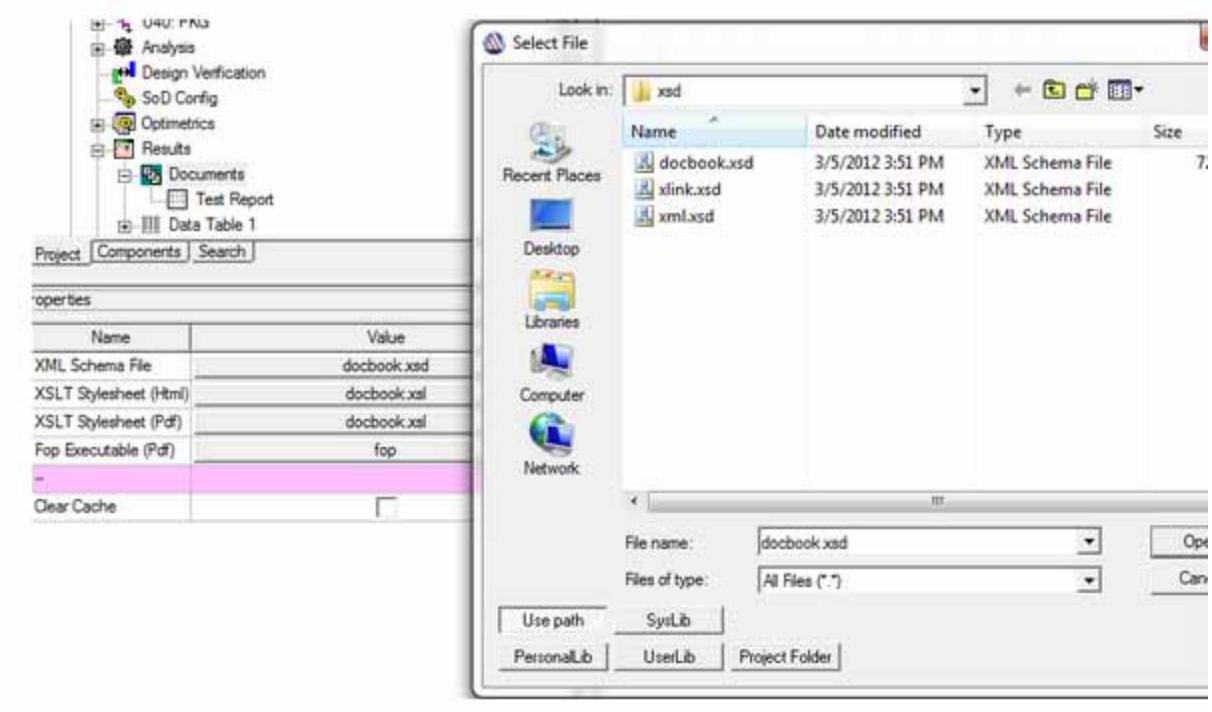
Right-click the documents folder has the menu options to **Update All Documents** or **Delete All Documents**. It also provides the option of creating a document from here.



Document Folder's 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.

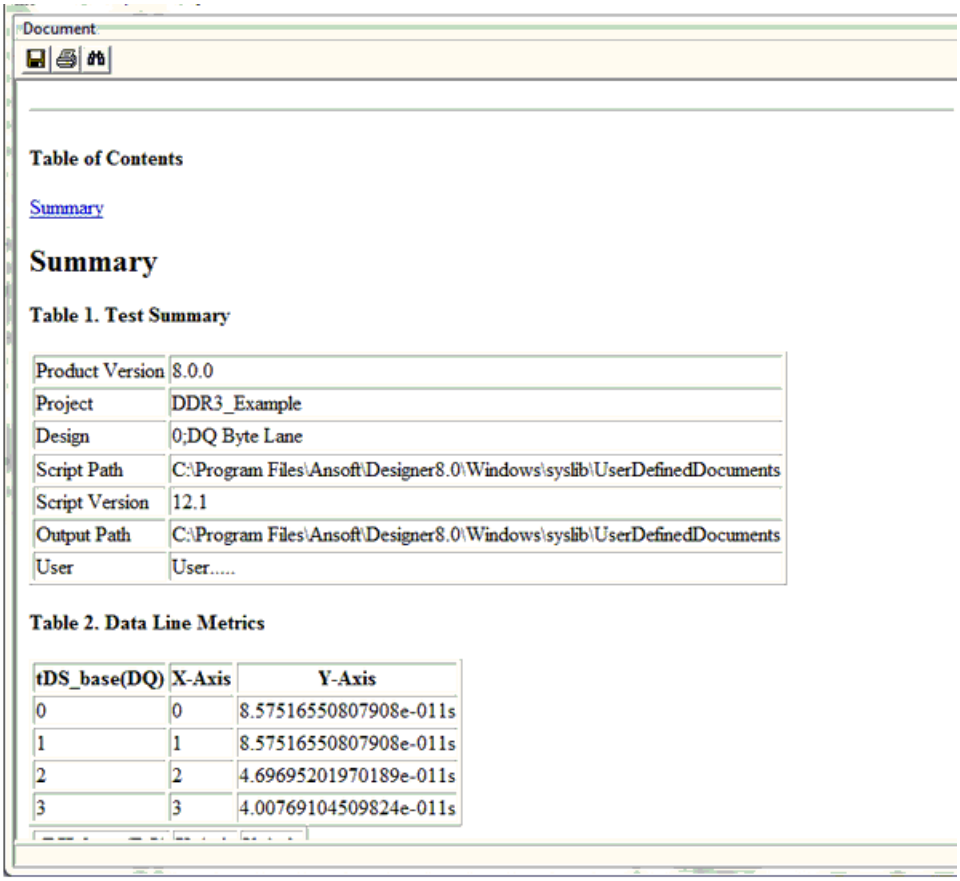

Document

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

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 through 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 lines 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")
```

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.

This API supports multiple data types in the forms of constants and classes. It also has several input interfaces.

User-Defined Document scripting commands are provided in this product's scripting guide. An complete example with a line by line explanation of how to use these methods is available.

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.Interfaces
import *
```

Data Types Used in UDD Python Scripts

There are several types that you must use while authoring a Python script. Some of them are used to pass data from the UI to a Python script and to provide interfaces for working with this data. Some are used to pass data from a Python script to the UI.

To pass data from a Python script to the UI, the objects of the C# class must be created in the Python script using their C# constructors. Then they can be set as a function's return values or set to the output parameters using their API.

Constants

- **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

Abstract Classes

- IProgressMonitor Abstract Class
- UDDExtension Class
- UDDInputData class
- UDDInputParams class

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It displays a task's calculated progress in the UI and checks if the user has requested to abort the computation. Use this object for computations that take a long time to complete or if the UI might freeze during 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

This class provides the following functionality and abort interaction:

- SetTaskName (string taskName)
- SetSubTaskName (string subTaskName)
- BeginTask (string name)
- SetTaskProgressPercentage(int progressPercent)
- CheckForAbort()
- 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)
```

> **Note:**
>
> There should be the same number of calls to BeginTask as to EndTask.

SetTaskName

Sets the name of the task whose progress is being monitored.

UI Access	NA		
Parameters	Name <*TaskName*>	Type String	Description Name of the task
Return Value	None		

Python Syntax	SetTaskName (<*TaskName*>)
Python Example	`progMon.SetTaskName("Compute UI")`

SetSubTaskName

Sets the name of the subtask whose progress is being monitored.

UI Access	NA		
Parameters	Name <*TaskName*>	Type String	Description Name of the task
Return Value	None		

Python Syntax	SetSubTaskName (<*TaskName*>)
Python Example	`progMon.SetSubTaskName("Compute UI segments")`

BeginTask

Sets the name of the task whose progress will be monitored. Call this method only once for each task and subtask.

UI Access	NA		
Parameters	Name <*Name*>	Type String	Description Name of the task
Return Value	None		

Python Syntax	BeginTask (<*Name*>)
Python Example	`progMon.BeginTask("Process DQS")`

SetTaskProgressPercentage

Sets the progress percentage of a task to a specific value.

UI Access	NA		
Parameters	Name <Percent>	Type Integer	Description Percent of progress
Return Value	None		

Python Syntax	SetTaskProgressPercentage (<*Percent*>)
Python Example	`progMon.SetTaskProgressPercentage(33)`

CheckForAbort

Checks if a user has aborted the task. Call this method if the quantities are computationally expensive. This method can be called multiple times.

UI Access	Abort button on the progress dialog.
Parameters	None
Return Value	Boolean flag that if true indicated that the user requested an abort and you should call EndTask. If false, allow the task to continue.

Python Syntax	CheckForAbort
Python Example	`should_abort = progMon.CheckForAbort`

EndTask

Stops a task from running. Call this method only once for each task and subtask. In a script, there should be the same number of BeginTask calls as **EndTask** calls.

UI Access	NA		
Parameters	Name <PassFail>	Type Boolean	Description Flag that indicates whether to continue the main task or not. If true, the task will continue to process the next subtask. If false, the main task stops and an error icon appears in the UI.
Return Value	None		

Python Syntax	EndTask (<*PassFail*>)
Python Example	`progMon.EndTask(True)`

IUDDPluginExtension Abstract Class

The **IUDDPluginExtension** abstract class declares the following abstract methods which must be implemented in the UDDExtension class or one of its base classes. If any of these methods are not implemented, a run-time error will occur and the UDD will not function.

- GetUDDName
- GetUDDDescription
- ShowDefaultSetupDialog
- GetUDDInputParams
- Generate

This class also provides for the following optional methods:

- SetupUDDInputParams
- HandleUDDEvents
- GetUDDSchema
- GetUDDStyleSheetForHtml
- GetUDDStyleSheetForPdf
- GetFopExecutable
- GetUDDAppContext
- GetUDDDesignContext

GetUDDName

Retrieves a prefix to use for all solution instances created using this UDD.

UI Access	NA
Parameters	None
Return Value	String that is the prefix

Python Syntax	GetUDDName()

Python Example	`def GetUDDName(self):` `  return "MinMaxAvg"`

GetUDDDescription

Retrieves a description for the UDD, its purpose, etc.

UI Access	NA
Parameters	None
Return Value	String that is the description

Python Syntax	GetUDDDescription()
Python Example	`def GetUDDDescription(self):` `  return "Sample UDD"`

ShowDefaultSetupDialog

Retrieves whether to show the default setup dialog box. If not, the user might want to implement or show a customized setup dialog.

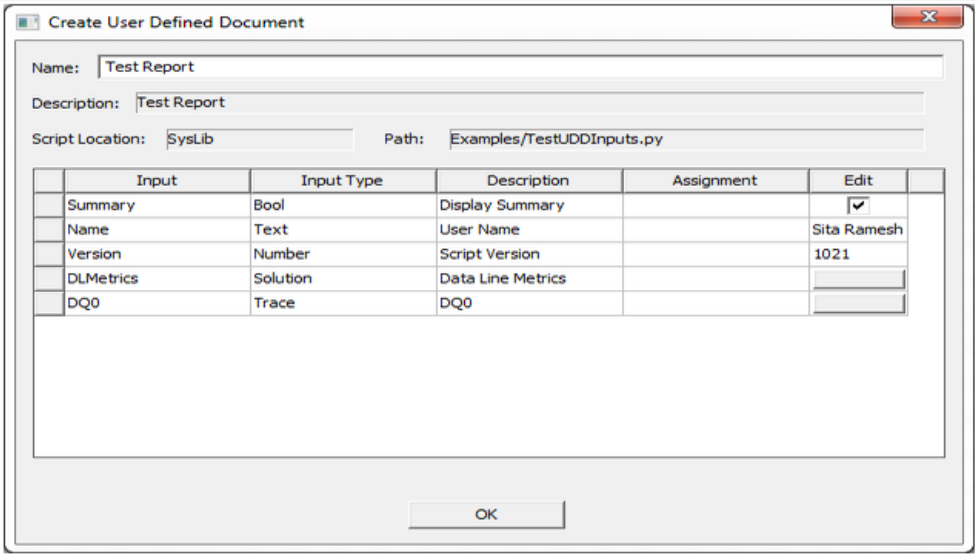
UI Access	NA
Parameters	None
Return Value	Boolean. If true, the default dialog should be shown. If false, the user does not want to see the default dialog.

Python Syntax	ShowDefaultSetupDialog()
Python Example	`def ShowDefaultSetupDialog(self):` `  return True`

GetUDDInputParams

Retrieves inputs parameters for the User Defined Document. These parameters are the input's name, description, and data type. The UDD user specifies the value of the inputs through a

dialog or within the script.

UI Access	Depending on the input parameters, the following dialog may be displayed when you click **Q3D Extractor**> **Results**> **Create Document**. The name and description of the UDD are also displayed in this dialog. 
Parameters	Name: *<uddInputs>* Type: List< UDDInputParams >* Description: The UDD script is expected to add one instance of UDDInputParams for each input definition that it will to display. When creating the UDD, the user will assign a matching value to each such input.
Return Value	Boolean. If true, the method was successful. If false, it was not.

Python Syntax	GetUDDInputParams(*<probeName>*)

Python 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
```

Generate

This is the main method which accesses the data from the uddInputs and generates the document.

UI Access	NA		
Parameters	Name	Type	Description
	<uddInputs>	List< UDDInputData>	The list of inputs that the user setup in the dialog box. They are now available to query for data.
	<generator>	IUDDGenerator	This is the document generator object which used to create different elements of the document like titles, sections, tables, images and write the data too. See the Document Generator Interfaces.
	<progressMonitor>	IProgressMonitor	This can be used to set progress for long running calculations and check for any user-initiated aborts.
Return Value	Boolean. If true, the method was successful. If false, it was not.		

Python	Generate(*<uddInputs>*, *<generator>*, *<progressMonitor>*)

Syntax	

Python 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
```

SetupUDDInputParams

This optional method displays a customized dialog and returns the user choices for the input params.

UI Access	NA		
Parameters	Name *<uddInputs>*	Type List<UDDInputParams>	Description .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. This method 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.
Return Value	Boolean. If true, the event was handled successfully. If false, it was not.		

Python Syntax	SetupUDDInputParams(*<uddInputs>*)

Python 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

This optional 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. The tags associated with the event are received by plugin using this abstract class.

UI Access	NA

Parameters	Name Type Description <*eventTags*> List<string> Event tags.
Return Value	Boolean. If true, the event was handled successfully. If false, it was not.

Python Syntax	HandleUDDEvents(<*eventTags*>)

Python 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

This optional method retrieves the file path of the schema to validate the XML. This will override the default schema used.

UI Access	NA
Parameters	None
Return Value	String containing the full file path of the schema.

Python Syntax	GetUDDSchema()

Python Example	`def GetUDDSchema(self):`
	`  return "C:\\Program Files\\ANSYS Inc\\v251\\AnsysEM\\common\\docbook\\schema\\xsd\\docbook.xsd"`
	`schemaPath = pluginExt.GetUDDSchema()`

GetUDDStyleSheetForHtml

This optional method retrieves the file path of the style sheet used to generate the HTML document. This will override the default stylesheet for HTML.

UI Access	NA
Parameters	None
Return Value	String containing the full file path of the style sheet.

Python Syntax	GetUDDStyleSheetForHtml()

Python Example	`def GetUDDStyleSheetForHtml(self):`
	`  return "C:\\Program Files\\ANSYS Inc\\v251\\AnsysEM\\common\\docbook\\"`
	`styleSheet = pluginExt.GetUDDStyleSheetForHtml()`

GetUDDStyleSheetForPdf

This optional method retrieves the file path of the style sheet used to generate the PDF document. This will override the default stylesheet for PDF.

UI Access	NA
Parameters	None
Return Value	String containing the full file path of the style sheet.

Python Syntax	GetUDDStyleSheetForPdf()

Python Example	`def GetUDDStyleSheetForPdf(self):`
	`  return "C:\\Program Files\\ANSYS Inc\\v251\\AnsysEM\\common\\docbook\\xsl\\fo\\docbook.xsl"`
	`styleSheet = pluginExt.GetUDDStyleSheetForPdf()`

GetFopExecutable

This optional method retrieves the file path of the fop executable used to generate the PDF document.

UI Access	NA

Parameters	None
Return Value	String containing the full file path of the fop executable.

Python Syntax	GetFopExecutable()

Python Example	`def GetFopExecutable(self):`
	`  return "C:\\Program Files\\ANSYS Inc\\v251\\AnsysEM\\common\\ApacheFOP\\fop-1.0\\fop"`
	`fopExe = pluginExt.GetFopExecutable()`

GetUDDAppContext

This optional method retrieves the UDD Owner if it is set. This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the desktop app.

UI Access	NA
Parameters	None
Return Value	the IDispatch for the desktop object

Python Syntax	GetUDDAppContext()
Python Example	`oDesktop = self.GetUDDAppContext()`

GetUDDDesignContext

This optional method retrieves the UDD Owner if it is set. This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the design.

UI Access	NA
Parameters	None

Return Value	IDispatch for the specified design object

Python Syntax	GetUDDDesignContext()
Python Example	`oDesign = self.GetUDDDesignContext()`

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:**
>
> This class is modeled after the UDO class UDOExtension, therefore the usage is similar.

UDDInputData Class

This class contains user input data and can be used in the generated document. UDDInputData has the constructor UDDInputData(string name) and the following properties accessors:

Property accessor	Purpose	Form
Name	Get/Set the name of an input.	string Name();
Type	Get/Set the type of an input	string Type();

The data can be any of the following types, based on the UDDInputParams type specified in the script.

- UDDInputBool
- UDDInputDouble
- UDDInputSolution
- UDDInputText
- UDDInputTrace

UDDInputBool

Method purpose	Return Value	Data type of the returned value in C#
Retrieves boolean value of the input data.	a boolean value of input data	string Data();

UDDInputDouble

Method purpose	Return Value	Data type of the returned value in C#
Retrieves double data.	a double value of input data	double Data();

UDDInputSolution

Method purpose	Return Value	Data type of the returned value in C#
Retrieves x and y double data for a probe	Data collection	IDictionary<double, double> DoubleData (string name);
Retrieves x and y double data for a probe for a particular variation	Data collection	IDictionary<double, double> DoubleData (string name, IDictionary<string, string> variation);
Retrieves x and y complex data for a probe	Data collection	IDictionary<double, double[]> ComplexData (string name);
Retrieves x and y complex data for a probe for a particular variation	Data collection	IDictionary<double, double[]> ComplexData (string name, IDictionary<string, string> variation);
Retrieves x and y text data for a probe	Data collection	IDictionary<string, string> TextData(string name, IDictionary<string, string> variation);

Retrieves x and y text data for a probe for a particular variation	Data collection	IDictionary<string, string> TextData(string name);
Retrieves a list of category names	A string collection	IList<string> CategoryNames();
Retrieves a list of quantity names given a category name	A string collection	IList<string> QuantityNames(string category);
Retrieves a list of variable values	A string collection	IList<Dictionary<string, string>> VariableValues();

UDDInputText

Method purpose	Return Value	Data type of the returned value in C#
Retrieves string data	a string value of input data	string Data();

UDDInputTrace

Method purpose	Return Value	Data type of the returned value in C#
Retrieves x and y double data	Double	IDictionary<double, double> DoubleData();
Retrieves x and y double data for a particular variation	Double	IDictionary<double, double> DoubleData (IDictionary<string, string> variation);
Retrieves x and y complex data	Double	IDictionary<double, double[]> ComplexData();
Retrieves x and y complex data for a particular variation	Double	IDictionary<double, double[]> ComplexData (IDictionary<string, string> variation);
Retrieves x and y text data	Text	IDictionary<string, string> TextData();
Retrieves x and y text data for a particular variation	Text	IDictionary<string, string> TextData (IDictionary<string, string> variation);
Retrieves a list of variable values	A string collection	IList<Dictionary<string, string>> VariableValues();
Retrieves an image of the trace data in a plot	An image	string Image();

UDDInputParams Class

The objects of this class must be created in Python script in the GetUDDInputParams and SetUDDInputParams functions.

Attributes :

- Input Name (string)
- Input Description (string)
- Input Type (string) (This can be boolean, number, text, trace, or solution.)
- 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 description 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

UDD Input Interfaces

The Generate function takes in a list of inputs which allows 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 9 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 11 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()
```

Explication of a Sample UDD Script

This VB script defines a document. It is a portion of one of the UDD example VB scripts.

```
Array("NAME:Test Report",          ' Name of the document
"Test Report",                     ' Description of the document
```

```
"SysLib",                                   ' Location of the python script: Syslib, Userlib, PersonalLib, etc.
"TestUDDReport",                            ' Relative path of the script in the UserDefinedDocuments folder

' This array is the start of the input definition.
Array("NAME:Inputs",                        ' Document Inputs keyword

' This array contains the Solution input.
Array("NAME:DLMetrics",                     ' Input name
"Solution",                                 ' Solution Input Type
"Data Line Metrics",                        ' Input Description
-1,                                         ' Solution ID
-1),                                        ' Report ID

' This array contains the trace input.
Array("NAME:DQ0",                           ' Input name
"Trace",                                    ' Trace Input Type
"DQ0",                                      ' Input Description
-1,                                         ' Solution ID
-1),                                        ' Report ID

' This array contains the text input.
Array("NAME:Name",                          ' Input name
"Text",                                     ' Text Input Type
"User Name",                                ' Input Description
Array("Sita Ramesh")),                      ' Default Value

' This array contains the Bool input.
Array("NAME:Summary",                       ' Input name
"Bool",                                     ' Boolean Input Type
"Display Summary",                          ' Input Description
Array(true)),                               ' Default Value

' This array contains the number input.
Array("NAME:Version",                       ' Input name
"Number",                                   ' Number Input Type
"Script Version",                           ' Input Description
Array(1021)))),                             ' Default Value
```

```
' This array contains trace selection for the solution and trace inputs.
Array("NAME:DocTraces",                    ' Document traces keyword

' This array has input for "DLMetrics".
Array("NAME:DLMetrics",                    ' Input name

' This array defines a trace 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())),

' This array is for input "DQ0".
Array("NAME:DQ0",

' This array defines a trace similar to the UDO. This trace definition 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())))
```

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. Use this interface to get 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. Use this interface to get 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 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 this interface. In 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.
     ```
     image1.SetFormat("format")
     ```

- IUDDText interface

  Calling AddText() on the IUDDSection interface provides you with the this interface. In 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. In 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", and "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. In 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. In 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. In 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**.

PinToPinUtility

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.

Exporting Equivalent Circuit Data

You can export Q3D and 2D Extractor equivalent circuit (lumped RLGC) data.

Exporting Equivalent Circuit Data in Q3D Extractor

You can export lumped RLGC data from a solution to the following circuit data file formats:

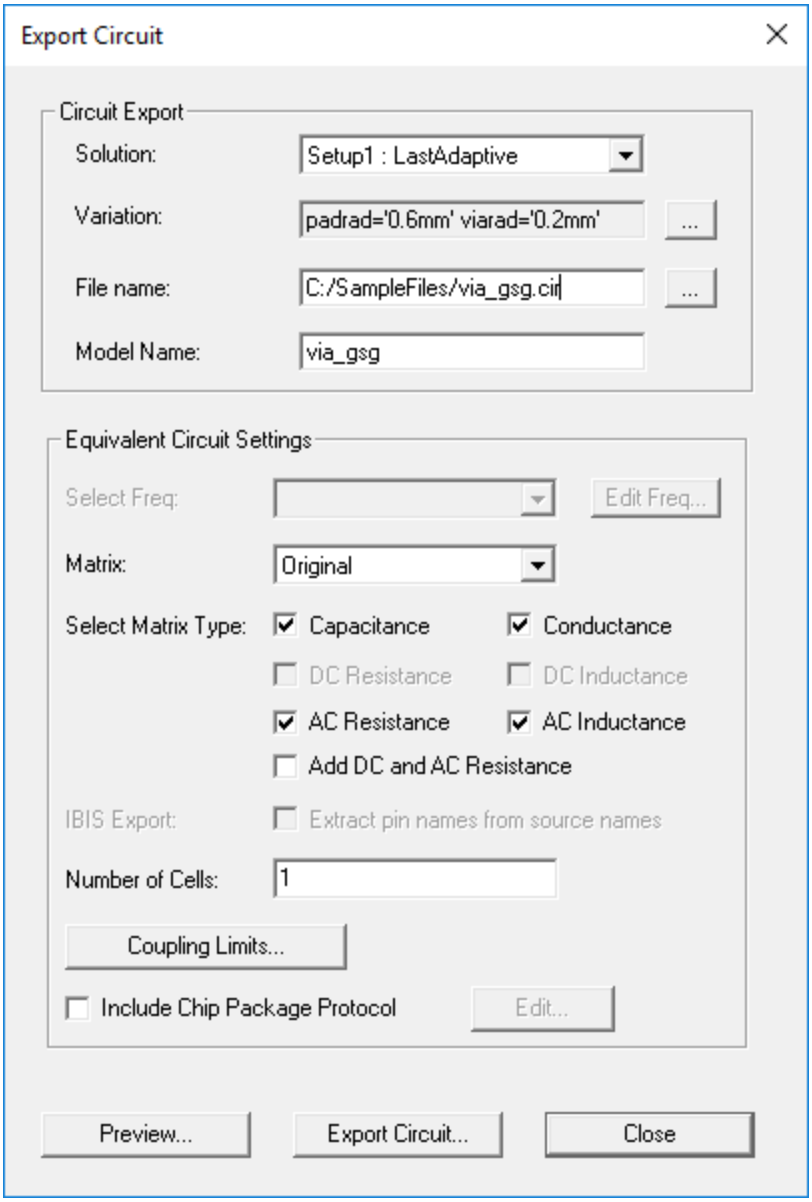
- **Ansoft Designer Netlist (*.cir)**
- **Berkeley Spice Circuit (*.bsp)**
- **Cadence DML (*.dml)**
- **HSPICE Circuit (*.sp)**
- **IBIS ICM (*.icm)**
- **IBIS Package Model (*.pkg)**
- **Maxwell Spice Circuit (*.spc)**
- **PSpice Circuit (*.lib)**
- **Simplorer Netlist Files (*.sml)**
- **Spectre Circuit (*.ckt)**

After importing the new data file to PSpice, HSPICE, or Maxwell Spice, you can include wave effects in circuit simulations.

To export a circuit file:

1. Click **Q3D Extractor** > **Analysis Setup** > **Export Circuit**.

   The **Export Circuit** window appears.



2. In the **Circuit Export** area, set the following:

   - **Solution** – use the drop-down menu to select the solution from which to export data.
   - **Variation** – use the **...** button to select a variation from all variations in the design.

- **File Name** – enter a file path and name for the circuit file to be exported.

  > **Important:**
  >
  > To save the circuit as an IBIS package model, use all lowercase names to facilitate portability between operating systems. The length should not be more than 20 characters.
  >
  > To save the circuit in IBIS ICM format, use all lowercase names to facilitate portability between operating systems. The extension should be no more than three characters in length. There is no length restriction on the base file name; however, only lowercase letters, digits 0 through 9, underscore and hyphen are valid characters.

- **Model Name** – enter a name for the model within the circuit file. Upper and lowercase letters, digits 0 through 9, and underscore are valid characters. Spaces are not accepted. Any invalid characters will be replaced with underscores.

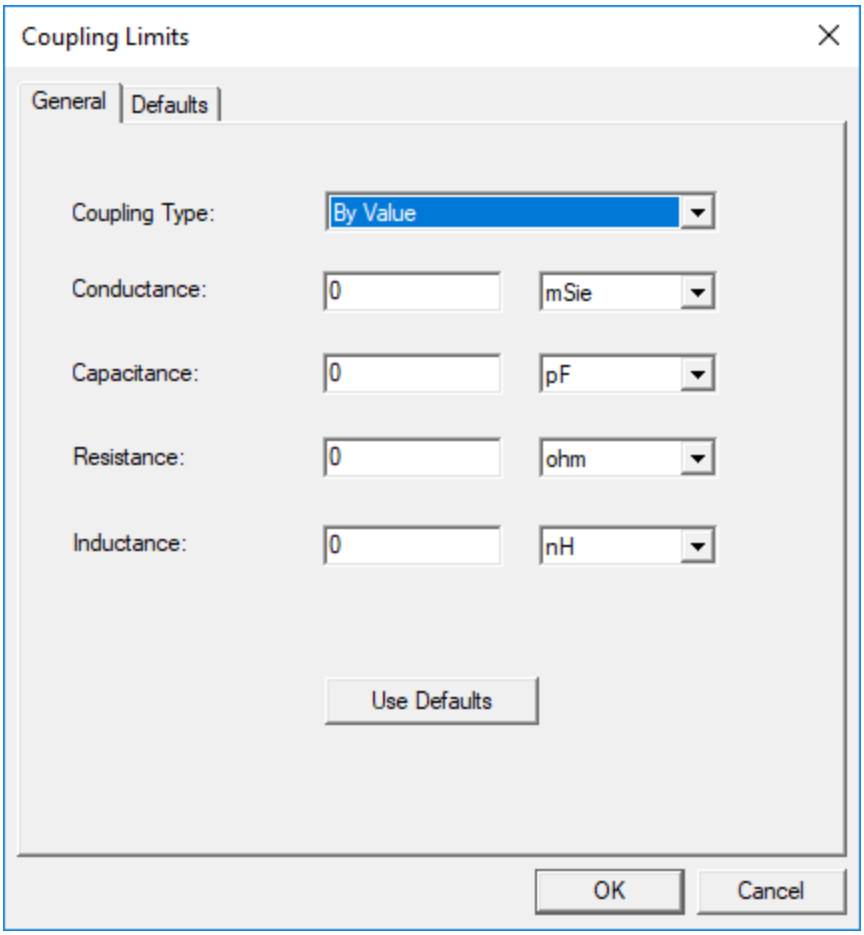
3. In the **Equivalent Circuit Settings** area, set the following:

   - **Select Freq** – for discrete or interpolating sweep solutions, use the drop-down menu to select a frequency. This option is unavailable for other solution types.

     For interpolating sweeps, you can click **Edit Freq** to open the **Edit Sweep** dialog box and enter start and end values.

   - **Matrix** – select either the original solved matrix or any reduced matrices.
   - **Select Matrix Type** – select one or more matrix types for export. Note that selecting DC Resistance or DC Inductance disables the option for AC Resistance or AC Inductance, and vice versa. Select **Add DC and AC Resistance** if you want to create a circuit with a total resistance computed from AC and DC values.
   - **IBIS Export** – by default, names are created in the form _NAME_src. Selecting the **Extract pin names from source names** check box will change that to just NAME.
   - **Number of Cells** – enter the number of cells (that is, sections) to be used while exporting the circuit. Finer discretization (i.e., a larger number of sections) gives a closer approximation to the underlying Partial Differential Equations. However, this also leads to increased simulation time. See the technical notes for Equivalent Circuits.

- **Coupling Limits** – click to open the **Coupling Limits** dialog box.



  Select a **Coupling Type** and enter limit values. RLCG values smaller than those specified will be ignored.

- **Include Chip Package Protocol** – select this option and click Edit to specify Chip Package Protocol header information for SPICE formats.

4. Click **Preview** to preview the Circuit model.
5. Click **Export Circuit** to complete the export.
6. Click **Close** to exit the window.

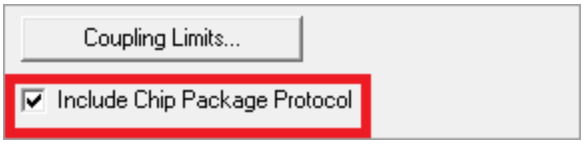
Specifying Chip Package Protocol Header Information in Q3D Extractor

Chip Package Protocol (CPP) is a set of statements specifying the location, the net information, and the SPICE node name for each package pin (the landing pad of flip-chip bumps or the chip landing pad of bonding-wires). It is embedded within a SPICE-compatible file header as comments.

The CPP is primarily used to help automate the co-simulation of IC (die) and package or package/board. For example, Apache's Redhawk utilizes CPP to import the package or package and PCB model into the full chip dynamic voltage drop analysis, and the global IO-SSO analysis. Package/PCB simulation tool can utilize the CPP to import the chip model in the system level analysis.

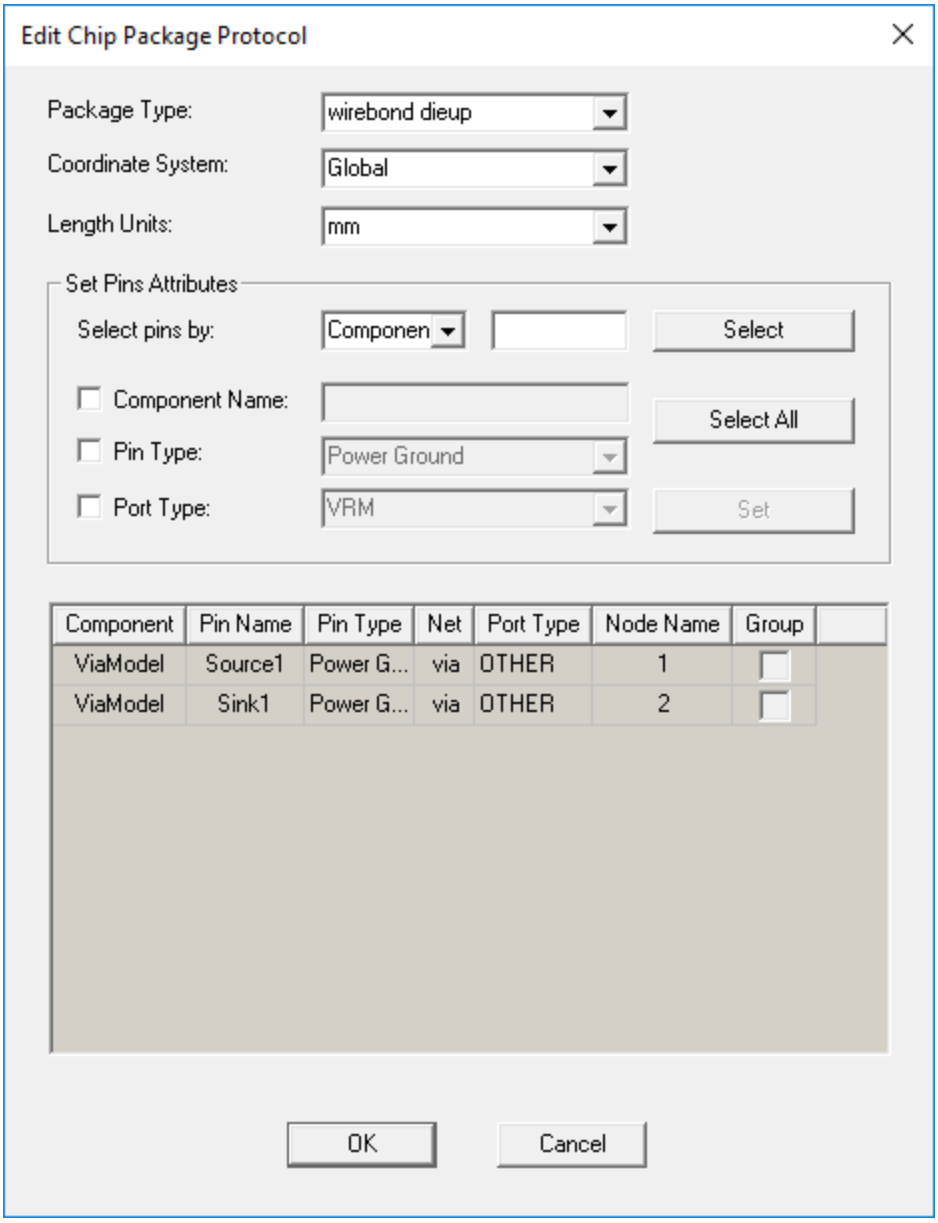
To specify the CPP information:

1. When exporting circuit data, select the **Include Chip Package Protocol** check box.



2. Click **Edit**.

   The **Edit Chip Package Protocol** window appears.



3. Set the following options:
    - **Package Type** – Choose from wirebond dieup, wirebond diedown, flipchip dieup, and flipchip diedown.
    - **Coordinate System** – Select either Global or a custom coordinate system.
    - **Length Units** – Select the unit of measure.
4. Use the options under **Set Pins Attributes** to select pins.
    - **Select pins by** – Use the drop-down menu to sort pins by component name, pin name (number), pin type (signal or power ground), net name, or port type (DIE,

PCB, VRM, OTHER). Enter a regular expression into the search field and click **Select** to select the applicable pins.

- **Select All** – Click to select all pins in the design.

5. Use the **Component Name**, **Pin Type**, and **Port Type** fields to modify the information for selected pin(s). Click **Set** to apply changes.
6. Click **OK** to exit the **Edit Chip Package Protocol** window.

Exporting Equivalent Circuit Data in 2D Extractor

You can export distributed or lumped RLGC data from a solution to the following circuit data file formats:

- **Berkeley Spice Circuit (*.bsp)**
- **Cadence DML (*.dml)**
- **HSPICE Circuit (*.sp)** – produces a lumped RLC ladder model at a single frequency.
- **IBIS ICM (*.icm)**
- **Intel LCF (*.lcf)**
- **Maxwell Spice Circuit (*.spc)**
- **Nexxim/HSPICE W Element (*.sp)**
- **Nexxim/HSPICE RLGC W Element (*.sp)** – produces a frequency-dependent W element with RLGC, R_s and G_d. R_s and G_d are the high-frequency asymptotes due to skin effect and loss tangent, respectively.

  $$\mathrm{R}(\mathrm{f}) = \mathrm{R_o} + \mathrm{R_s} * \sqrt{\mathrm{f}}$$

  $$\mathrm{G}(\mathrm{f}) = \mathrm{G_o} + \mathrm{G_d} * \mathrm{f}$$

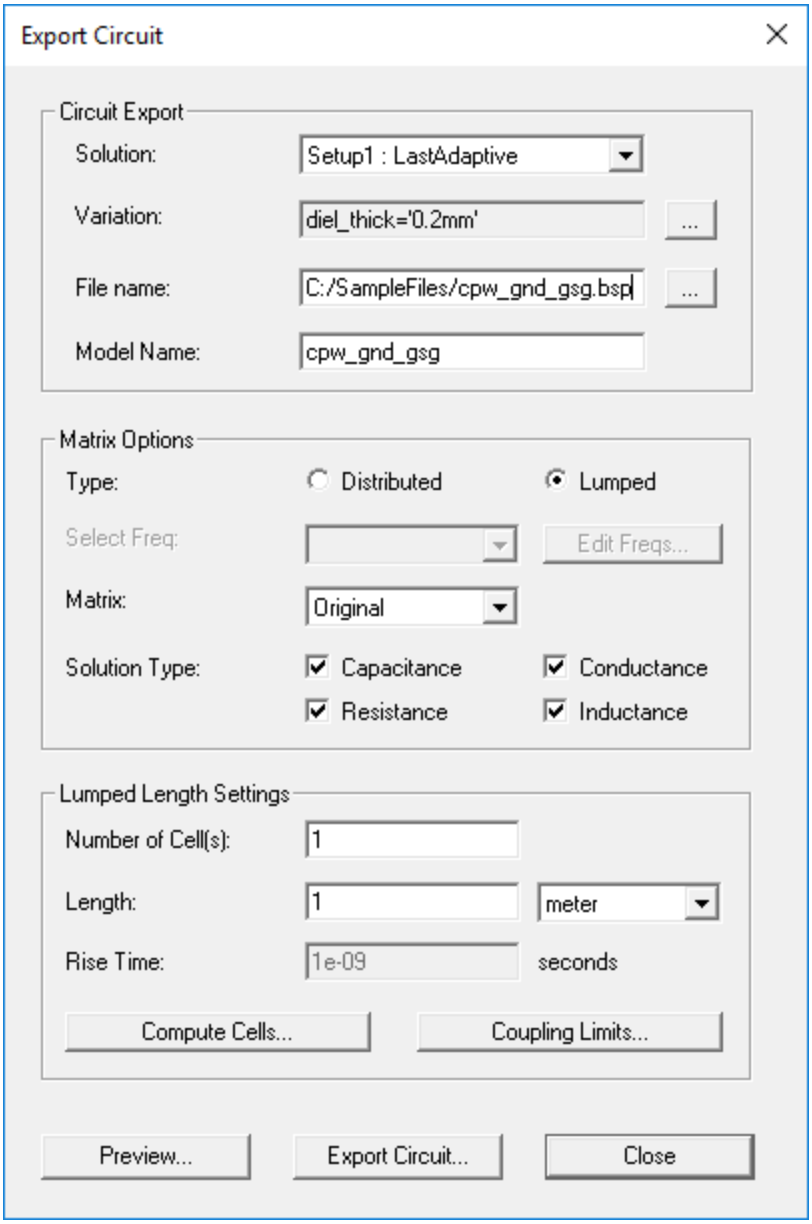
- **PSpice Circuit (*.lib)**
- **Simplorer Netlist Files (*.sml)**
- **Spectre Circuit (*.ckt)**

After importing the new data file to PSpice, HSPICE, or Maxwell Spice, you can include wave effects in circuit simulations.

To export a circuit file:

1. Click **2D Extractor** > **Analysis Setup** > **Export Circuit**.

   The **Export Circuit** window appears.



2. In the **Circuit Export** area, set the following:
    - **Solution** – use the drop-down menu to select the solution from which to export data.
    - **Variation** – use the **...** button to select a variation from all variations in the design.

- **File Name** – enter a file path and name for the circuit file to be exported.

  > **Note:**
  >
  > To save the circuit in IBIS ICM format, use all lowercase names to facilitate portability between operating systems. The extension should be no more than three characters in length. There is no length restriction on the base file name; however, only lowercase letters, digits 0 through 9, underscore and hyphen are valid characters.

- **Model Name** – enter a name for the model within the circuit file. Upper and lowercase letters, digits 0 through 9, and underscore are valid characters. Spaces are not accepted. Any invalid characters will be replaced with underscores.

3. In the **Matrix Options** area, set the following:

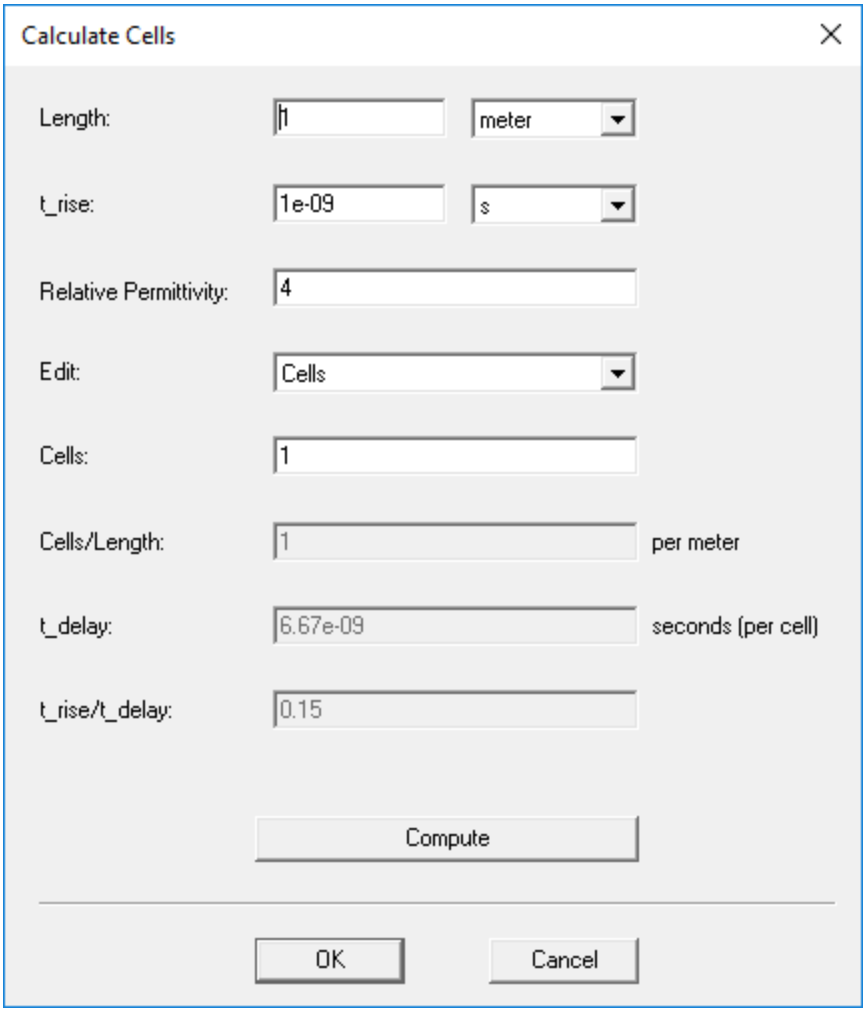
   - **Type** – select either **Distributed** or **Lumped**. Both export types contain nodes for the near and far ends of signal lines, but lumped exports contain nodes for non-ideal grounds only, while distributed exports contain nodes for all ground conductors. See Lumped Circuit Models.
   - **Select Freq** – for discrete or interpolating sweep solutions, use the drop-down menu to select a frequency. This option is unavailable for other solution types.

     For interpolating sweeps, you can click **Edit Freq** to open the **Edit Sweep** dialog box and enter start and end values.

   - **Matrix** – select either the original solved matrix or any reduced matrices.
   - **Solution Type** – select one or more solution types for export.

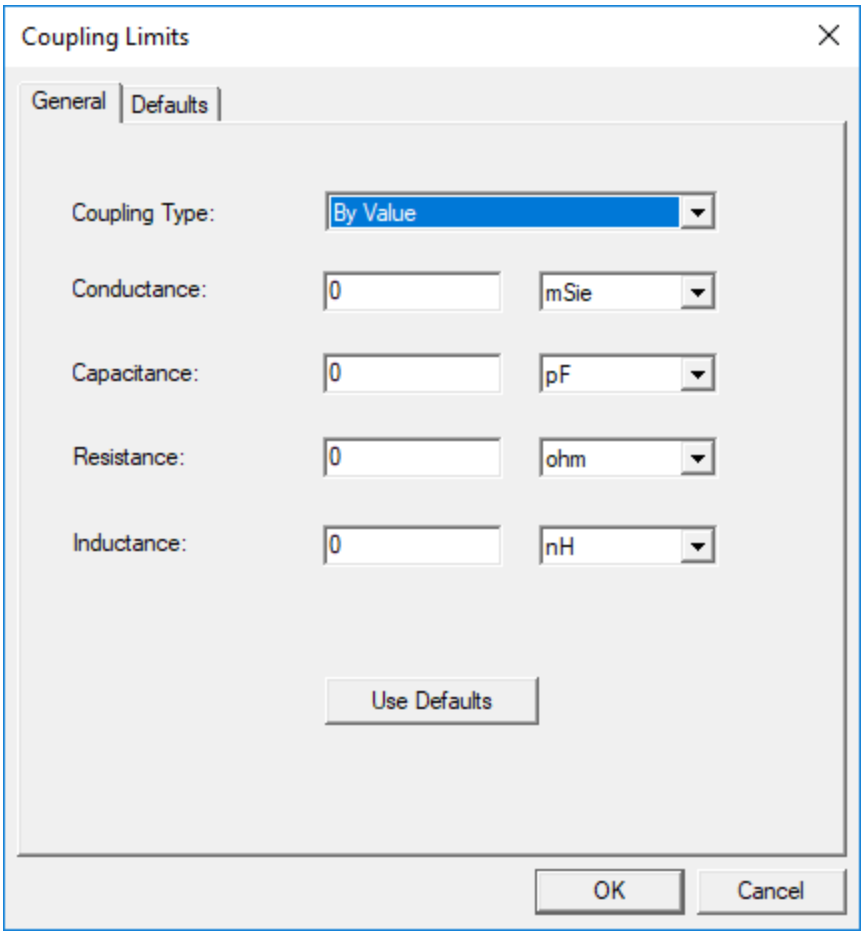
4. If you chose the **Lumped** export, in the **Lumped Length Settings** area, set the following:

   - **Number of Cells** – enter the number of cells (that is, sections) to be used while exporting the circuit. Finer discretization (i.e., a larger number of sections) gives a closer approximation to the underlying Partial Differential Equations. However, this also leads to increased simulation time. You can calculate the number of cell using the **Compute Cells** option. See the technical notes for Equivalent Circuits.
   - **Length** – specify the length of the transmission line by entering a value and selecting a unit of measure from the drop-down menu. This option is also available for certain distributed exports.
   - **Rise Time** – specify the rise time, in seconds.
   - **Compute Cells**– click to open the **Calculate Cells** dialog box, which allows you to calculate how many cells to use in a lumped ladder model.



Enter the **Length** of the transmission line, and select the units. Enter the minimum rise/fall time of the signals on the line in the **t_rise** box, and select the units. From the **Edit** drop-down menu, select the quantity you want to edit. Enter the number of cells into which the lumped ladder model will be broken in the **Cells** box. Enter the number of cells per meter in the **Cells/Length** box. Enter the delay through a single cell in the **t_delay** box. This is computed as t_delay = $1/c_0$ * length/cells; where c_0 = speed of light in a vacuum, length = length of the transmission line and cells = number of cells into which the lumped ladder model is broken. Enter the ratio of **t_rise** to **t_delay** in the **t_rise/t_delay** box. This should be 3 or more for good results.

- **Coupling Limits** – click to open the **Coupling Limits** dialog box.



Select a **Coupling Type** and enter limit values. RLCG values smaller than those specified will be ignored.

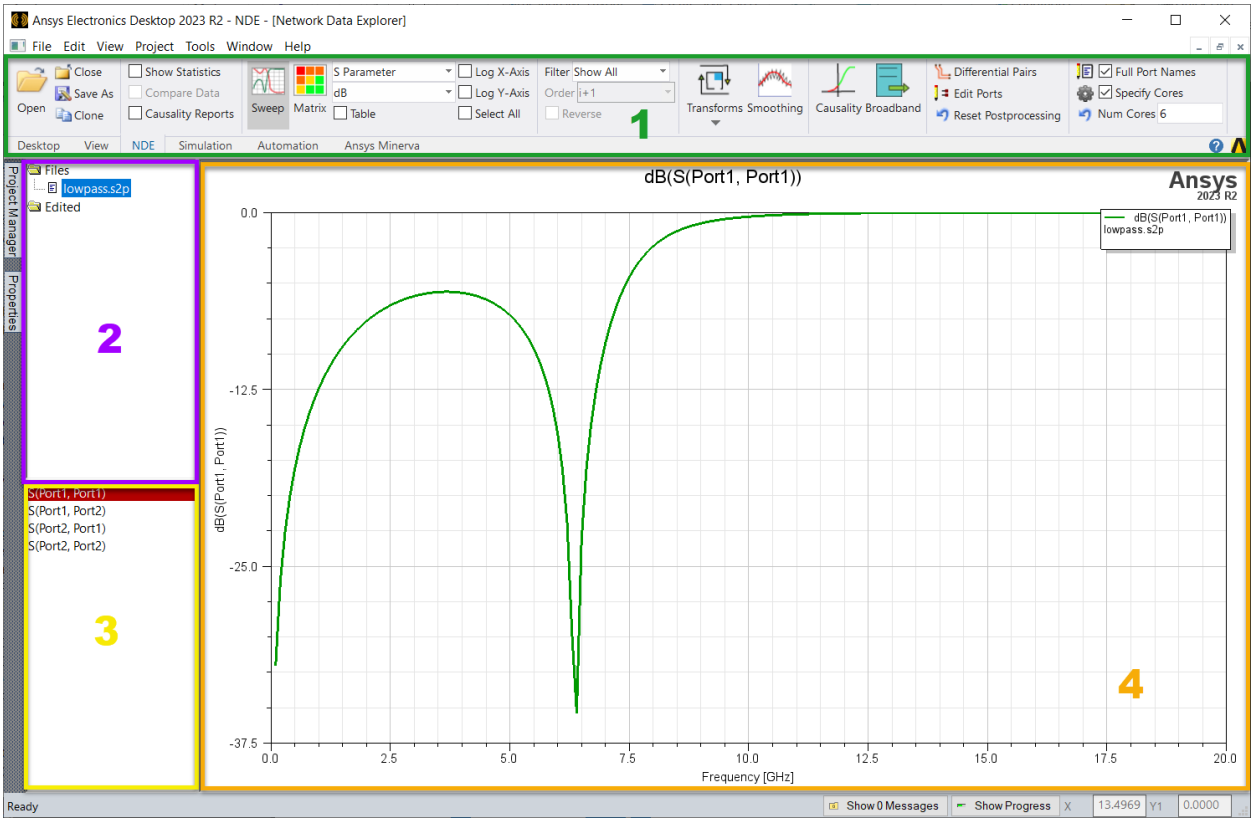
5. Click **Preview** to preview the Circuit model.
6. Click **Export Circuit** to complete the export.
7. Click **Close** to exit the window.

18 - Network Data Explorer

The Network Data Explorer provides visualization, analysis, and manipulation tools for network data.

To access Network Data Explorer, click **Tools** > **Network Data Explorer**.

The **Network Data Explorer** window appears.

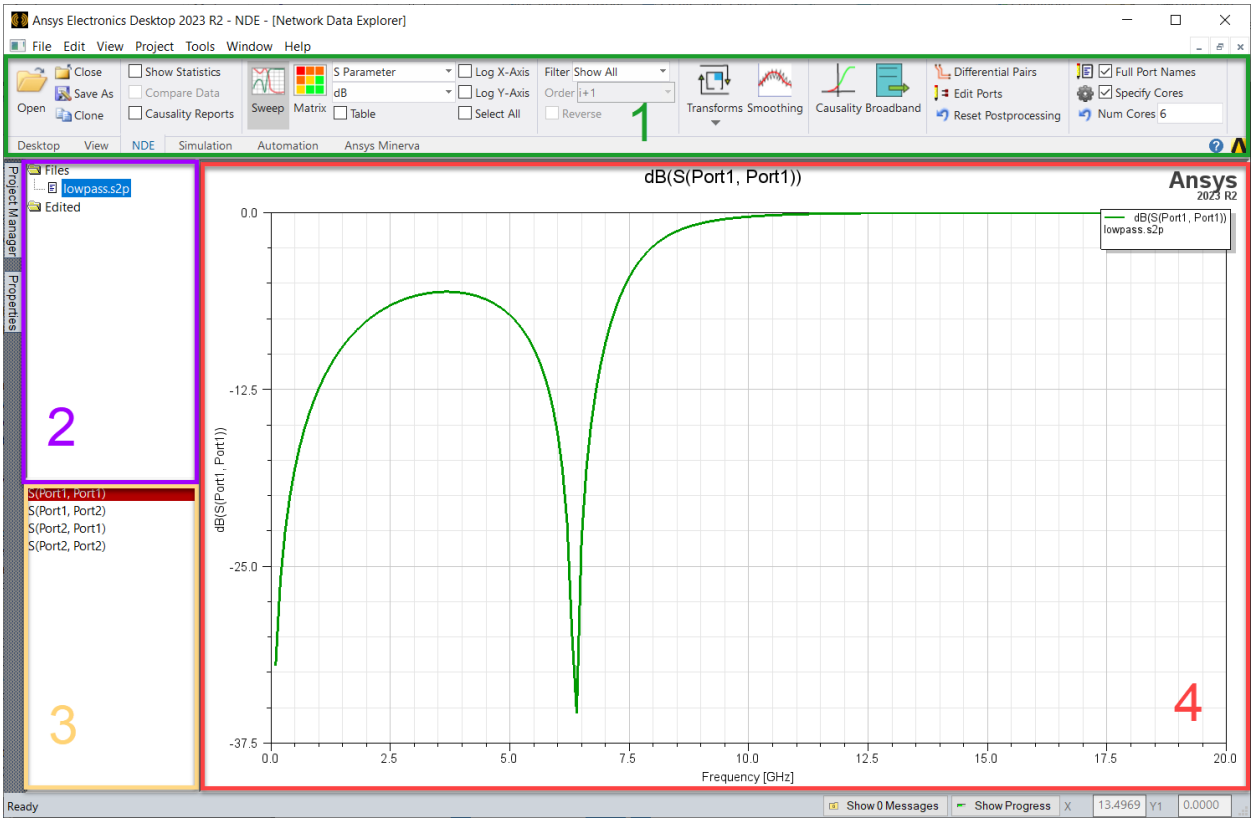


Network Data Explorer Overview

The Network Data Explorer window is divided into the following panes:

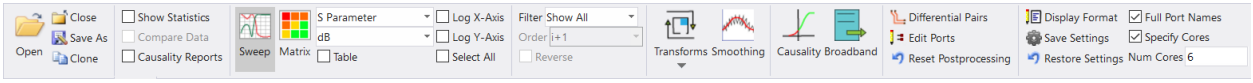
1. **NDE ribbon** – allows you to perform many functions of the Network Data Explorer.
2. **Network Data Selection pane** – allows you to select a network data file.
3. **Cell and Frequency Selection pane** – allows you to narrow your selection.
4. **Data View pane** – displays data in table or plot 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.



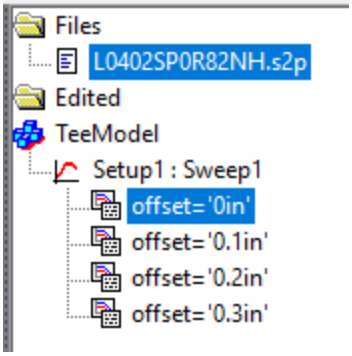
On this ribbon, you can control:

- 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.

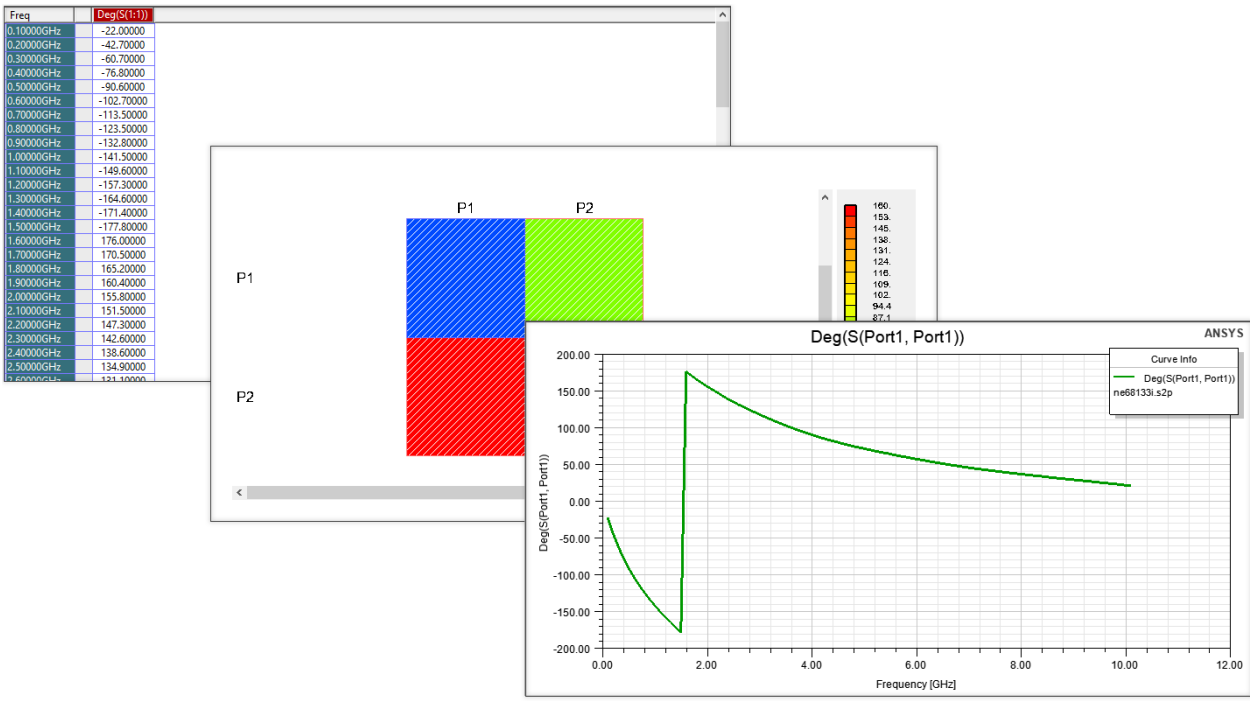
Use the **Shift** and **Ctrl** keys to select and display data sets from multiple files simultaneously. Multiple data sets can be selected to display multiple traces in sweep plots, but not in tables. Click the **Files** icon to select all data sets under **Files**. Click the **Edited** icon or a design/setup 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.

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

You can launch **Network Data Explorer** from within several Ansys products. In the Project Manager, open <project> > **Analysis** > **Setup***n* > **Sweep***n*. Right click on the sweep and click **Network Data Explorer** in 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.

> **Note:** When solution data that is loaded into NDE is modified and resimulated in another Ansys product, the NDE data automatically updates.

You can import the following file types:

- Touchstone Format (*.s*p)
- Touchstone 2 Format (*.ts, *.sp)
- 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

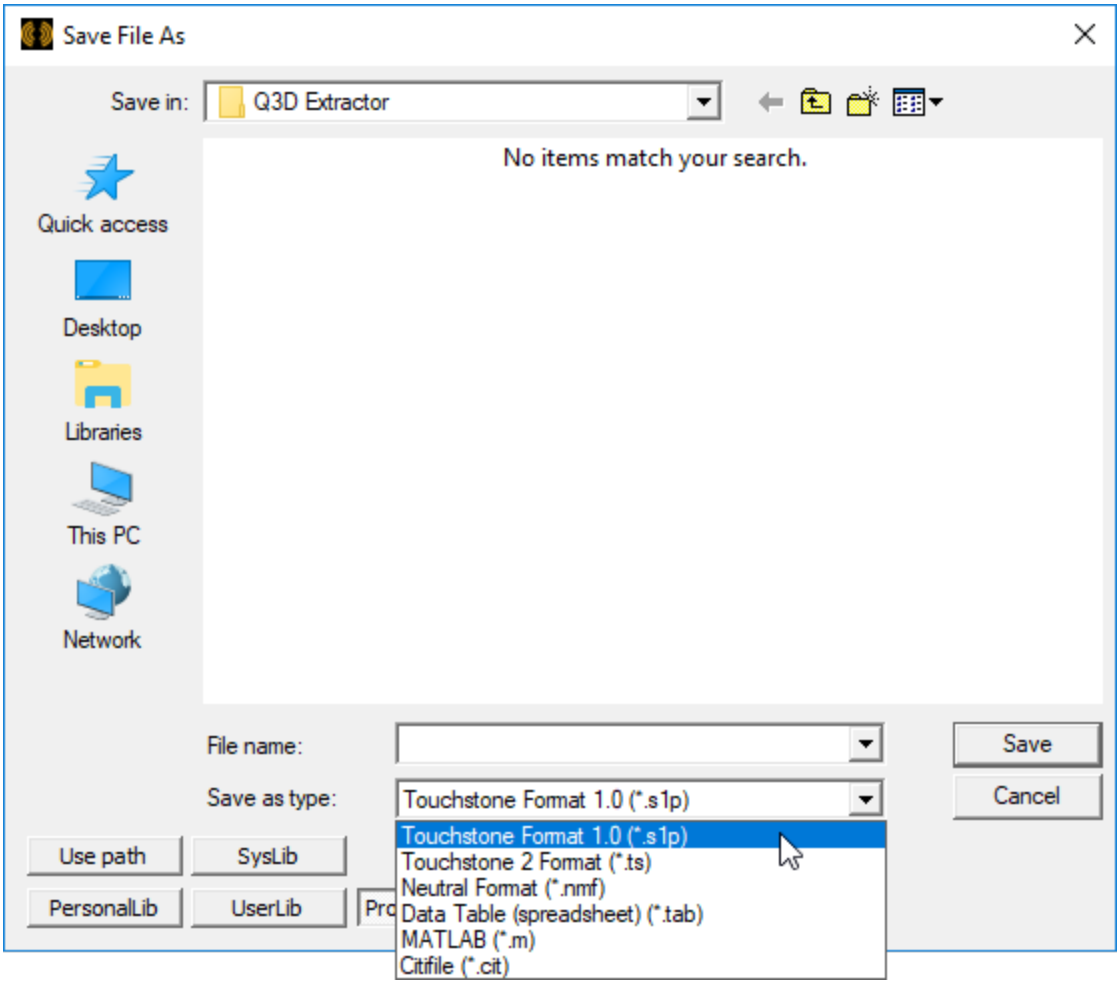
Exporting Macro Model

Creating an NPort Model

Scripting for Network Data Explorer

Exporting SYZ Data

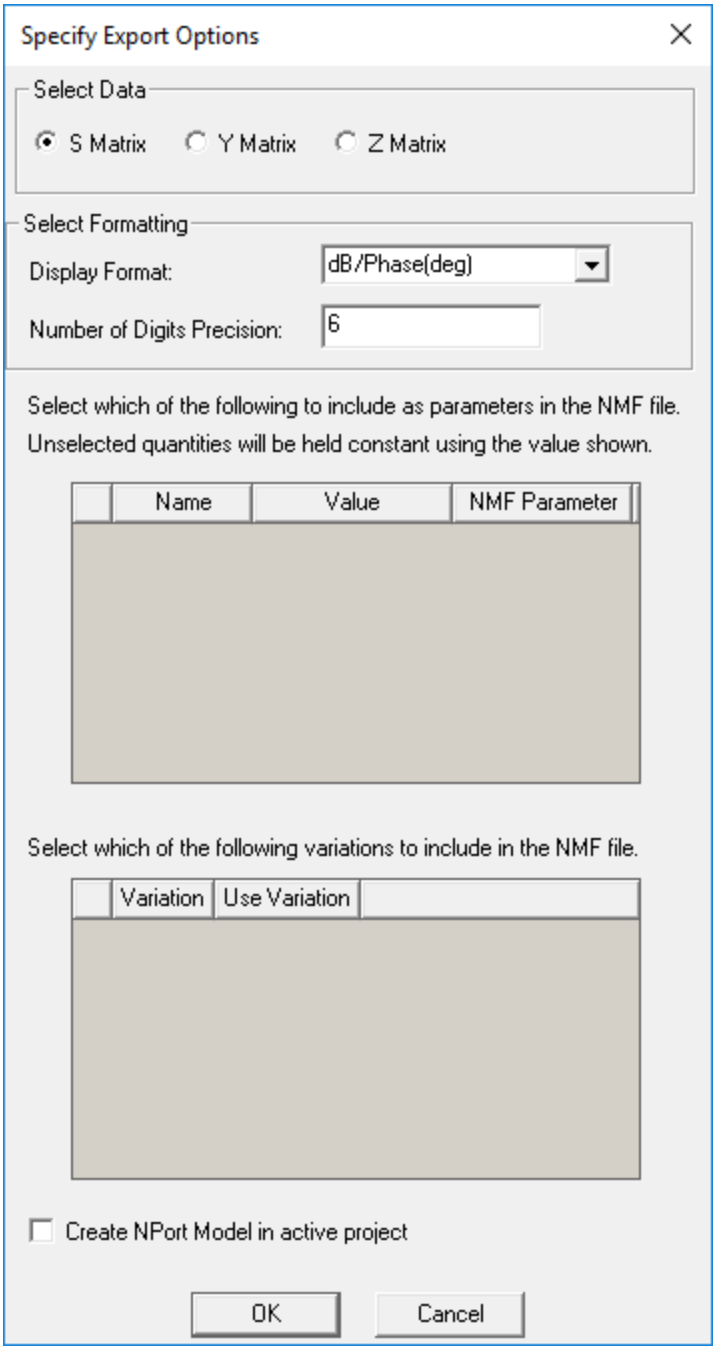
To export SYZ data from within Network Data Explorer, click the **Save As** icon  on the **NDE** ribbon. The **Save File As** window appears.

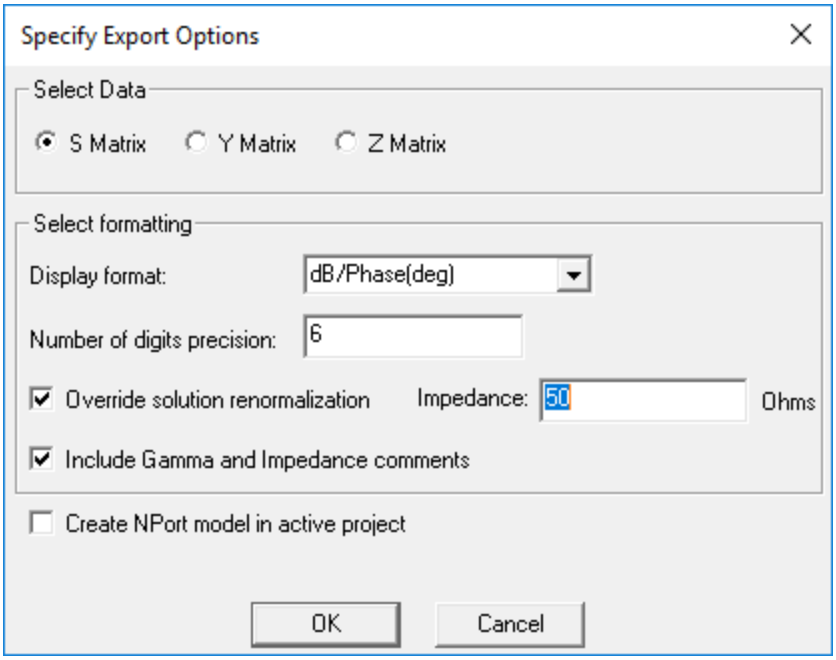


You can export data in any of six file types:

- 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:

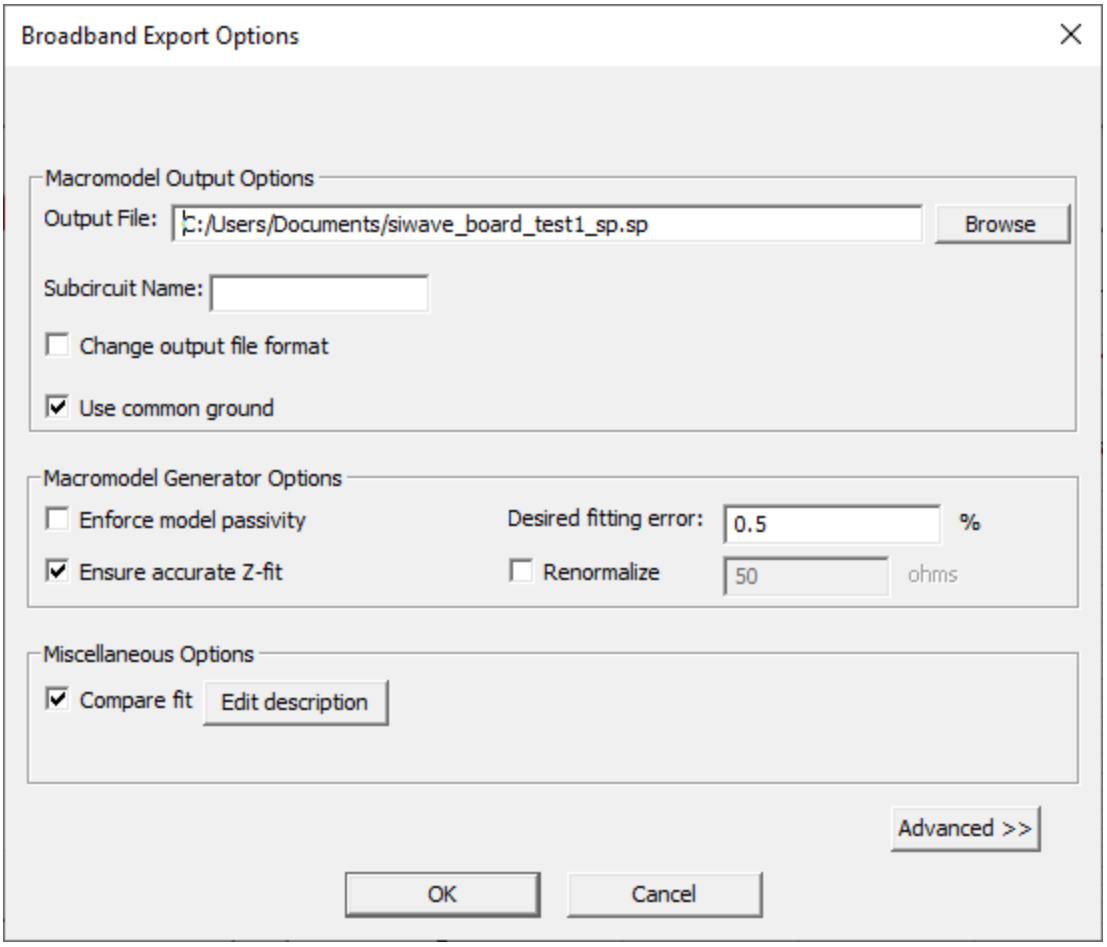
- Select from **S Matrix**, **Y Matrix**, and **Z Matrix** data.
- Select the **Display Format**.

Some types allow you to create an NPort Model in the active project.

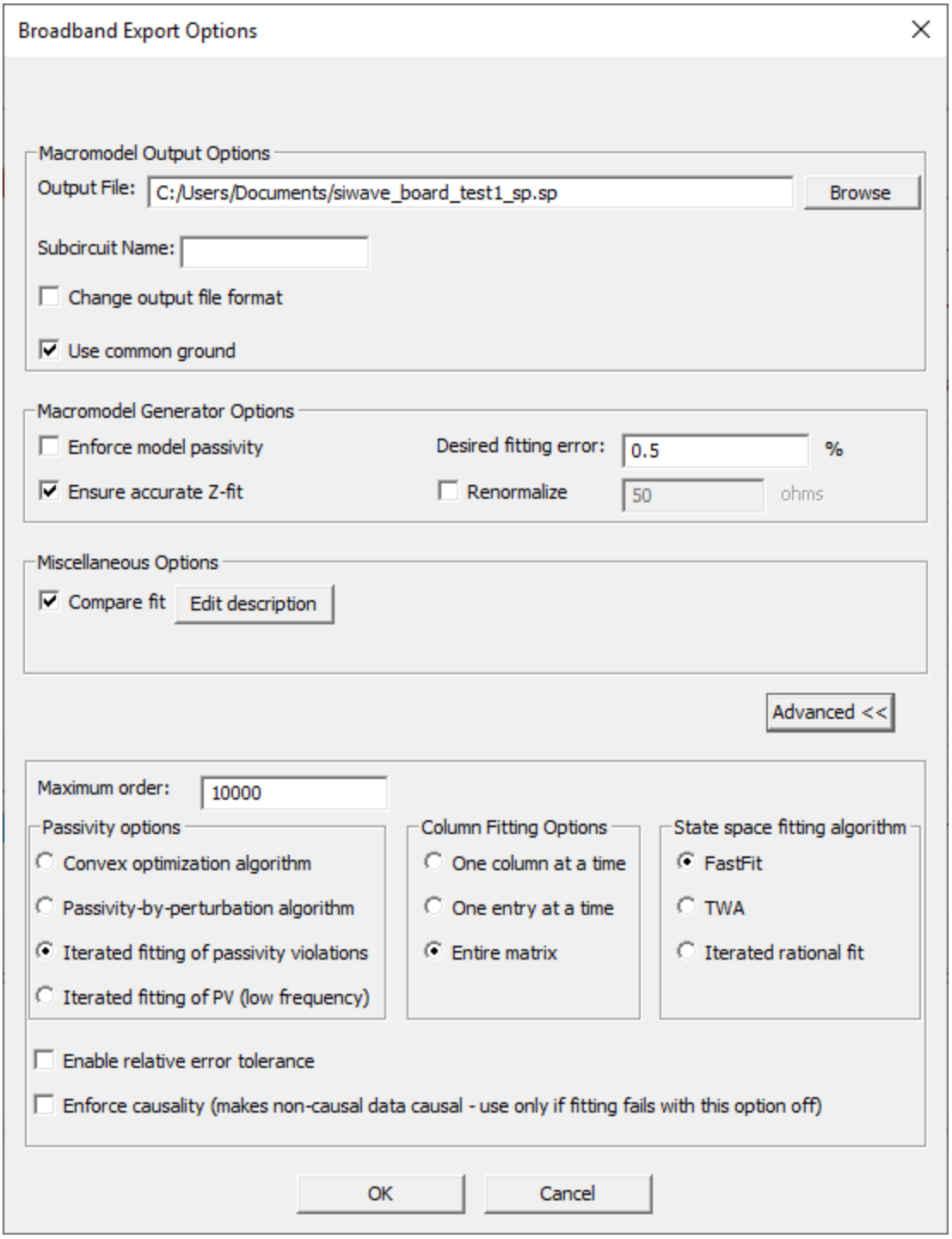
Exporting Macro Model

Network Data Explorer lets you export macro model data. 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.

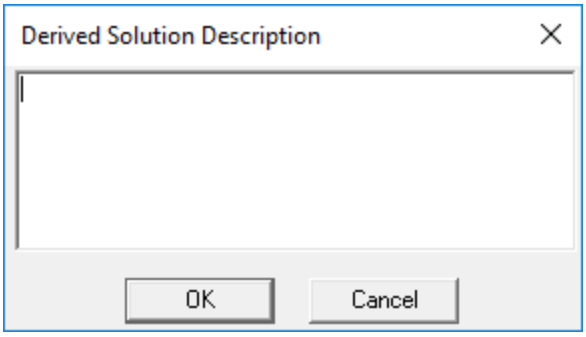
Macromodel Generator Options include:

- **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.
  - **Convex optimization algorithm** – guarantees a passive state-space realization, but is very slow and memory-intensive. Not practical for numbers of ports beyond ten.
  - **Passivity-by-perturbation algorithm** – designed for systems with a large number of ports. Less accurate than the Convex optimization method.
  - **Iterated fitting of passivity violations (IFPV)** (default) – less accurate than other algorithms but more suitable for larger numbers of ports.
  - **Iterated fitting of PV (low frequency)** – similar to IFPV while improving the fit to “Z” at DC and low frequencies. A better choice of passivity enforcement when the fit to corresponding “Z-data” is important such as power delivery, EMI/EMC applications.

    > **Note:**
    >
    > For a more detailed explanation on any of the passivity options, see the technical note section of the Circuit help.

- **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.

> **Note:**
>
> - 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.

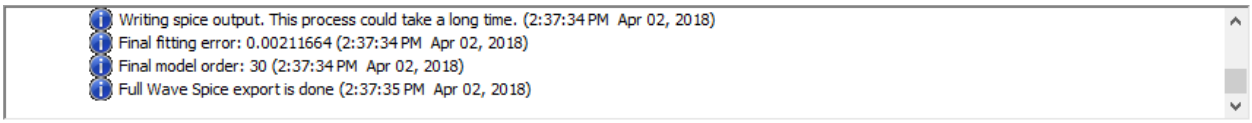
- **State Space Fitting Algorithm** – Allows you to select FastFit, TWA, Iterated rational fitting.
  - **FastFit** (default) – FastFit is the Ansys-proprietary method for state-space fitting. Network Data Explorer uses FastFit for calculating the state-space matrices from the network data. The FastFit 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.

> **Note:**
>
> 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.

> **Note:**
>
> Broadband models are built from a rational-function approximation of 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.

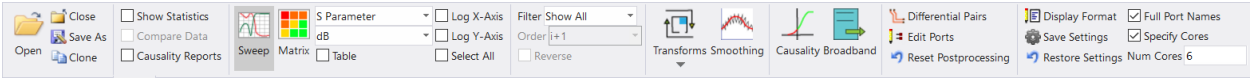
When there is no design available, export functionality and scripting are not available.

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 Ribbon

The **NDE** ribbon provides access to many of the Network Data Explorer's functions.



In this section, you will learn about:

Data Sources

Setting Display Format

Displaying Full Port Names

Saving or Resetting Default Settings

Smoothing All Frequencies

Cell Filtering

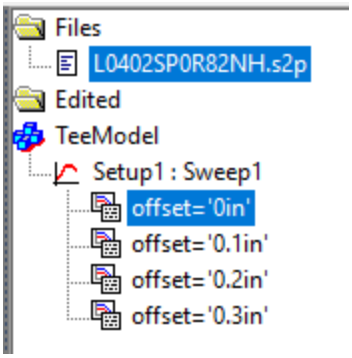
Editing Port Properties

Defining Differential Pairs and Displaying Mixed Mode Parameters

Resetting All Port Properties

Network Data Explorer Data Sources

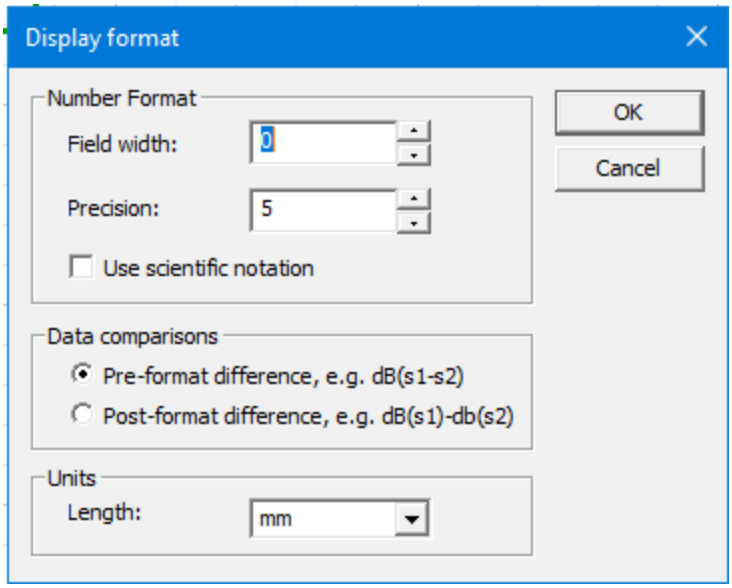
Network Data Explorer allows you to easily view all data sources in the **Network Data Selection** pane.



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.

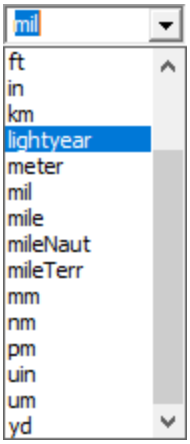


The **Number Format** options allow you to specify **Field Width** (the minimum number of characters used to display a number) and **Precision** (the number of decimals to display). You can also check the **Use scientific notation** check box, if desired.

The **Data comparisons** options allow you to choose Post-format difference or Pre-format difference.

- **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.
- **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.

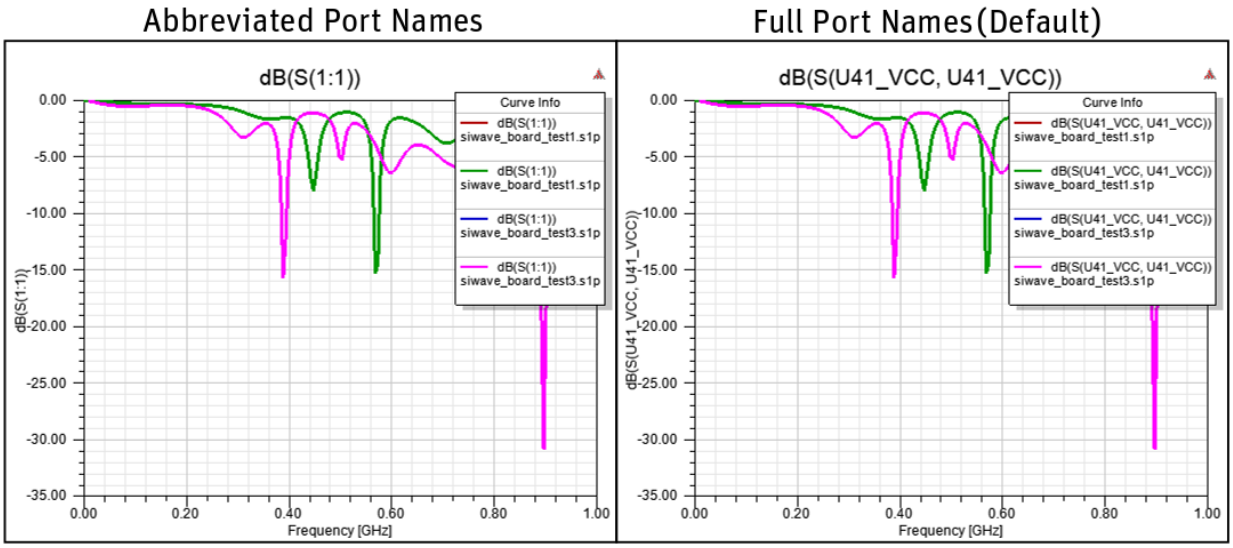
The **Units** option allows you to specify the **Length** unit (the unit used to display and interpret length values). The default is mm.



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.

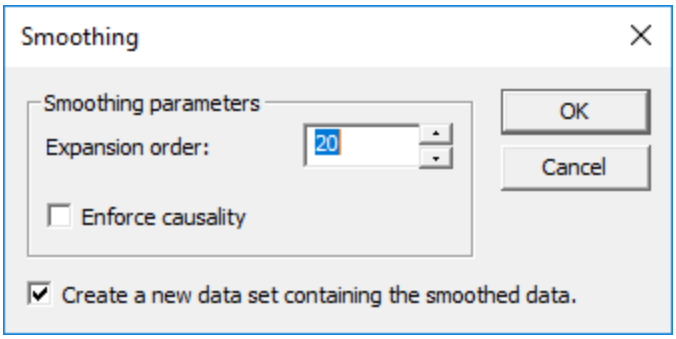
Smoothing

To access data smoothing options:

1. Click the **Smoothing** icon on the **NDE** ribbon.

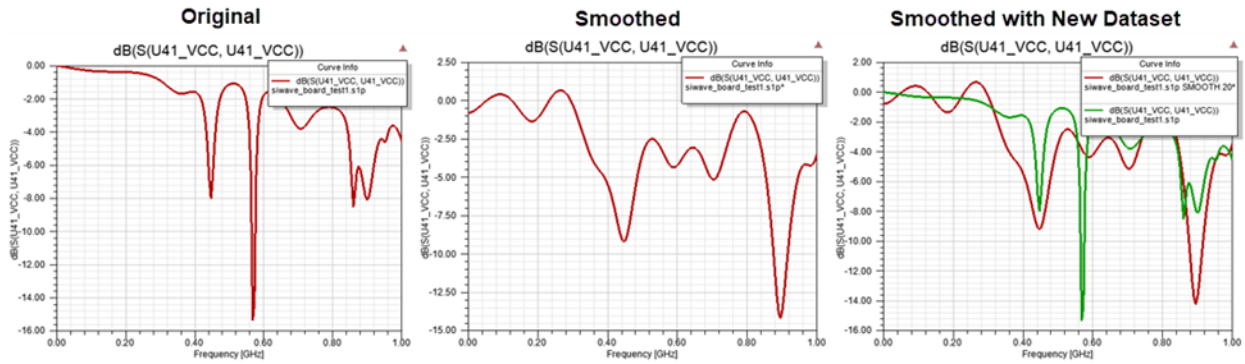
   

   The **Smoothing** window appears.



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** check box.
4. If desired, check the **Create a new data set containing the smoothed data** check box. If selected, the smoothed data appears alongside the original data.
5. Click **OK**.

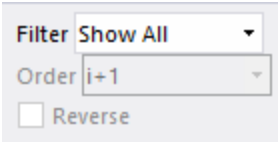
The Data View Pane updates. The following image shows a plotted result with **Create a new data set containing the smoothed data** unchecked and checked.



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:

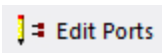
- **Show all** – display all available cells. There are n-squared choices.
- **Return loss** – show S(i, i). There are n choices.
- **Insertion loss** – show S(i, i+1). There are n choices.
- **Lower triangle** – show S(i, j) for all j < i. There are n(n-1)/2 choices.

Three pin arrangements are recognized:

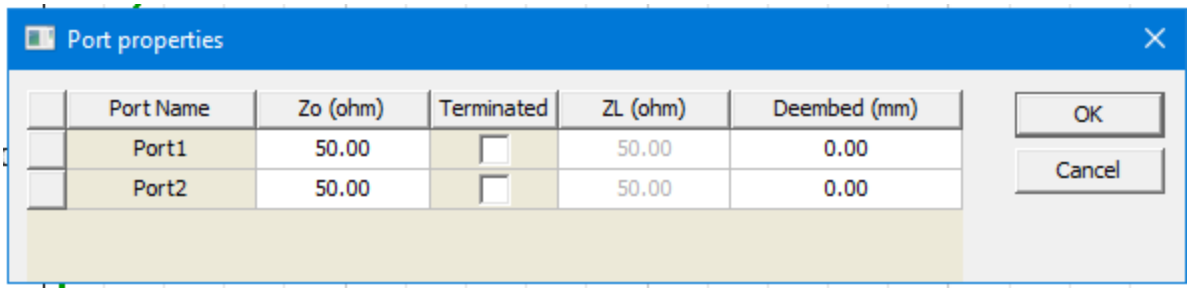
- S(i, **i+1**) and its **Reverse order**, S(i, i-1)
- S(i, **i+n**) and its **Reverse order**, S(i, i-n)
- S(i, **2*n-i+1**) and its **Reverse order**, S(i, 2*n-i-1)

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.



> **Note:** For HFSS Driven Terminal designs, selecting the solution data will disable this ribbon button. The user must make port/differential pair changes directly in the HFSS Design and re-import the solution.

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 check box 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.

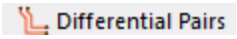
To save changes, click **OK**.

Displaying Mixed-Mode Parameters using Differential Pairs

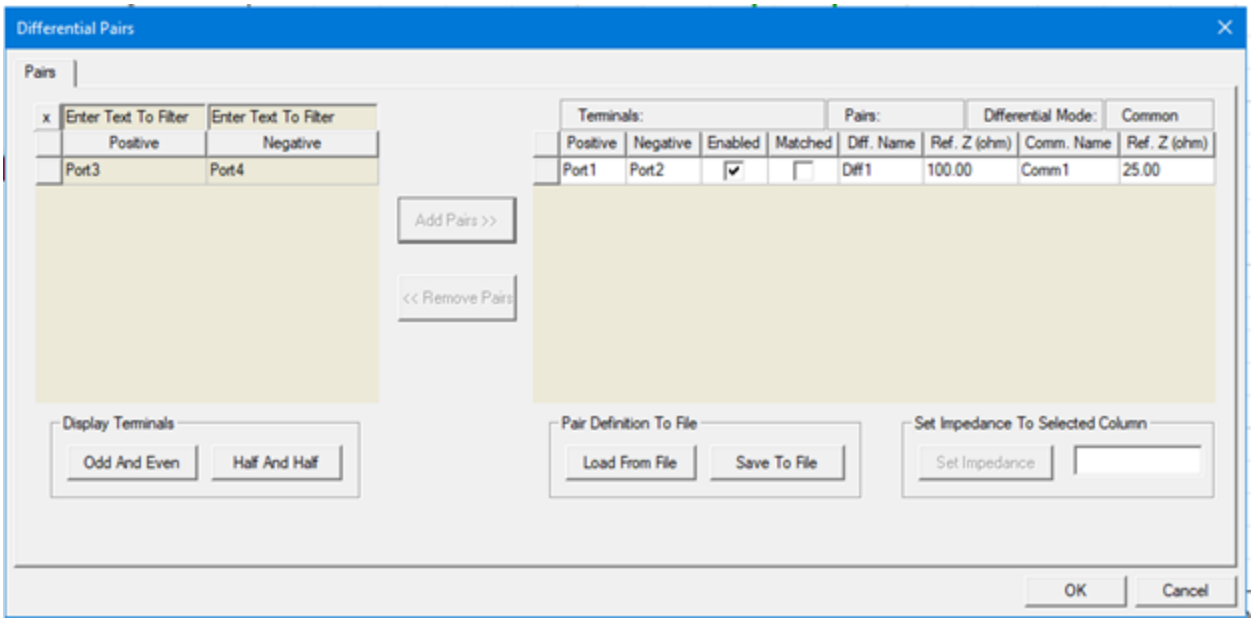
Network Data Explorer displays mixed-mode parameters when differential pairs are both defined and activated.

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**.

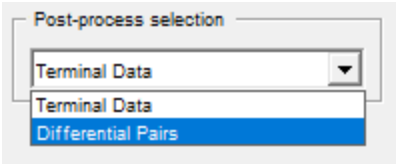


The **Differential Pairs** window appears.



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.

For HFSS Driven Terminal designs, selecting the solution data will disable this ribbon button. The user must make port/differential pair changes directly in the HFSS Design and re-import the solution. When the HFSS Driven Terminal Design has differential pairs defined, NDE shows this drop down menu and you can change between showing reports for Terminals or Differential Pairs.Select **Differential Pairs** from the **Post-process selection** field to view mixed-mode parameters.



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.

> **Note:** For HFSS Driven Terminal designs, selecting the solution data will disable this ribbon button. The user must make port/differential pair changes directly in the HFSS Design and re-import the solution.

Data View Pane Context Menus

The Data View pane presents different right-click menu options, depending on the context. Some commands are the same as those on the NDE ribbon. Others appear only in the context menus.

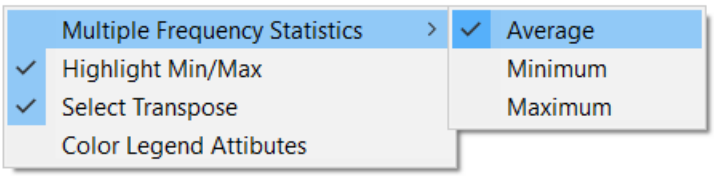
Network Data Explorer commands unique to the context menus are:

- 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:

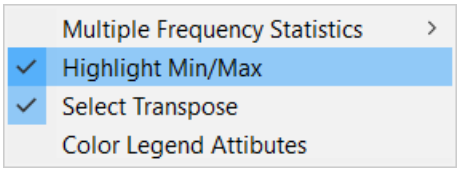
- **Average** – display the average of the matrix values across selected frequencies.
- **Minimum** – display the minimum matrix values across selected frequencies.
- **Maximum** – display the maximum matrix values across selected frequencies.

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.

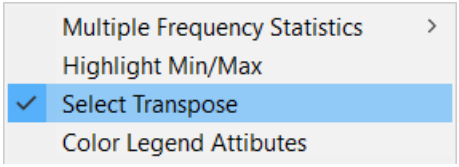
This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



Select Transpose

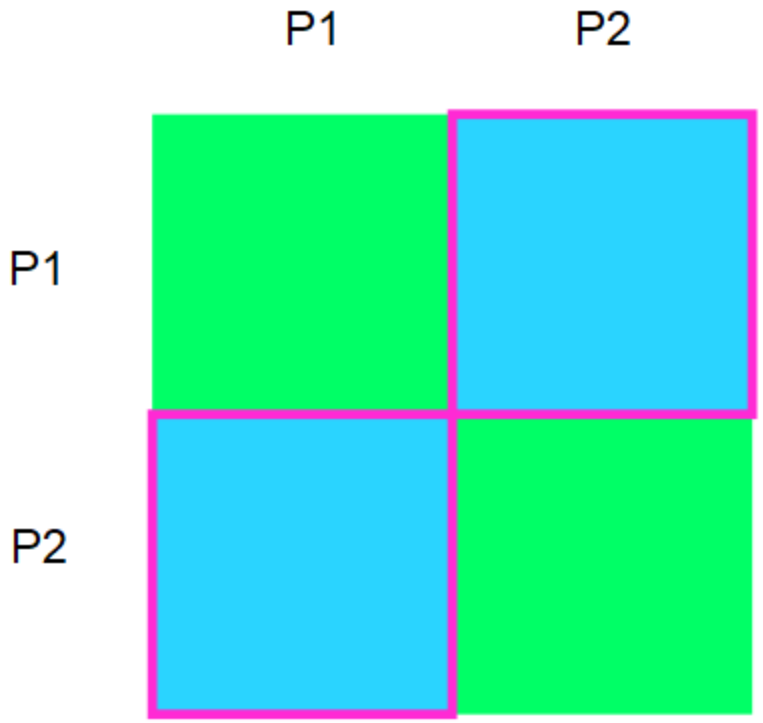
The **Select Transpose** menu option determines whether transpose cells are highlighted along with selected cells.

This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



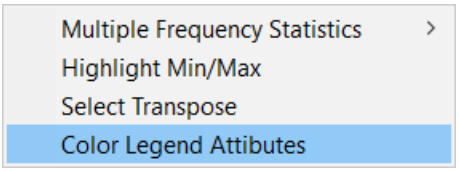
Transpose cells are highlighted in Table or Plot view, as shown below.

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

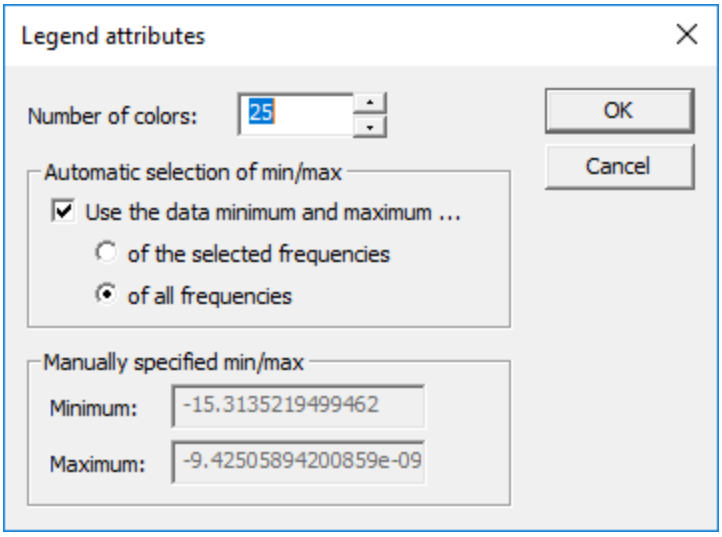


Color Legend Attributes

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 plot's legend to open the **Legend Attributes** window.



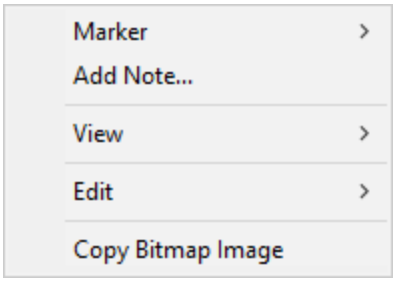
Options include:

- **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** check box 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.

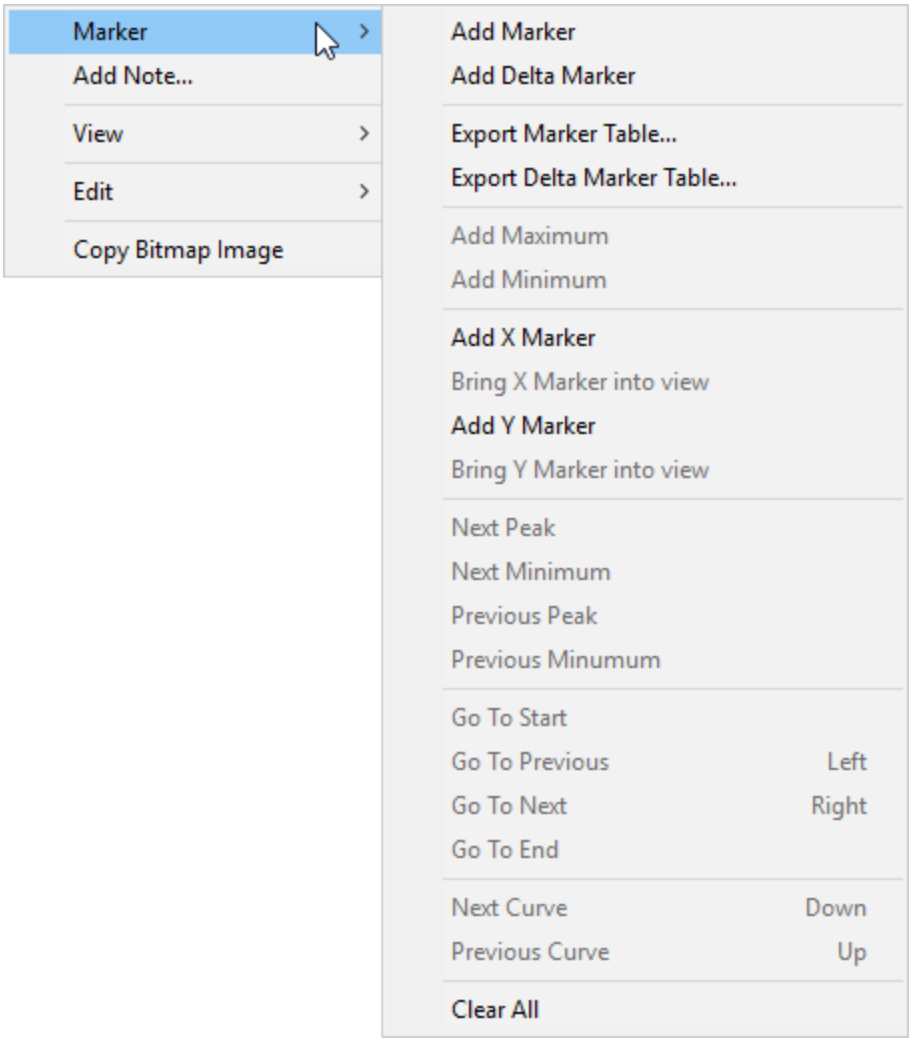
Using a standard range across all frequencies permits you to quantitatively compare plots, and ndExplorer remembers legend settings for each data-type and display-format pair.

Matrix Entries Plot Menu

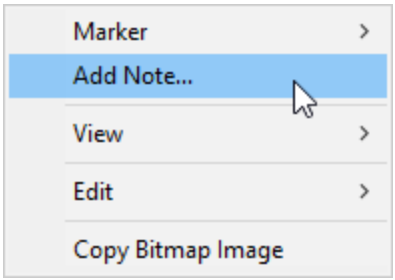
On the **Matrix** plot, several new right-click menu options appear.



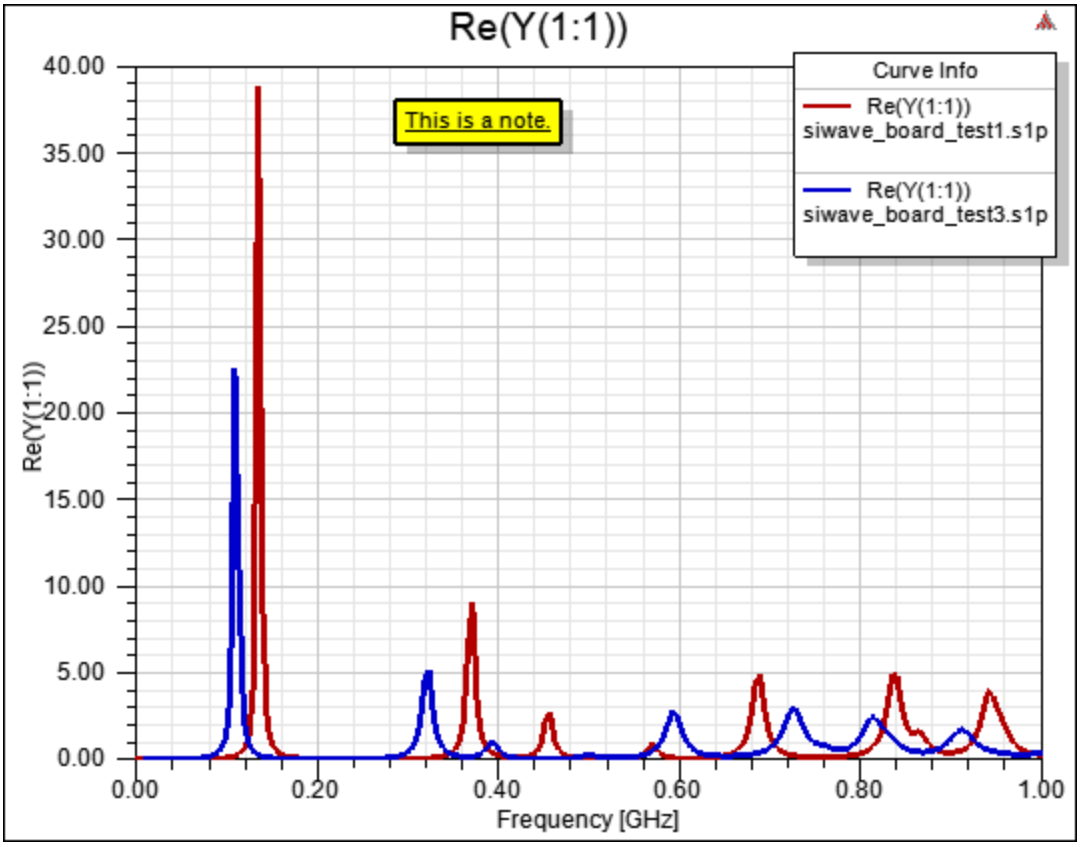
The **Marker** sub-menu provides commands for adding markers to plots.



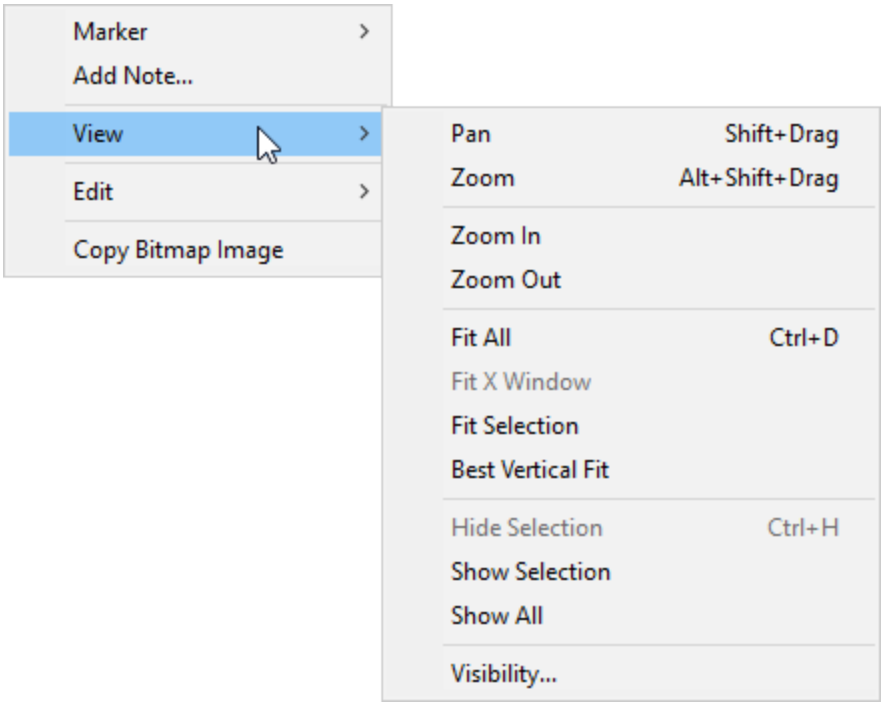
The **Add Note** menu option allows you to add a note to the plot.



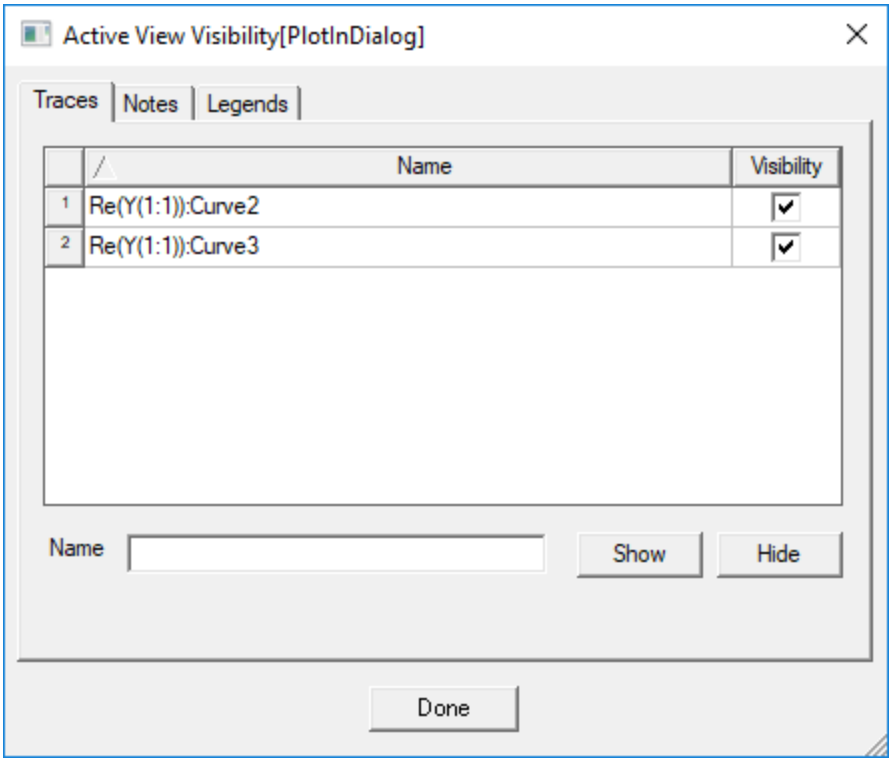
The note appears at the location you right-clicked. You can click and drag the note to a new location, or double-click the note to change its color and font.



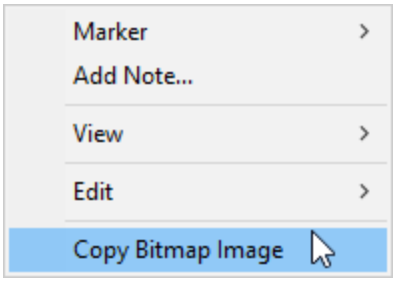
The **View** sub-menu provides commands for viewing, panning, zooming, and fitting elements to your plot.



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:

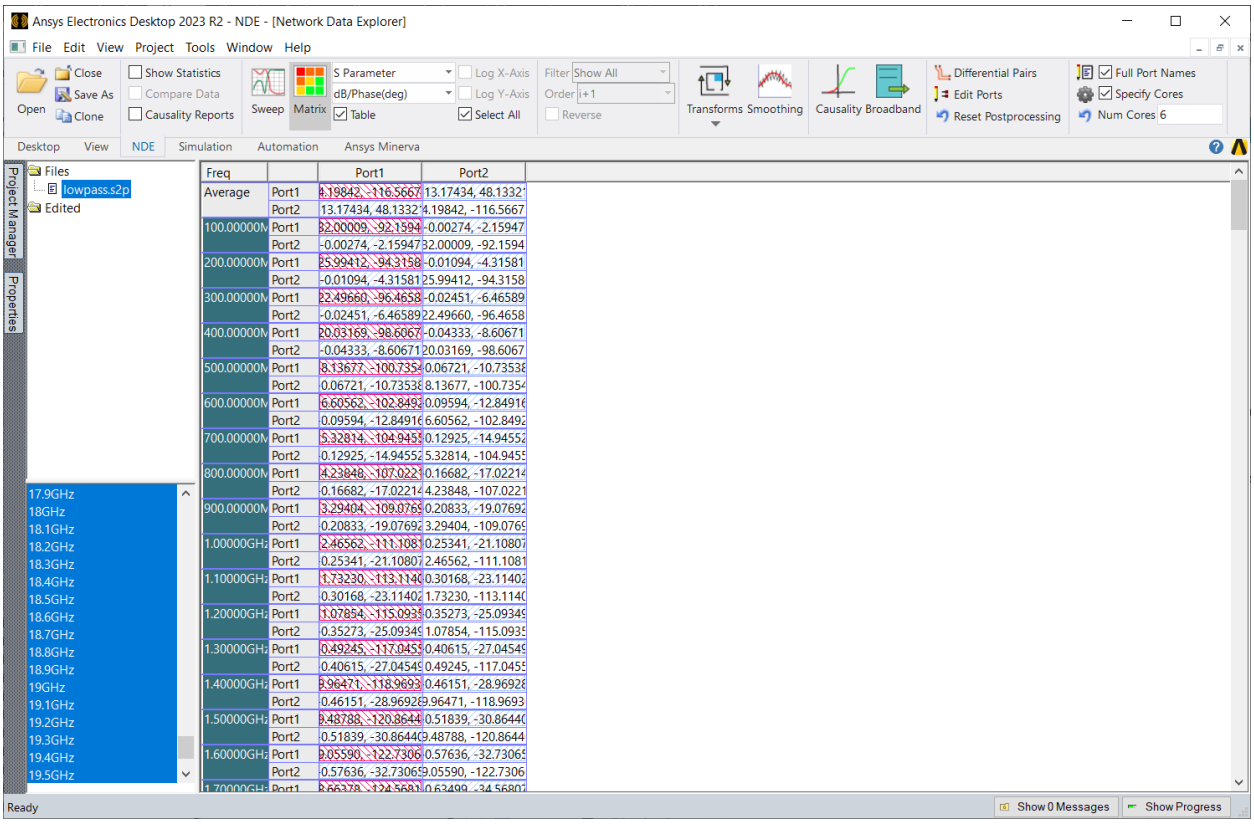
- Viewing the S, Y, or Z Matrix for a Frequency
- Viewing a Color-coded Matrix Plot
- Displaying a Cell Graph Across All Frequencies

- Displaying Matrix Statistics by Frequency
- Displaying Individual Statistics for All Frequencies
- Creating a Statistics Plot
- Comparing Variations

Viewing a Matrix Table

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.

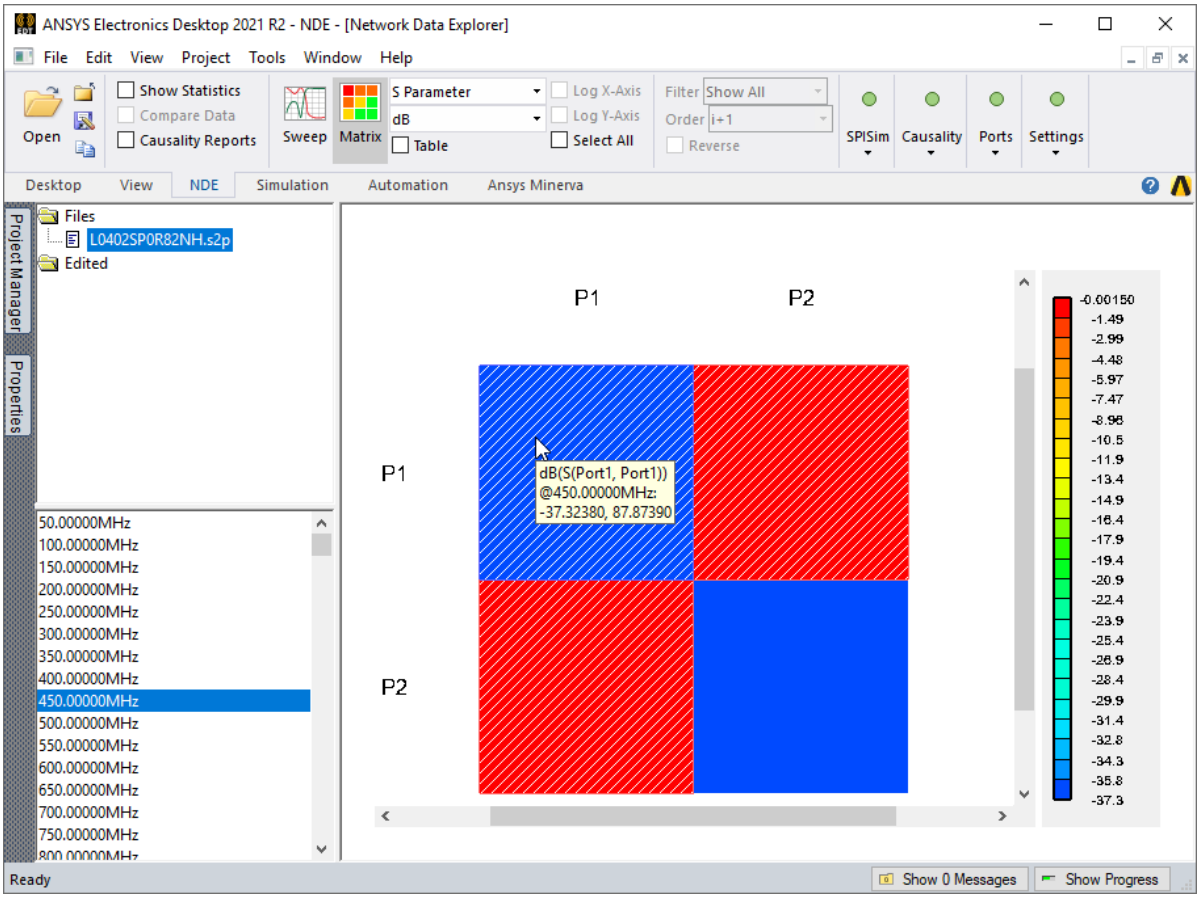
Selected cells appear in solid blue. If Select Transpose is enabled, the transpose is highlighted in blue as well.

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 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.

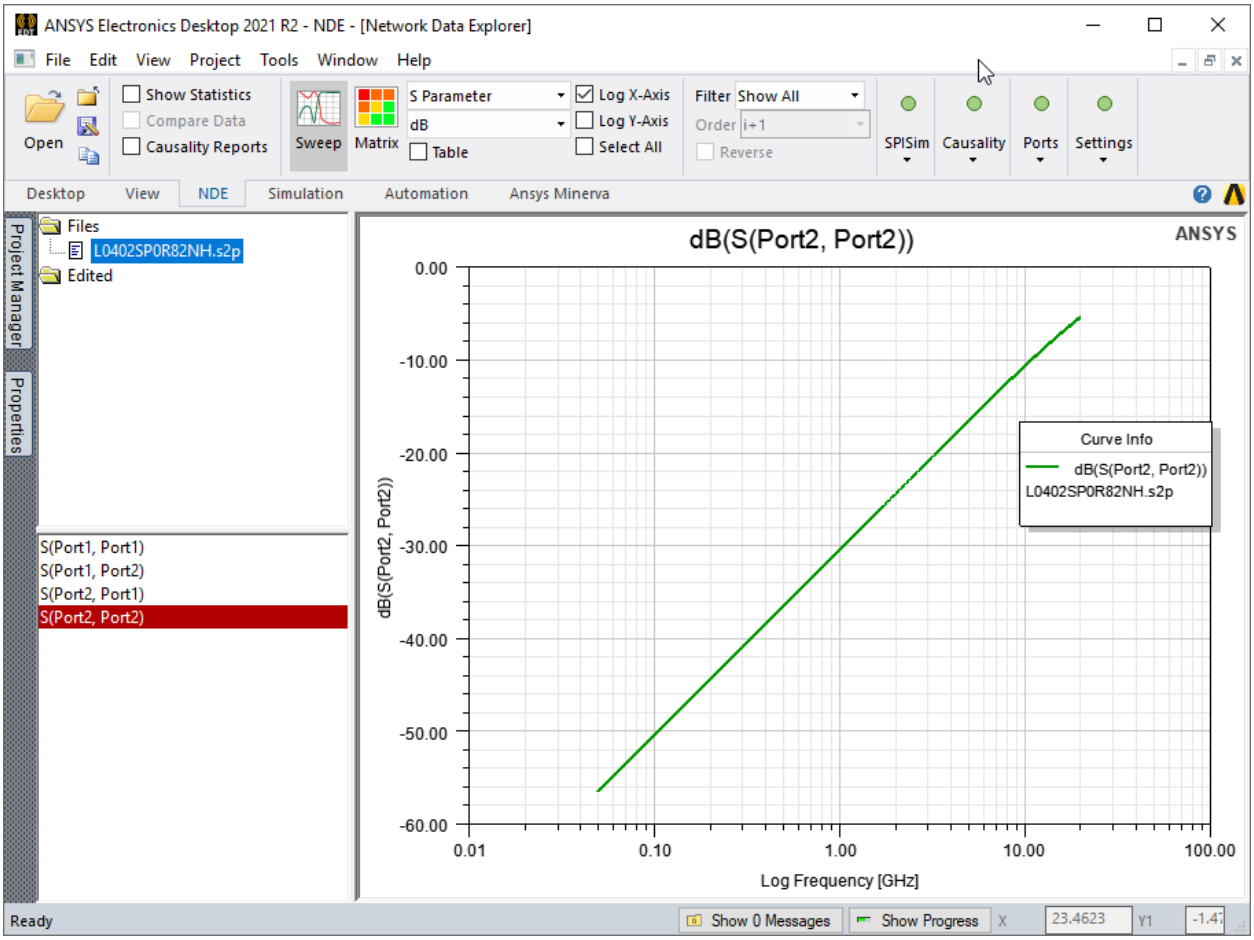
Click any cell to select it. Selected cells appear with a pink outline. If Select Transpose is enabled, the transpose is selected in pink as well.

Double-click any cell to view a matrix cell plot in which all frequency values for that matrix cell are displayed as a graph.

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.



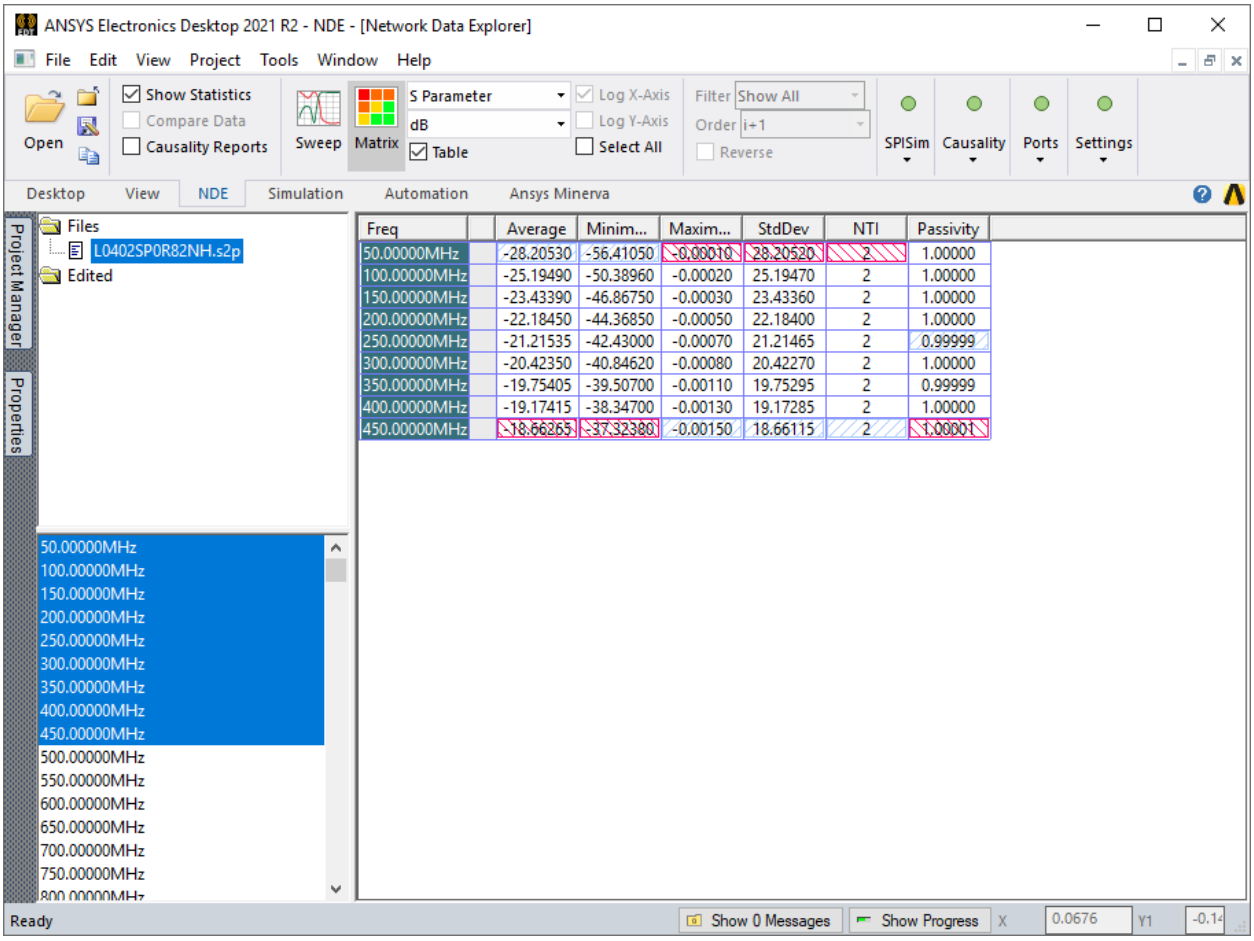
5. To add a log scale to the Y-axis, click **Log Y-Axis** on the **NDE** ribbon.
6. Right click to see a right click menu with other commands.

Displaying Matrix Statistics by Frequency

Network Data Explorer can display various statistical measurements.

1. On the **NDE** ribbon:
   a. Click **Matrix**.
   b. Click **Table**.
   c. 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.

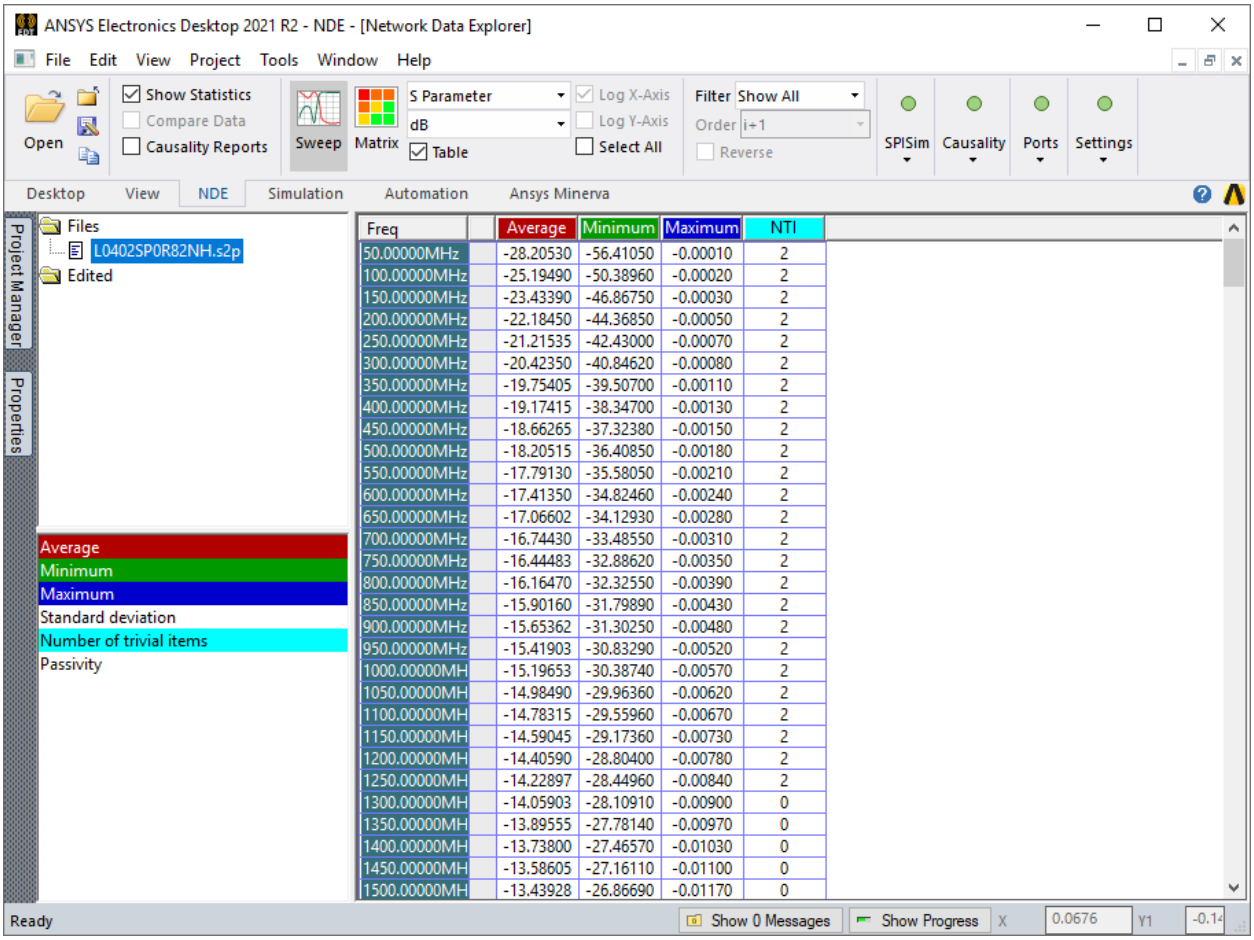
Hover the cursor over a cell to view information about it.

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:
    a. Click **Sweep**.
    b. Click **Table**.
    c. 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).

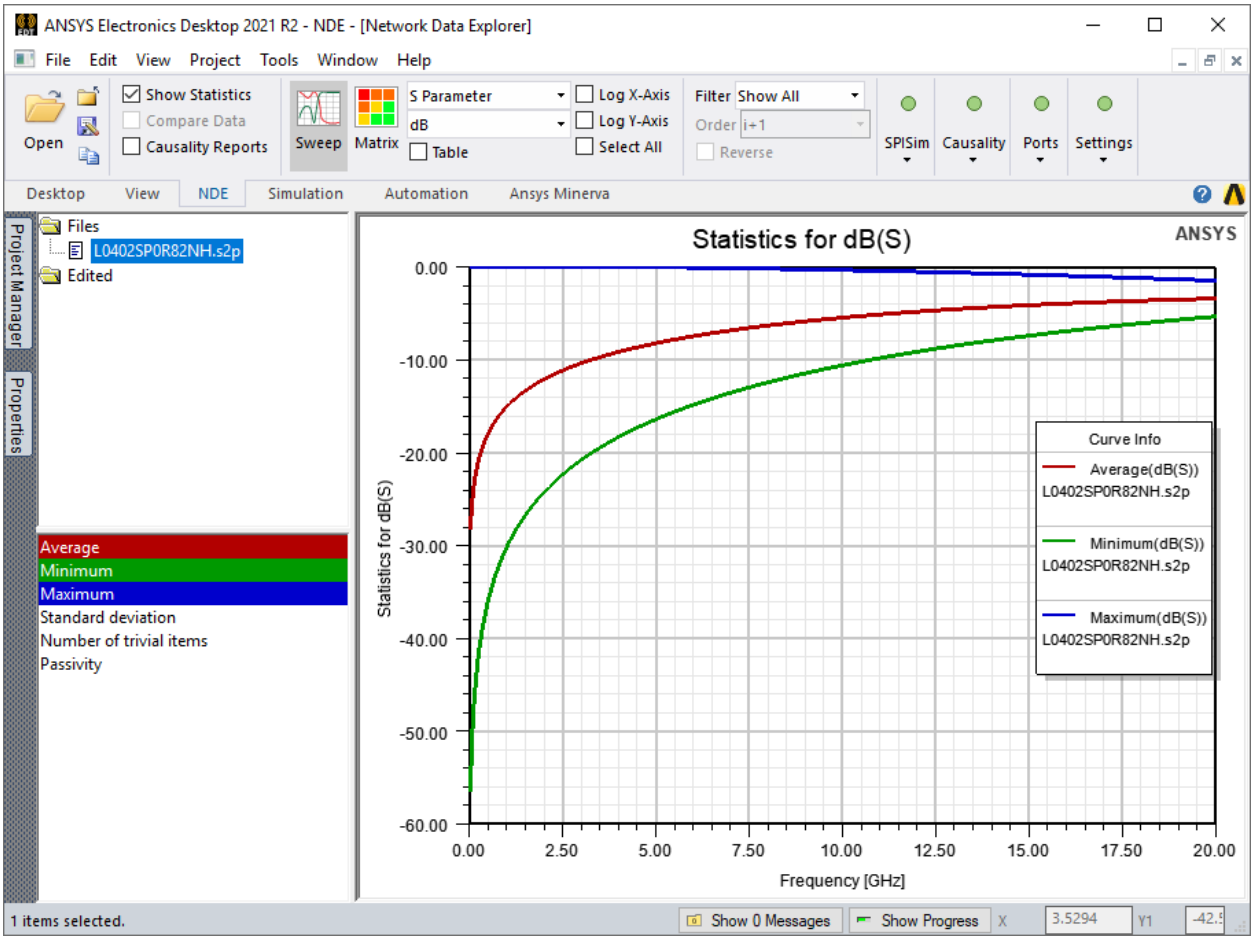
For S-parameters, the **Number of Trivial Items (NTI)** includes all values of 0 and 1. For other data and data comparisons, only values of 0 are counted as trivial.

Creating a Statistics Plot

Network Data Explorer can display a graph of selected statistical measures across all frequencies.

1. On the **NDE** ribbon:
    a. Click **Sweep**.
    b. 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 cloned.* Then, you can compare the original data set to the clone.

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.

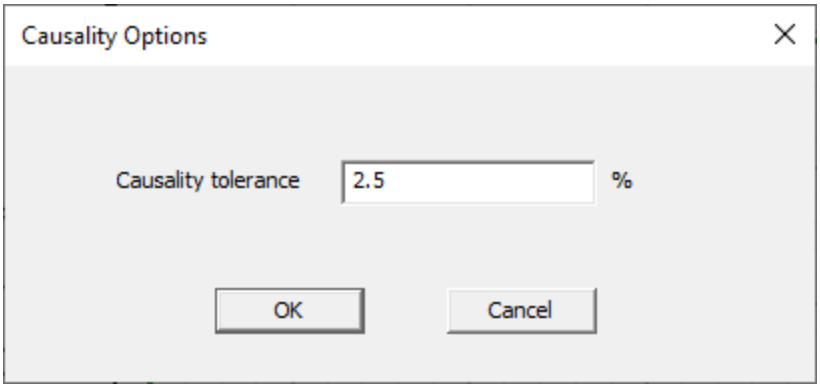
If **dB** format is shown, you can either subtract values before or after applying the dB function. See Display Format to make this choice.

Causality Checking and Plots

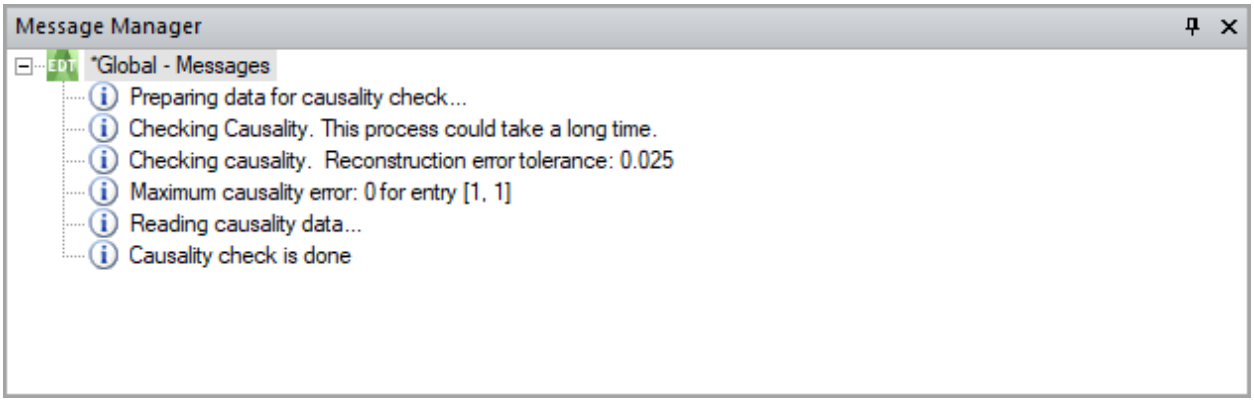
Network Data Explorer can perform a causality check on S-parameter data from any source (solution or file), and provide plots of the results in various formats.

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.

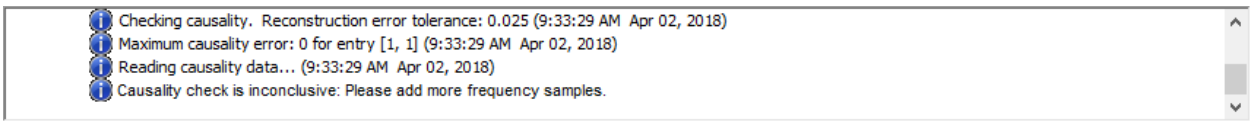


When the check completes, the Message Manager updates to display a summary of results.

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.

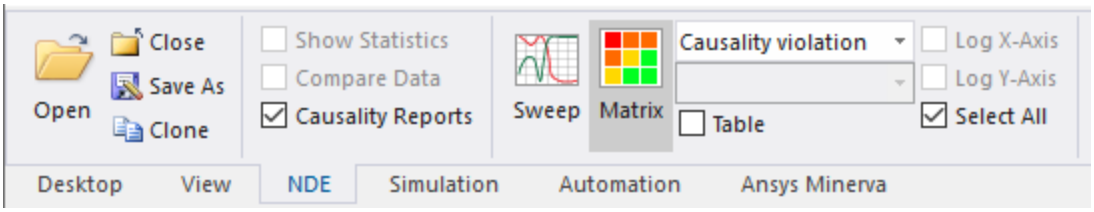
Maximum Causality Error – The maximum causality error for all port pairs and all frequencies, along with the matrix indices (port numbers) where the maximum noncausality occurs. A noncausal response is one where all matrix entries can be conclusively analyzed, and at least one entry exceeds the causality 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. Network Data Explorer will also report the data set as inconclusive if any cells are inconclusive, even if other entries exhibit causality violations.

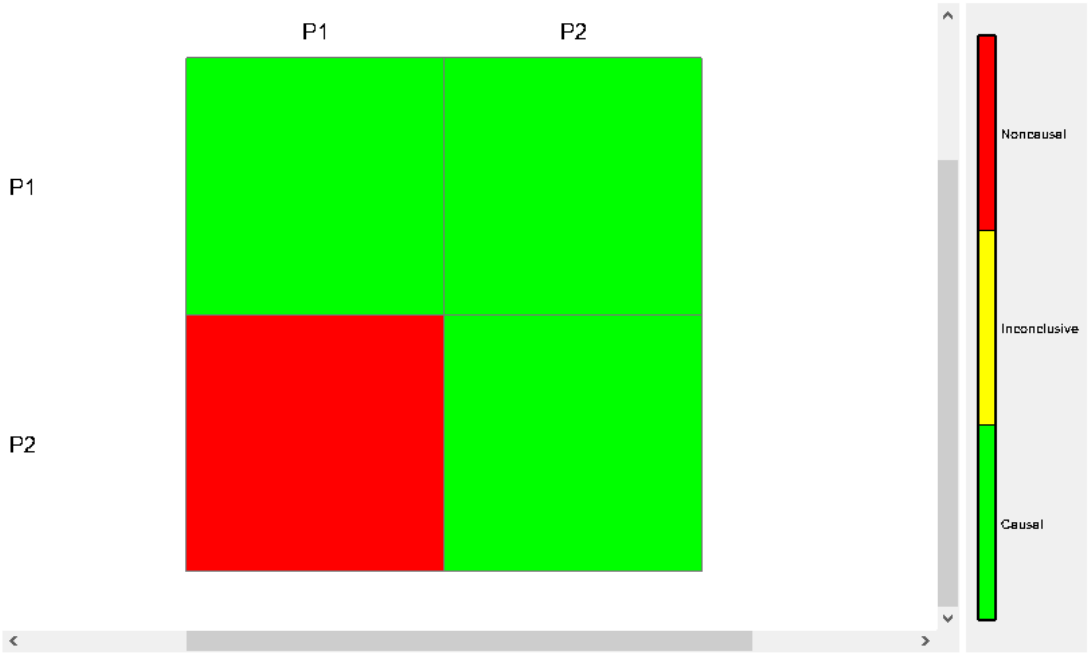


To see the plotted results of the causality check:

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.



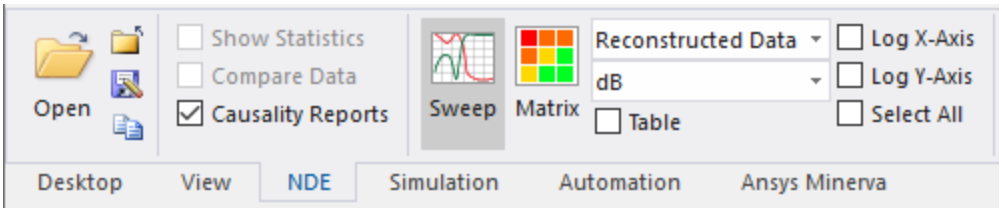
In this plot, the cells go from (Port 1, Port 1) at the upper-left area to (Port N, Port N) at the lower-right corner. The result shows the causality over all frequencies in the data. In this example, the matrix is asymmetric, so that S12 is noncausal, while S11, S21, and S22 are causal.

To see the details for each frequency, click the **Table** check button on the **NDE** ribbon.

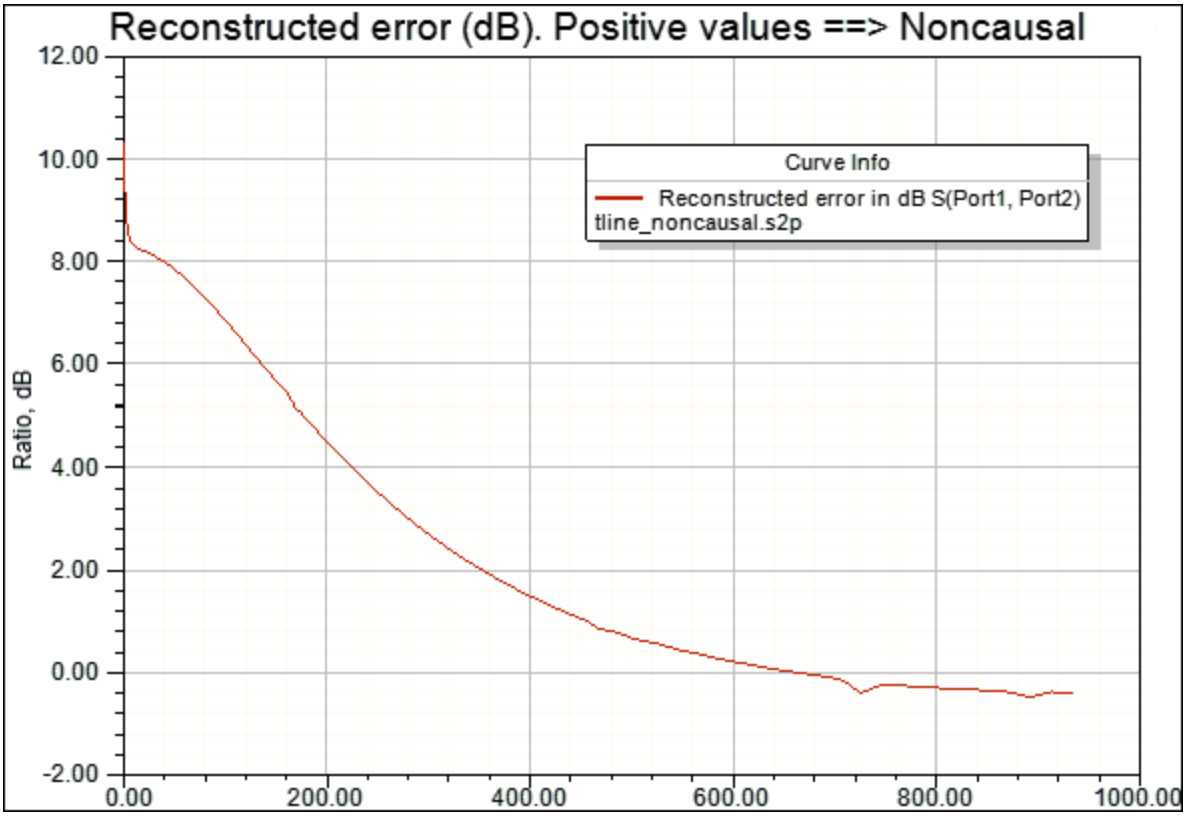
Freq		Port1	Port2	Port1	Port2
0.10000GHz	Port1	0.01715, 0.00000	0.00050, 0.00000	0.01493, 0.00000	0.00098, 0.00000
	Port2	0.53111, 0.00000	0.00579, 0.00000	0.11471, 0.00000	0.00832, 0.00000
0.20000GHz	Port1	0.01367, 0.00000	0.00059, 0.00000	0.00736, 0.00000	0.00052, 0.00000
	Port2	0.27237, 0.00000	0.00394, 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.

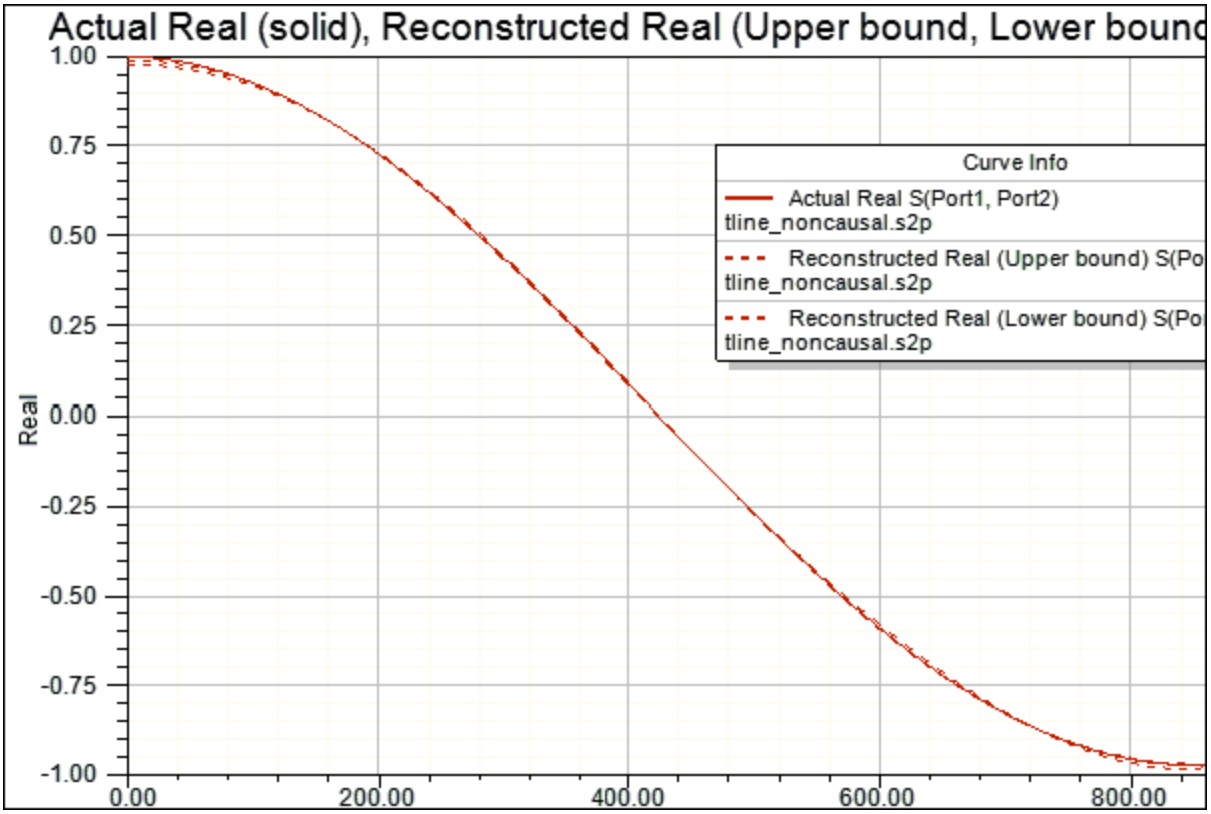


The plot appears, showing the reconstruction error at each frequency divided by the tolerance.



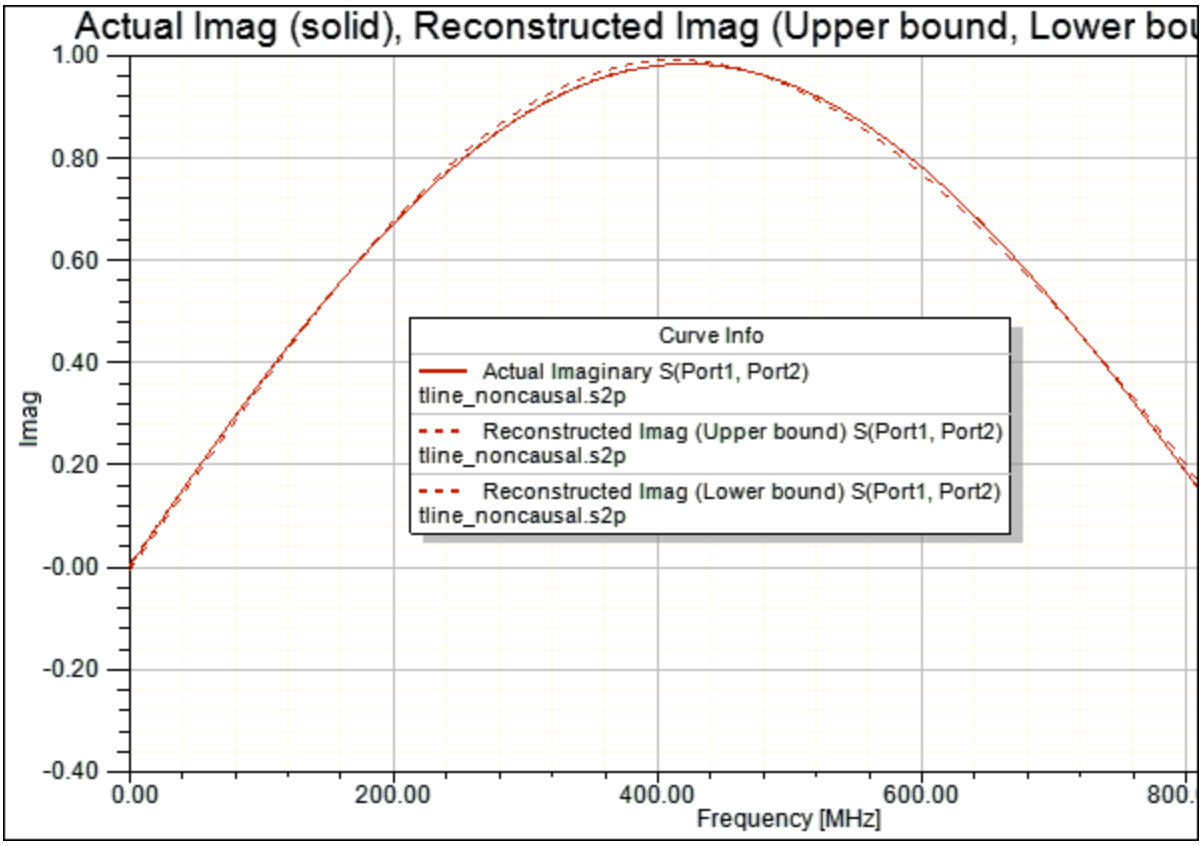
The reconstruction error ratio for parameter S12 is positive for frequencies less than about 680MHz, indicating a broad range of noncausal behavior.

4. To compare the real part of the reconstructed data to the real part of the actual data, set the **Format** to **Real**.



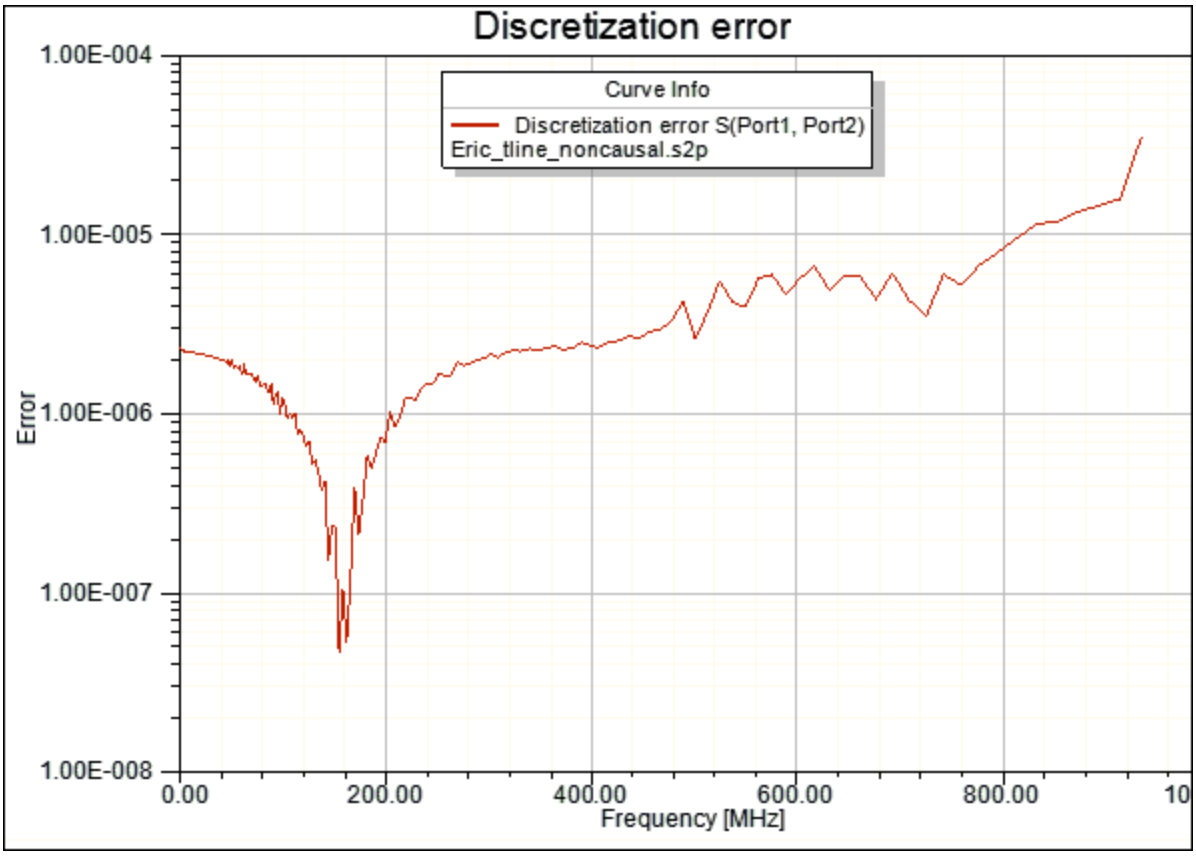
For a causal frequency response, the actual data (solid line) will be within the upper and lower bounds of the reconstruction (dotted lines) at all frequencies.

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**.



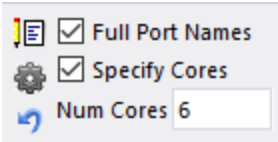
The discretization error is the error that is due to the fact that the data are available only at discrete frequencies rather than for a continuous spectrum. A discretization error near or greater than the causality tolerance renders the causality check inconclusive. Data at more frequencies could reduce 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).

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.

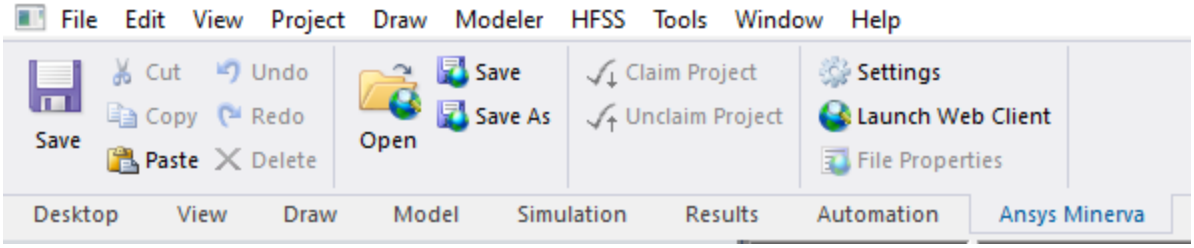
For best performance, disable hyper-threading on your computer. When hyper-threading is enabled, the number of cores includes the physical cores and an equal number of logical cores. With hyper-threading disabled, the display shows only half the number of physical cores.



To disable multithreading, uncheck the check box.

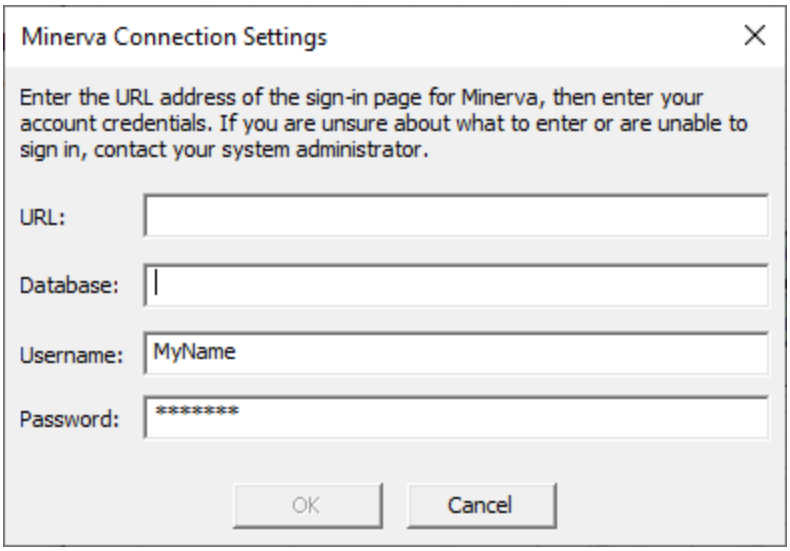
19 - Minerva Remote Storage Environment

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.

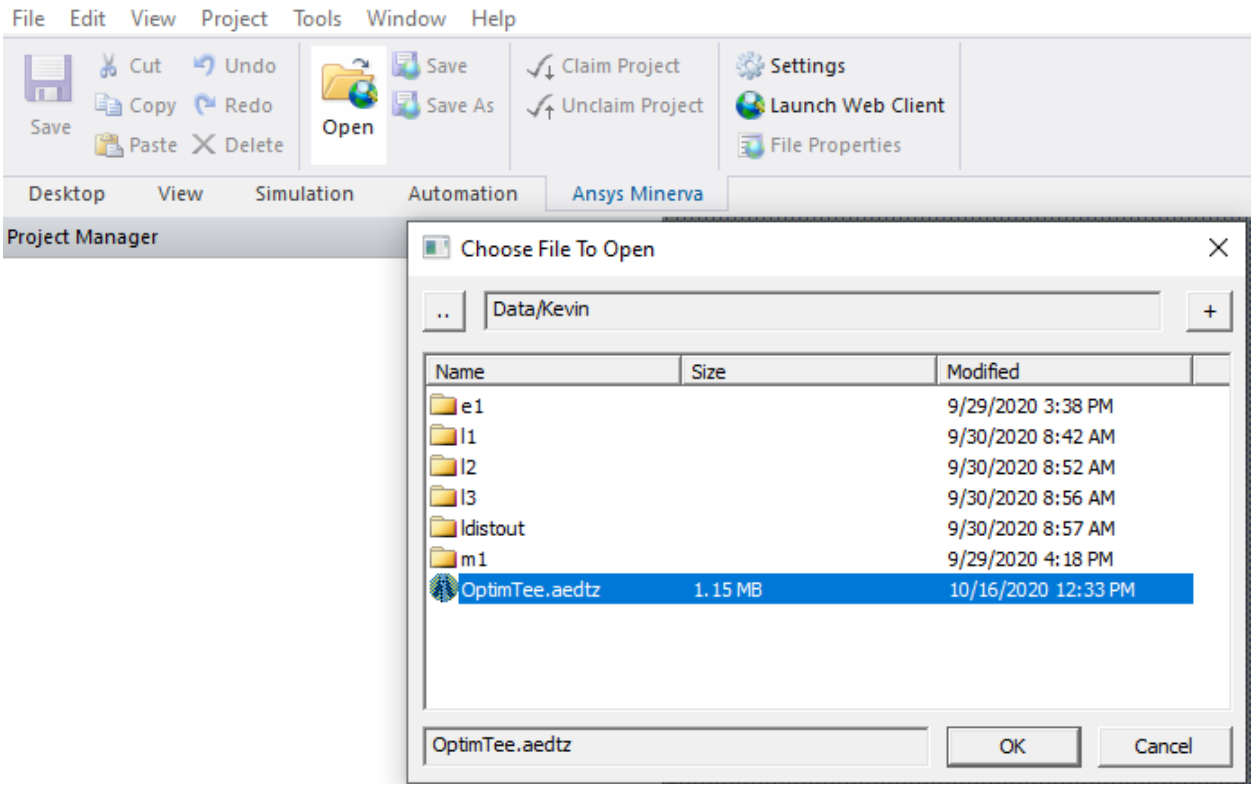
Minerva Settings



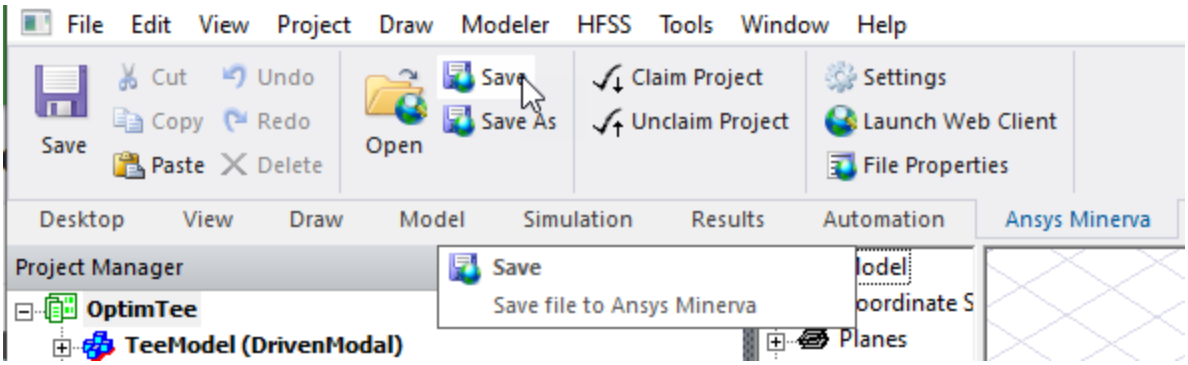
When you first login via the desktop, or to the Web Client, you must provide a Minerva URL, a database name, a user name, and a password. You will need to obtain these from your system

administrator. Once you provide the connection settings, the other Minvera commands are enabled.

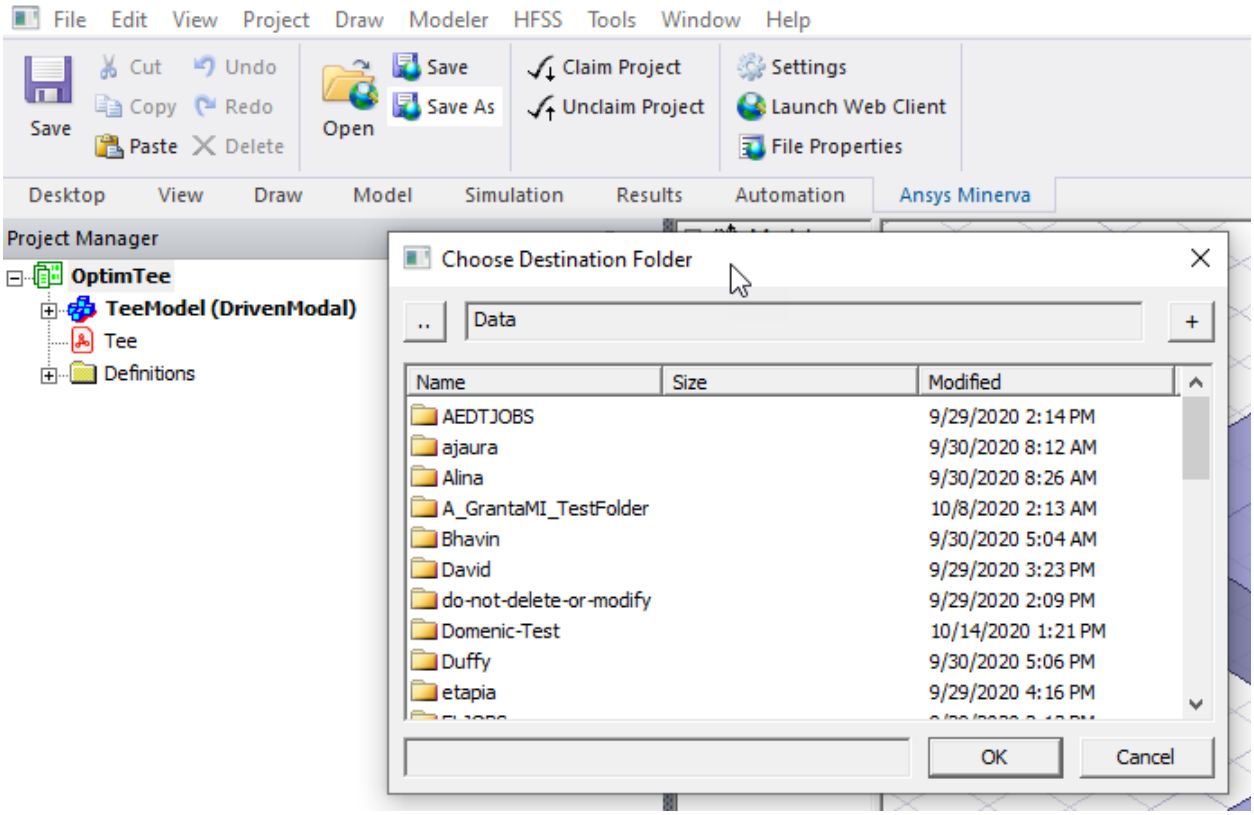
Open : opens the Choose File to Open browser that lets you select and download an archive file, extract it, and open it. The [..] button moves up one level in the file hierarchy. The [+] button creates a new folder at the current level.



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.

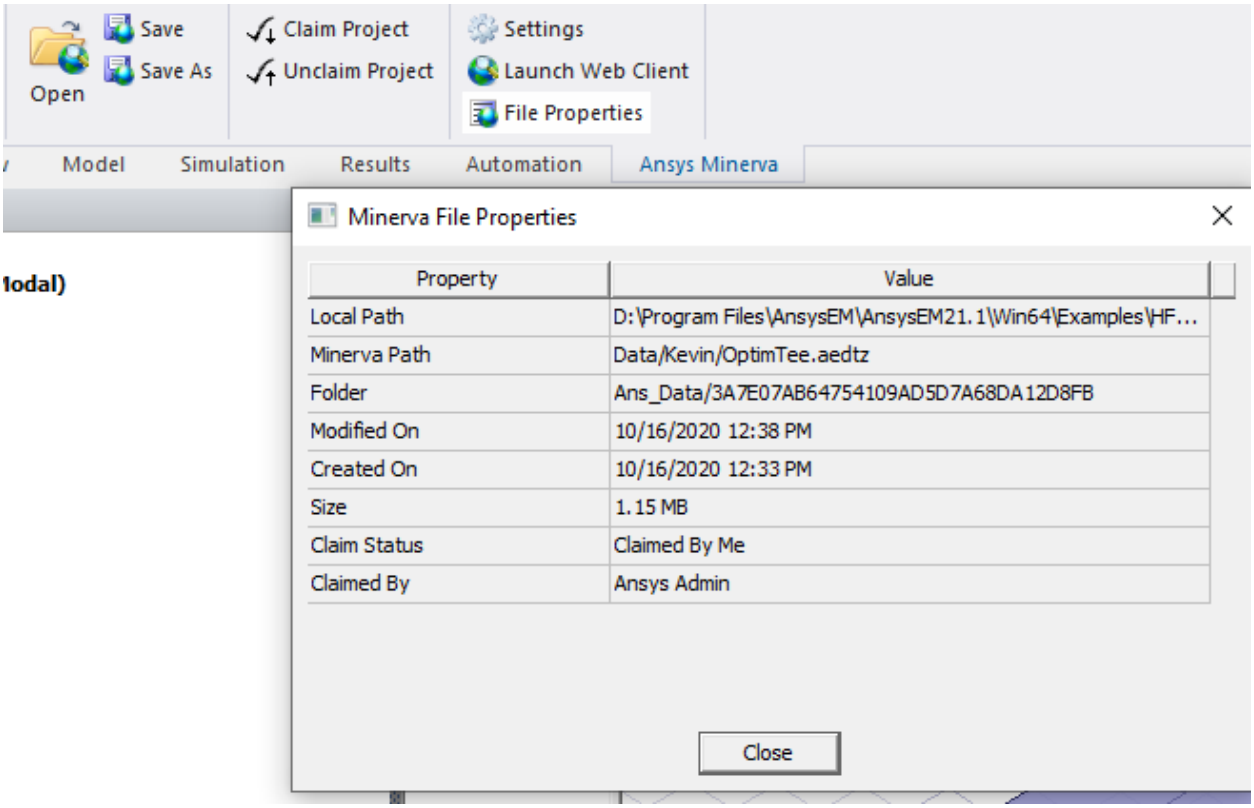
The recommended flow when working on a project is:

1. Claim the file, make 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.

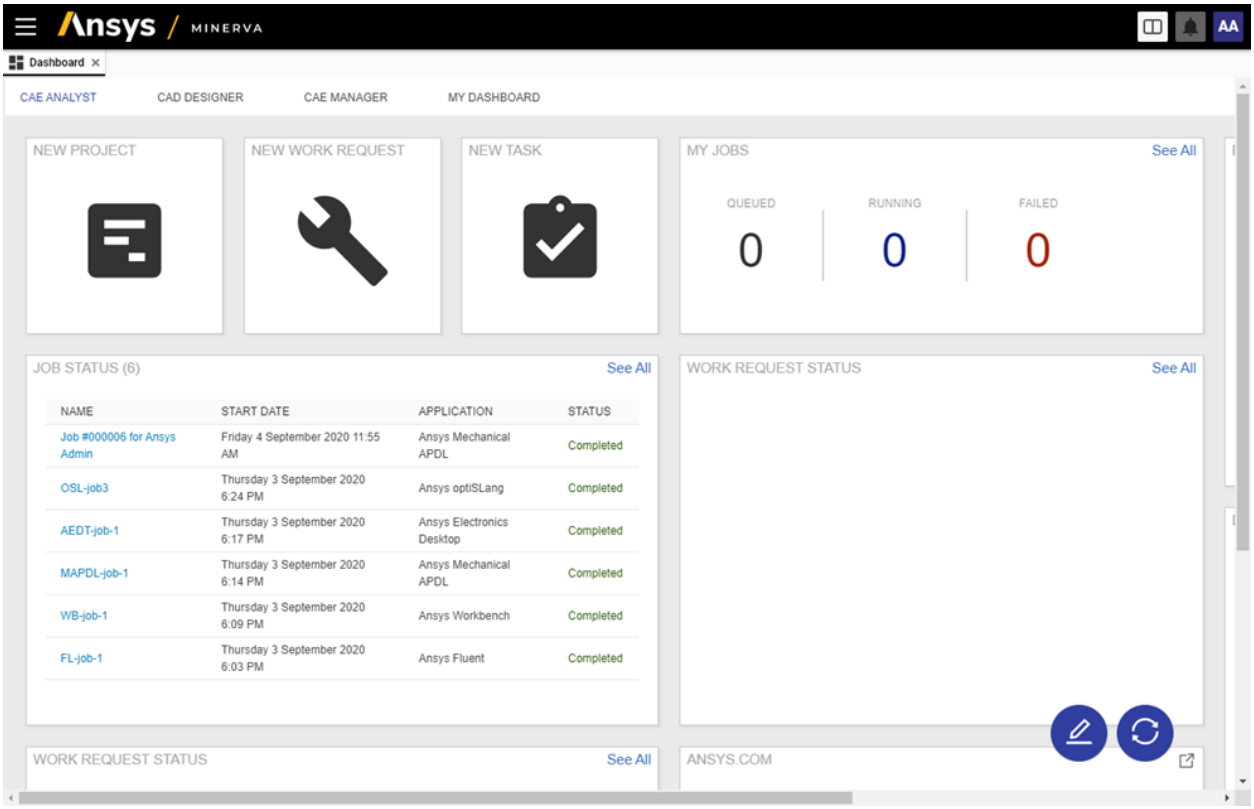
Unclaim Project: releases claimed file.

File properties: opens a dialog letting you view remote file properties. Notice the Claim Status and Claimed By properties.

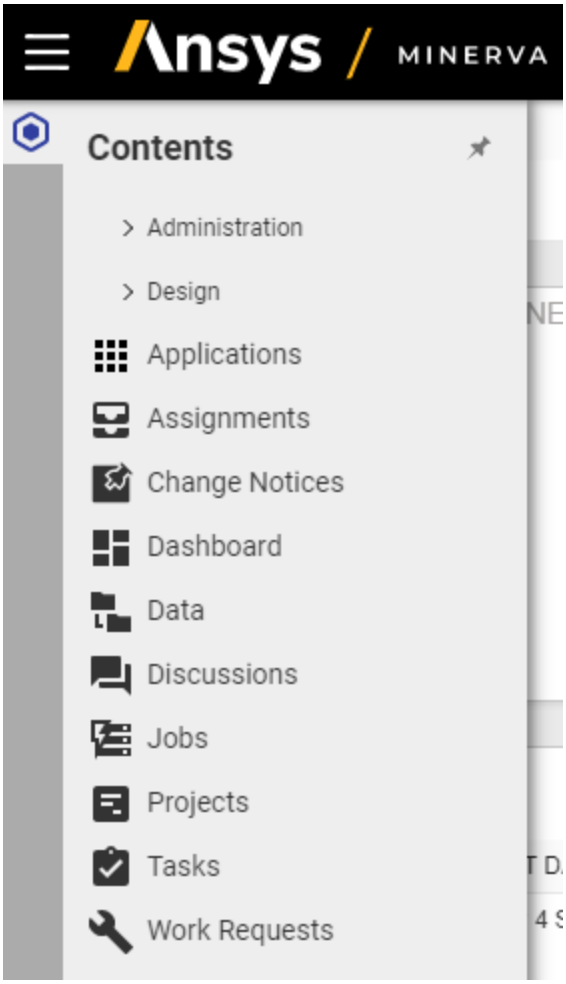


Launch Web Client:

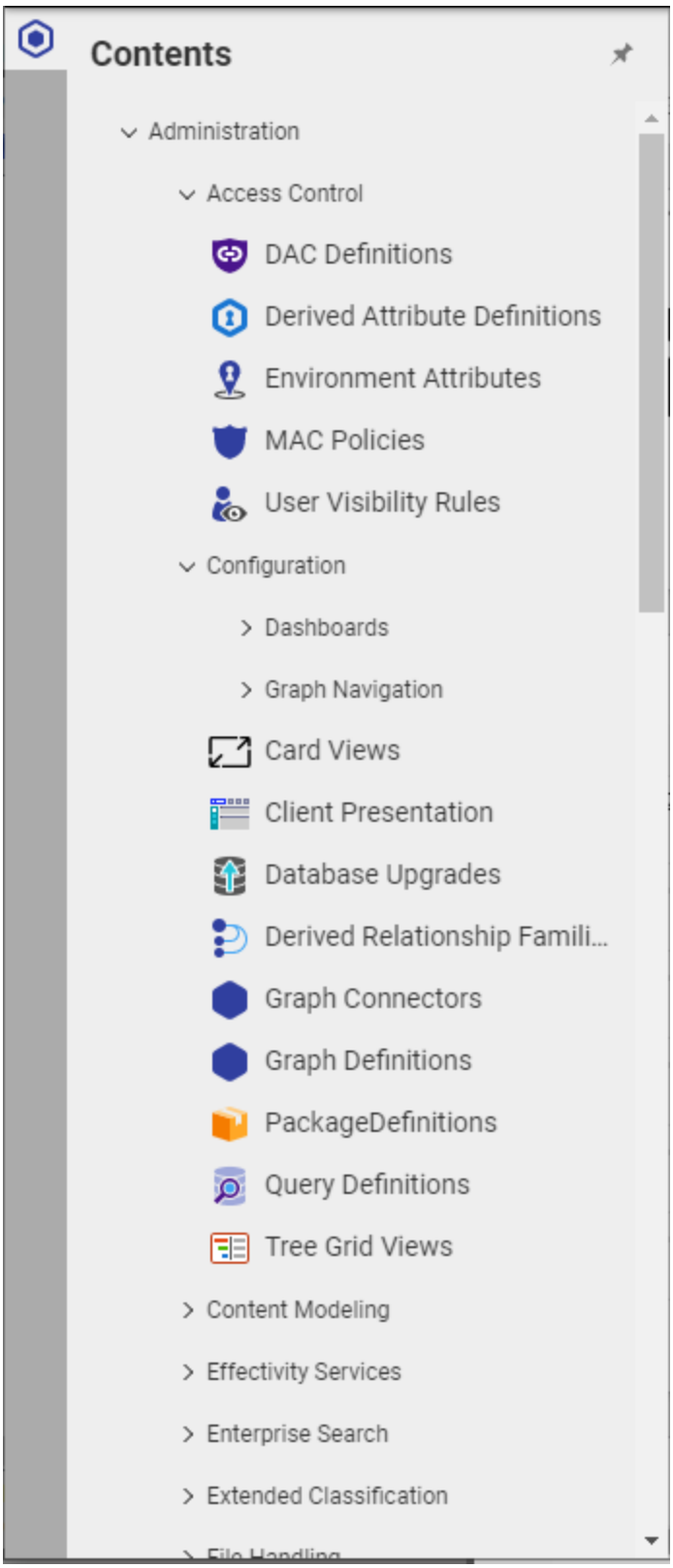
Ansys Minerva, powered by Aras, is a knowledge management application that secures critical simulation data, and provides simulation process and decision support to simulation teams across geographies and functional silos. Launching the Minerva Web Client gives access to these features. This opens a dialog calling for you to specify 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.



20 - 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, Icepak, Maxwell/RMxprt, Mechanical, Q3D Extractor, and Twin Builder.

> **Note:**
>
> For detailed information on working with Ansys Workbench, please refer to the Workbench documentation.

Integrating Ansys EM Suite with Ansys Workbench

You can integrate the Ansys Electromagnetics Suite with the Ansys Workbench application after both have been installed on your system. For successful integration, both preinstalled applications must be of the same version. A single operation completes the integration process for all supported design types.

The same procedure is used to integrate the applications or to decouple them (if they had previously been integrated). The procedure is as follows:

1. From the Windows Start Menu, select **All Programs** > **Ansys EM Suite 20*yy* R*n*** > **Modify Integration with Ansys 20*yy* R*n***.

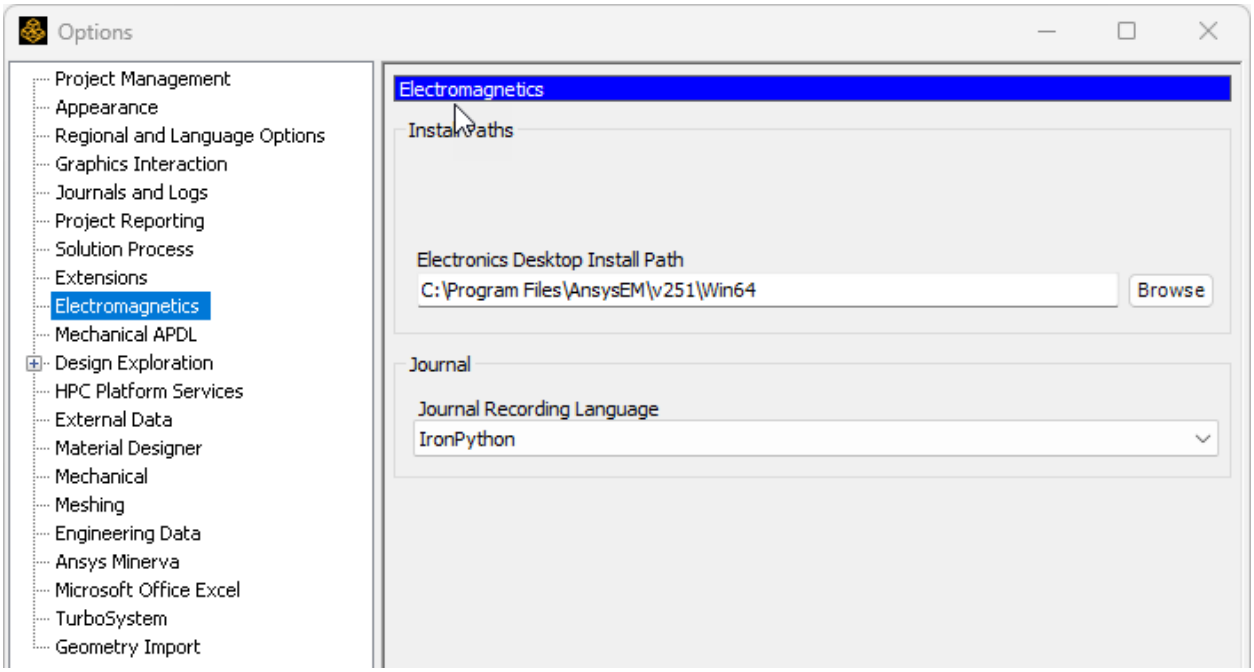
   A *Modify Integration with Ansys 20yy Rn* command window appears.

- If both applications are present on the system, their installation paths will be reported, and you will be asked if you wish to integrate them.
- Alternatively, if integration has already been preformed, you will be asked if you wish to decouple them.

2. Type **yes** to perform the prompted action or **no** to abort the operation without making any changes. Then press **Enter**
   - If you enter *"yes,"* a confirmation line appears in the command window indicating that integration (or decoupling) has been completed, and the window remains open.

     Press **Enter** again to close the command window.
   - If you enter *"no,"* the operation aborts and the command window closes immediately.

> **Note:**
>
> You can confirm that the Ansys 20*yy* R*n* application is “aware” of the integrated application via the *Options* dialog box in Workbench. The path to the integrated Electronics Desktop application is listed in the *Electromagnetics* group of settings.

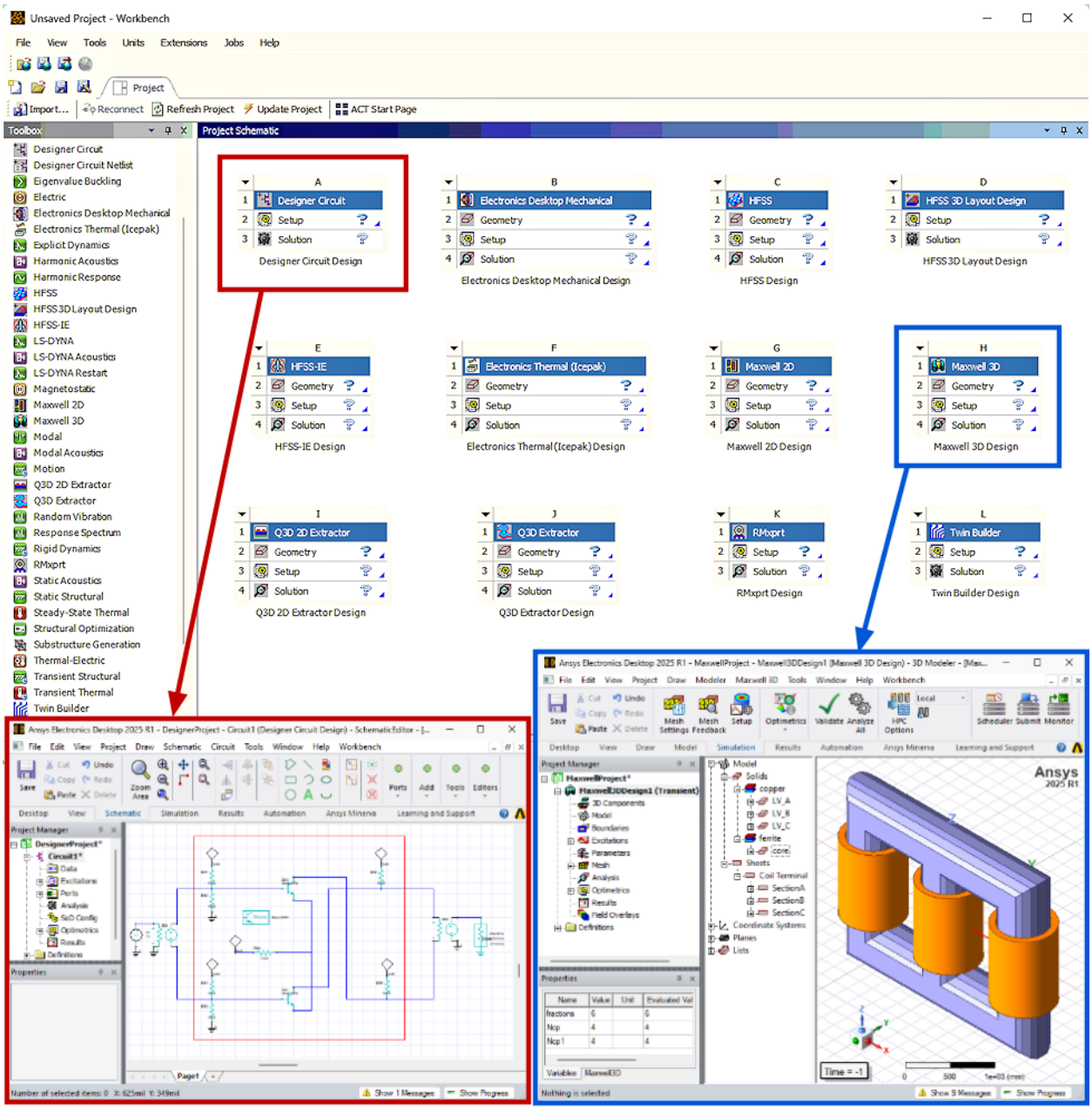


In this group of options, you can also specify Workbench's default **Journal Recording Language** for use when interoperating with the Ansys Electromagnetics products. **Iron Python** is the default language, and the alternative choice is **Visual Basic** (VB). Changes applied here

affect any new Workbench or Ansys Electronics Desktop session launched after the change. Any open instance will continue recording in the language that was in effect when the program was launched.

Workbench Data Integration Overview

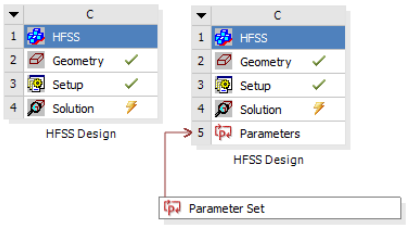
Ansys EM data-integrated applications can reside on a Workbench **Project Schematic**, as shown.



Objects placed on a Workbench Project Schematic (such as instances of Ansys Electromagnetics projects) are referred to as *systems*. The number of cells in the system depend on the desigh type, as follows:

- **RMxprt, Circuit, HFSS 3D Layout,** and **Twin Builder:** Appear on the Workbench Project Schematic as systems with three cells: **<*System_Type_Name*>, Setup,** and **Solution**.
- **HFSS, Maxwell, Q3D Extractor, Icepak,** and **Mechanical:** Include an additional **Geometry** cell along with the **<*System_Type_Name*>, Setup,** and **Solution** cells, for a total of four cells.

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 Workbench help for details on working with systems, cells, and parameter sets.



All supported Ansys Electromagnetics Suite products integrate with Workbench commands, services, and DesignXplorer in a similar manner. Some capabilities include:

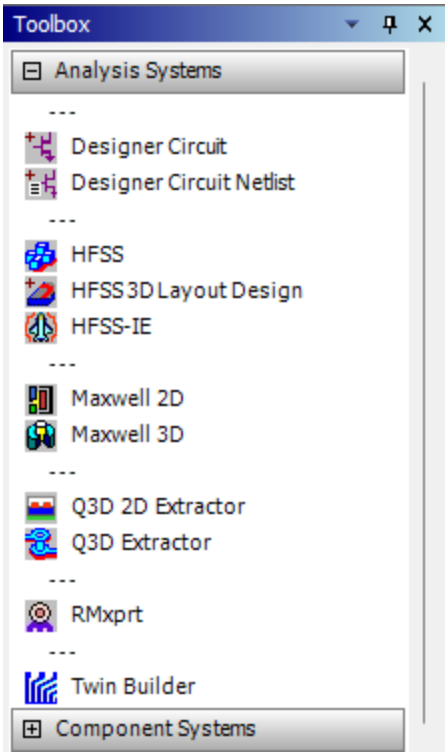
- Adding New Analysis Systems
- Importing Existing Ansys EM Projects into Workbench
- Editing Ansys EM Models in Workbench
- Ansys EM to Workbench Geometry Transfer
- Ansys EM to Workbench Material Data Transfer
- Analyzing Ansys EM Models in Workbench
- Performing Parameter Studies in Workbench
- Scripting in Workbench

In addition to these major features, Workbench can archive, save, back up, duplicate, and delete Ansys Electromagnetics projects used in a Workbench project. Progress information and messages from integrated Ansys EM projects are also displayed in Workbench.

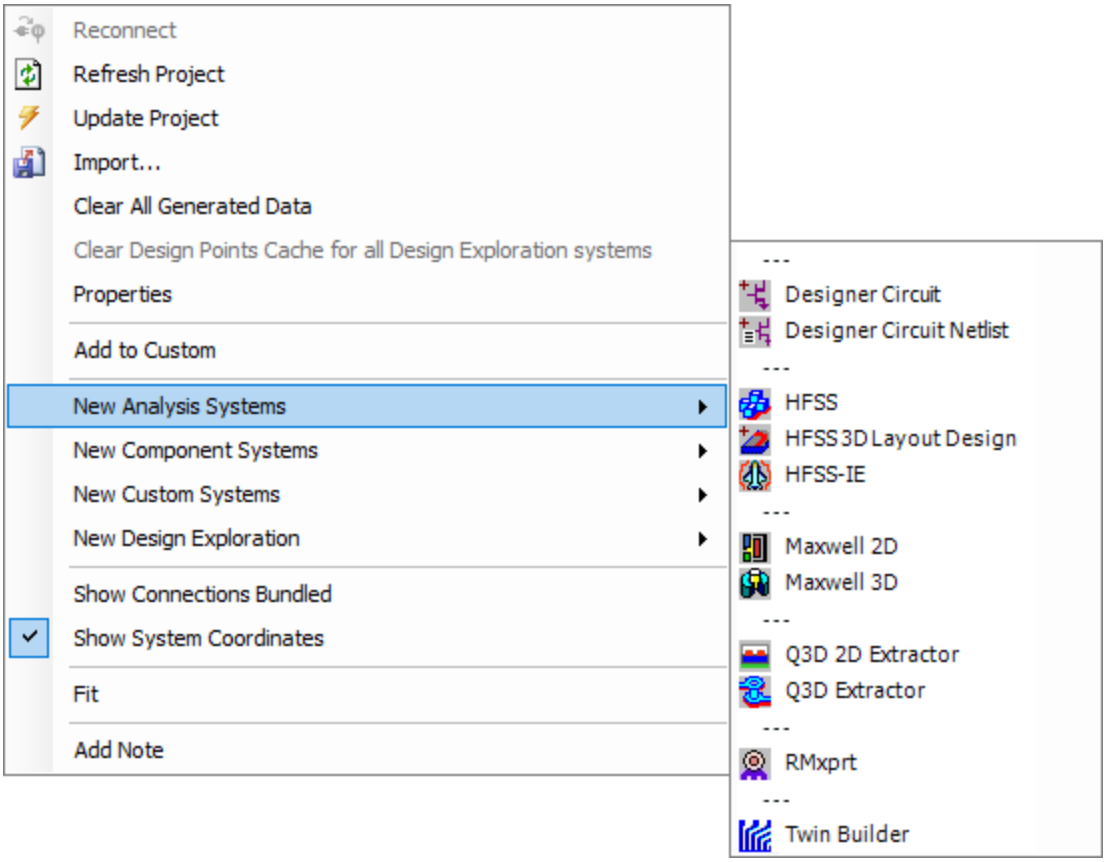
> **Note:**
>
> Look in the *Ansys Workbench Help* to find detailed information about performing the above-listed operations.

Adding New Analysis Systems

A new Ansys Electromagnetics Analysis System can be added to a Workbench Project Schematic by dragging and dropping it from the Toolbox:



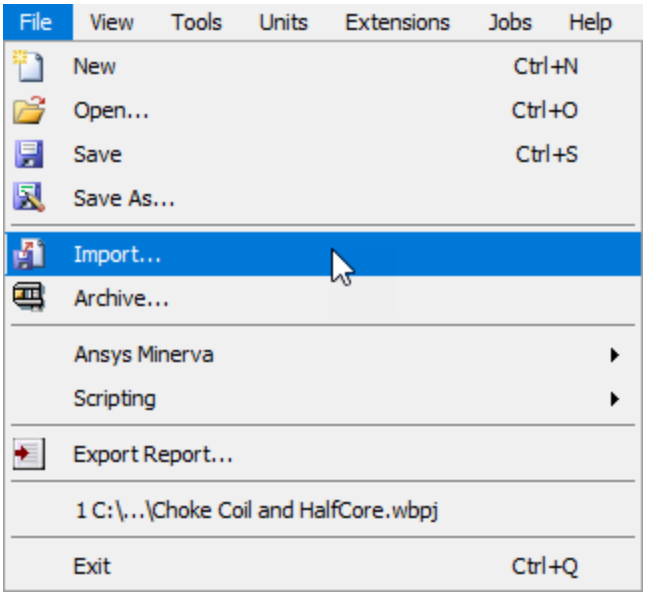
Alternately, you can add a system by right-clicking in the Workbench Project Schematic area and selecting **New Analysis Systems** > **[System Type]**.



Importing Existing Ansys EM Projects into Workbench

You can import existing Ansys Electromagnetics Suite projects into a Workbench project schematic using **File** > **Import**. When you do, a copy of the Ansys Electromagnetics project is placed in the Workbench project folder. The original Ansys Electromagnetics project remains intact.

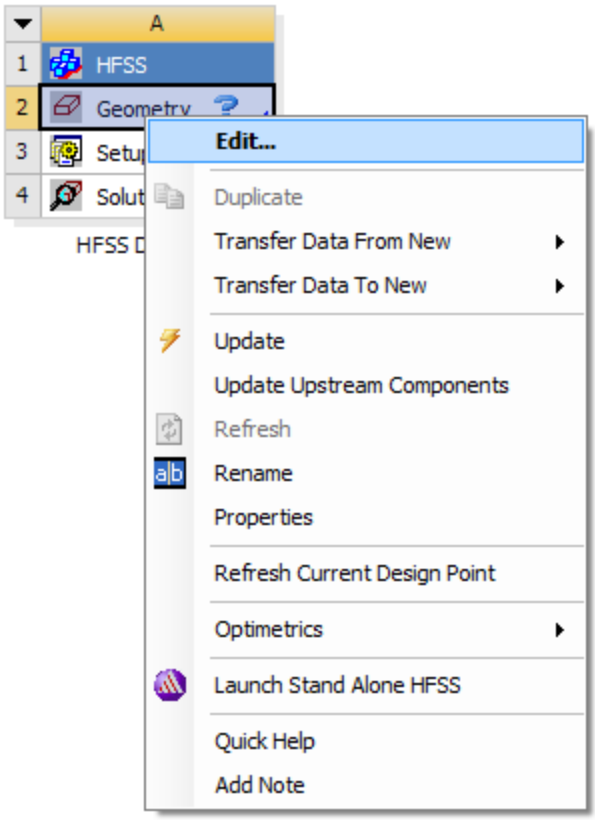
> **Note:**
>
> Object, material, and parameter names with non-ASCII characters are not allowed for data transfer. Such transfers fail and produce an error message.



Editing Ansys EM Models in Workbench

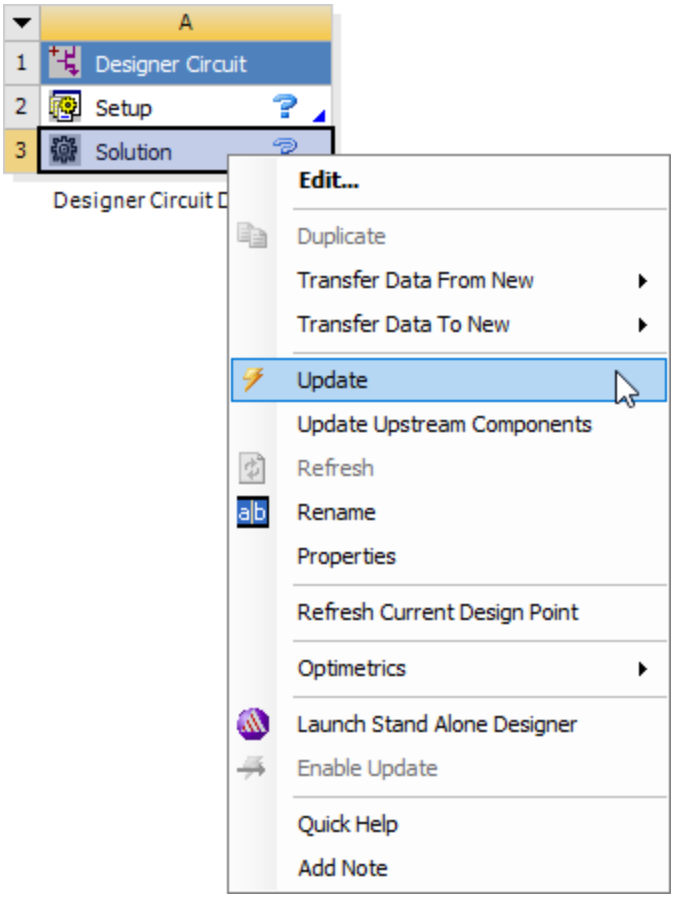
You can edit various properties and parameters of the Ansys Electromagnetics project (geometry, setup, solution, etc.) either by right-clicking the project in Workbench and selecting **Edit**, or by double-clicking the project.

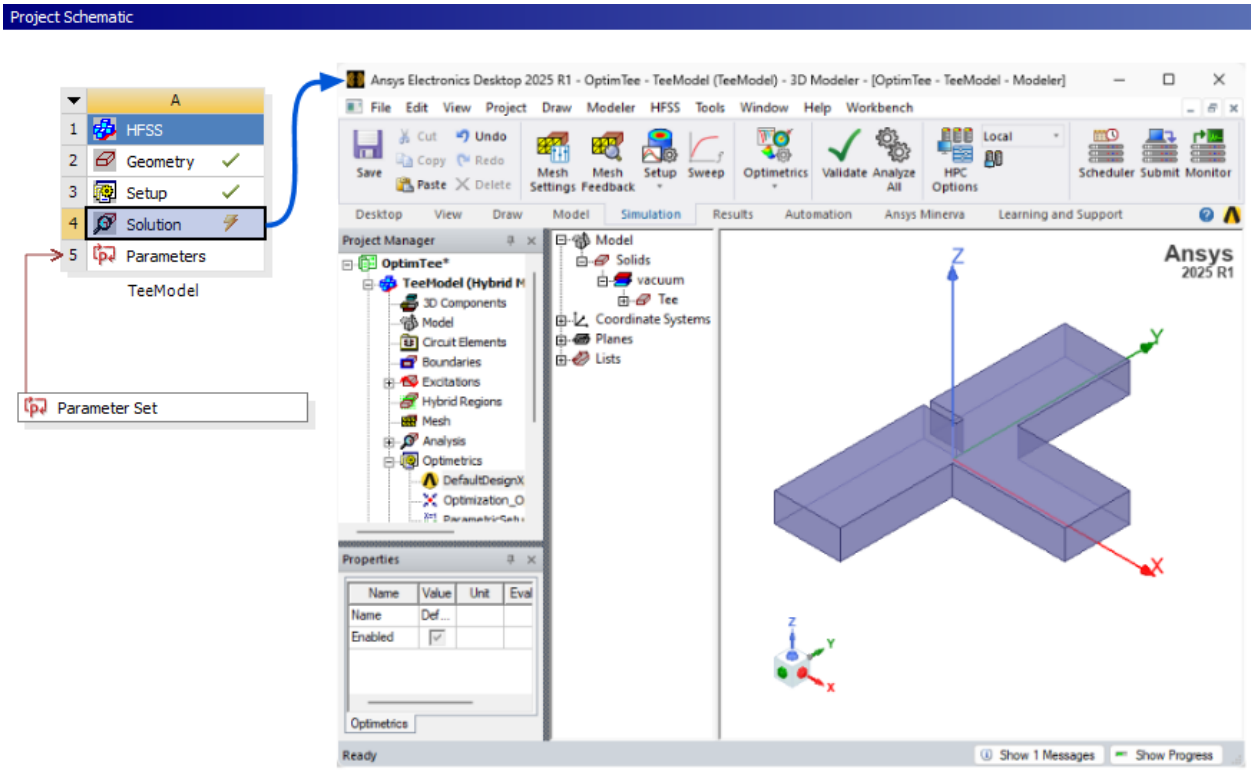
Doing so launches the Ansys Electromagnetics desktop application and loads the project so that you can set up your project in a familiar desktop environment. Changes made to the Ansys Electromagnetics project are saved to the project instance in the Workbench project folder.



Analyzing Ansys EM Models in Workbench

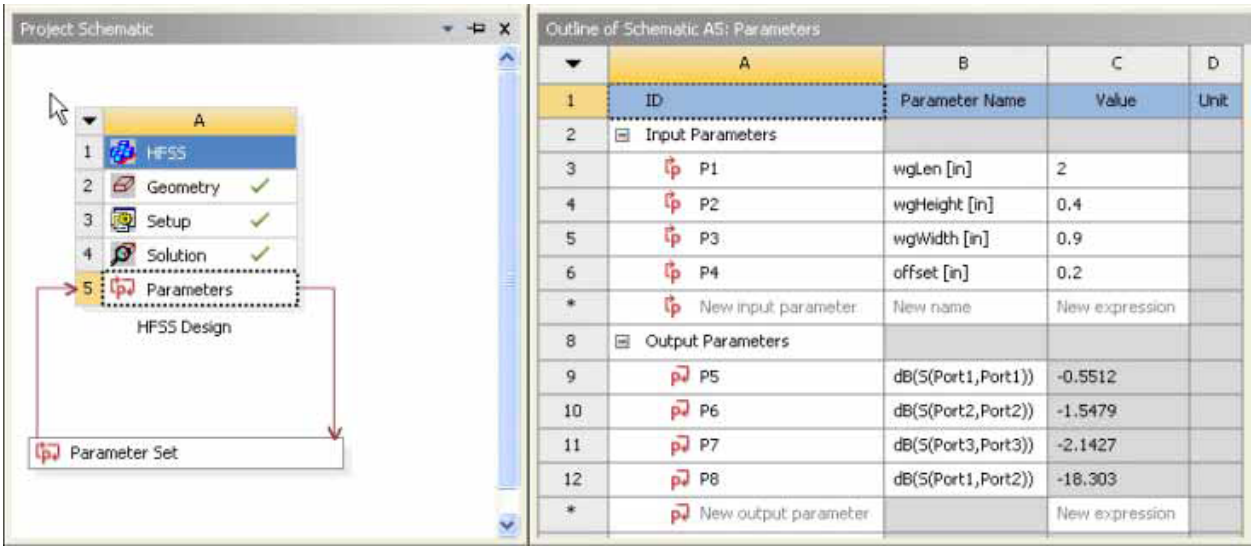
You can use Workbench's **Update** command to run analyses in the integrated Ansys Electromagnetics project. Progress information is also shown in Workbench.





Performing Parameter Studies in Workbench

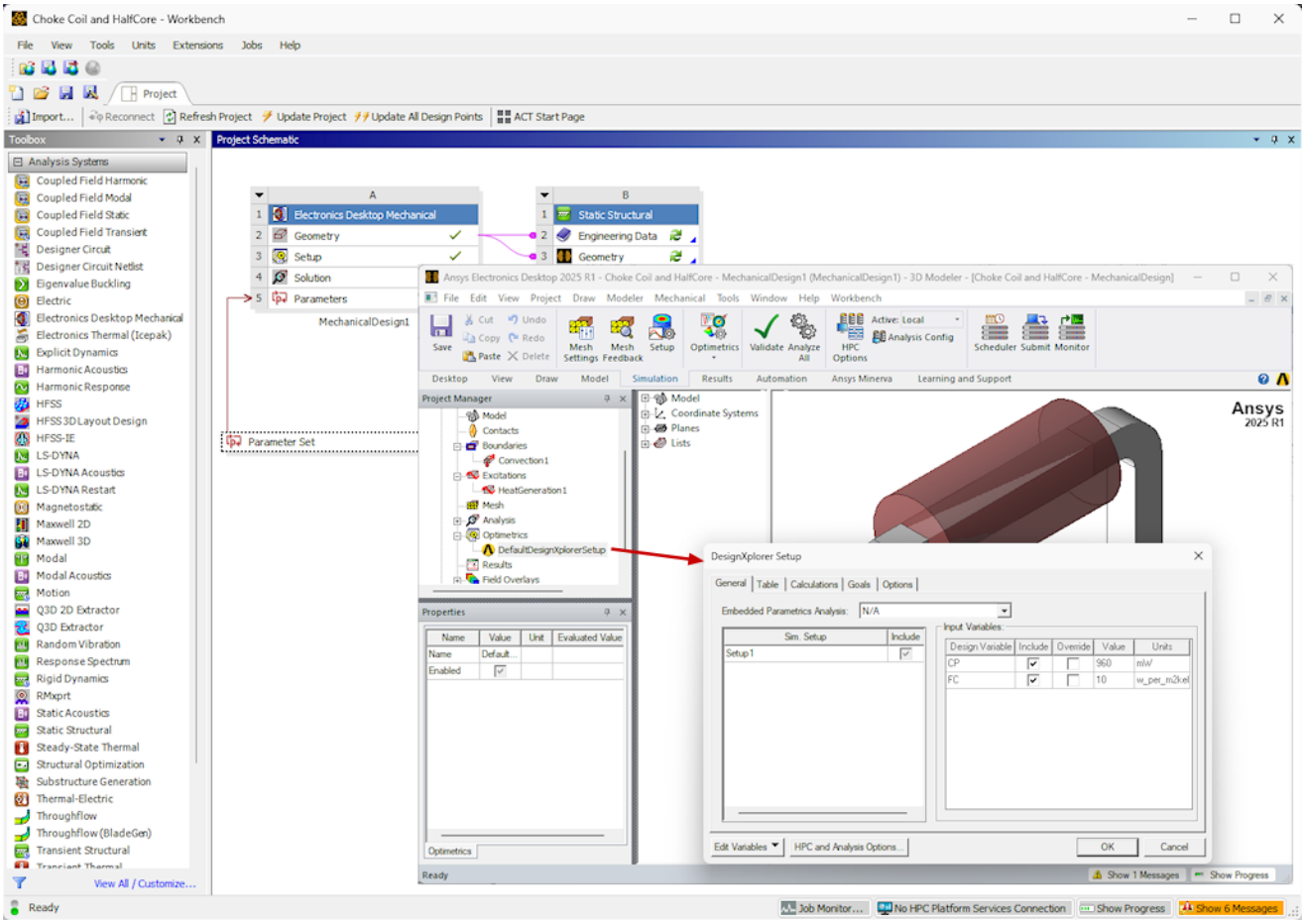
Workbench **Parameter Sets** allow you to change parameter values and units of measure, or add new parameters. Parameter data is passed back to the Ansys Electromagnetics application for updated analyses. After analysis, the Workbench Parameter table should be updated correctly for all design points.



Note:

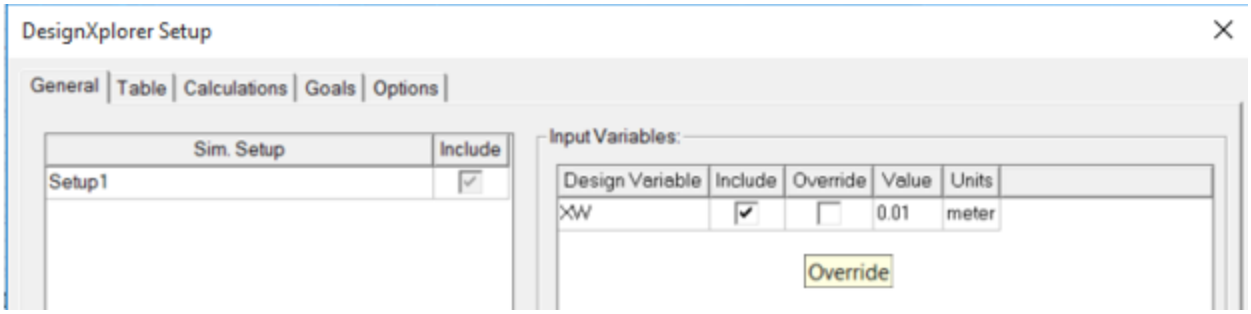
Because Workbench and Ansys Electromagnetics Desktop have different unit systems, units defined in Desktop are Dimensionless in the Workbench Parameters table and thus cannot be changed. Refer to the *Ansys Design Modeler User's Guide* > *Project Schematic Operations* > *Parameters in Project Schematic* > *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. The following example is a Mechanical, Steady-State Thermal design in which two variables have been defined. One variable (CP) is the coil power (960 mW heat generation) and the other (FC) is the film coefficient of the assigned convection boundary:

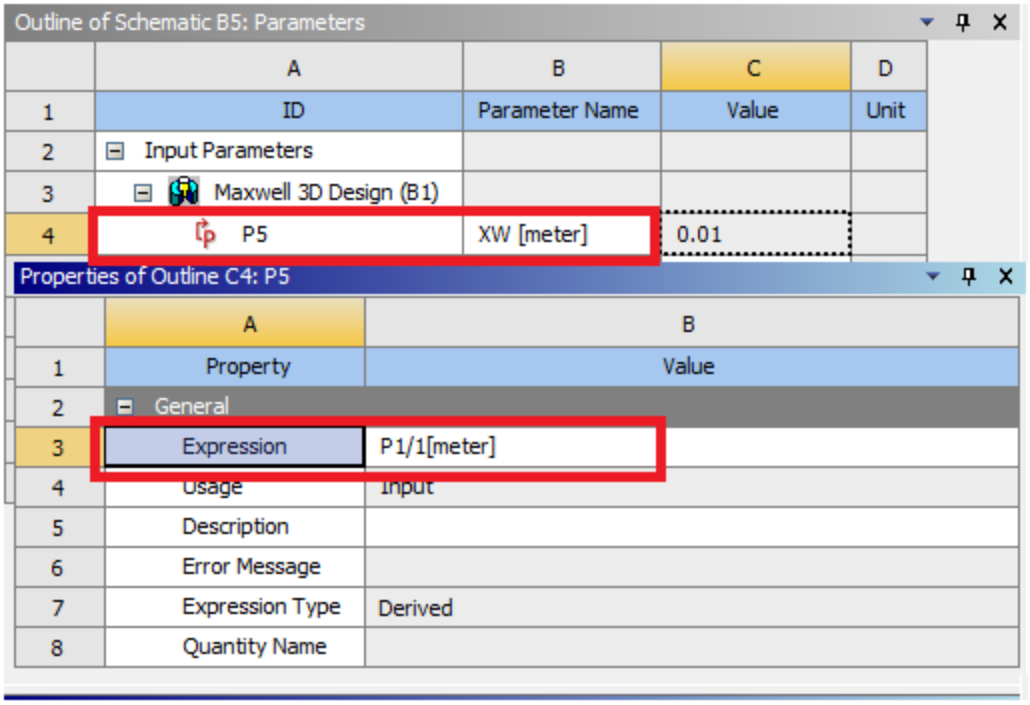


The workflow for using Electronics Desktop systems with the Workbench to take advantage of Distributed Analysis is as follows:

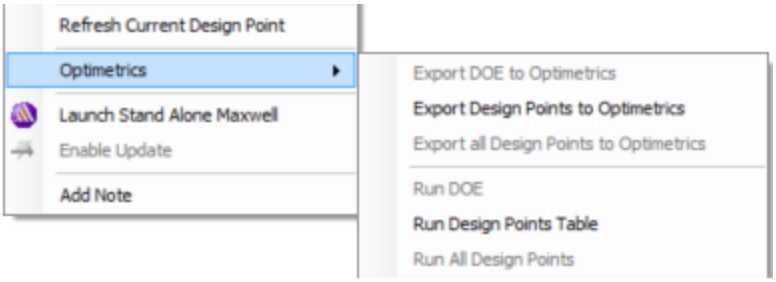
1. In the Ansys Electronics Desktop, specify a variable an as input parameter. The following figure shows a DesignXplorer Setup that includes a geometry variable called XW This variable is mapped to a variable in the corresponding Workbench Design, in this example, P5.



2. In the Workbench Design, change the P5 expression to a new value (in this example figure, the P5 parameter is assigned an expression of P1/1[meter] to tie it to the geometry system's variable.



In Workbench, the **Export Design Points to Optimetrics** and **Run Design Points Table** commands are enabled.

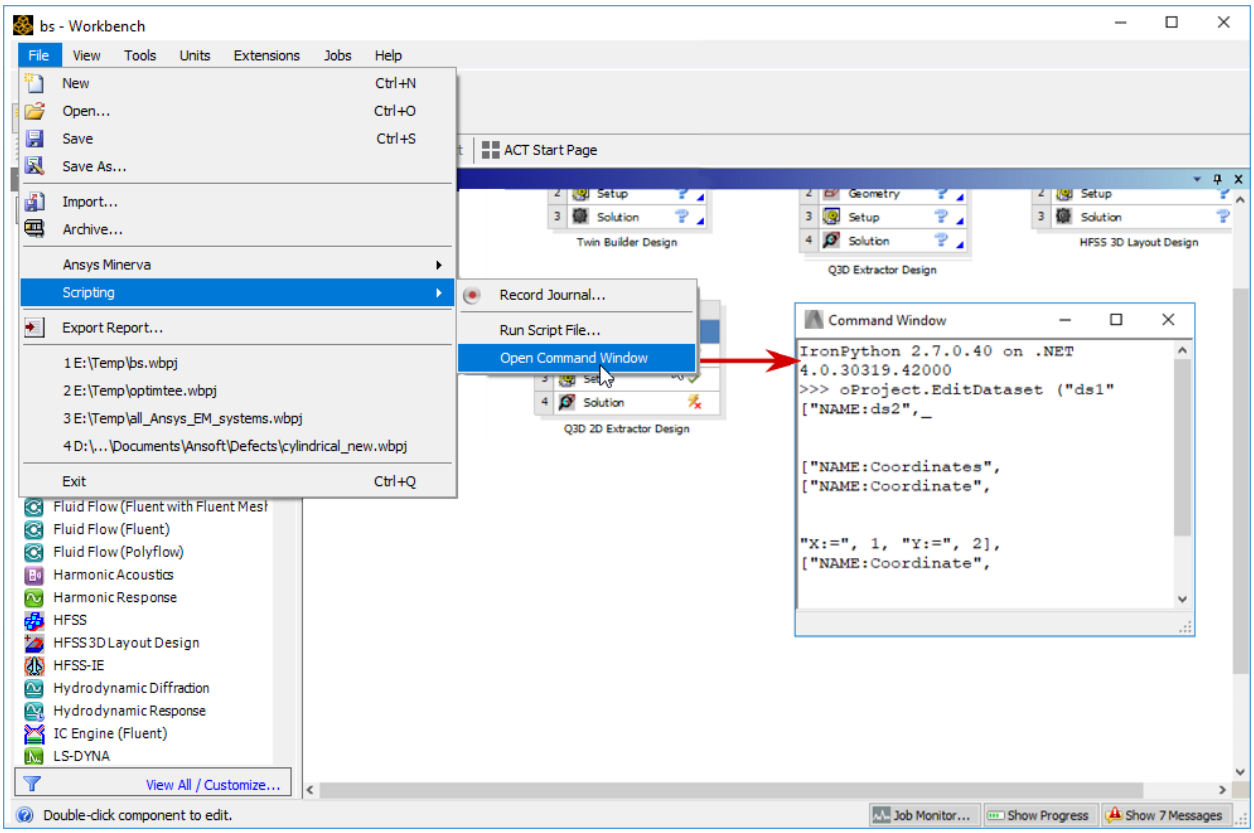


3. In Electronics Desktop, you can click on the default DesignXplorer Setup, and choose **Generate Variation Data**.

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.

Scripting in Workbench

Scripts that include Ansys Electromagnetics projects can be recorded and run in Workbench.



When a Workbench journal is run from the command line, Electronics Desktop allows the Ansys EDT portion to also run in batch mode (that is, when using RunWB2 -b).

Example: From a Windows Command prompt, enter the following command to replay the journal named "myReplayJournal.wbjn":

```
RunWB2 -b -r myReplayJournal.wbjn
```

Ansys EM - Ansys Multiphysics Coupling

Data integration provides an 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, and there is no need to export or import transfer XML files. Edits made in Ansys Electromagnetics applications are automatically transferred to the Ansys application through the Workbench **Refresh** command. Workbench commands also enable easier automation of iterative coupling of thermal feedback.

The following sections provide some examples of multiphysics coupling.

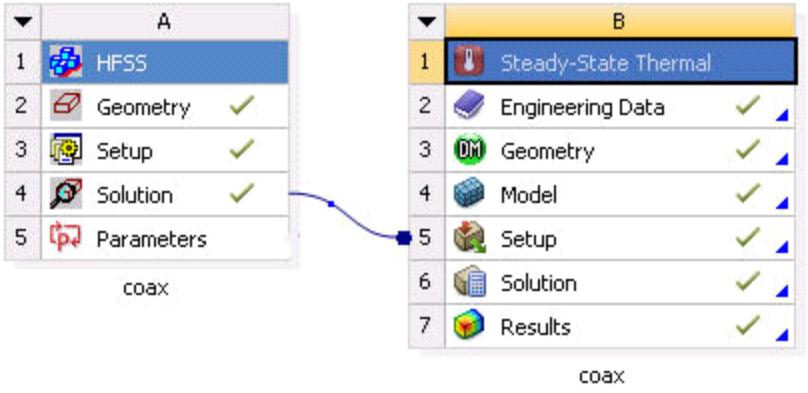
Multiphysics Coupling on Workbench with Ansys Thermal

Multiphysics Coupling on Workbench with Ansys Structural

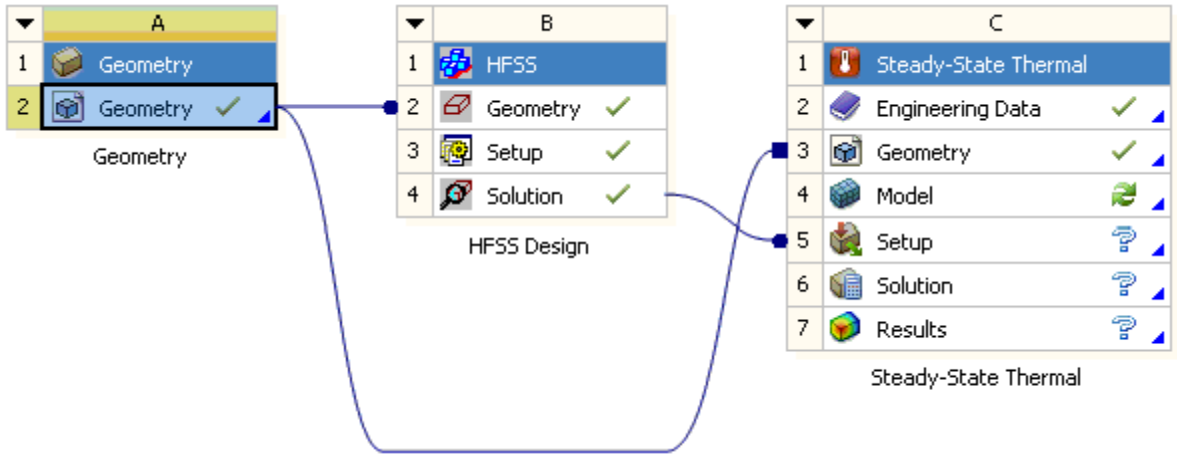
Multiphysics Coupling on Workbench with Ansys Thermal

Using data integration, HFSS, Maxwell, and Q3D Extractor provide heat losses (heat generation and heat flux) to Ansys Thermal. In HFSS, both Driven and Eigenmode projects can be coupled for multiphysics. You first need to enable feedback as described in Setting the Temperature of Objects.

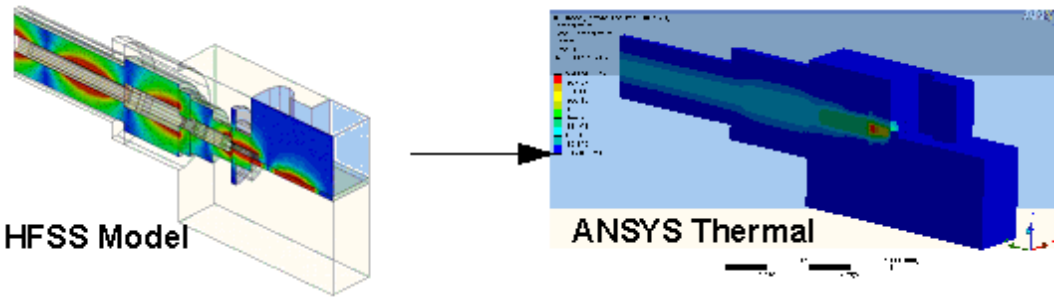
Note how the HFSS design is linked visually to Ansys Thermal on the Workbench project schematic.



Geometry sharing is possible, provided you activate Beta Options in Workbench. A geometry from the Component Systems in the Toolbox in Workbench Schematic can be shared as shown in the image below.



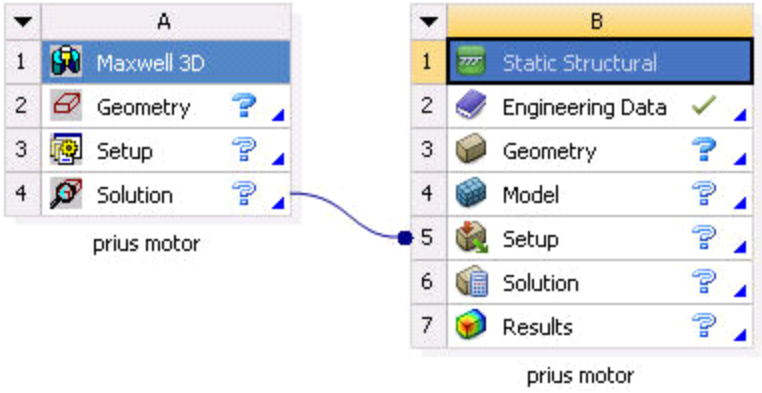
In this example, HFSS coax model Solution provides heat loss data as a thermal load to the Ansys Thermal Setup. The resulting analysis shows a thermal "hotspot", providing the user with the information needed to adjust the design's material to fix the problem.



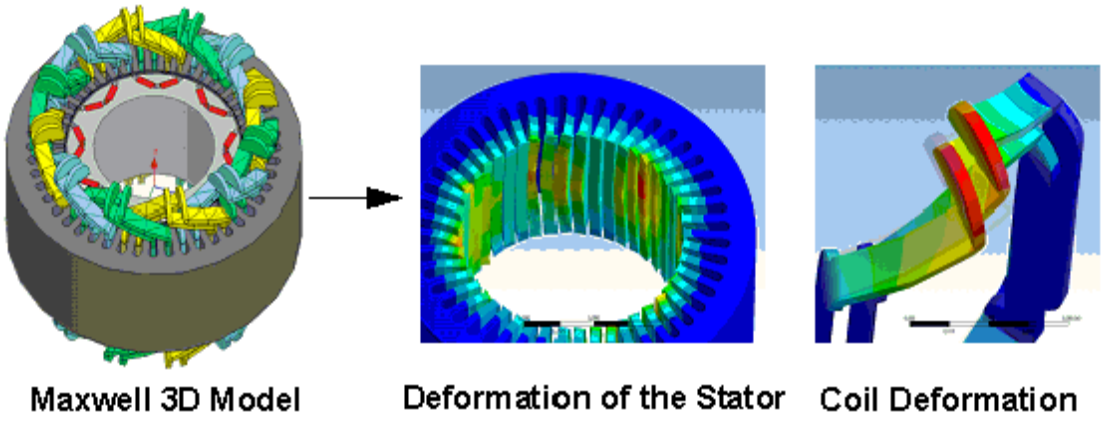
You enable this feature by checking Enable Feedback in the dialog for Setting the Temperature of Objects.

Multiphysics Coupling on Workbench with Ansys Structural

Using data integration, Maxwell 2D and Maxwell 3Dcan provide forces to Ansys Structural.



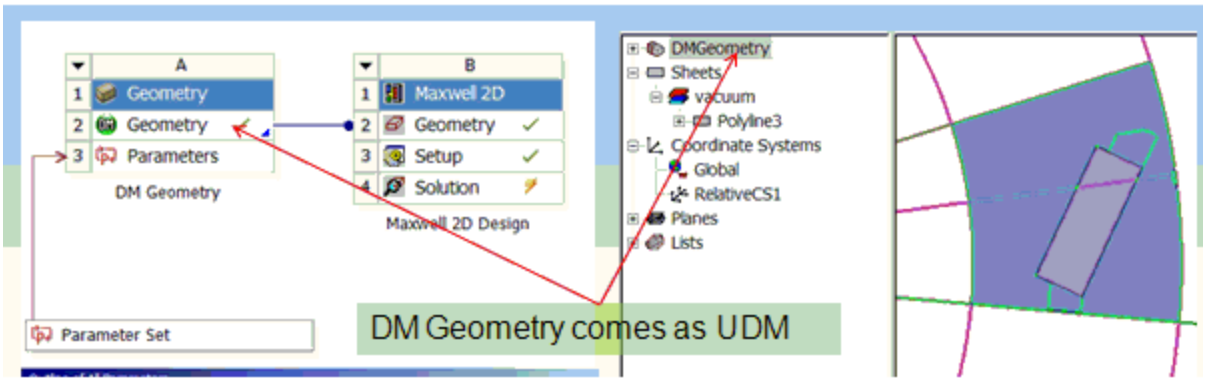
In this example, the Maxwell 3D electromagnetic force density **Solution** is used as the load in Ansys Structural to determine how these forces deform the motor's stator and coils.



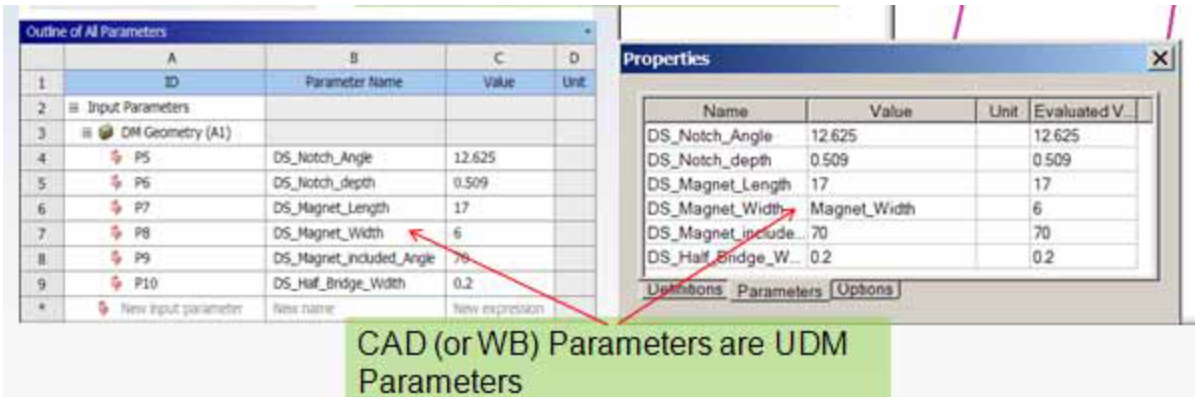
Ansys EM CAD Integration through Workbench

Ansys Electromagnetics CAD integration is a Workbench feature available for Ansys Electromagnetics 3D Products - HFSS, Maxwell, Q3D, Icepak, and Mechanical, as the Ansys Framework. The feature is available only through Workbench, and is not available via standalone Ansys Electromagnetics products.

Ansys CAD integration provides a bi-directional dynamic link through Workbench. This makes it possible to get updated geometry from CAD, modify CAD parameters in Ansys Electromagnetics products, and return an updated geometry. The feature is non-associative due to a need to reassign boundaries if a modified CAD model is used. The process creates a User Defined Model (UDM) for each geometry source.

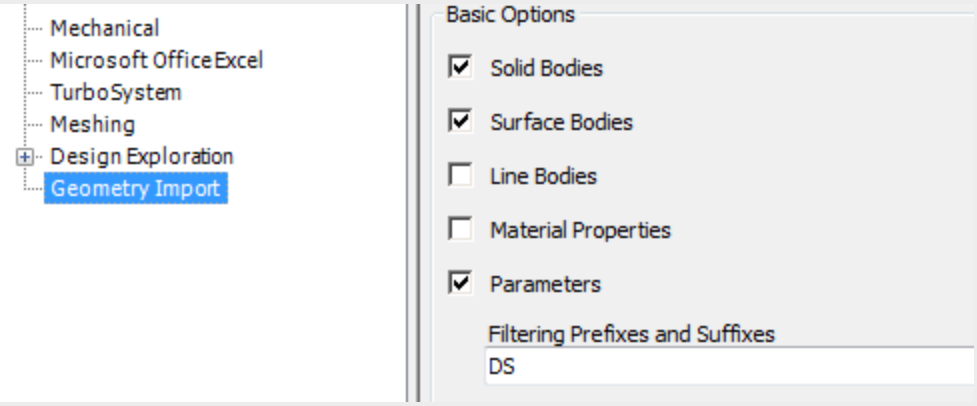


The UDM format makes it possible to exchange parameters. Ansys online help contains further description of the UDM feature and function.



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**. To import parameters with different prefixes or names, assign an appropriate prefix or clear the **Filtering Prefixes and Suffixes** field.



Ansys Electromagnetics CAD integration makes it possible to consume geometry from multiple upstream sources (any CAD or Ansys Electromagnetics product).

This feature supports direct interfaces with major CAD systems, including

- Creo Parametric
- UG NX
- CATIA V5
- SOLIDWORKS
- Autodesk Inventor
- Ansys Design Modeler (DM)
- Discovery UDM model

Note:

The CAD software must be installed on the user machine (not required for solve nodes).

The following sections contain additional information:

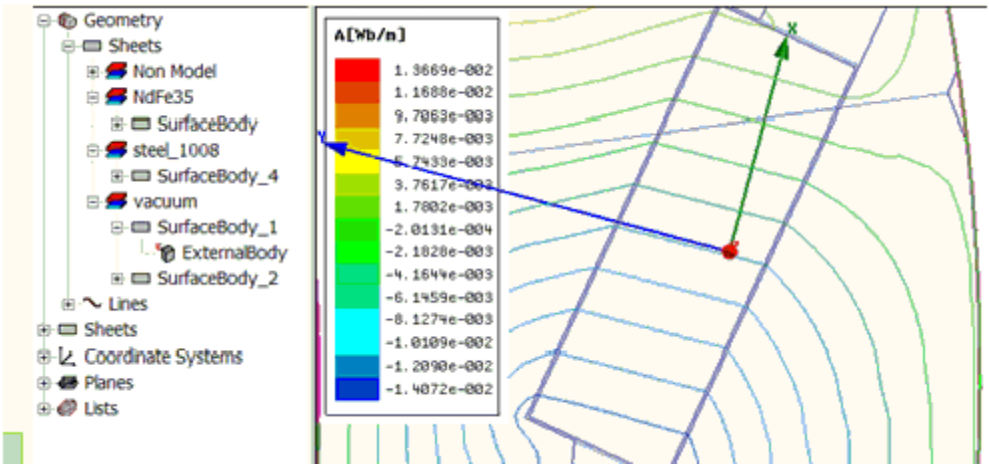
CAD Integration Functionality

CAD Integration and Geometry Sharing

CAD Integration Functionality

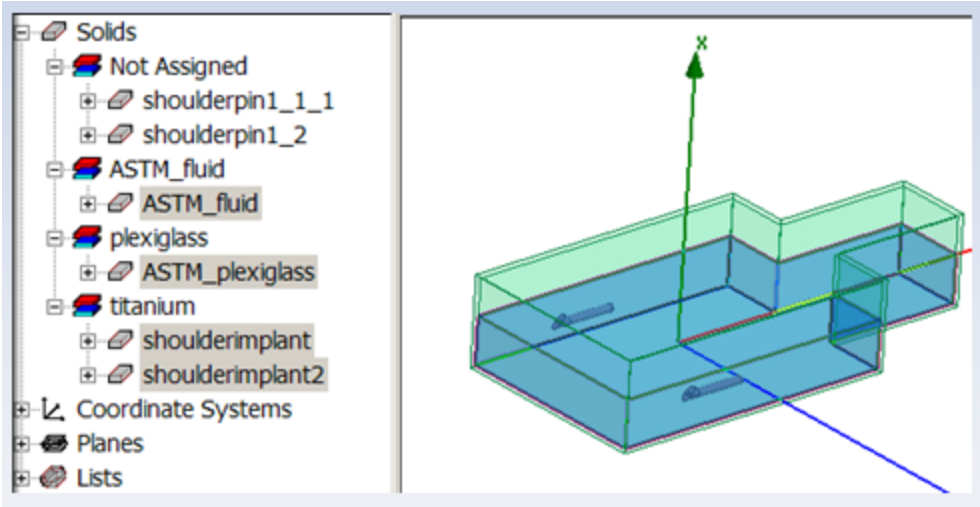
CAD Integration includes the following functionality:

- DX analysis
- WB Update Project
- WB Update All Design Points
    - Parametric analysis with DSO
    - Animation
    - Geometry
    - Field plots

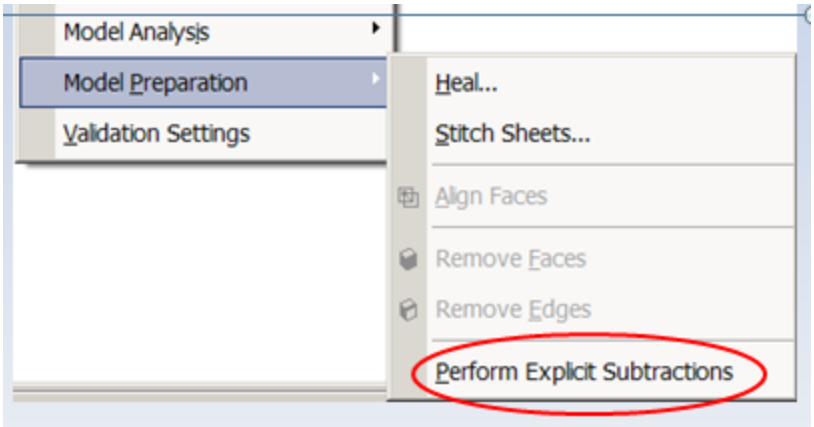


Ansys Mechanical does not do implicit subtraction. Ansys Electromagnetics products can do explicit subtractions before sending geometry to Ansys using **Modeler > Model Preparation > Perform Explicit Subtraction**.

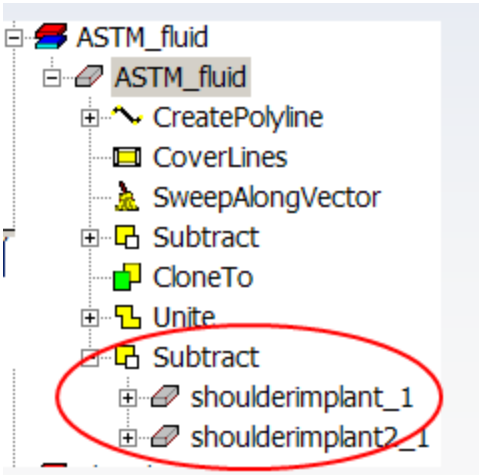
For example, consider the following model.



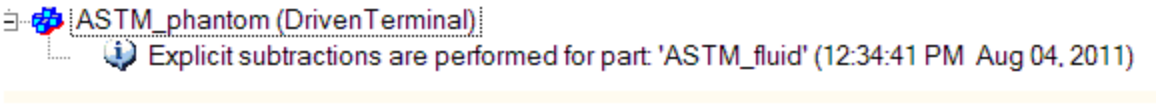
Perform Explicit Subtractions can be performed.



Results appear in the History tree, as shown:



The **Messages** window reports this action.

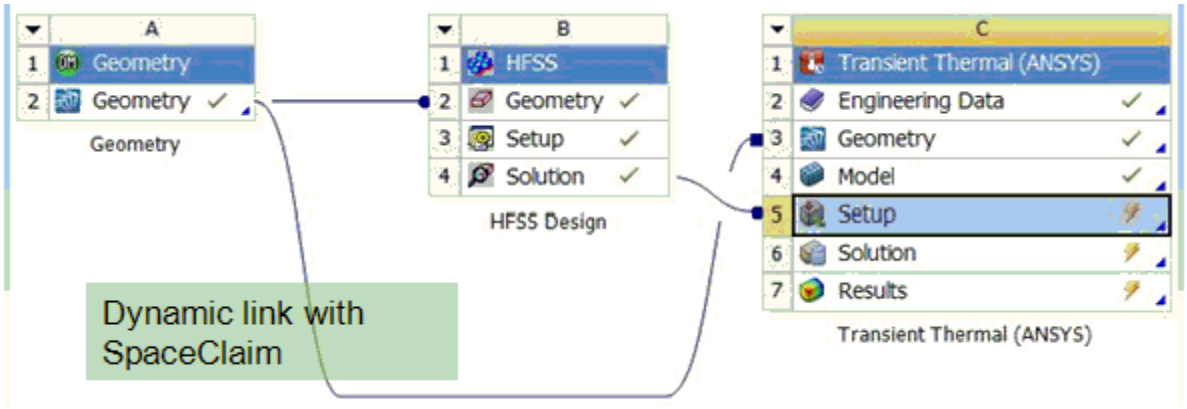


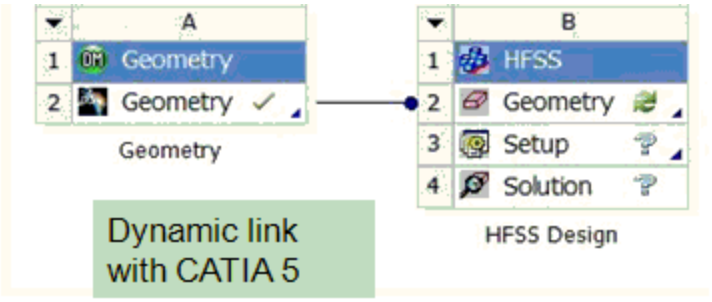
CAD Integration and Geometry Sharing

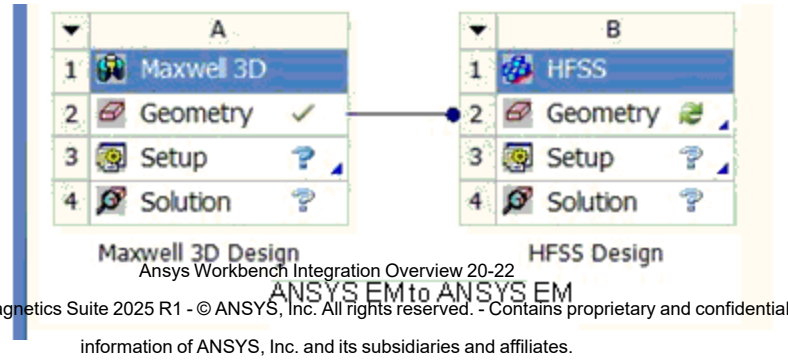
A CAD model comes into Ansys Electromagnetics as a User Defined Model (UDM).

The inputs to Ansys Electromagnetics from CAD are:

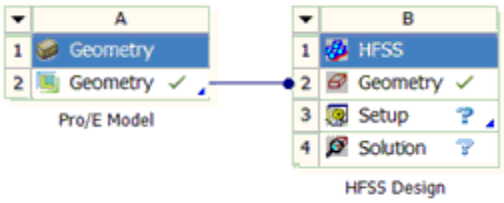
- Geometry/Topology with persistent IDs
- CAD parameters
- Material assignment
- Attributes, such as name and color



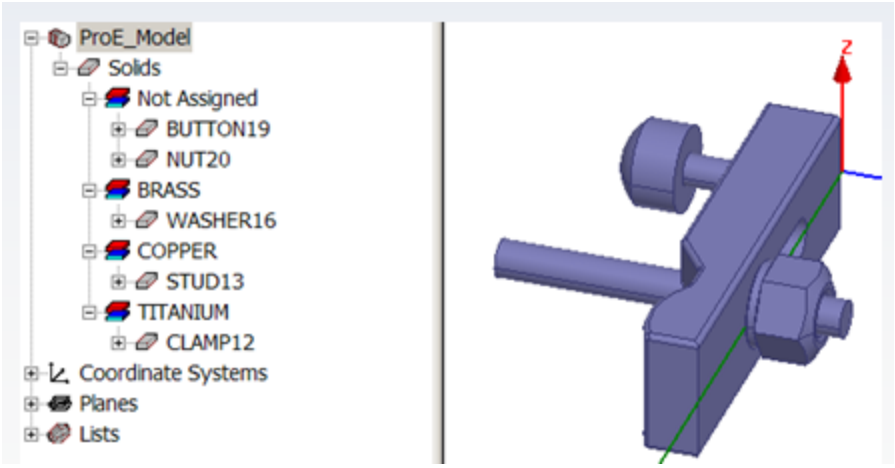




For example, in Workbench, a Pro/E Model can be linked to Ansys.



The geometry can then be viewed in HFSS as a UDM.

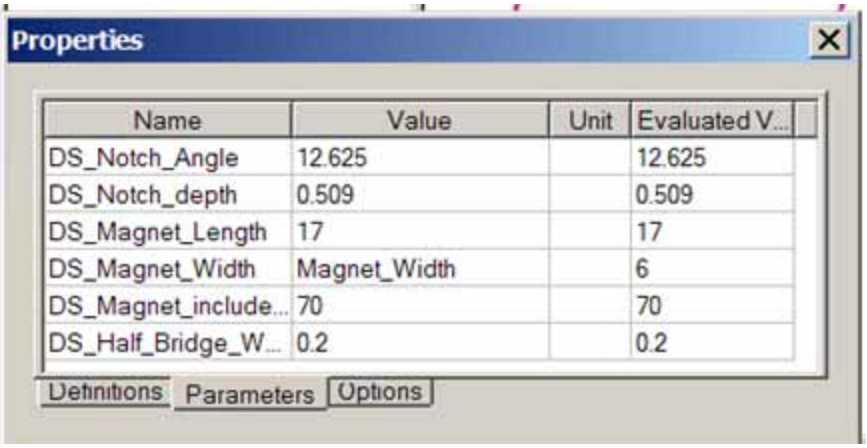


The CAD or WB model parameters appear in the Workbench:

Outline of All Parameters

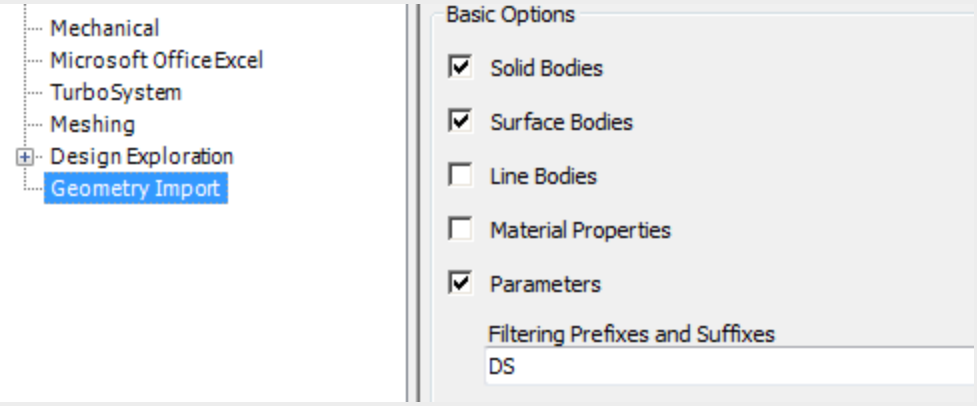
	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	Input Parameters			
3	DM Geometry (A1)			
4	P5	DS_Notch_Angle	12.625	
5	P6	DS_Notch_depth	0.509	
6	P7	DS_Magnet_Length	17	
7	P8	DS_Magnet_Width	6	
8	P9	DS_Magnet_included_Angle	70	
9	P10	DS_Half_Bridge_Width	0.2	
*	New input parameter	New name	New expression	

Through the Ansys Electromagnetics CAD integration, the linked UDM includes the same parameters.

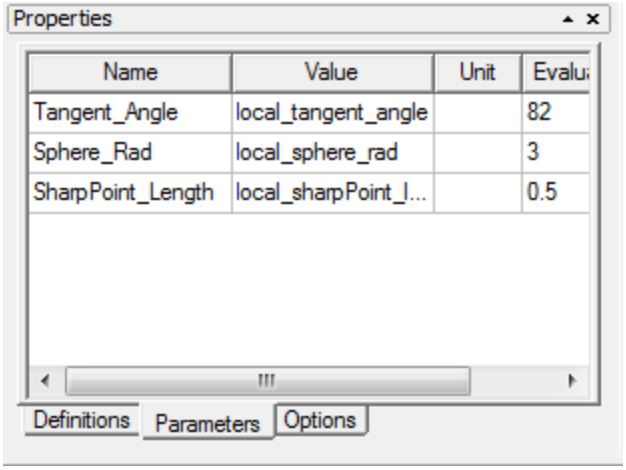


Note:

The parameters shown in these examples all have a DS prefix. This is the default for the Workbench **Tools>Options** for **Geometry Import**. To import parameters with different prefixes or names, assign an appropriate prefix or clear the **Filtering Prefixes and Suffixes** field.



Once you import a geometry with parameters to an Electromagnetics application in the desktop, you need to map to local variables. In the target Electromagnetics application, select the geometry associated with the parameters. Update the geometry and view the **Parameters** tab in the **Properties** window. The **Value** column shows the values of the imported parameters. Type names for local variables in the **Name** column.



Once you have local variables for the imported Design Modeler parameters, you are ready to use Design Xplorer for multiple design parameters.

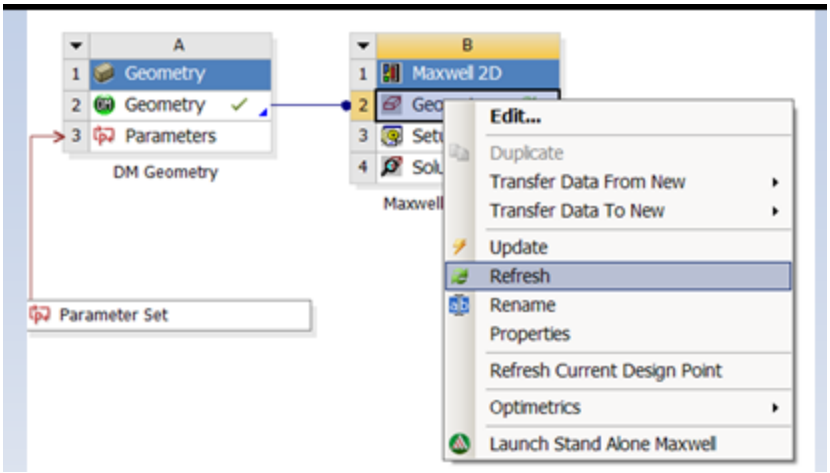
Table of Schematic D2: Design of Experiments (Custom)

	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
5	4	80	2.7

Bi-Directional CAD Integration

Ansys Electromagnetics CAD Integrations uses **Refresh** or **Generate** to pass updates.

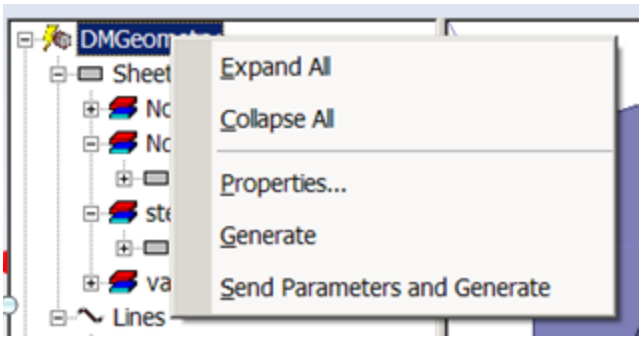
You make an edit in a CAD application and run **Refresh** on an Ansys Electromagnetics Geometry Cell.



Refresh pulls the current state of CAD model (geometry, parameters, materials etc) and updates the corresponding data in the Ansys Electromagnetics application.

Alternatively, you can run **Generate** on the UDM in the Ansys Electromagnetics window.

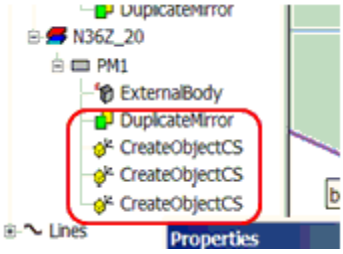
If you edit UDM (CAD) parameters in the Ansys Electromagnetics modeler window you can run the **Send Parameters and Generate** command.



The command passes the edited parameters to the linked CAD application and then pulls corresponding CAD geometry.

CAD Integration Model Edits

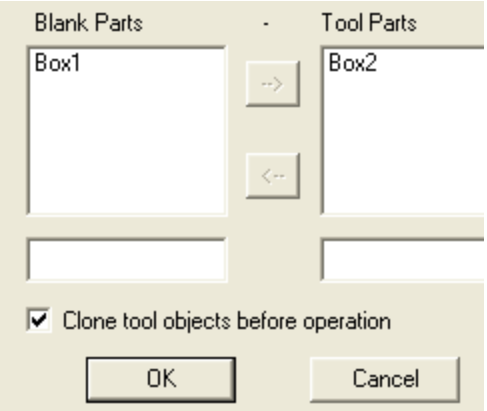
Several modeling operations are allowed on a CAD model in the Ansys Electromagnetics Modeler window. Operations are included in the History tree and retained during model **Refresh**.



The following operations are not allowed:

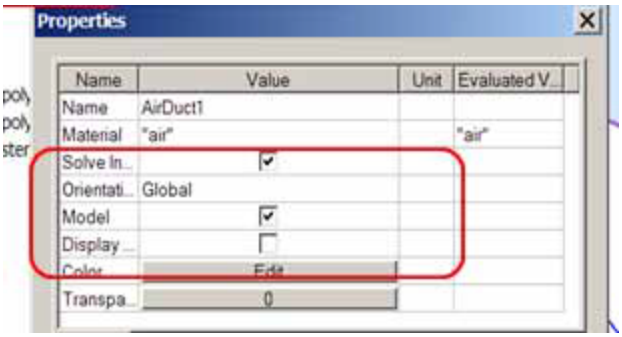
- Non-history tree operations like Heal or defeature.
- Operations that use UDM parts as tools, such as Sweep

- Boolean operations like Split or Unite (allowed when you select the clone tool option)



The following part attributes can be modified for UDM parts:

- Model/Non Model flag
- Part orientation
- Color
- Display Wireframe

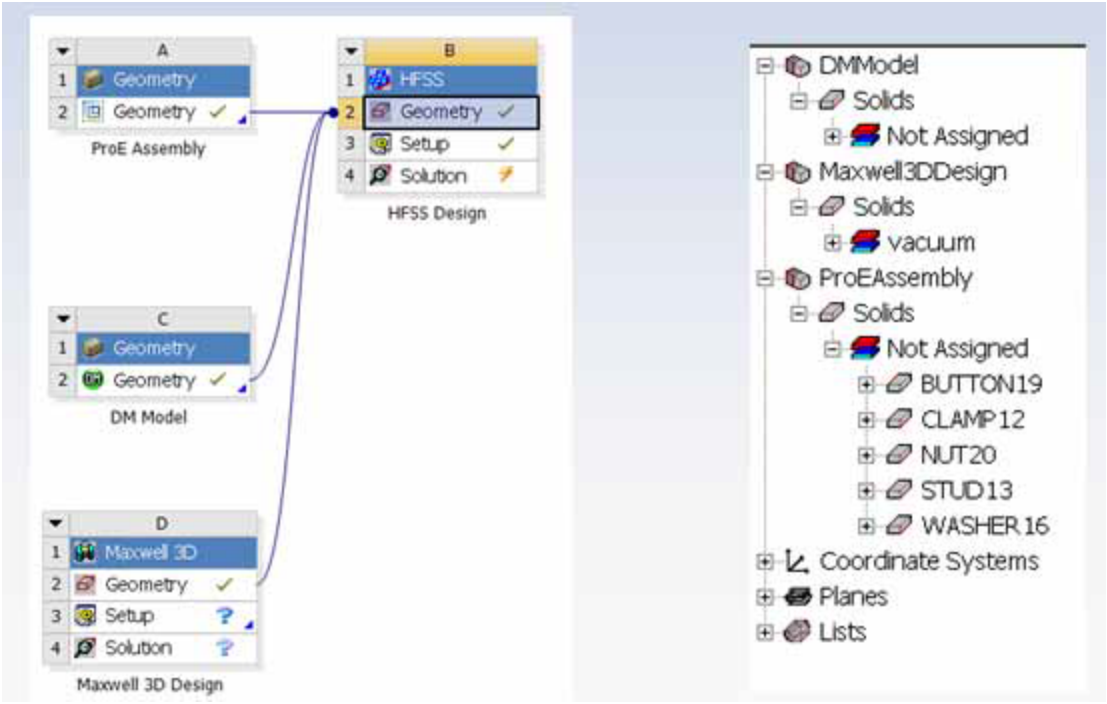


It is not possible to delete individual parts of UDM.

Multiple Geometry Links for CAD Integration

With CAD Integration you can consume geometry from multiple upstream sources. The source can be any of supported CAD or Ansys Electromagnetics products. This creates a UDM for each geometry source.

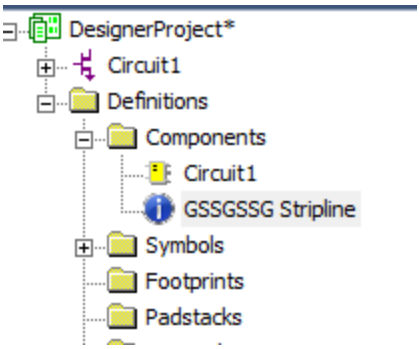
For example, the following figure shows a DM Model, a Maxwell model, and ProE model linked to HFSS in Workbench, and displayed in the HFSS History tree as three UDMs.



Creating Dynamic Links to EDT Designs

If you have a Circuit Design and one or more HFSS, 2D Extractor, Q3D, Icepak, or Mechanical designs open in a Workbench project, you can create dynamic links to those designs.

Update the Circuit Design to open the Electronics Desktop. In the menu bar, select **Workbench > Update Dynamic Link Components**. This command will create components in the project tree for each design in the Workbench project.



Note that in this example, a GSSGSSG Stripline component was created corresponding to the GSSGSSG Stripline design in WorkBench. Once created, you can drag the component into a schematic to place it.

Adding a Design Variable for Dynamic Links

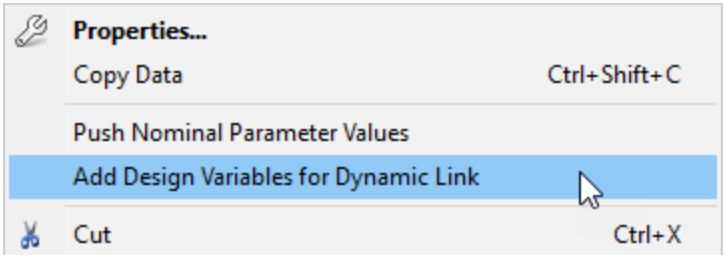
Design variables from an HFSS, Q3D, 2D Extractor, Icepak, or Mechanical design can be imported though a dynamic link to a Circuit or 3D Layout design. New design variables are created in the Circuit or 3D Layout design and correspond to the dynamic link variables. When the new design variables are created, the dynamic link variables are set to the Circuit or 3D Layout design variables.

For example, imagine an HFSS design with a variable wire_rad of 3mm. If a dynamic link is created for this design in a Circuit design, then a design variable “d_wire_rad” is created in the Circuit design and the component instance of the HFSS design has a parameter wire_rad set to d_wire_rad.

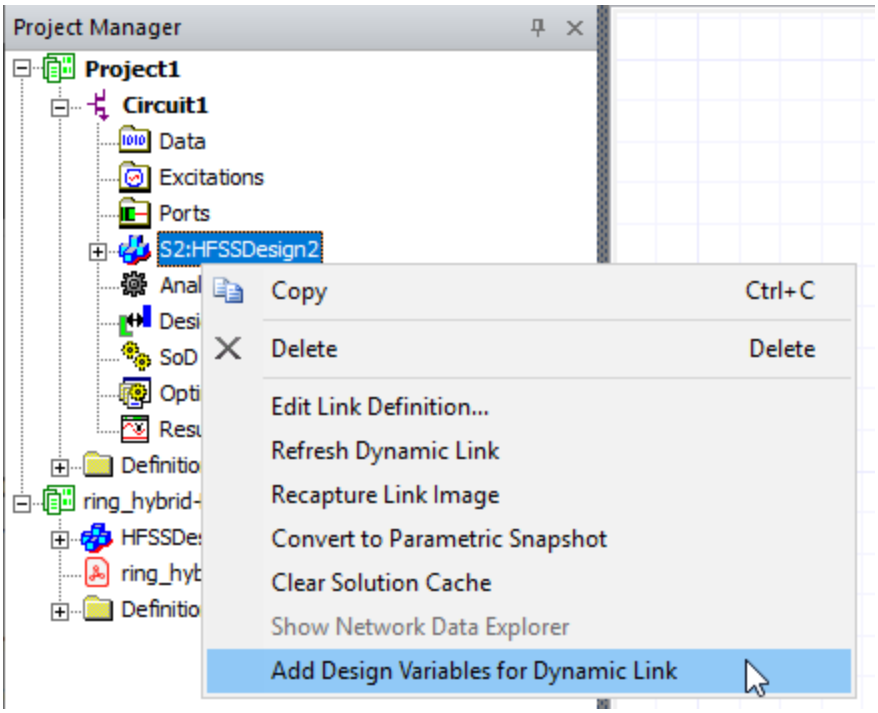
Design variables can be created from existing or newly added variables in the dynamic link design.

To create design variables based on pre-existing variables in a dynamic link design:

1. In the Circuit or 3D Layout design, create a dynamic link to the target design.
2. Either:
    a. In the Schematic or Layout editor, right-click the new instance of the linked design and click **Add Design Variables for Dynamic Link**.



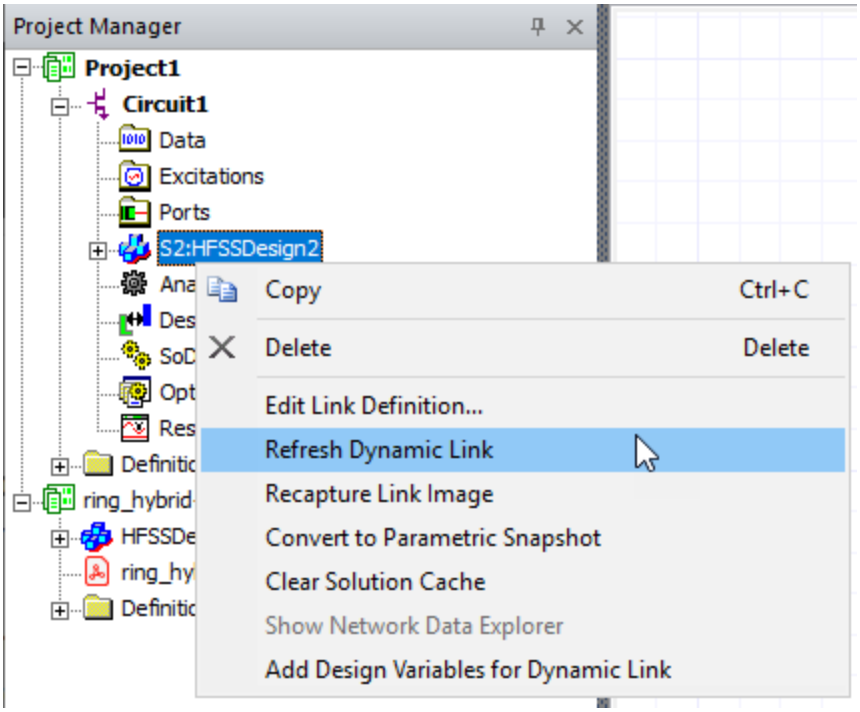
b. From the **Project Manager** window, right-click **Project Name** > **Design Name** > **Dynamic Link Design** and select **Add Design Variables for Dynamic Link**.



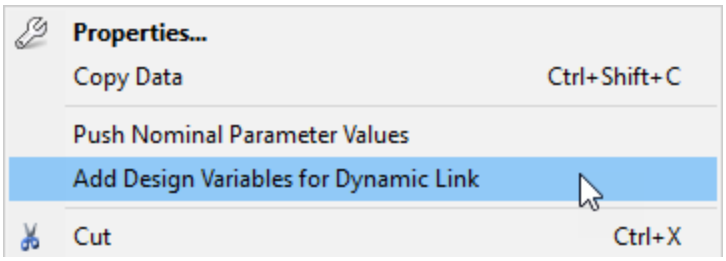
To create design variables based on variables added to a dynamic link design after it was linked:

1. Create a new design variable in the dynamic link project.

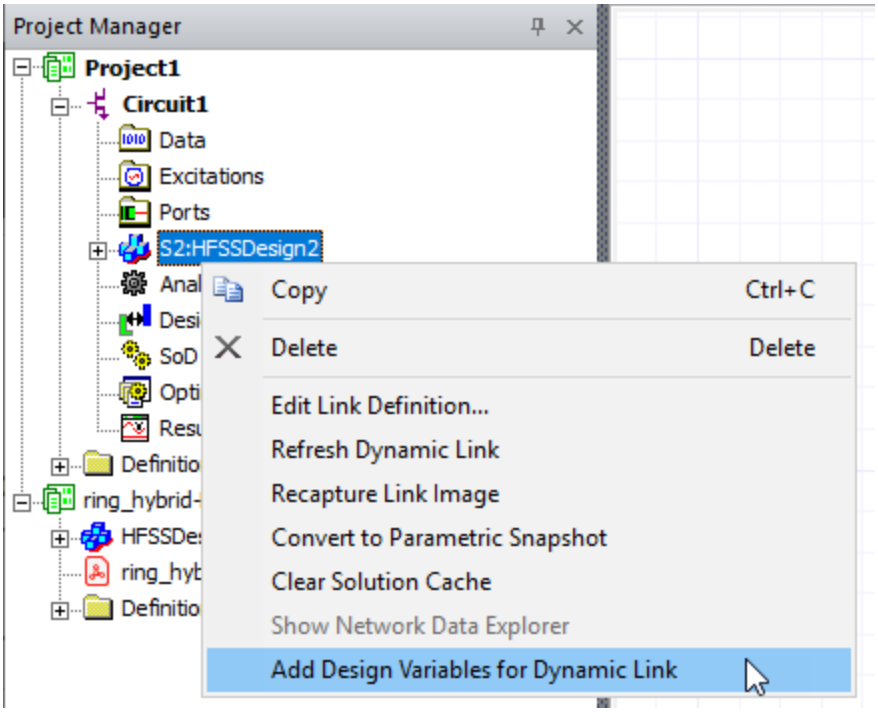
2. In the Project Manager of the Circuit or 3D Layout design, Right-click *Project Name* > *Design Name* > *Dynamic Link Design* and click **Refresh Dynamic Link**.



3. Either:
   a. In the Schematic or Layout editor, right-click the new instance of the linked design and select **Add Design Variables for Dynamic Link**.

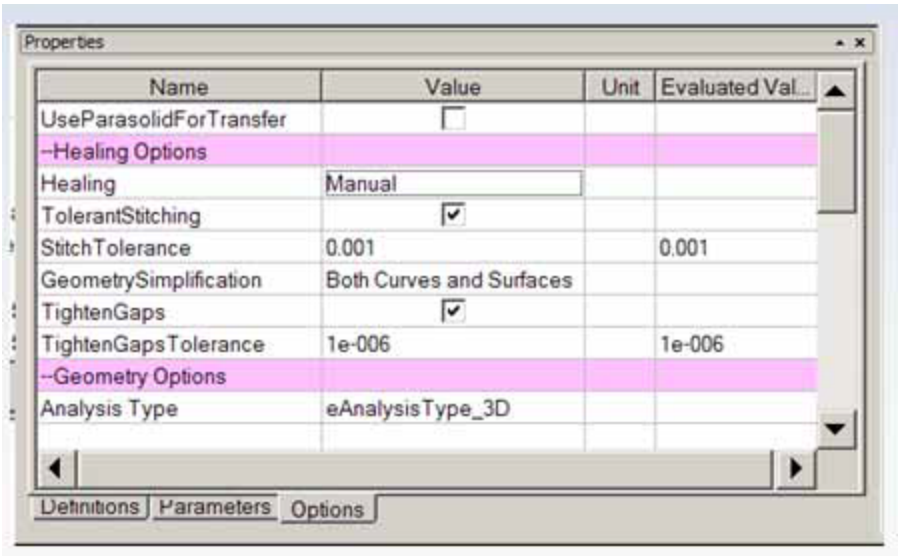


b. From the **Project Manager** window>, right-click **Project Name** > **Design Name** > **Dynamic Link Design** and select **Add Design Variables for Dynamic Link**.



Healing with CAD Integration

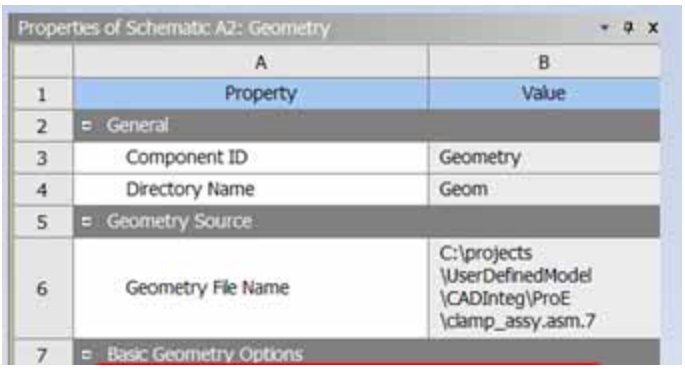
It is not possible to use the Heal command in Ansys Electromagnetics Modeler. Instead similar healing options are available under UDM properties option tab.



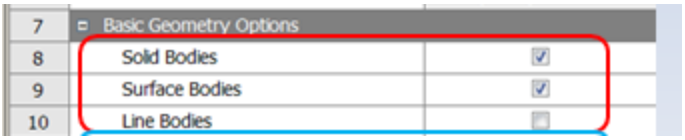
Healing options are None, Auto and Manual. By default, healing is off (None) and should be turned on only if required.

Important Geometry Options for CAD Integration

Select a Geometry Cell in Workbench to see options in the Properties window.

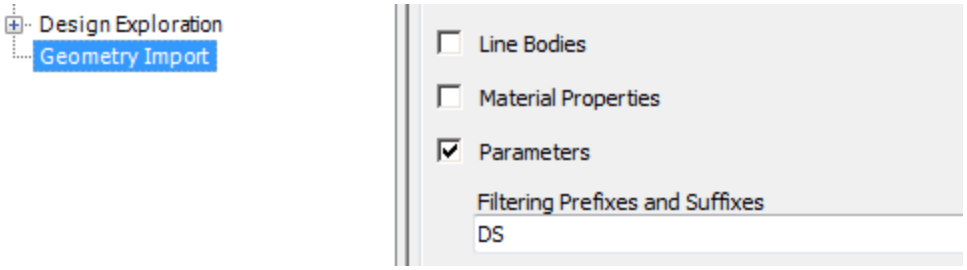


Control dimension of bodies coming from CAD



Make sure **Parameters** is checked and that the parameter key (filter) is appropriate for CAD parameters.

The default parameter key filtering means that only parameters whose names start with DS will come through. This is the default for the Workbench **Tools>Options** for **Geometry Import**. To import parameters with different prefixes or names, assign an appropriate prefix or clear the **Filtering Prefixes and Suffixes** field.



The **Attribute Key** should be empty or "Color" to bring in CAD colors.

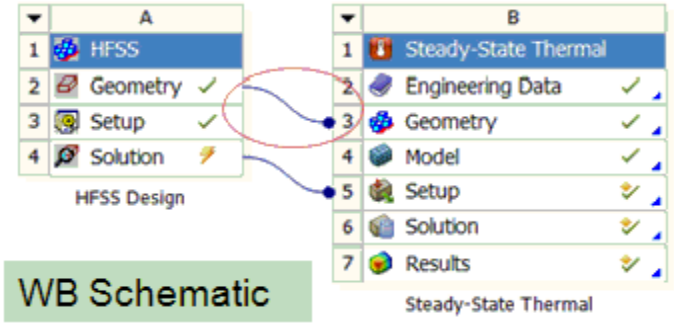
Material Properties must be checked to bring in the material assignment.

The **Mixed Import Resolution** option is used to resolve parts with mixed dimension (typically from Pro/E)

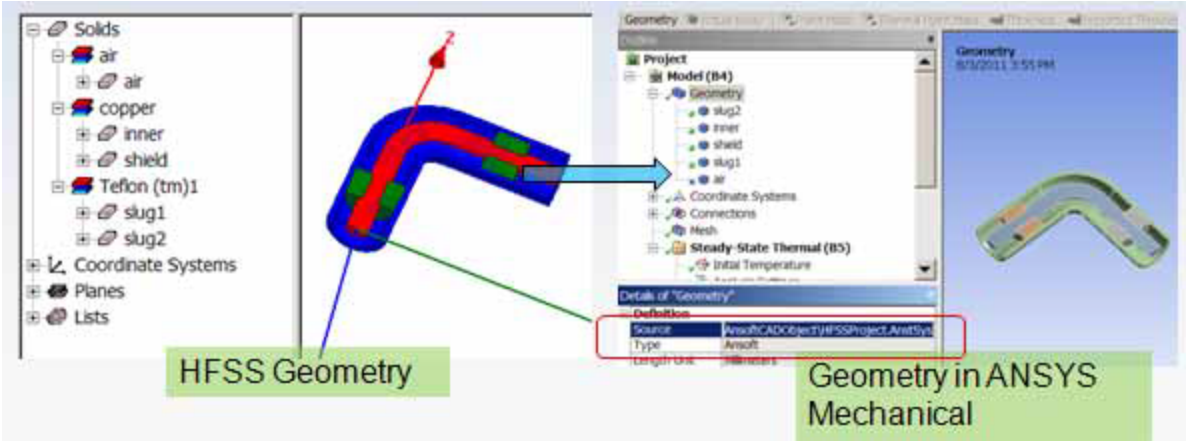
Refer to Ansys Help for details.

Ansys EM to Ansys Geometry Transfer

Ansys Electromagnetics CAD Integration transfers model information based on Workbench links.

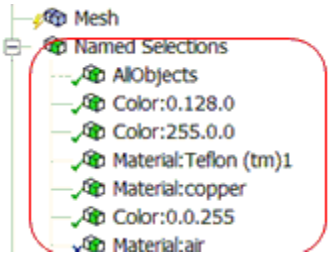


The following figure shows how the information is transferred between simulators.

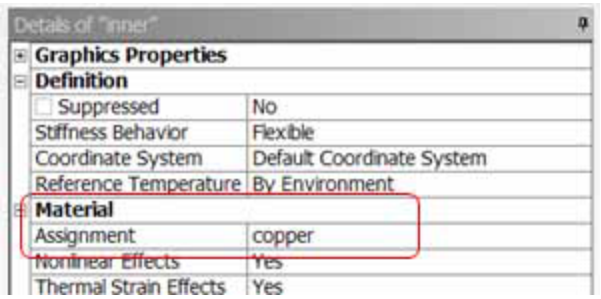


Information transferred includes:

- Geometry
- Ansys Electromagnetics lists and material assignment as Named Selection



- Material assignment



The CAD Integration geometry link is dynamic (you can get updated geometry from Ansys Electromagnetics) and associative (IDs persist between Ansys Electromagnetics model and Ansys model during model refresh).

Boundary conditions in Ansys are preserved. Object, material, and parameter names with non-ASCII characters are not allowed for data transfer. Such transfers fail and produce an error message.

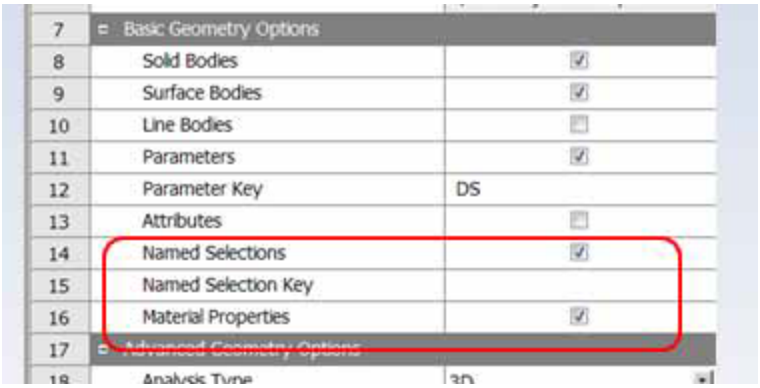
Material Assignment Transfer

There are two prerequisites for material assignment transfer:

- Engineering Data should have materials used in Ansys Electromagnetics model and material names should match (including case)
- Material Properties in Geometry Options must be checked

Before transferring as a named selection, **Named Selections** must be checked in **Basic Geometry Options**.

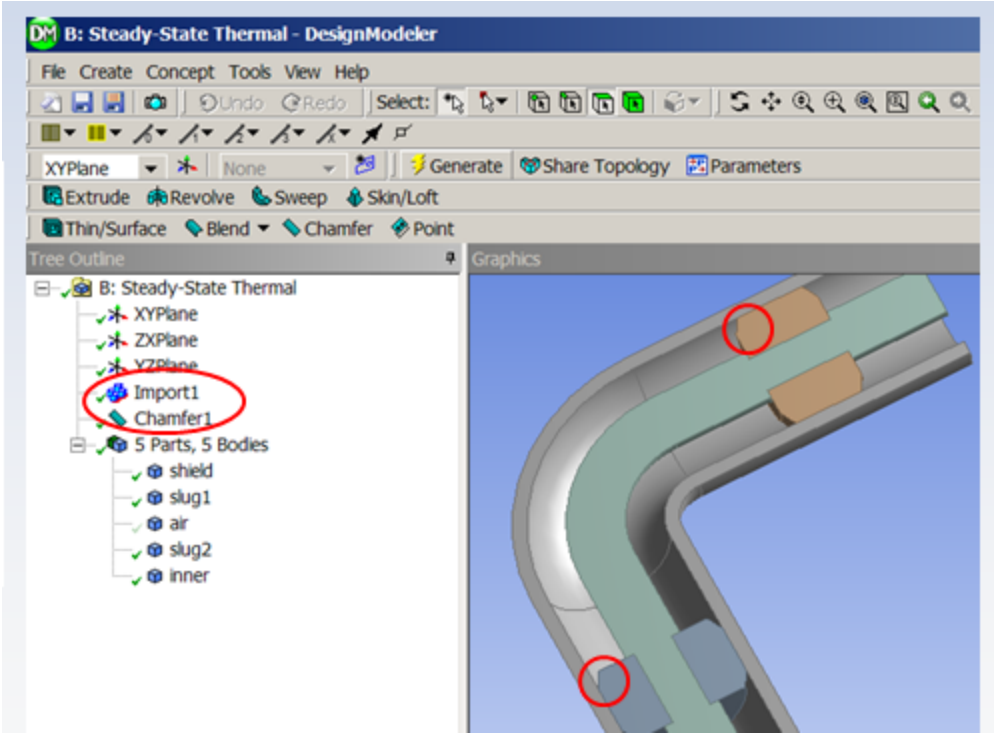
The Named Selection Key should either be empty or contain "Material."



Geometry Transfer through Ansys DesignModeler (DM)

It is possible to edit geometry in Ansys DesignModeler (DM) before consuming it in Mechanical. This is useful when geometry requires pre-processing for Ansys simulation.

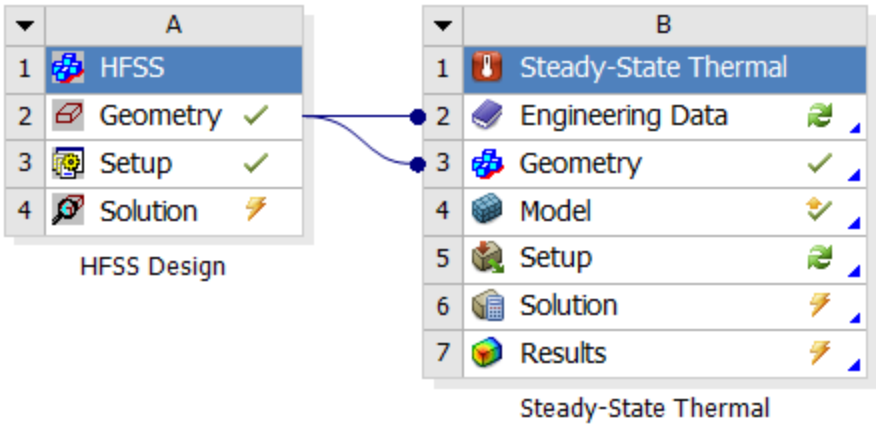
Consider an HFSS model linked to DM through the Workbench. In the following example, a chamfer operation on geometries is being imported.



Workbench Material Data Transfer

A link can be created from an Ansys Electromagnetics Desktop system (e.g., HFSS) geometry cell to a downstream Engineering Data (ED) cell. When the downstream ED cell is refreshed, it updates with materials (names and properties) 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 by dragging the Ansys Electromagnetics Desktop system geometry cell to the ED cell of a downstream system. When the downstream system's ED cell refreshes, the upstream system generates a MatML format XML file, which is consumed 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 Ansys Electromagnetics Desktop products.

For material assignments to work with the ED in the downstream model, the downstream geometry cell must have the **Material Properties** check box selected. A convenient way to do this is to have this property checked by default, by setting this in **Tools > Options > Geometry Import**. Systems created after this default option is set will have **Material Properties** automatically checked for the geometry cell.

When the downstream model cell is refreshed or updated, material properties from the ED in the same system are used in the model's materials, based on material assignments set during the geometry cell refresh. Note that if material assignment fails, the Workbench default material (Structural Steel) is 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 system uses a material named "Structural Steel", this will conflict with the existing downstream default material of the same name. The update will result in the Ansys Electromagnetics Desktop material being named "Structural Steel 2" in the downstream ED, but the downstream model will still use the properties supplied by Ansys Electromagnetics Desktop for the material.

Note that some materials may not satisfy all requirements for downstream system physics. For example, vacuum has no thermal conductivity, 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 modified after refresh to ensure that validity criteria can be met. Any property added to a material in the downstream ED is preserved in subsequent ED cell updates. However, any existing property that is edited in the downstream ED will be overwritten by a subsequent ED cell update.

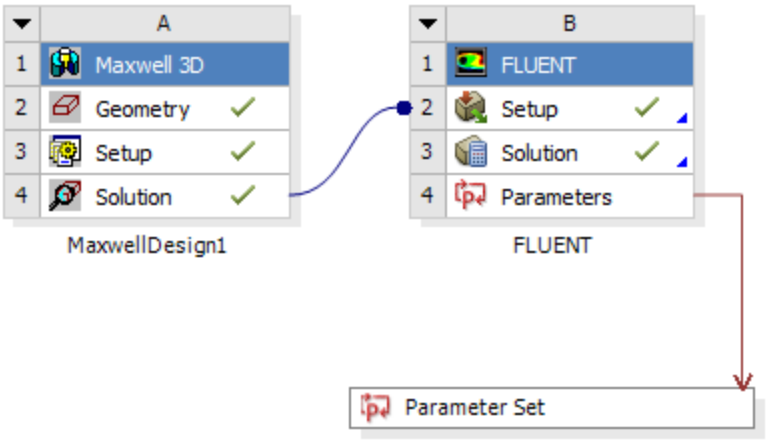
Feedback Iterator

Background Information

Prior to the introduction of the Feedback Iterator, Ansys Workbench supported a two-way loose-coupling protocol with Ansys Electromagnetics products.

SystemCoupling already uses the word coupling to mean low-level solver coupling. Existing Electromagnetics product coupling is loose compared to SystemCoupling and is limited to file transfers at the end of a complete solve in a standalone system/product. No communication occurs during a solve. The coupling is two-way.

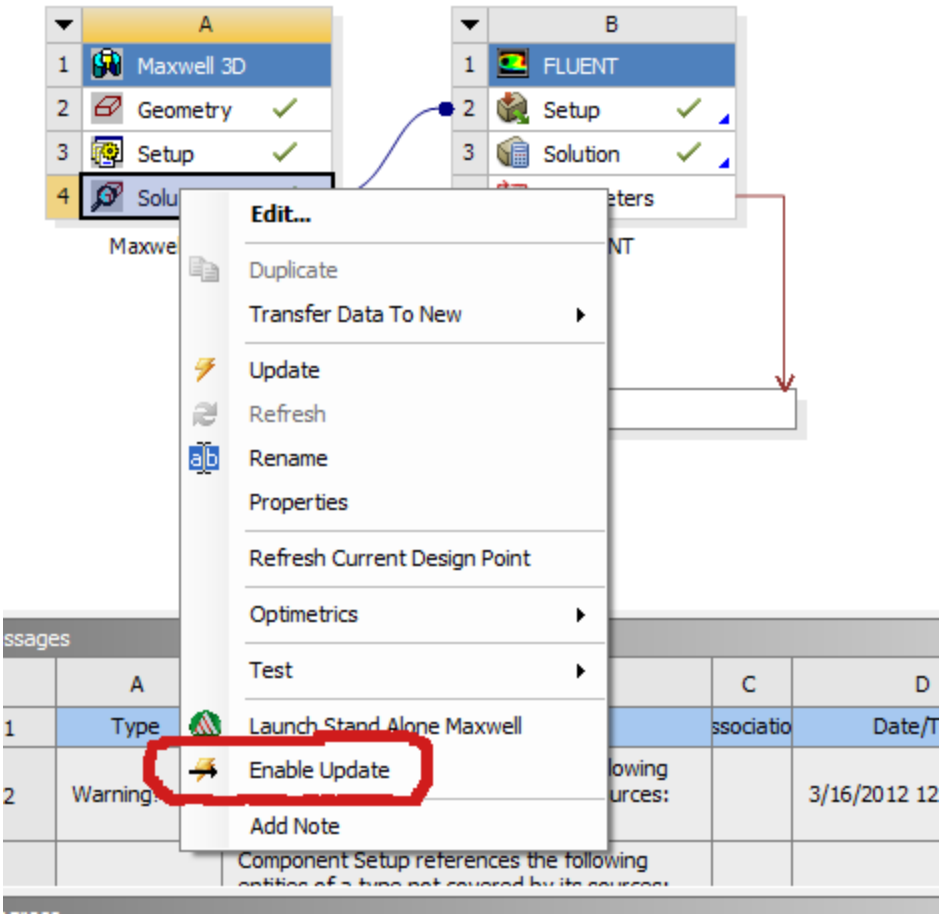
The one way portion (Upstream source component to Downstream target component) is handled via existing workbench data/transfer connection mechanism.



The round trip is handled by a separate protocol, agreed upon by the participating systems, whereby the downstream system exports a set of files 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.

To run the next coupled solve iteration, invoke the **Enable Update** operation, as shown below, which updates all systems involved.

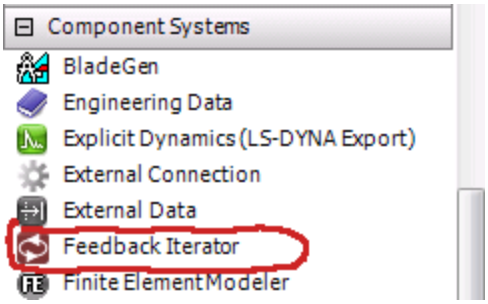
These steps (**Enable Update**, **Update Project**) are continued as long as needed.



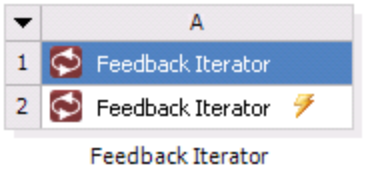
The Feedback Iterator System

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 control over the number of iterations (iteration termination criteria), target temperature, and displacement convergence criteria.

The **Feedback Iterator** appears in the Workbench Toolbox user interface under **Component Systems**.



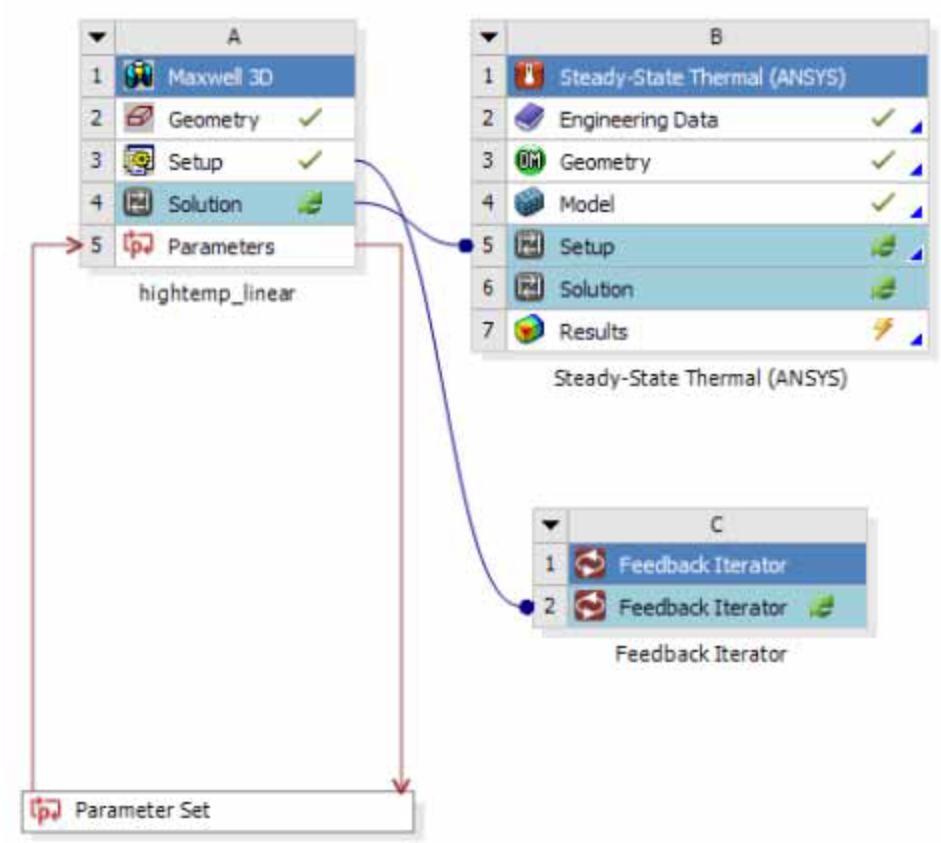
If you drag and drop an instance of the Feedback Iterator into Workbench Schematic, the default system name is Feedback Iterator.



Feedback Iterator in Use

The Feedback Iterator system attaches to the Workbench schematic like any other system.

In the following example, a Maxwell system provides AnsoftHeatLossData to a fluent system. Dropping the Maxwell setup cell onto the Feedback Iterator component results in the links shown.



The Feedback Iterator Addin functions the same way with all systems that participate in Electromagnetics product two-way loose coupling (Maxwell and HFSS for sources and Fluent, Thermal, and Structural for targets).

Feedback Iterator Component Properties

The Feedback Iterator's primary role is to control iterative solves and its properties target this end.

For Maxwell and HFSS, the Feedback Iterator properties are

- **Iterations Completed** – a read-only property that displays the number of iterations completed.
- **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. The default value is 100.
- **Target Delta Temperature %** – specifies the maximum *Target Delta Temperature %* that signifies convergence. The default value is 5%, and the value must be 0.01 or greater.

  > **Note:**
  >
  > • The delta temperature error is calculated in Kelvins.
  >
  > • The Maxwell and HFSS design Profile tab displays 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 %* that signifies convergence. The default value is 5%, and the value must be 0.01 or greater.

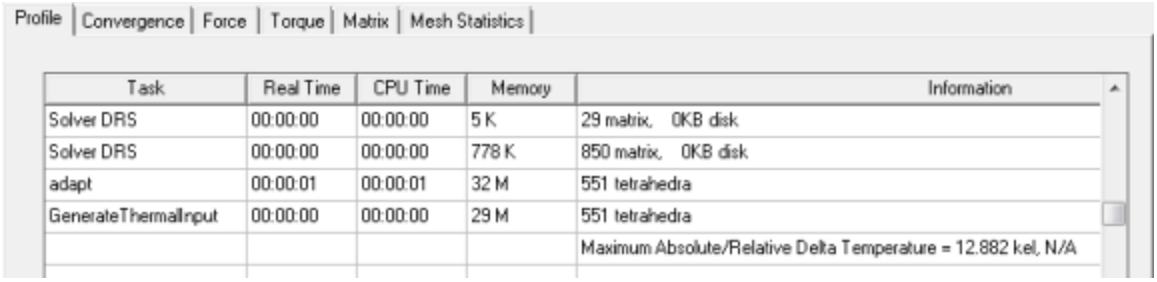
	A	B
1	Property	Value
2	⊟ General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	⊞ Notes	
7	⊞ Used Licenses	
9	⊟ Iterations	
10	Iterations Completed	0
11	Max Iterations	100
12	⊞ Callback	
14	⊟ Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	Not Available
17	⊟ Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	Not Available

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 satisfied with the achieved deltas.

> **Note:**
>
> Both the Temperature and Displacement Convergence properties sections are always shown irrespective of which feedback types are enabled for the project.

Running a two-way feedback simulation from Workbench while keeping the Maxwell or HFSS Profile tab open shows that the 3D solver keeps track of two feedback related quantities:

- Maximum Absolute/Relative Delta Temperature (if temperature feedback is enabled)
- Maximum Absolute/Relative Delta Displacement (if displacement feedback is enabled)

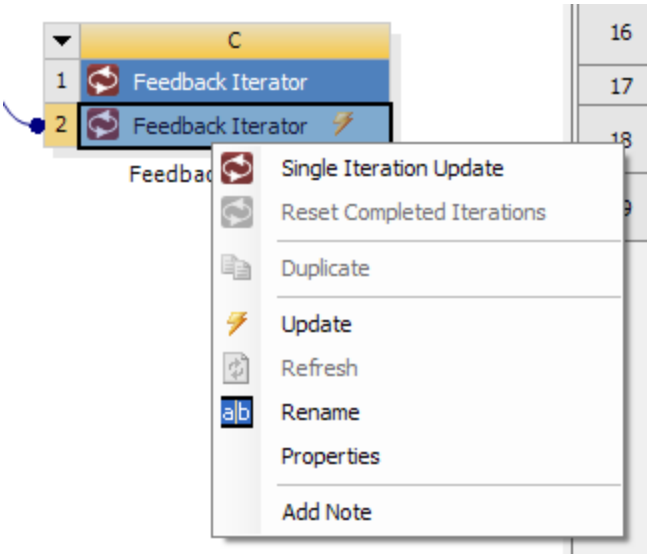


These signify the maximum difference on the solution mesh of the quantity in question from one feedback iteration to the next (thus the value is only available from the second iteration onward). This data forms the basis for the convergence controls in the FeedbackIterator. These controls allow the user to base the termination of feedback iterations on a target max delta T and/or max delta displacement. The target max delta forms the convergence condition.

Feedback Iterator GUI Operations

The **Single Iteration Update** action allows you to run a single iteration worth of updates, as follows:

1. Updates the Electromagnetics product system, which will incorporate any previous feedback from the downstream system.
2. Updates the downstream system.
3. All coupled systems (Electromagnetics product, Electromagnetics product Downstream, and Iterator) are in the UpToDate state.
4. Iterator's “Iterations Completed” property increments by 1.



Right-clicking 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.

Callback Interface

The callback interface allows you to react to each step of the iteration process as implemented by the FeedbackIterator component. This is useful for implementing special iterations or solving transient simulations with a limited scope.

There are four main API features provided:

- Callback functions, which the FeedbackIterator calls at various points in each iteration
- Utility functions, which the Callback functions can use to extra properties of the containers being processed
- Output functions, which allow script debugging and supply additional messages
- Limited state management, which allows the script to store and retrieve state across callback functions and iterations

Callback and State API

State is managed 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, 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 box with the supplied string with an **OK** button

Example Scenarios for Feedback Iterator

This section describes several scenarios for using the Feedback Iterator:

Setting 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 Electromagnetics product OR • Create Feedback iterator system and connect Electromagnetics product setup component to the TwoWay iterator component AND	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 • The cell states of all coupled cells (master and clients) are synchronized to be the

Scenario	Steps to execute	Outcome
	User clicks on the Iterator component and sets the desired Feedback Iterator properties.	most pessimistic state among any of them (e.g., after an update, any modification will set all of them to the "Modified" state).
Electromagnetics product solution provides to multiple downstream setup cells	same as above	All the downstream components (setup and 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.

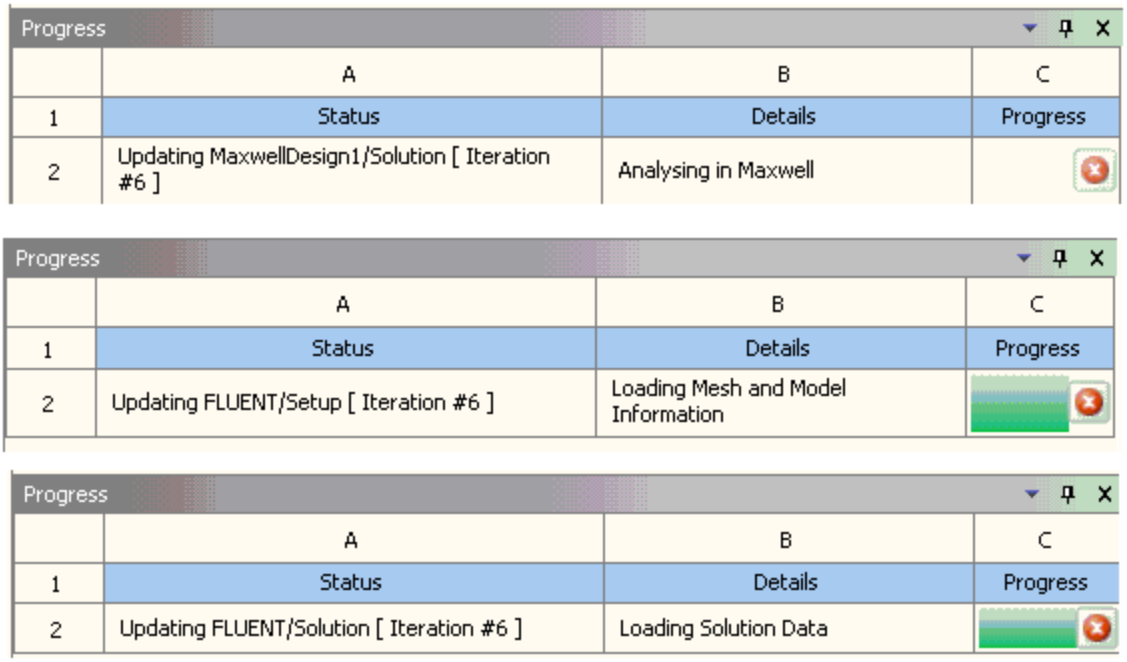
Breaking Iteration Control

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 iterator OR • Delete Electromagnetics product solution consumer 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)

Scenario	Steps to execute	Outcome
	OR • Delete Iterator system	
Electromagnetics product setup cell is connected to iterator but Electromagnetics product solution does not provide anything. No components are coupled to the iterator	• Break Electromagnetics product setup link to iterator • Delete Iterator system	No coupling exists in this scenario so nothing visible changes beyond what is expected from the user actions

Starting an Iterative Update

Scenario	Steps to execute	Outcome
Standard update scenario	• RMB on iterator component and select Update GUI 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)



Running a Single Iteration

Scenario	Steps to execute	Outcome
NumCompleted < NumIterations	RMB on iterator component and select Single Iteration Update	One iteration is run. "Completed Iterations" property is incremented.
NumCompleted ≥ NumIterations	same as above	same as above

Interrupting an Iterative Loop

Currently, there is no special progress or interruption control. The progress monitor for individual components displays as they are updated and any control they choose to provide (interrupt, abort or both) is available. Choosing to abort any of the client component updates aborts the current iteration, and the Completed Iterations property remains unchanged from the previous iteration.

Resuming an Interrupted Iterative Loop

Scenario	Steps to execute	Outcome
Single step	RMB on iterator component and select Single Iteration Update	One iteration is always run and the Completed iterations property is incremented.

Scenario	Steps to execute	Outcome
Run until completion	• RMB 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.

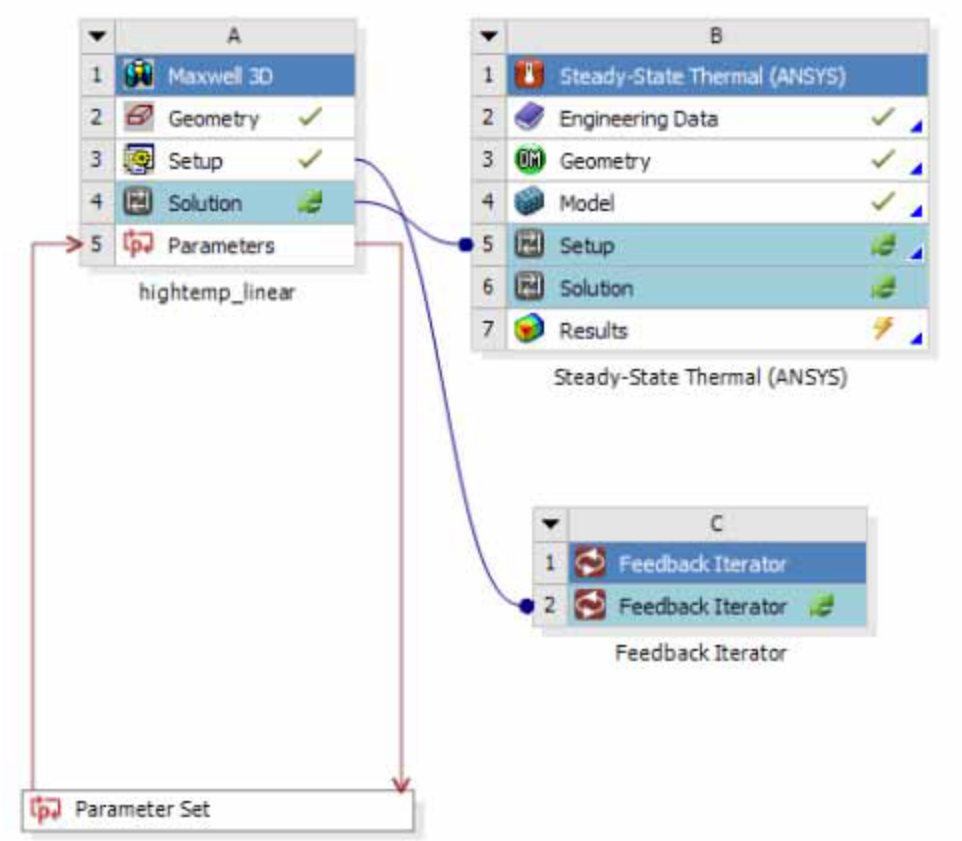
Modifying any of the Systems Involved in Iteration (Coupled Clients)

When any coupled client components are modified (either in Workbench or in an external editor), the coupled clients and master are marked as modified. The Completed Iterations property is 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 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)

The following temperature convergence example uses 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, because it has no previous iteration, cannot return a meaningful delta value.

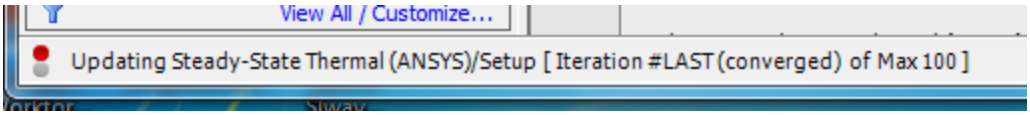
		started. (9:51:41 PM Jun 11, 2013)		
6	Informational	After iteration#: 1, Max temperature delta % = NaN, Max displacement delta % = NaN		6/11/2013 9:51:39 PM

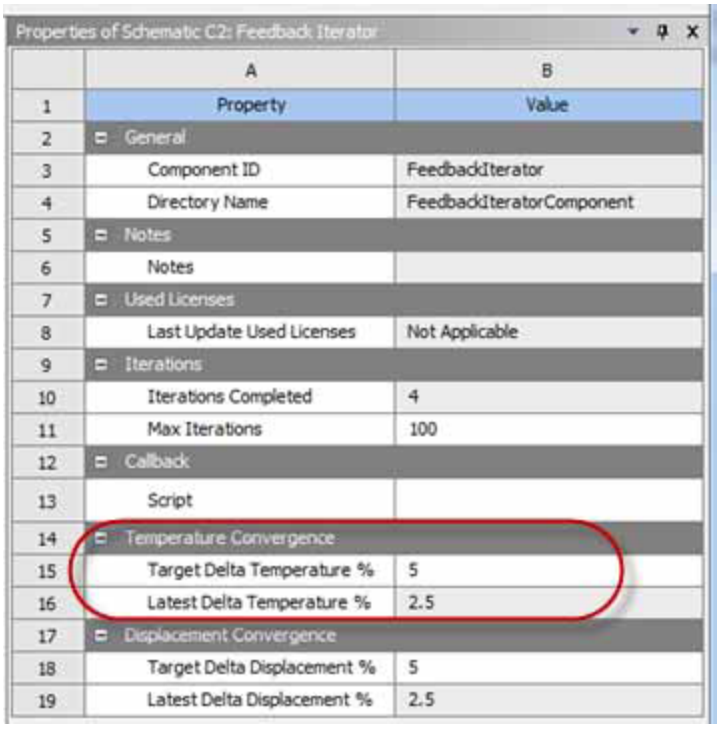
At the end of this iteration, the Latest Delta value is listed as Not Available.

	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

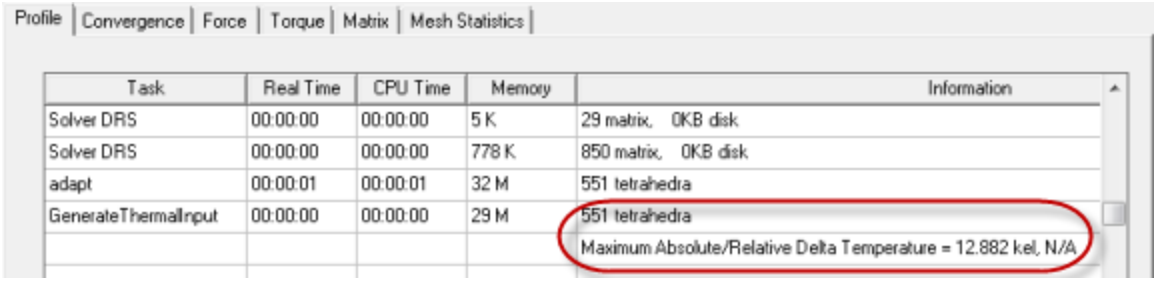
In this example, convergence has been achieved by the third iteration.





When convergence criteria are met, the simulation stops. If you attempt to update after convergence has been achieved, the simulation will not be launched since the current solutions satisfy the convergence criteria.

You can monitor the iteration progress by opening the Maxwell editor and selecting **Results** > **Solution data**. Choose the **Profile** tab and keep the window 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.



21 - Scripting

Information about creating, editing, and using IronPython scripts and VB scripts is in the product's scripting guide in the help. You can also access PDF versions of the scripting guide from the help system.

22 - Q3D Extractor Technical Notes

This section describes, in more technical detail, the solution processes used by the 3D field solvers in Q3D Extractor.

Please select a subtopic:

AC vs. DC Analysis

Q3D Extractor can compute resistance and inductance matrices for AC or DC problems.

AC Region

In the AC region, inductance is nearly constant with frequency. The AC self-inductance is lower than the DC inductance because skin effect reduces the magnetic fields (and the corresponding stored inductive energy) inside the conductors. Only the stored inductive energy in the region between conductors remains.

Resistance in the AC region increases proportionately with the square root of frequency. This occurs because the skin depth decreases with frequency, reducing the effective cross-section for current flow.

DC Region

In the DC region, resistance and inductance are both nearly constant with frequency.

Transition Region

Between the DC and AC regions of operation is a region that spans about 1 decade of frequency where neither the DC nor the AC models are truly valid. Here the skin depth is an appreciable fraction of the conductor depth. See: Estimating R and L Values in the Transition Region.

Frequency Ranges in AC Calculations

AC solvers compute high frequency asymptotes of inductance and resistance. While external inductance values do not grow with frequency, the resistance values are unbounded and increase with the square root of frequency. Therefore, the operating frequency must be specified to allow the software to calculate a finite resistance. The default operating frequency is 1 GHz. Alternatively, you can alter the frequency through matrix reduction. The skin depth used by the software is given by the following relationship (for reference, remember that the skin depth in copper (σ = 5.8 x 10^7 S/m) at 1 GHz is 2.1 µm):

where:

$$\delta = \sqrt{\frac{2}{\omega\sigma\mu_0\mu_r}}$$

- ω is the angular frequency, which is equal to $2\pi f$ (*f* is the frequency in hertz)
- σ is the conductor 's conductivity in S/m.
- μ_r is the conductor 's relative permeability. For most metals, $\mu_r = 1$.
- μ_0 is the permeability of free space, which equals $4\pi \times 10^{-7}$ Wb/Am (or $4\pi \times 10^{-7}$ H/m).

Thus, given a conductor with thickness ***d***, you can calculate the lower bound of the AC Region by calculating the smallest frequency that produces a skin depth (δ) much smaller (say 3 times smaller) than this thickness. Using the formula for d and the definition of $\omega = 2\pi f$, you have the frequency limit:

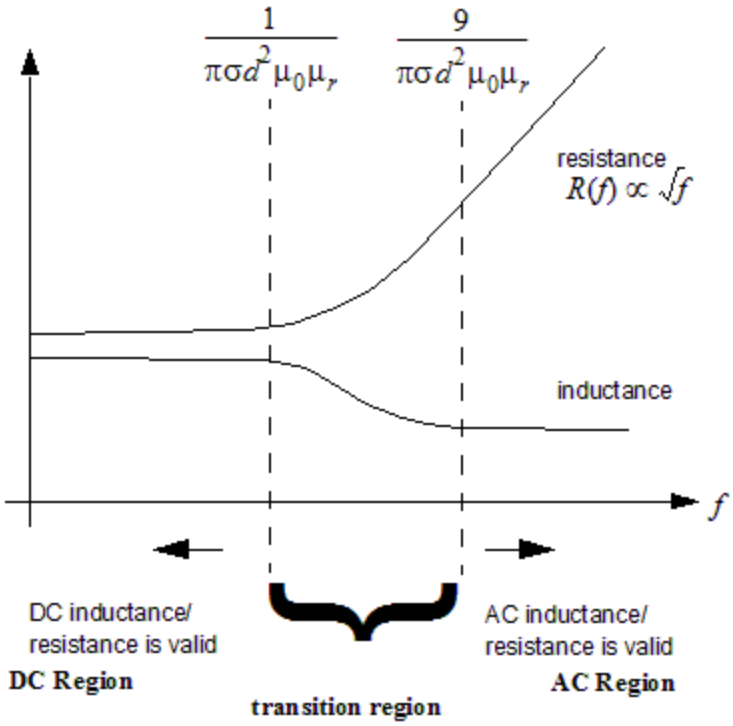
$$f \geq \frac{9}{\pi\sigma d^2\mu_0\mu_r}$$

For copper, this reduces to $f \geq (0.04 \text{ Hz m}^2)/d^2$. For a 1 mm thick conductor, the AC inductance/resistance calculation is good for frequencies of 40 kHz or more. For a 100 µm thick conductor, the minimum frequency rises to 4 MHz, and so forth.

A similar calculation can be performed to determine the upper frequency bound for DC resistance and inductance calculations. By assuming that the skin depth must be greater than the conductor thickness *d*, the frequency becomes:

$$f \leq \frac{1}{\pi\sigma d^2\mu_0\mu_r}$$

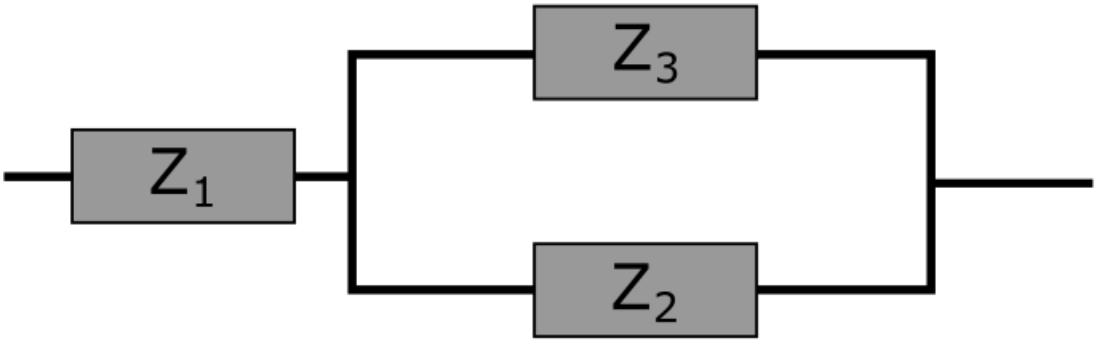
The following figure shows how typical circuit parameters vary according to frequency and illustrates the different frequency regions where the parameter calculations are valid:



Estimating the R and L Values in the Transition Region

Q3D Extractor estimates the inductance and resistance values in the transition region. The estimate is based upon the DC and AC asymptotes.

An equivalent circuit model (shown below) is used to interpolate between the DC and AC solutions:



$$Z_1 = R_{dc} + j\omega L_{ac}$$

$$Z_2 = j\omega \left(L_{dc} - L_{ac} \right)$$

$$Z_3 = R_{ac} \sqrt{j\omega / \pi f_{ac}}$$

where:

- R_{dc} and L_{dc} are the DC RL values
- R_{ac} and L_{ac} are the AC RL values at frequency f_{ac}
- f is the frequency

This circuit has the correct limiting behavior at low and high frequencies. At intermediate frequencies, it has a frequency-dependent impedance:

$$Z\left(j\omega\right) = R\left(\omega\right) + j\omega L\left(\omega\right)$$

The effective resistance *R(ω)* and inductance *L(ω)* can be extracted from the real and imaginary parts of this impedance.

When there are multiple conduction paths in the problem, all of the elements in the above circuit become multi-port devices, but the same basic procedure applies.

The interpolation scheme described above is not a solution of the underlying physics, but rather is intended as a rough estimate of the behavior at intermediate frequencies. A more rigorous solution can be obtained by employing a finite element field solver, such as Maxwell Field Solver, to solve the eddy current equation within the conductors.

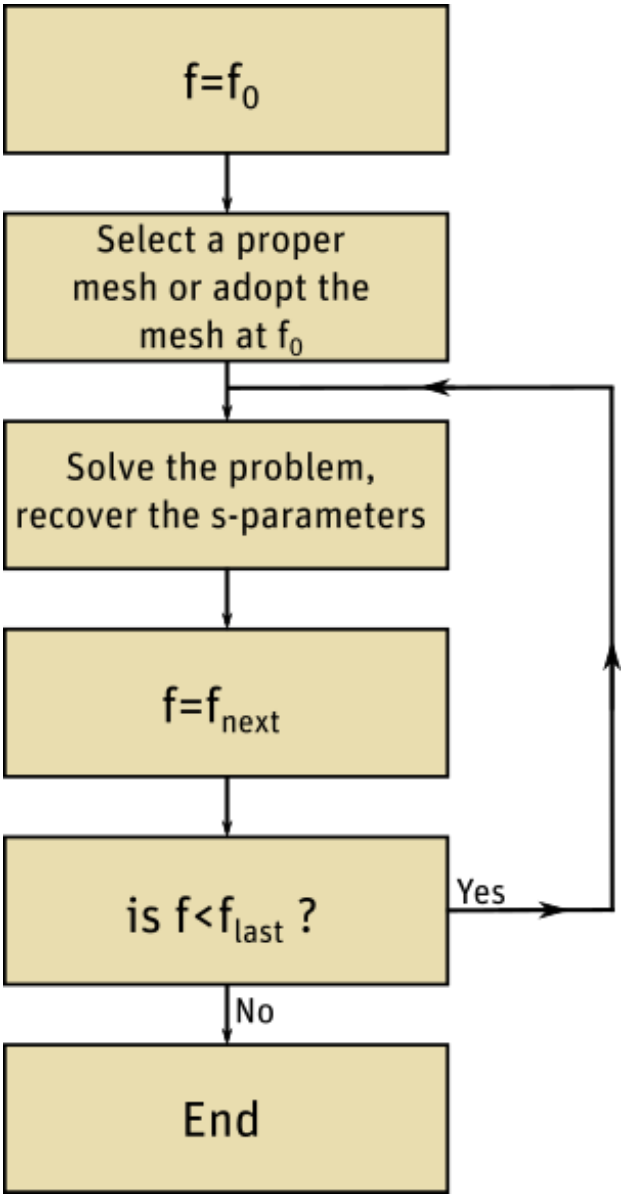
Discrete Frequency Sweeps in Q3D Extractor

A Discrete sweep generates field solutions at specific frequency points in a frequency range. For example, if you specify a range of 1000 MHz to 2000 MHz then a Step Size of 2.5, the result would be solutions at 1000, 1250, 1500, 1750, and 2000 MHz. Select the **Save Fields** option when setting up the points to solve if you want to save the fields (only applies to CG fields). The more steps requested, the longer it takes to complete the frequency sweep.

Choose a discrete sweep if only a few frequency points are necessary to accurately represent the results in a frequency range.

Be aware that Q3D Extractor uses the finite element mesh refined during an adaptive solution at the solution frequency. The solution frequency for the Solution Setup should be chosen taking into consideration that the well-established skin effect is assumed at that frequency; therefore, the solution frequency has to be chosen so that this assumption is true and correct. The frequency must be in the AC region for the model under analysis.

The procedure for a discrete frequency sweep is shown below, where *n* equally spaced frequencies are included in the sweep.



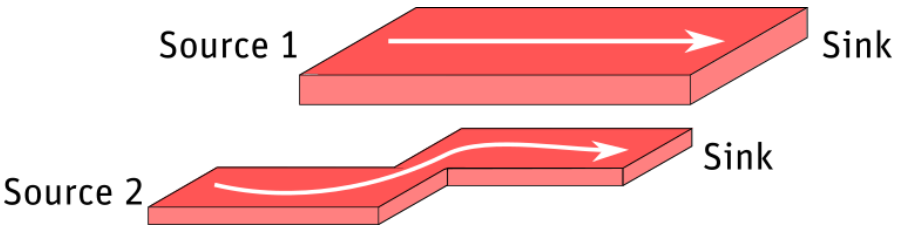
Equivalent Circuits

When you export a SPICE subcircuit, the Q3D Extractor constructs an equivalent circuit for your model using the circuit parameters you have requested.

Single-Source Conductors

If all conductors have just one source terminal and one sink terminal, the system creates a balanced circuit to model the transmission structure — that is, a circuit whose impedance is the same regardless of the direction of current flow. For instance, a two-conductor transmission

structure is shown below. Each conductor has a single source terminal through which current flows.



The circuit parameter matrices for this model are:

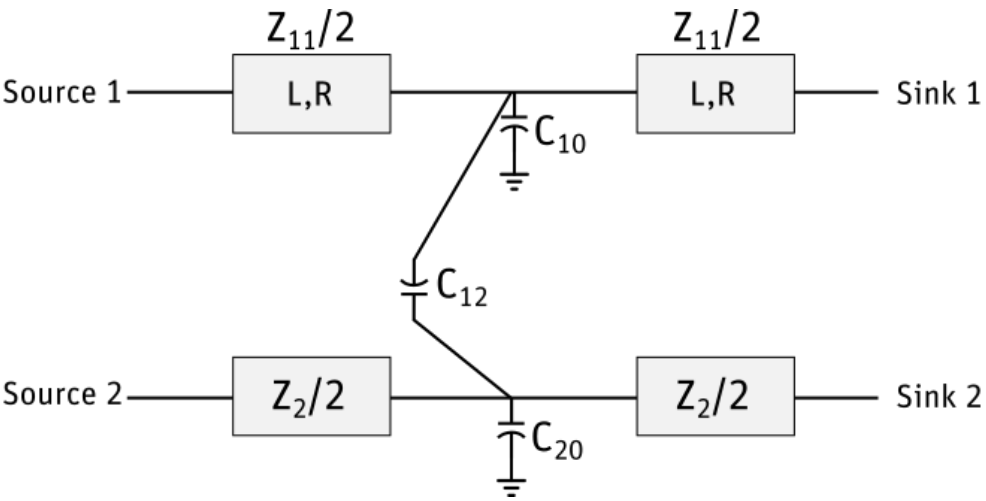
$$\begin{bmatrix} L_{11} & L_{12} \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} \\ C_{12} & C_{22} \end{bmatrix}$$

In the equivalent circuit for this structure, the negative of the mutual capacitance (C_{12}) is used directly as a circuit element. The self-capacitances of each conductor (C_{11}, C_{22}) are used to compute the capacitances between the conductor and ground (C_{10}, C_{20}).

$C_{10} = C_{11} + C_{12}$

$C_{20} = C_{22} + C_{12}$

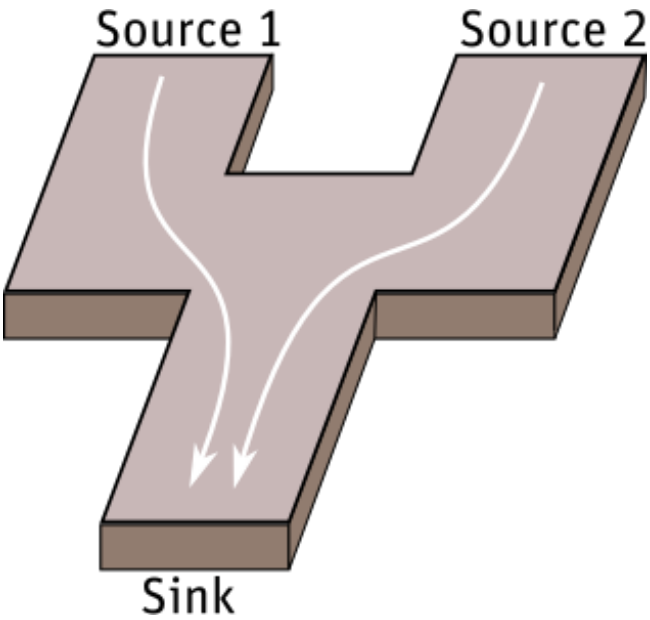
To create a balanced circuit, each inductance or resistance matrix entry is divided into two series inductors or resistors and placed in the circuit as shown in the following figure:



All sinks on a conductor are considered to be connected to each other. Sinks on different conductors are independent.

Multiple-Source Conductors

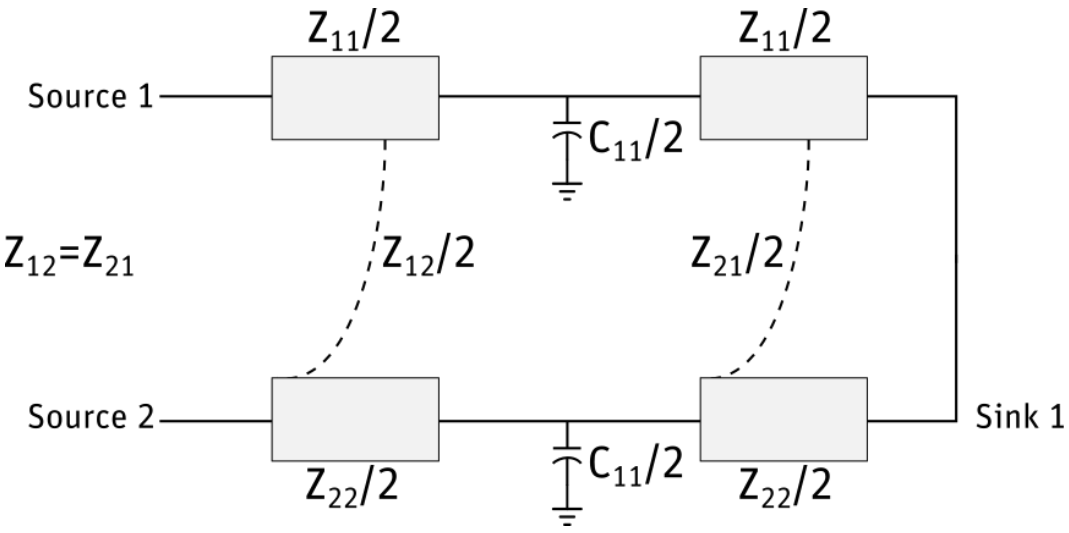
Balanced models are used for multiple-source nets. The conductor in the following figure has two source terminals.



The circuit parameter matrices are:

$$\begin{bmatrix} L_{11} & L_{12} \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{bmatrix} \begin{bmatrix} C_{11} \end{bmatrix}$$

To create a balanced model of this structure, two "T" sections are used (just as if there were separate nets) and then the two are shorted together at the sink point:



The capacitance of the net is evenly divided between the two "T" sections.

Understanding Equivalent Circuit Models

This section explains the topology of the RLGC circuit models produced by Q3D Extractor. It uses a combination of quasi-static field solvers to model interconnect structures. An electrostatic solver is used for capacitance (C), and a magneto-quasi-static solver for inductance (L) and resistance (R). The RLGC matrices are then combined into a circuit model to represent the overall behavior of the structure. Usually this quasi-static model is a good approximation to the structure's full-wave response up to a frequency corresponding to about ¼ wavelength.

When the dimensions of the RLGC matrices become large, these circuit models become very lengthy, making it difficult for users to untangle the basic topology of the equivalent circuits. While this is inevitable, it's also unfortunate, because the topologies are simple.

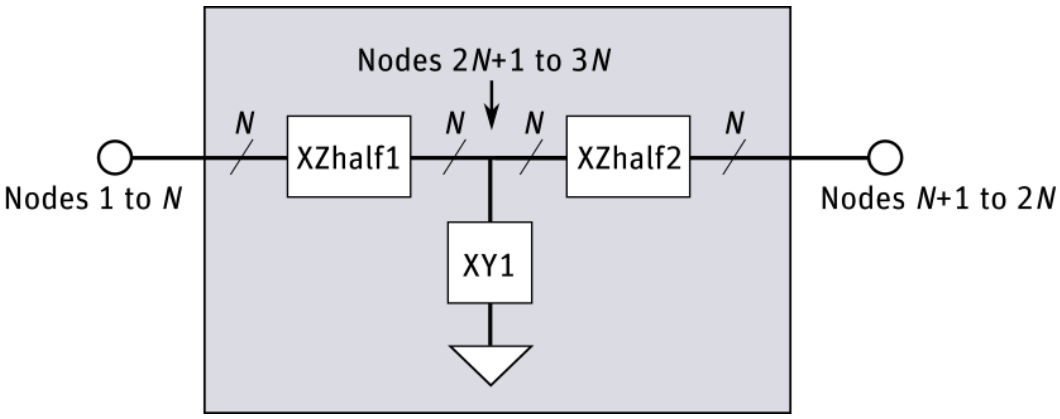
The T Model

To begin, let's assume that we have N separate nets (independent electrically connected paths) and that each net has exactly one source and sink. Later we will expand the discussion to consider nets with multiple sources.

The most commonly used equivalent circuit model is the "T" model. It is used when the number of lumps requested is 1. It consists of a grounded "shunt" element (XY1) representing the capacitance matrix and a pair of "series" elements (XZhalf1 and XZhalf2) representing the inductance and resistance matrices. Note that we are assuming that you have actually solved for both sets of matrices-if not, then the model will be simplified further. We postpone discussing these special cases to a later section.

All three subcircuits are contained within a single top-level subcircuit. The terminals of the top-level subcircuit are nodes 1 through 2N. Nodes 1 to N correspond to source terminals and nodes to correspond to the sink terminals. To provide connection points between the internal subcircuits, the top-level subcircuit has a set of internal nodes (nodes 2N+1 through 3N.)

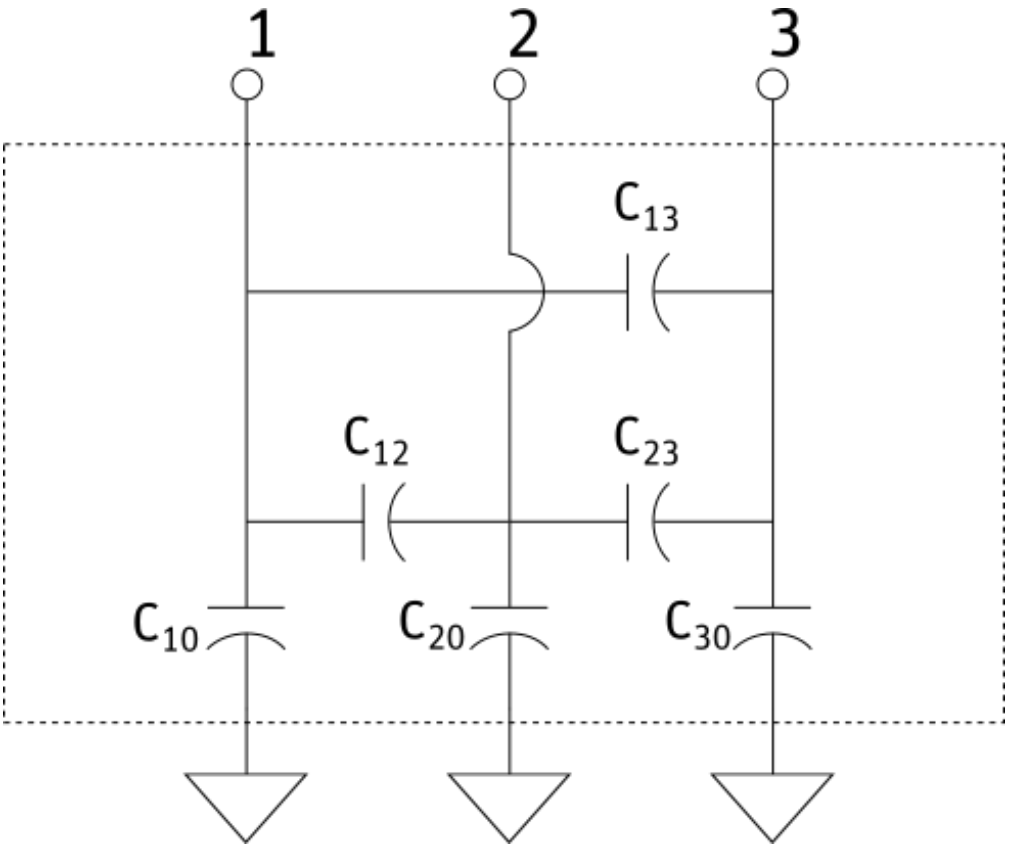
The T model is shown below.



The "T" model is used to represent single-lump equivalent circuits. The gray box in the figure represents the top-level subcircuit. It contains 3 lower-level subcircuit instances. The bold lines represent N-wire bundles or "buses" of wires.

The Shunt Subcircuit

The shunt subcircuit (XY1 in the T model figure) represents the capacitance matrix. The basic topology is illustrated in the figure below for the case of 3 different nets.



The topology of the shunt subcircuit XY1, illustrated for the case of 3 nets.

The values of the two-terminal capacitors in the subcircuit do not correspond directly to the entries of the Maxwell capacitance matrix displayed in the software. The "floating" capacitors (C_{12}, $C_{13}$and C_{23}) are the same as the corresponding off-diagonal entries of the capacitance matrix, but the signs are flipped from negative to positive. The grounded capacitors (C_{10},C_{20} and C_{30}) are actually the sums across the corresponding row of the Maxwell capacitance matrix. If the row sum is zero (this happens when a "floating at infinity" matrix reduction operation is used), then the capacitor is eliminated from the circuit.

> **Note:**
>
> If you only solve for capacitance, then only the shunt subcircuit will be produced and it will be the top-level subcircuit.

The Series Subcircuit

The series subcircuits represent the inductance and resistance matrices. For the T model, divide the total L and R matrices by 2 and put half on each side so that the internal structures of

subcircuits XZhalf1 and XZhalf2 are identical.

> **Note:**
>
> If you do not have to solve for capacitance, then you don 't need to divide the series elements in 2. There will be only a single series subcircuit in this case.

The self and mutual inductances are modeled in the standard way, using Spice's L and K elements. The self resistance is modeled using Spice's R elements, but there is no element in Spice corresponding directly to the mutual resistance (the off-diagonal entries of the R matrix). If we consider two mutually coupled resistors, their defining equations are:

$v_1=R_{11}i_1+R_{12}i_2$

$v_2=R_{21}i_1+R_{22}i_2$

We can rewrite these as:

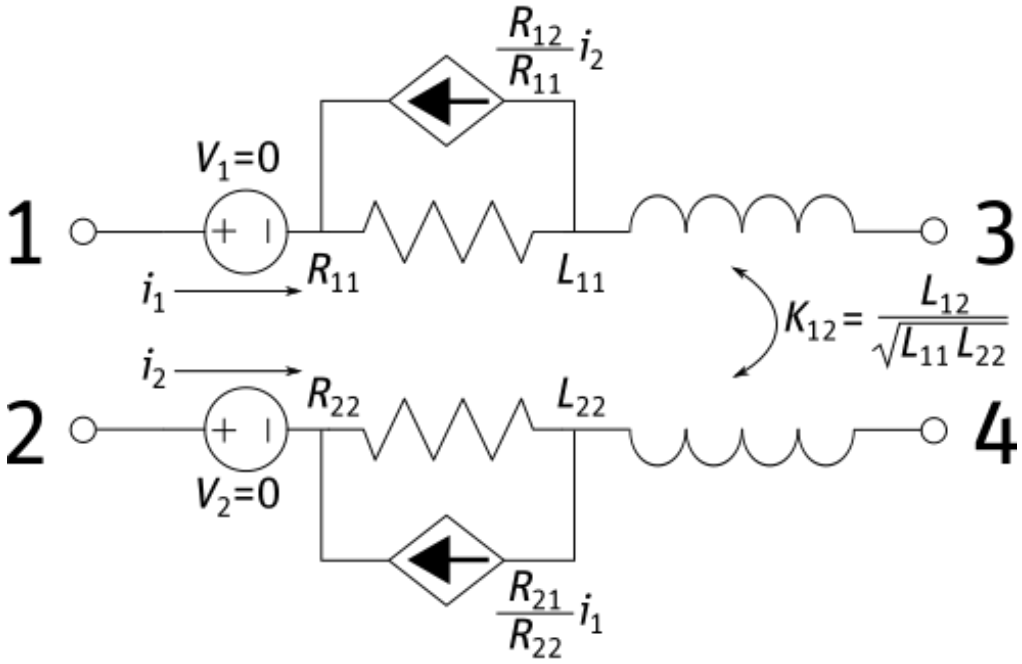
$v_1=R_{11}[i_1+R_{12}/R_{11}i_2]$

$v_2=R_{22}[R_{21}/R_{22}i_1+i_2]$

The desired voltage drop v_1 can be obtained by forcing a current equal to $i_1+(R_{12}/R_{11})i_2$ through the resistance R_{11}.

Similarly, we can force a current $(R_{21}/R_{22})i_1+i_2$ through a resistance R_{22} to get the voltage drop v_2.

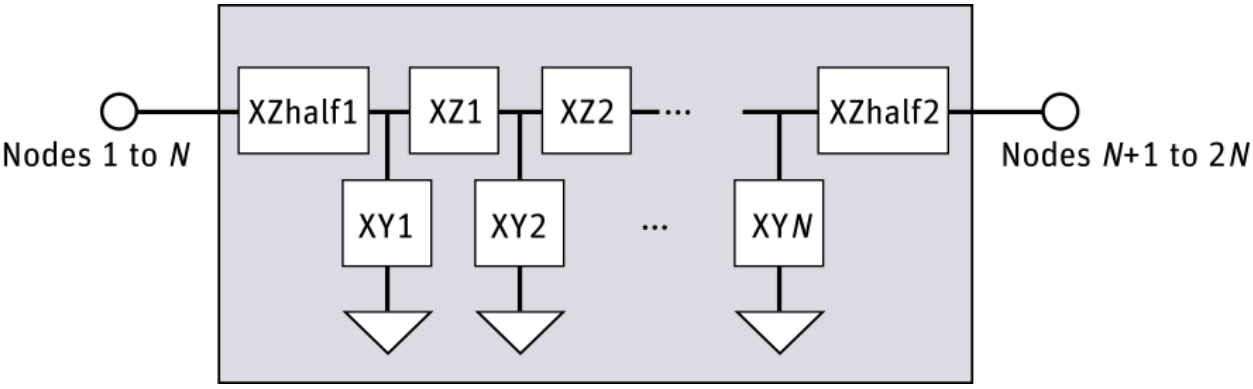
This can be accomplished by putting a current-controlled current source (the Spice F element) in parallel with the self-resistances R_{11} and R_{22}. But in order to use a current-controlled current source, we must have a way to measure the currents i_1 and i_2. This can be done by placing zero-valued voltage sources in series with R_{11} and R_{22}. These zero-valued sources don't change the total voltage drop, but do provide a convenient "ammeter" for measuring the currents. Thus we arrive at the final series circuit model shown in the following figure.



The series subcircuit, shown for the case of 2 nets. If there are more than 2 nets, additional controlled current sources will be added in parallel to model the mutual resistance coupling to them.

The Ladder Model

If you request a circuit with more than 1 lump, Q3D Extractor uses a ladder model. The basic structure of the ladder model is shown in the following figure.



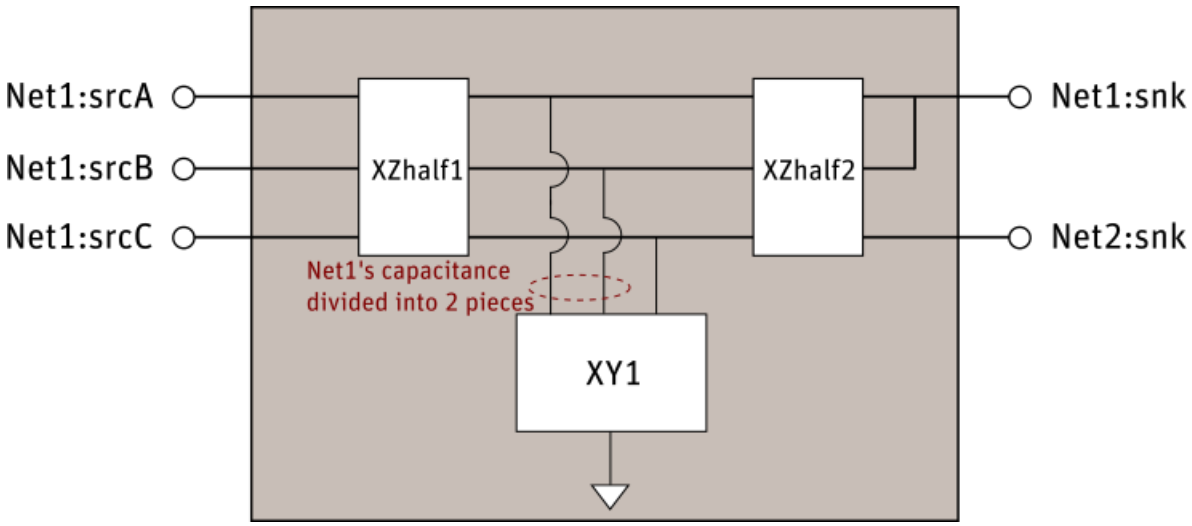
For an N-lump model, it consists of *N+1* series elements and *N* shunt elements. Each of the shunt elements (XY1, XY2, …, XYN) represents 1/N times the original capacitance matrix. Thus the total capacitance of the structure is equal to that in the original matrix.

The series elements are not all equal. On the left and right ends we use "half" elements (XZhalf1 and XZhalf2) whose R and L values are equal to 1/2N times the original resistance and inductance matrices. Between these we have (N-1) series elements (XZ1, XZ2, …, XZN-1) whose R and L values are equal to 1/N times the original. Again, the total resistance and inductance will be equal to that in the original matrices.

The ladder model is a generalization of the T model. Its chief advantage is that it provides a higher bandwidth model of structures with uniform cross-sections (such as transmission lines). The structures of the shunt and series elements are the same, and only the RLGC element values are scaled differently.

Presence of Multi-Source Nets

When a net has more than 1 source, the circuit topologies described above are modified to represent the fact that the different current paths on the same net converge at the sink point. This is conveniently handled by the top-level subcircuit. As an example, consider a problem with 2 nets. Net 1 has 2 sources and Net 2 has 1. There are 3 sources total and 2 sinks. The top-level circuit will have 3 terminals on the left representing the sources and 2 terminals on the right representing the sinks. Internally, the circuit structure is unmodified, but the connections from the last series subcircuit on the right to the top-level circuit terminals are changed as shown in the following figure.



An example of an equivalent circuit for nets that have multiple sources (in this case, Net1 has 2 sources). The beige box represents the top-level subcircuit.

Notice that the wires in the figure are not bundled into buses. Net1's capacitance is divided by 2 within the shunt subcircuit XY1 and assigned to its two source-to-sink current paths.

To make this procedure work, the shunt capacitance subcircuit must be "expanded" so that its number of terminals matches the total number of sources. Internally Q3D achieves this by dividing up the self and mutual capacitances of nets with multiple sources. If there are M source terminals on a net, then each source-to-sink current path receives an equal share of 1/M times the total net capacitance. This procedure preserves the total capacitance on the net as well as the coupling capacitances between nets.

Field Simulation Methods

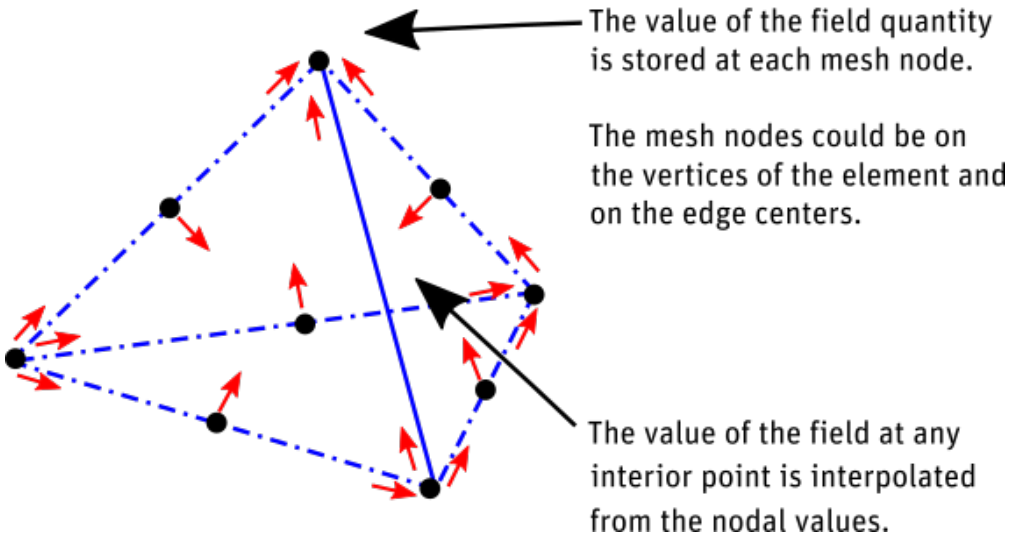
In order to generate an electromagnetic field solution, Q3D Extractor employs a combination of the Finite Element Method (FEM) and the Method of Moments (MoM).

In general, FEM divides the full problem space into smaller regions (tetrahedral elements) and represents the field in each sub-region with a local function. MoM divides up the surface (or volumes) of conductors and dielectrics into many triangular (or tetrahedral) elements to represent the charges and currents present.

The choice of which method to use depends on the quantity being calculated. Q3D solves DC resistance problems using FEM (to model current flow within the conductors), while inductance and capacitance problems are solved using MoM.

Representing a Field Quantity in the Finite Element Method (FEM)

A FEM solver stores the value of a field quantity (such as electric potential) at each mesh node of a triangular or tetrahedral element. Inside each element, the field is interpolated from the values stored at the mesh nodes using local finite element basis functions.



When field quantities are represented in this way, it is possible to transform Maxwell 's partial differential equations into a *sparse matrix* of linear algebraic equations ("sparse" because it has many zero-valued entries) that can be solved using traditional direct or iterative numerical methods.

Representing a Field Quantity in the Method of Moments (MoM)

Using MoM, field quantities are interpolated over elements, but typically with simpler basis functions than in Finite Element Method (FEM).

For capacitance problems, the field quantity of interest is the charge density on the surface of a conductor or dielectric interface. Triangular elements are used, and the charge density is approximated with a piecewise-constant basis function.

For inductance problems, the field quantity is a vector (current density), and again piecewise-constant basis functions are used. For DC inductance problems, the elements are tetrahedra (volume currents), and for AC inductance problems the elements are triangles (surface currents).

MoM differs significantly from FEM in the way that Maxwell 's equations are transformed into a matrix equation. A mathematical device called a *Green 's function* is used to represent the electrical interaction between any pair of elements. Using integrals of the Green 's function and the basis functions, a dense matrix of linear equations ("dense" because there are few or no zero-valued entries) is derived. A straightforward direct solution of such a dense matrix is prohibitively expensive, and so Q3D uses iterative matrix solvers and the Fast Multipole Method to make it practical to solve large problems.

Basis Functions in Q3D Extractor

Piecewise-constant basis functions are used for the method of moments solutions (capacitance, DC and AC inductance).

Piecewise-quadratic basis functions are used for the DC conduction solution (DC resistance). These have one unknown value at each vertex of a tetrahedron, and one unknown at the center of every tetrahedron edge.

Frequency-Dependent Material Properties in Q3D

The properties of some materials vary with the frequency of the field excitation. Q3D provides a number of ways to model this frequency dependence (see topics below). One consideration to be aware of when selecting a frequency-dependent material model for time-domain applications is causality.

Causality

Causality is a property of a system. Simply stated, a causal system initially in a steady state should not show any change in its output until its input signal is changed. Equivalently, a causal system cannot anticipate changes in its input signal and react before those changes occur. One criterion for deciding if a model is physically reasonable is to ask if it is causal.

$\varepsilon(j\omega)$

$\varepsilon(j\omega) = \varepsilon'(j\omega) - j\varepsilon''(j\omega)$

Then the Kramers-Krönig conditions for causality are given by:

$$\varepsilon''(j\omega) = \frac{1}{\pi} PV \int_{-\infty}^{\infty} \frac{\varepsilon'(j\omega')}{\omega' - \omega} d\omega'$$

and

$$\varepsilon'(j\omega) = -\frac{1}{\pi} PV \int_{-\infty}^{\infty} \frac{\varepsilon''(j\omega')}{\omega' - \omega} d\omega'$$

Here *PV* denotes the Cauchy principal value of the complex integral. The important point to notice about these relations is that the real and imaginary parts of the permittivity are not independent of one another: in fact the real part determines the imaginary part, and vice versa. Therefore, you can not choose the two independently.

The imaginary part of the permittivity is governed by the loss tangent (or, equivalently, the effective conductivity.) This means that the loss tangent and the relative permittivity must be chosen in such a way that the resulting complex permittivity satisfies the above relations. If you are not careful to do this, then the resulting model may not be causal.

When working in the frequency domain, a non-causal model may produce perfectly acceptable results. However, these results are used in a time-domain analysis, they will cause serious problems, such as non-physical transient responses or simulation failures.

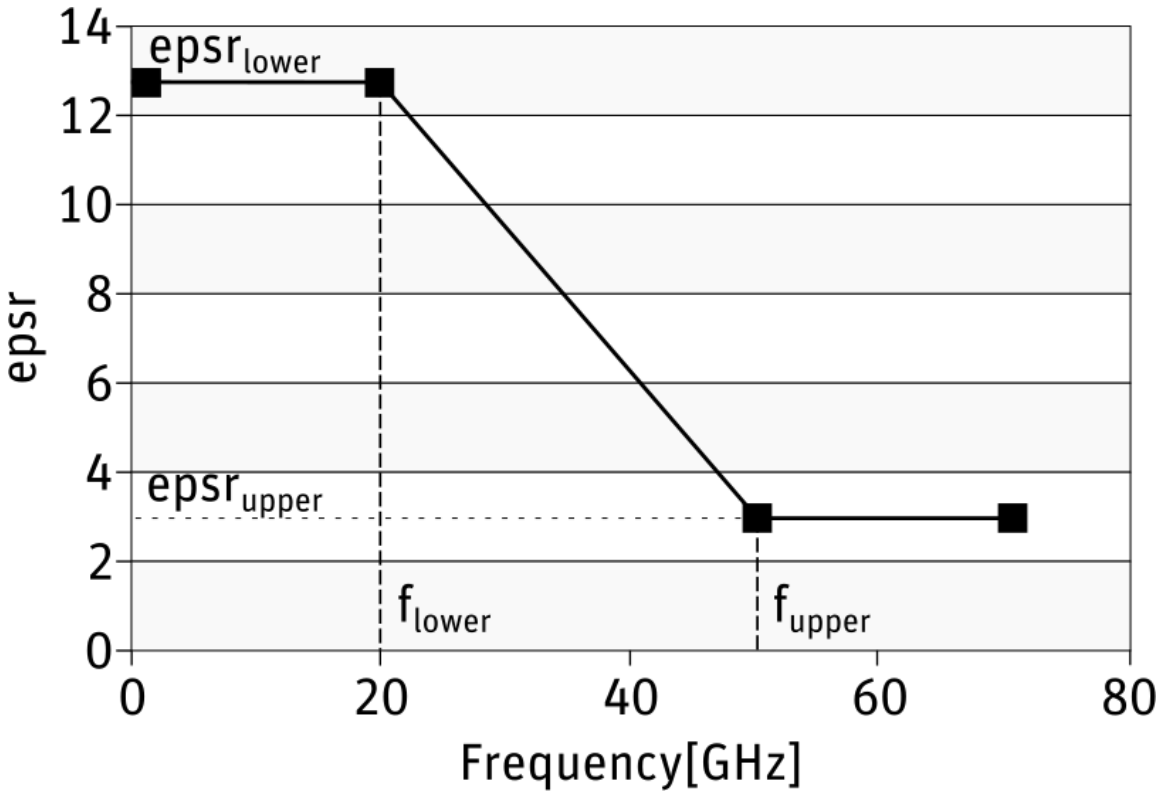
The Constant Permittivity, Constant Loss Tangent Model

It is common for people accustomed to working in the frequency domain (particularly with narrow-band microwave applications) to use a lossy dielectric model with a constant permittivity and a constant loss tangent. Such a model is easy to specify—only two numbers are required. Unfortunately, it is also non-causal.

The special case of a lossless material (zero loss tangent) with constant permittivity is causal, however.

The Piecewise Linear Model

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 to specify a material property 's values at frequencies below and above the frequency range. Based on these values, Q3D Extractor automatically creates a piecewise 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.

The piecewise linear model is quite general, and permits the relative permittivity to be specified independently from the loss tangent. As a result, it does not guarantee that the Kramers-Krönig causality conditions will be satisfied.

The Djordjevic-Sarkar Model

The Djordjevic-Sarkar model is a causal, frequency-dependent dielectric model developed specifically to model the FR-4 epoxy resin material used commonly in printed circuit boards. It is also useful for many other low-loss insulator materials. The formula for the complex permittivity in this model is given by Formula One (1):

$$\varepsilon(\omega) = \varepsilon_\infty + \frac{\Delta\varepsilon}{\ln(\omega_B/\omega_A)} \ln\left(\frac{\omega_B + j\omega}{\omega_A + j\omega}\right) + \frac{\sigma_0}{j\omega\varepsilon_0}$$

Here, ε_∞ denotes the high-frequency (or optical) limit of the relative permittivity and $\Delta\varepsilon$ is the difference between the low-frequency (DC limit) of the relative permittivity and ε_∞.

σ_0 is the DC conductivity (typically zero for low-loss dielectric materials). The model has lower and upper "corner frequencies" ω_A and ω_B respectively, which are described further below. All of these parameters are real-valued, and are computed for you from the material measurement data you provide.

The effective relative permittivity and conductivity for the Djordjevic-Sarkar model can be derived from the real and imaginary parts of Equation One (1) above.

Equation Two (2):

$$\varepsilon_r(\omega) = \varepsilon_\infty + \frac{\Delta\varepsilon}{2\ln(\omega_B/\omega_A)} \ln\left(\frac{{\omega_B}^2 + \omega^2}{{\omega_A}^2 + \omega^2}\right)$$

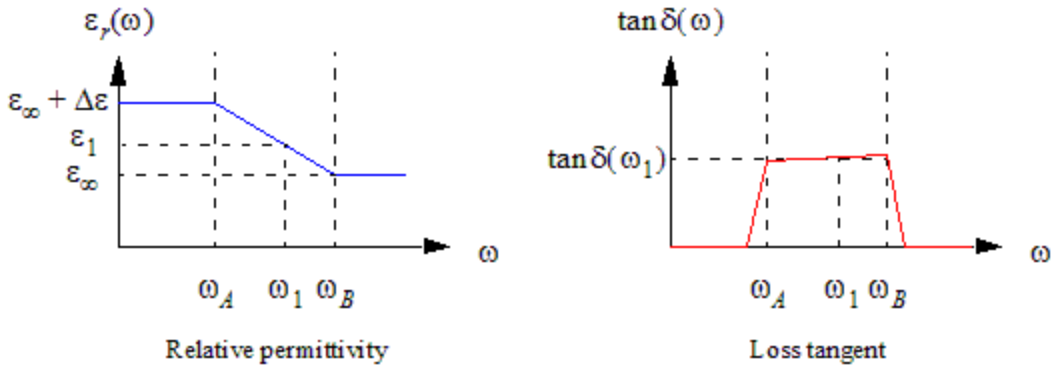
Equation Three (3):

$$\sigma(\omega) = \sigma_0 + \frac{\omega\varepsilon_0\Delta\varepsilon}{\ln(\omega_B/\omega_A)} \left[\tan\left(\frac{\omega}{\omega_A}\right) - \tan\left(\frac{\omega}{\omega_B}\right)\right]$$

The effective loss tangent is derived from the ratio of these:

$$\tan\delta(\omega) = \frac{\sigma(\omega)}{\omega\varepsilon_0\varepsilon_r(\omega)}$$

The frequency-dependent behavior of the relative permittivity and loss tangent are illustrated in the following figure. The frequency axis in this figure is shown on a *logarithmic* scale.



Some important points to note about these curves are:

- The low frequency relative permittivity is always higher than the optical permittivity.
- The relative permittivity decreases almost linearly (on a logarithmic frequency scale) between the two corner frequencies. Above and below this frequency range, it is essentially constant.
- The slope of the permittivity in the linear region is proportional to the loss tangent: a higher loss tangent results in a more rapid decrease of the permittivity.
- The loss tangent is nearly constant between the two corner frequencies. If the DC conductivity is zero, it rapidly decreases to zero outside this frequency range (If the DC conductivity is non-zero, then the loss tangent becomes infinite at DC).

Suppose that the relative permittivity ε_1 and loss tangent $\tan\delta(\omega_1)$ are given at a frequency ω_1 lying between the corner frequencies. Because the loss tangent determines the slope of the line, and the relative permittivity defines a point on the line, the line is completely determined by a single measurement. The only remaining parameters to be decided are the corner frequencies.

The upper corner frequency has not been observed in practice, and so it is set to a very high value ($10^{12}\,\mathrm{rad/s}$) The lower corner frequency will be determined by the DC permittivity, at the point where the flat DC part of the permittivity curve meets the linear part. If the user does not provide the DC permittivity, the software will pick a value such that the lower corner frequency is on the order of 10 krad/s.

References

[1] A. R. Djordjevic, R. D. Biljic, V. D. Likar-Smiljanic and T. K. Sarkar, "Wideband Frequency-Domain Characterization of FR-4 and Time-Domain Causality," *IEEE Trans. on Electromagnetic Compatibility*, Nov. 2001, p. 662.

Handling Complicated Models

Some complex models require special consideration.

RAM Settings for Very Large Geometries

Click **Modeler** > **Support Large Geometry Import** to reduce memory during file I/O and other operations for very large imported geometry. This is only necessary in exceptional cases.

Object Overlap Settings for Complicated Models

Complicated geometries often have small object overlaps. Click **Q3D Extractor** > **Design Settings** to 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.

The **Set Material Override** dialog box appears. Click **Allow metals to override dielectrics** if you want some intersections to be resolved automatically in the mesh. If metal intersects dielectric, the metal overrides the dielectric in the overlap region. (That is, the metal object is subtracted from the dielectric.) If objects with the same material overlap, the small object overrides the larger. (That is, the small object is subtracted from the larger.) All other intersections are treated as errors. Normally, the modeler considers any intersection between 3D objects to be an error.

In the meshing process, the dielectrics are locally overwritten by 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.

This feature allows 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.

Post-Processing for Complicated Models

Under **Tools** > **Options** > **General Options**, on the **Miscellaneous Options** tab...

- Deselect **Dynamically update postprocessing data during edits**. This will disable expensive updating of existing reports and plots.
- Deselect **Update reports on file open**. This will disable expensive updating of reports and plots when opening a project.

Interpolating Frequency Sweeps in Q3D Extractor

An interpolating sweep estimates a solution for an entire frequency range. Q3D Extractor chooses the frequency points at which to solve the field solution so that the entire interpolated

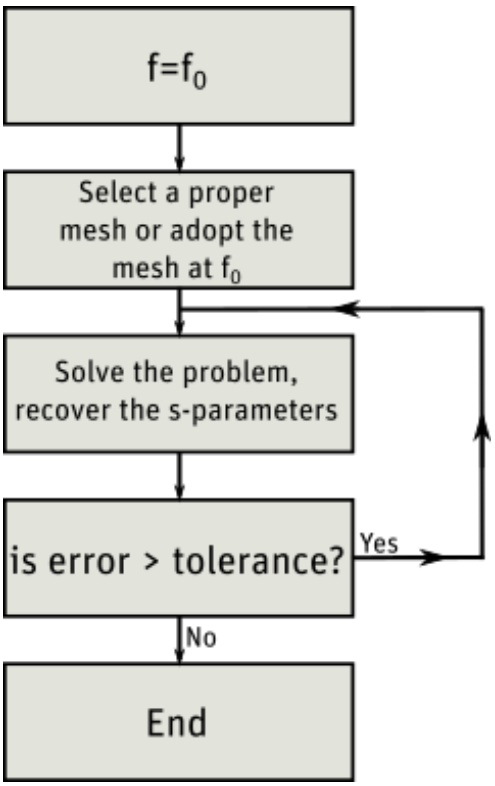
solution lies within a specified error tolerance. The sweep is complete when the solution meets the error tolerance criterion or generates the maximum number of solutions. To view more information about the solution, increase the number of steps and perform the sweep again.

The field solution for each point is deleted so that a new field solution can be generated for the next point. The full-field solution is only saved for the final frequency point computed. The S-parameters are saved for every solved frequency point.

Choose an interpolating sweep if the frequency range is wide and the frequency response is smooth, or if the memory requirements of a sweep exceed your resources. An interpolating sweep's time requirement is much less than a discrete sweep's because a solution for the entire frequency range is interpolated based on solutions for a minimal number of frequency points. The maximum time required for an interpolating sweep is the time required for a single frequency solution multiplied by the maximum number of solutions.

Be aware that Q3D Extractor uses the refined mesh during an adaptive solution at the solution frequency or, if you did not request an adaptive solution, the initial mesh generated for the problem. It uses this mesh without further refinement.

The procedure for an interpolating sweep is shown below, where *n* frequencies determined by the system are included in the sweep.



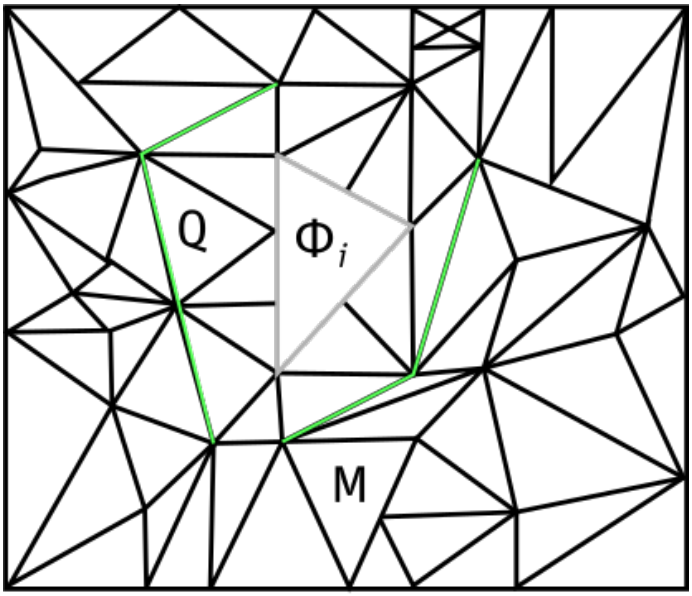
Multipole Acceleration

Multipole expansion is a way to accelerate the computation of the potential distribution produced by a known distribution of charges.

The relationship is described by:

$$\Phi = \int_V \frac{\rho}{r} dV$$

Computing the potential at ***n*** locations due to the charge on ***n*** triangles without multipole acceleration would require $O(n^2)$ computations, because every pair of interacting triangles must be accounted for. Multipole acceleration reduces this computation to $O(n)$. Instead of directly computing the potential due to each source at each observation point, the multipole method does the following:



Φ_i = Potential on triangle *i*
Q = Directly computed charge
M = Multipole approximation

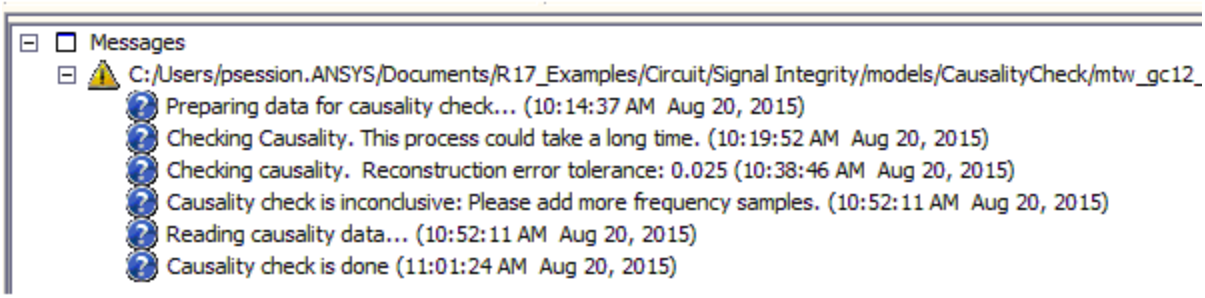
- The charges on triangles that touch the triangle where the potential is being evaluated contribute significantly more to the result than the charges on triangles that are farther away. The contributions to the potential from these charges are computed directly.

- The charges on triangles that are farther away have a significantly smaller contribution to the result. They are approximated using a multipole expansion, which groups the charges into a single set of equivalent terms.

Directly computing the potential on the triangles that are near neighbors minimizes the error in the solution. Using a multipole expansion to compute the potential on triangles that are farther away is somewhat less accurate than directly computing them but greatly speeds up the solution and uses less computer memory. The accuracy of the overall solution can be controlled by using distance criteria to decide where to apply the multipole expansion technique.

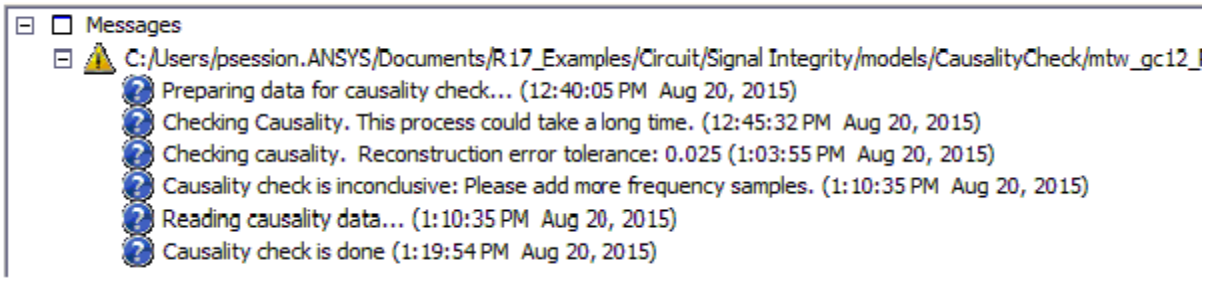
Multithreading Technical Notes

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.



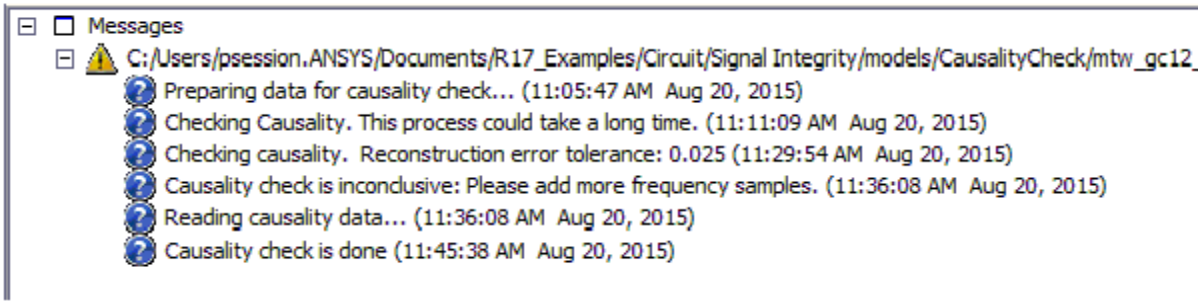
The causality check itself happens between the message `Checking causality. Reconstruction error tolerance: 0.025 (time 10:38:46)` and the message `Causality check is inconclusive: Please add more frequency samples (time 10:52:11).` The difference between these two times is the time for the causality 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 eight cores:



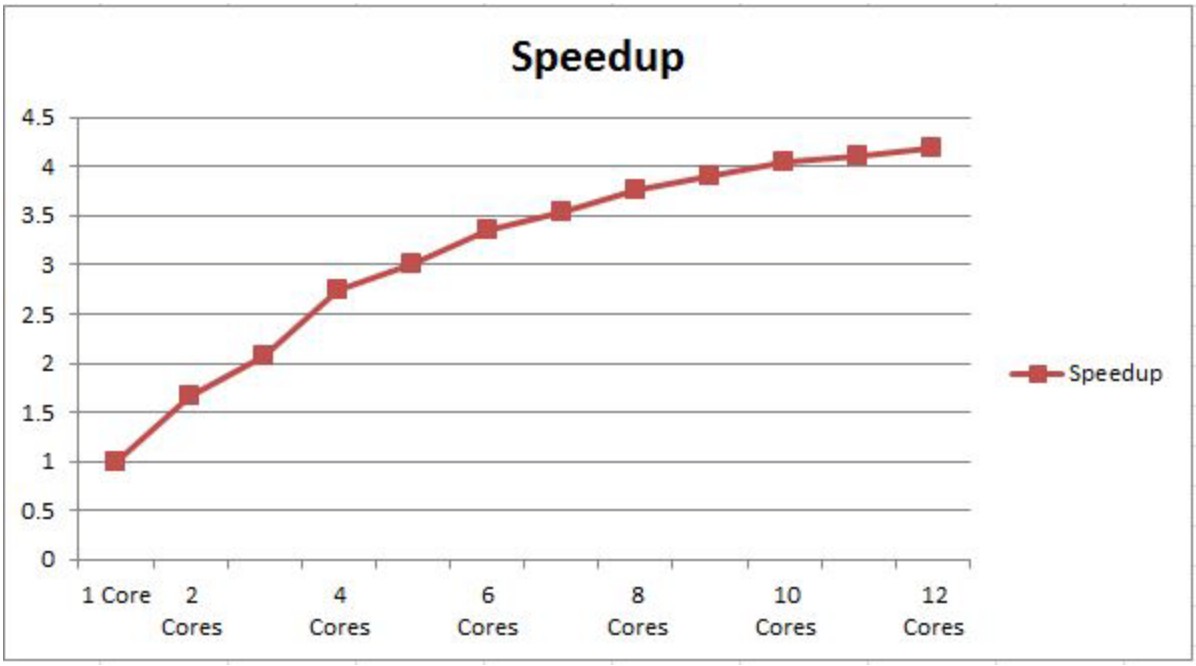
Now the causality check happens between times (1:03:55) and (1:10:35), a difference of (6:40) or six minutes and forty seconds, about half the time for the two-core example.

The speedup from multithreading is not linear in practice. Here are the messages from the same 278-port file using sixteen 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.



RLGC Parameter Matrices

This section includes information on the following matrices:

- Capacitance Matrices
- Conductance Matrices
- Resistance Matrices
- Inductance Matrices

To compute the entries of a matrix, Q3D solvers individually excite each conductor net or source terminal you have defined. Conducting objects, whether part of a net or not, influence the matrix calculations. Non-net conductive objects are grounded.

Since a model may contain many nets, the electromagnetic interaction between nets is described by a matrix, calculated as follows:

- As each source is excited, the field solution is used to compute a capacitance, inductance, or resistance for that conductor relative to the reference ground and to the other conductors. Each excitation yields one column of the matrix.
- The resistance and inductance matrices have one row for each source terminal. However, the capacitance and conductance matrices contain one row for each net.

Capacitance Matrices

At the simplest level, capacitance is proportional to the amount of energy stored in the electric field in and around a structure. In a circuit, this is the energy stored in the electric field resulting from the voltage differential across a dielectric:

$$U_e = \frac{1}{2}Cv^2$$

where:

- U_e is the energy stored in the electric field
- C is the capacitance
- v is the voltage across the dielectric

Q3D Extractor uses AC conduction to compute the capacitance between two lines by simulating the electric field that arises when a voltage differential is applied and then computing the energy stored in the field. The solver uses the following equation to solve for capacitance in terms of U_e:

$$C = \frac{2U_e}{v^2}$$

To compute the elements in the capacitance matrix for an *n* conductor transmission line, the Q3D Extractor performs a sequence of field simulations. In each field simulation, 1 volt is applied to a single conductor and 0 volts are applied to all other conductors. The energy stored in the electric field is given by the following relationship:

$$U_{ij} = \frac{1}{2}\int_{\Omega} D_i \cdot E_j d\Omega$$

where:

- U_{ij} is the energy stored in the electric field linking conductor *i* with conductor *j*.
- D_i is the electric flux density produced when 1 volt is applied to conductor *i* and 0 volts are applied to all other conductors.
- E_j is the electric field produced when 1 volt is applied to conductor *j* and 0 volts are applied to all other conductors.

The capacitance coupling *i* and *j* is therefore:

$$C_{ij} = \frac{2U_{ij}}{v^2} = \int_{\Omega} D_i \cdot E_j d\Omega$$

Since a standard SPICE component only has two terminals, the software derives a SPICE matrix format from the Maxwell matrix capacitance elements.

Conductance Matrices

Conductance matrices are analogous to capacitance matrices. In the case of the conductance matrix the total currents I_1, I_2, etc. replace the total charges Q_1, Q_2, etc.

The relationship is therefore:

$$\begin{bmatrix} I_1 \\ I_2 \\ \dots \\ I_n \end{bmatrix} = \begin{bmatrix} G_{11} G_{12} \dots G_{1n} \\ G_{21} G_{22} \dots G_{2n} \\ \dots \ \dots \ \dots \ \dots \\ G_{n1} G_{n2} \dots G_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix}$$

More concisely:

$$I = GV$$

The entries of the conductance matrix are specified in units of Siemens (Amps/Volt.)

Resistance Matrices

In a device with ***n*** source terminals, the resistance relationship is expressed by an ***n x n*** matrix:

$$\begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1n} \\ R_{21} & R_{22} & \dots & R_{2n} \\ \dots & \dots & \dots & \dots \\ R_{1n} & R_{2n} & \dots & R_{nn} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \dots \\ I_n \end{bmatrix}$$

Or, more concisely:

V = RI

The entries of the resistance matrix are in Ohms.

Q3D Extractor computes partial resistances—that is, the resistance only of the specific part of the structure being modeled. The rest of the current loop is not considered. The total resistance of the loop is the sum of the partial resistances of each section of it.

Resistance

Resistance in the Q3D Extractor refers to the ohmic losses in the signal conductors. Conductance is the inverse of the shunt resistance, and is calculated in the AC conduction solver.

To compute resistance, the eddy current solver calculates ohmic loss, ***P***, using the following relationship:

$$P = \frac{1}{2} Re \int_V J^* \cdot \sigma E dV$$

and

$$P = \frac{1}{2} Re \int_V \frac{1}{\sigma} J^* \cdot J dV$$

Since ohmic loss is related to resistance by:

$$P = I^2_{rms} R$$

Resistance is therefore:

$$R = \frac{P}{I^2_{rms}} = \frac{2P}{I^2_{peak}}$$

Since the software assumes that the object for which resistance is being computed has a peak current of one amp per coil turn flowing through it, the resistance is simply ***2P***.

> **Note:**
>
> The computed resistance for an eddy current problem is higher than the equivalent DC resistance, due to the skin concentration of currents.

This results in the following matrix equation, which gives the relationship between ***I*** and ***V*** for the two current loops:

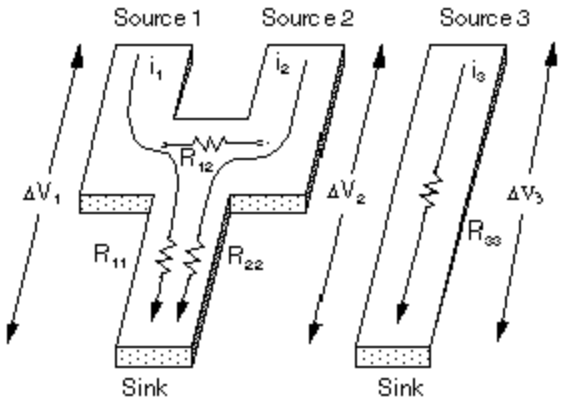
$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

Where:

- V_i and I_i are voltage and current phasors.
- $Z_{11} = R_{11} + j\omega L_{11}$ (the self-resistance of loop 1).
- $Z_{12} = R_{12} + j\omega L_{12}$ (the mutual resistance between loops 1 and 2).
- $Z_{22} = R_{22} + j\omega L_{22}$ (the self-resistance of loop 2).

DC Resistance Matrices

The resistance matrix gives the relationship between voltage drops and currents. For instance, the transmission structure shown below consists of two regular conductors, one of which has multiple sources of current.



The voltage drops are given by:

$\Delta V_1 = i_1 R_{11} + i_2 R_{12}$

$\Delta V_2 = i_2 R_{22} + i_1 R_{12}$

$\Delta V_3 = i_3 R_{33}$

This can be expressed in matrix form as:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & 0 \\ R_{12} & R_{22} & 0 \\ 0 & 0 & R_{33} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}$$

where:

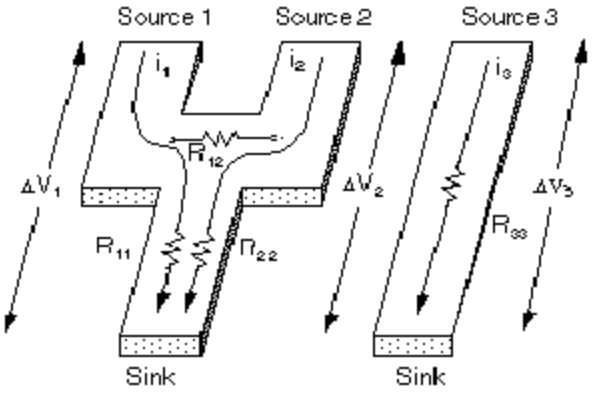
- R_{11}, R_{22}, and R_{33} are the self-resistances of each current path.
- R_{12} is the mutual resistance between current paths 1 and 2. For DC resistance matrices, the mutual resistance is only nonzero for current paths within the same net (that is, conductor nets that have multiple sources of current).

> **Note:**
>
> Mutual DC resistance between regular conductors on distinct nets is zero.

AC Resistance Matrices

The resistance matrix gives the relationship between voltage drops and currents.

For example, the transmission structure below consists of two regular conductors, one of which has multiple sources of current.



The voltage drops are given by:

$\Delta V_1 = i_1R_{11} + i_2R_{12}+i_3R_{13}$

$\Delta V_2 = i_2R_{22} + i_1R_{12}+i_3R_{13}$

$\Delta V_3 = i_3R_{33} +i_1R_{13}+i_2R_{23}$

where:

R_{11}, R_{22}, and R_{33} are the self-resistances of each current path.

$R_{12}$is the mutual resistance between current paths 1 and 2. Mutual resistance only applies to current paths within the same net (that is, conductor nets that have multiple sources of current).where:

R_{13} is the mutual resistance between paths 1 and 3. R_{23} is the mutual resistance between paths 2 and 3.

> **Note:**
>
> Mutual AC resistance between regular conductors on distinct nets is non-zero.

At non-zero frequency, two phenomena begin to affect the resistance matrix:

- Current within a conductor begins to accumulate near the periphery (skin effect). This reduces the effective cross-section through which the current travels, thus increasing the resistance.
- Current in one net induces currents in adjacent nets. Induced currents cause additional power loss, thus increased self-resistance. These induced currents create a non-zero mutual resistance between conductors.

AC resistance is computed after determining the surface currents on a net. Surface currents are computed as a step in determining AC inductance.

Resistance Matrices with Perfect Conductors

If a project contains only perfect conductors, a DC solution cannot be run because the results will be the same as the AC solution: all self and mutual resistances will equal zero.

In an AC Inductance/Resistance solution containing regular and perfect conductors, there will be a mutual resistance between all conductors due to induced currents. In addition, the perfect conductors will also have a small self resistance.

The main reason to use perfect conductors in Q3D Extractor is when you have an application with both good and poor conductors. By defining the good conductors as perfect conductors, induced currents are simulated in these conductors, while only volume currents are simulated on the poor conductors.

Inductance Matrices

At the simplest level, inductance is proportional to the energy stored in the magnetic field when current flows:

$$U_m = \frac{1}{2} L I^2$$

where U_m is the energy stored in the magnetic field, L is the inductance, and I is the current flowing in the circuit. Q3D Extractor uses the eddy current simulator to compute the inductance between two lines by simulating the magnetic field that arises when a current is applied and then computing the energy stored in the field.

The solver uses the following equation to solve for inductance in terms of U_m:

$$L = \frac{2U_m}{I^2}$$

To compute the elements in the inductance matrix for an n conductor transmission line, Q3D Extractor performs a sequence of field simulations. In each field simulation, 1 amp is allowed to flow in a single conductor, and no current is allowed to flow in any other conductor.

The energy stored in the magnetic field associated with the inductance between two conductors is given by the following relationship:

$$U_{ij} = \frac{1}{2} L I^2 = \frac{1}{2} \int_{\Omega} B_i \cdot H_j d\Omega$$

where:

- U_{ij} is the energy stored in the magnetic field linking conductor ***i*** with conductor ***j***.
- B_i is the magnetic flux density produced when one amp is allowed to flow through conductor ***i*** and no current flows in the other conductors.
- H_i is the magnetic field produced when one amp is allowed to flow through conductor ***j*** and no current flows in the other conductors.

The inductance coupling conductors ***i*** and ***j*** is therefore:

$$L_{ij} = \frac{2U_{ij}}{I^2} = \int_{\Omega} B_i \cdot H_j d\Omega$$

Admittance in Terms of Time-Varying Conductance

Conductance in the Q3D Extractor refers to the shunt resistance used in the ladder circuit model. Resistance is the series resistance, and is calculated in the eddy current solver used by the inductance matrices.

To compute conductance, the AC Conduction solver calculates power, ***P***, using the following relationship:

$$P_{rms} = Re\left[\frac{1}{2}\int_v J^* \bullet E dV\right] = \frac{1}{2} Re\left[\int_v (\sigma - j\omega\varepsilon)\nabla\Phi^* \bullet \nabla\Phi \, dV\right]$$

Since power is related to conductance by:

$$P_{rms} = \frac{1}{2} Re\left[i^* v\right]$$

and:

$$i = \mathrm{G}\mathrm{v}$$

conductance is therefore:

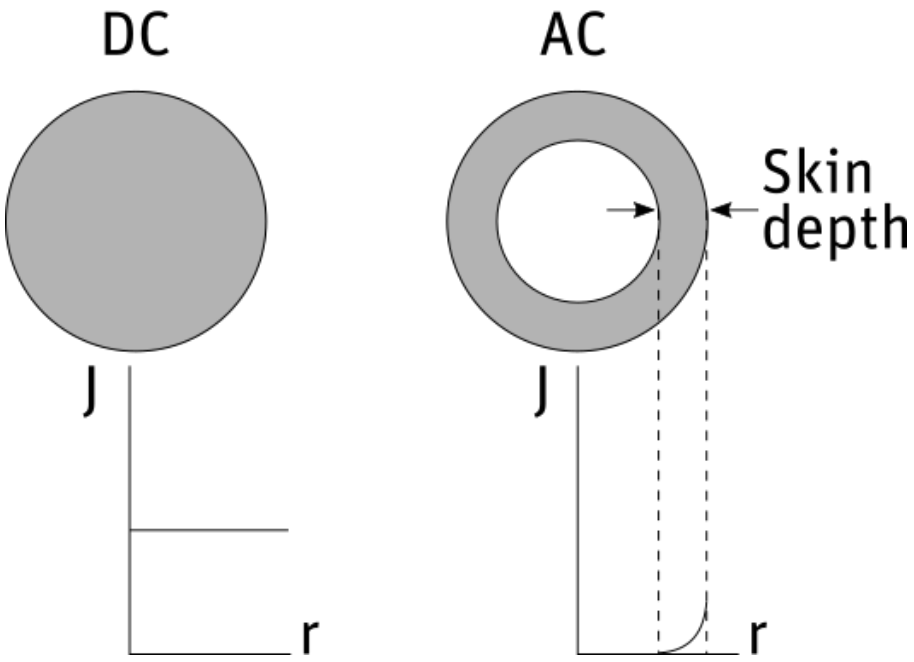
$$\mathrm{G} = \frac{2P}{Re[v^* v]}$$

Since the software assumes that the object for which resistance is being computed peaks at one volt, the conductance is simply ***2P***.

> **Note:**
>
> The computed resistance for an eddy current problem is higher than the equivalent DC resistance, due to the skin concentration of currents.

Eddy Currents and Skin Depth

The inductances and resistances computed during an resistance matrix solution are different from those computed for the equivalent DC case. The eddy current simulator includes these eddy effects when it computes inductance during a resistance solution. The following figure shows how current density (**J**) varies with position along the radius (**r**) of the conductor cross-section for both cases:



The current density in the DC case (on the left) is evenly distributed throughout the cross-section of the conductor. The current density in the AC case (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, resistance to the current is higher in the AC case than in the DC case.

In the DC example, eddy currents do not occur because the magnetic field created by the current flowing through the conductor is static. In the AC example, the oscillating magnetic field induces currents in the conductors. These induced currents affect the computation of inductance, causing it to be different from (generally lower than) the DC case. At higher frequencies the magnetic field can propagate completely outside the conductor, whereas current (the movement of electrons) still must travel in the conductor, though with a smaller cross-sectional area. Therefore, at higher frequencies, there is less interaction of the magnetic field with the conductor, so it is impeded less, thus there is less inductance. The inductance flattens out because the magnetic field still interacts with the surface of the conductor but has avoided pulling fields along through the metal.

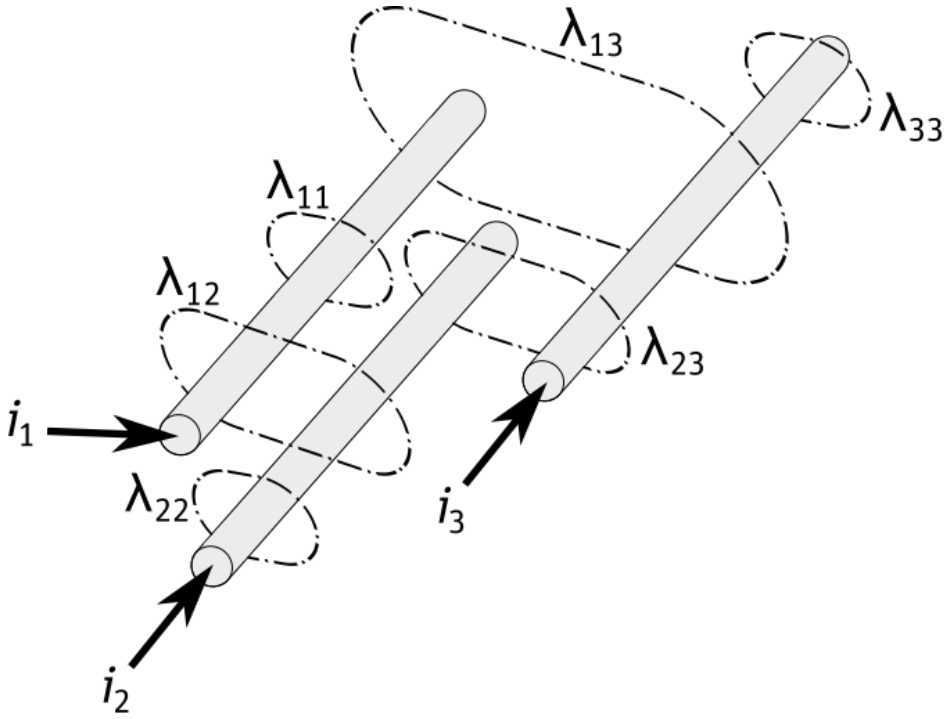
Inductance in Terms of Flux Linkage

The inductance matrix represents the magnetic flux linkage within a group of conductors. For example, given the three conductors in the following figure, the relationship between induced flux and currents is as follows, where **λ** is the flux linkage:

$\lambda_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3$

$\lambda_3 = L_{13}i_1 + L_{23}i_2 + L_{33}i_3$

$\lambda_1 = L_{11}i_1 + L_{12}i_1 + L_{13}i_3$



This results in the following matrix equation, which gives the relationship between **λ** and ***i*** for the three conductors:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} \lambda_{11}\lambda_{12}\lambda_{13} \\ \lambda_{12}\lambda_{22}\lambda_{23} \\ \lambda_{13}\lambda_{23}\lambda_{33} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}$$

Inductance matrix values are in henries. In a device with ***n*** conductors, this relationship would be expressed by an ***n x n*** inductance matrix. The diagonal terms in the matrix (such as λ_{11}) represent the self-inductance of each signal line. The off-diagonal terms (such as λ_{12} and λ_{13})

represent the mutual inductances between the signal lines. Note that the inductance matrix is symmetric about the diagonal. This indicates that the mutual effects between any two signal lines are identical. Since the software assumes that the object for which resistance is being computed has a peak current of one amp per coil turn flowing through it, the inductance is simply $4U_{av}$.

Inductance in Terms of Time-Varying Current

A inductance matrix also represents the relationship between voltage and time-varying current in a system of conductors. At the simplest level, inductance is proportional to the energy stored in the magnetic field when current flows:

$$U_m = \frac{1}{2}Li^2$$

where U_m is the instantaneous value of the energy stored in the magnetic field, L is the inductance, and $i = I_{peak}\cos(\omega t + \theta)$— the instantaneous value of the current flowing in the circuit, related to the peak value of the current.The average value for the energy can then be found by integrating U_m.

$$= \frac{1}{4}\int_{\mathrm{v}} B \cdot H dV$$

$$= \frac{1}{2\pi}\int_0^{2\pi} U_m d\omega t$$

and

$$= \left(\frac{L}{2}\right)\left(\frac{1}{2\pi}\right)\int_0^{2\pi} I_{peak}^2[\cos(\omega t + \theta)]^2 d\omega t$$

From this, the average energy of the system is equal to:

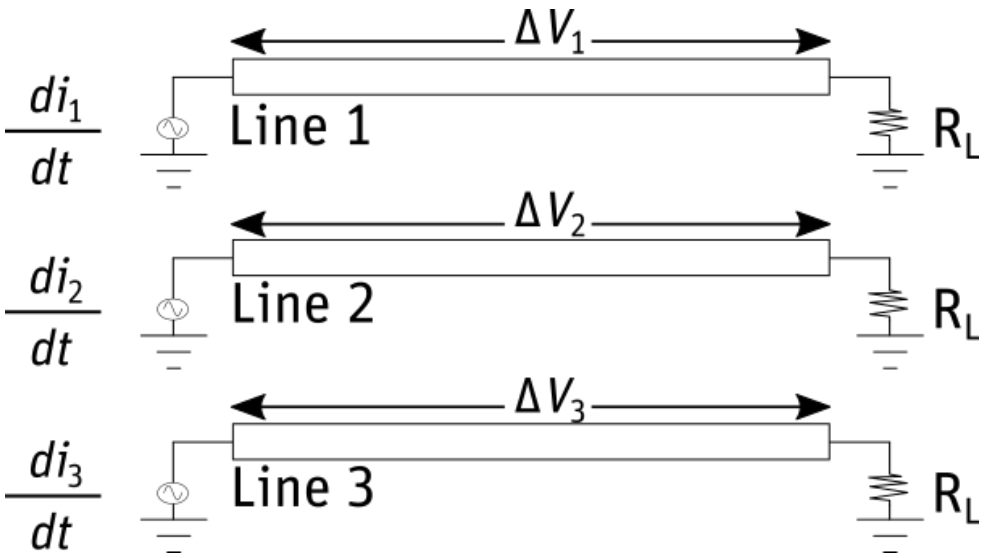
$$U_{av} = \left(\frac{L}{2}\right)I_{rms}^2 = \left(\frac{L}{2}\right)\left(\frac{I_{peak}}{\sqrt{2}}\right)^2 = \left(\frac{L}{4}\right)I_{peak}^2$$

The inductance is therefore:

$$L = \frac{4U_{av}}{I_{peak}^2}$$

Given the three transmission lines shown in the following figure, the voltage changes caused by the time-varying current source on each line are given by the following relationship:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{bmatrix} = \begin{bmatrix} L_{11} L_{12} L_{13} \\ L_{12} L_{22} L_{23} \\ L_{13} L_{23} L_{33} \end{bmatrix} \begin{bmatrix} di_1 / dt \\ di_2 / dt \\ di_3 / dt \end{bmatrix}$$



If ***di2/dt*** and ***di3/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 inductance matrix gives the voltages 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 lines or the open circuit inductance. Inductance matrix values are in henries.

Sign of the Partial Inductances

The partial self inductance is always a positive quantity. The partial mutual inductance may be positive, negative or zero depending upon the polarity of the inductive voltage drop produced on the observation current path by the source path. If the two paths have the same orientation, the mutual inductance is positive.

Now suppose we exchange the locations of the source and sink terminals on one current path, so that the two current paths have the opposite orientation. The mutual inductance will now become negative. In the partial inductance example above, the loop inductance would be given by the modified formula:

$$L_{total} = L_{P11} - L_{P12} - L_{P21} + L_{P22}$$

This of course gives the same result as before, because the signs of the partial mutual inductance terms have changed.

Finally, suppose we take one conductor and turn it 90 degrees relative to the other. The mutual inductance will become zero.

Matrix Reduction Operations in Q3D Extractor and 2D Extractor

Matrix reduction operations provide a convenient way to explore different types of connections and grounding configurations without additional runs of the field solver.

This section of technical notes explains the meanings of various matrix reduction operations available in Q3D Extractor and 2D Extractor, as well as the assumptions behind them. It is divided into the following topics:

- **Introduction** – explains basic concepts from electromagnetics and circuit theory that are essential for understanding this section.
- **Grounding and Floating Conductors** – describes the effects of grounding and floating conductors on the capacitance and inductance matrices.
- **Connecting Elements** – describes the effects of connecting elements in parallel or in series on the capacitance and inductance matrices.
- **Joining Elements in Parallel** – describes the effects of joining elements in parallel on the inductance matrix. The Join Selected Terminals operation is also covered.
- **Floating at Infinity** – describes the Float at Infinity operation and its effect on the capacitance matrix.
- **Return Path Reduction** – describes the Return Path Reduction operation and its effect on the inductance matrix.
- **Redefining the Ground Reference** – describes redefining the ground reference and the effect of doing so on the capacitance matrix.
- **Operations for Multiterminal Conductors** – describes operations for multiterminal conductors (such as moving source and sink points or adding sink points) and their effect on the inductance and capacitance matrices.
- **Set Ground vs Add Ground in 2D Extractor** – describes the two different grounding options available in 2D Extractor and the effects of each.
- **Differential Pair Matrix Reduction in 2D Extractor** – describes the differential pair matrix reduction operation and its effects.

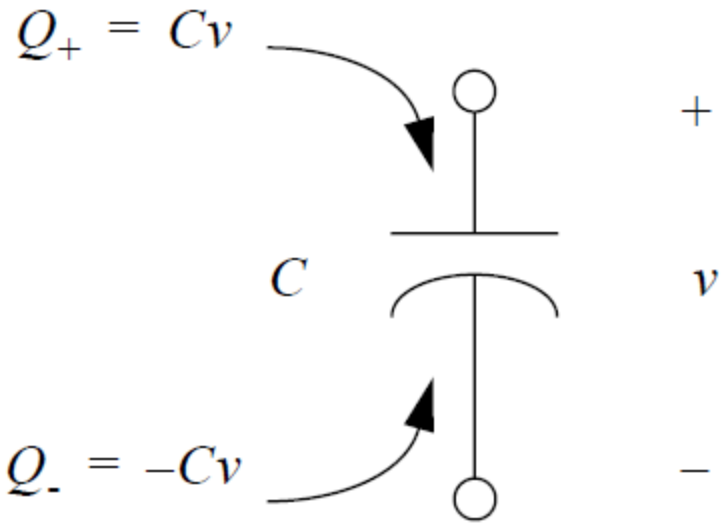
Circuit-oriented approaches will be used to explain the concepts involved, and the examples will distinguish between the cases of inductance and capacitance matrices because many (if not all) matrix reduction operations have very different effects on the two types of matrices.

Introduction

This topic reviews a few basic concepts from electromagnetics and circuit theory that are essential to this discussion.

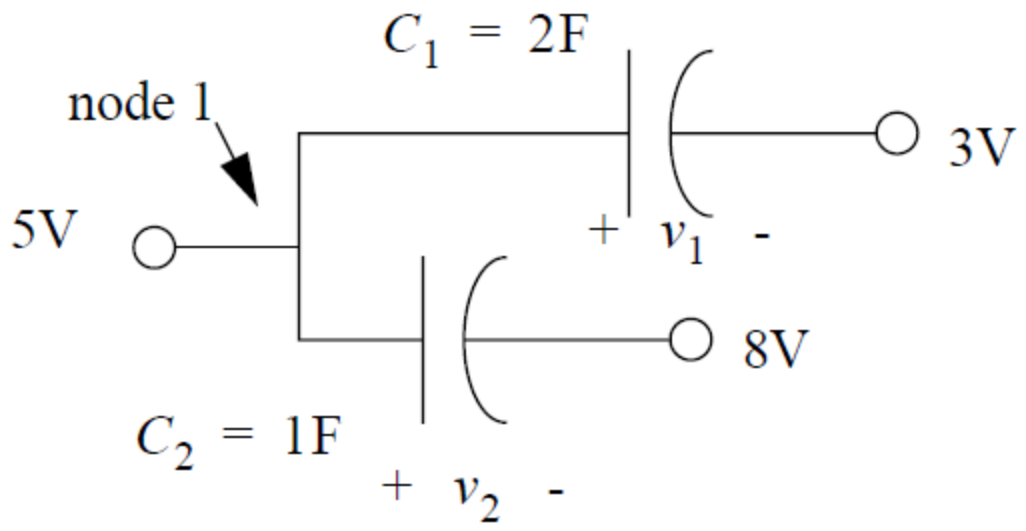
Two-Terminal Capacitor

A two-terminal capacitor of value is shown in the figure below, with the charges on the top and bottom plates shown. below. Recall from basic circuit theory that each "plate" of the capacitor has associated with it a certain charge. The "top" plate (the one that serves as the positive reference for the branch voltage) has a total charge $Q_+ = Cv$ spread over it. The bottom plate has an equal and opposite charge $Q_- = -Cv$. The charges are always equal and opposite because it is assumed that the field lines from the top plate terminate on the bottom plate.



When two or more capacitors are connected to the same node, the total charge on the node is simply the sum of the charges on the various capacitor plates connected to the node.

For example, consider the charge on node 1 in the following figure.



The charge on node 1 is found as follows:

$$v_1 = 5V - 3V = 2V$$

$$v_2 = 5V - 8V = -3V$$

$$Q_1 = C_1 v_1 + C_2 v_2$$

$$= (2\text{F})\,(2\text{V}) + (1\text{F})\,(-3\text{V})$$

$$= 4 \text{ coulombs} - 3 \text{ coulombs}$$

$$= 1 \text{ coulomb}$$

Multiterminal Capacitors: Maxwell Capacitance Matrix

The Maxwell capacitance matrix is an electromagnetics concept that assists in dealing with interactions between many different conductors. Say there are three different conductors, each at its own voltage. The charge on conductor 1 is partly determined by its own voltage and partly by the voltages on the other two. The relationship is linear:

$$q_1 = C_{11}^M v_1 + C_{12}^M v_2 + C_{13}^M v_3$$

Similar relationships hold for conductors 2 and 3. The role of a field solver is to compute the various coefficients C_{ij}^M, where C_{ij}^M indicates the charge induced on conductor i by a 1 volt voltage on conductor j. In linear, isotropic dielectric materials, these coefficients are reciprocal; that is, $C_{ij}^M = C_{ji}^M$.

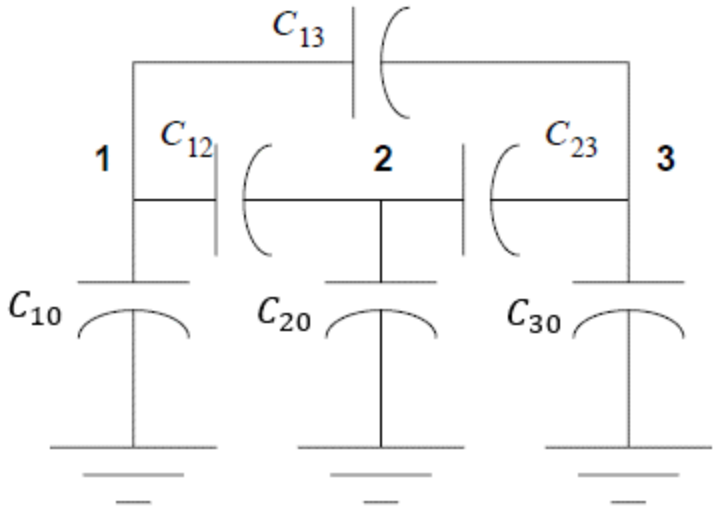
The collection of these capacitance coefficients is known as the Maxwell capacitance matrix $\mathbf{C}^M$. For the 3-conductor example, the Maxwell capacitance matrix could be written as:

$$\mathbf{C}^M = \begin{bmatrix} C_{11}^M & C_{12}^M & C_{13}^M \\ C_{12}^M & C_{22}^M & C_{23}^M \\ C_{13}^M & C_{23}^M & C_{33}^M \end{bmatrix}$$

This employs the reciprocity condition to replace C_{21}^M with C_{12}^M, etc. This is the capacitance matrix that the field solvers in Q3D Extractor compute.

Spice Capacitance Matrix

The figure below shows a set of two-terminal capacitors connected between ground and three other nodes (1, 2, 3).



Assume that the voltages v_1, v_2, and v_3 on nodes 1, 2, and 3 are known. The total charge on these nodes can be computed. For example, on node 1 the total charge q_1 is:

$$q_1 = C_{10}v_1 + C_{12}\left(v_1 - v_2\right) + C_{13}\left(v_1 - v_3\right)$$

$$= (C_{10} + C_{12} + C_{13})v_1 - C_{12}v_2 - C_{13}v_3$$

Therefore, there is a relationship between the two-terminal capacitor values and the entries of the Maxwell capacitance matrix:

$$C_{11}^M = C_{10} + C_{12} + C_{13}$$

$$C_{12}^M = -C_{12}$$

$$C_{13}^M = -C_{13}$$

And so on. The diagonal entries of the Maxwell capacitance matrix are found by summing the *total* of all two-terminal capacitances connected to the given node. The off-diagonal entries of the Maxwell capacitance matrix are found by taking the *negative* of the coupling capacitance between two nodes.

Thus, the Maxwell capacitance matrix is given in terms of two-terminal capacitor values as:

$$\begin{bmatrix} C_{11}^M & C_{12}^M & C_{13}^M \\ C_{12}^M & C_{22}^M & C_{23}^M \\ C_{13}^M & C_{23}^M & C_{33}^M \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix}$$

An alternative representation of the Maxwell capacitance matrix, known as the *Spice Capacitance Matrix*, is available in Q3D Extractor. The Spice Capacitance Matrix collects the values of the two-terminal capacitors that would give rise to the same charge relationships that the Maxwell capacitance matrix describes.

For the example above, the Spice Capacitance Matrix is:

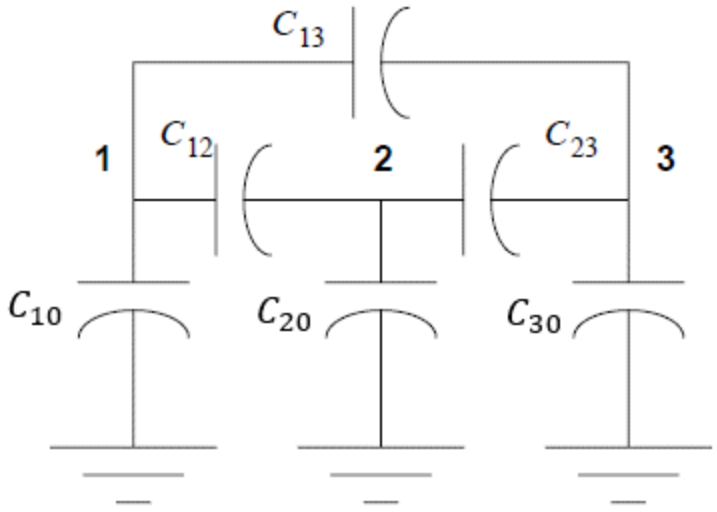
$$\mathbf{C}^S = \begin{bmatrix} C_{10} & C_{12} & C_{13} \\ C_{12} & C_{20} & C_{23} \\ C_{13} & C_{23} & C_{30} \end{bmatrix}$$

The diagonal entries of the Spice capacitance matrix are the sum of the rows of the Maxwell capacitance matrix: $C_{i0}^S = C_{i1}^M + C_{i2}^M + C_{i3}^M$. The off-diagonal entries of the Spice capacitance matrix are the negative of the corresponding entries of the Maxwell capacitance matrix: $C_{ij}^S = -C_{ij}^M$ for $i \neq j$.

Multithermal Capacitors in Lossy Dielectrics: Conductance (G) Matrix

Lossy dielectrics are imperfect insulating materials. Small "leakage currents" can flow in these materials at DC if the conductivity is non-zero, or at higher frequencies when the loss tangent is non-zero.

When Q3D Extractor solves a problem involving lossy dielectric materials, the losses give rise to a non-zero conductance (G) matrix. The G matrix can be visualized as a set of two-terminal resistors connected in parallel with each of the two-terminal capacitors in this figure:



These resistors are characterized in terms of their conductance values (the reciprocal of their resistance), measured in siemens.

The G matrix relates the leakage currents in the dielectrics to the conductor voltages. Thus, a system of two conductors gives a 2x2 conductance matrix:

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{12} & G_{22} \end{bmatrix}$$

The leakage currents are then given by:

$$i_1 = G_{11}v_1 + G_{12}v_2$$

$$i_2 = G_{12}v_1 + G_{22}v_2$$

Admittance (Y) Matrix

In many cases, when Q3D Extractor performs matrix reduction operations, the capacitance matrix and the conductance matrix are affected in the same way. In these cases, the same reduction formulas apply to both the G and C matrices. However, for certain operations these matrices must be combined to work in terms of the admittance (Y) matrix. The entries of the Y matrix are $Y_{ij} = G_{ij} + j\omega C_{ij}$ where $\omega = 2\pi f$ is the angular frequency.

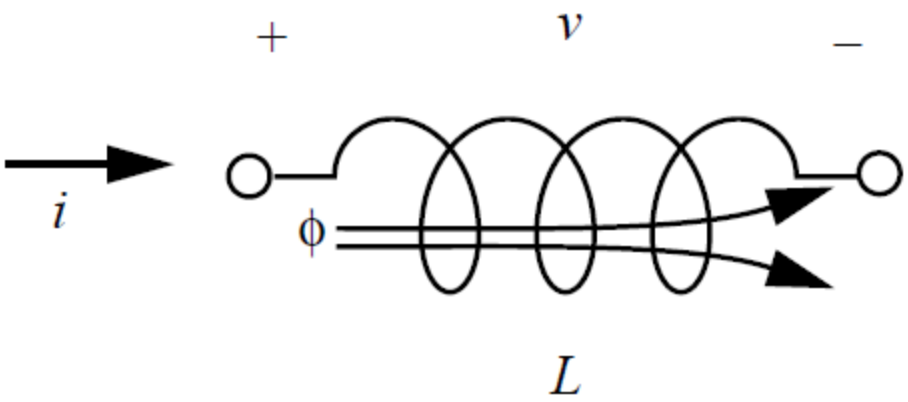
The admittance matrix must be used when a problem requires Q3D Extractor to enforce conditions on the total current. For example, when floating a conductor in a lossy problem, its total current must equal zero.

The matrix reduction operations that require use of the admittance matrix are: floating a conductor and floating at infinity.

To simplify the information that follows, these operations are explained in terms of the capacitance matrix, as if the problem were lossless. The results derived could also apply to a lossy problem if C_{ij} were replaced by Y_{ij}. The reduced Y matrix can be separated into the reduced G and C matrices by taking the real and imaginary parts, respectively, and then dividing the imaginary part by ω to obtain C.

Two-Terminal Inductor

The figure below shows an ideal two-terminal inductor with its branch voltage, branch current, and magnetic flux.



A two-terminal inductor is described by the following relationship between its branch voltage v and the time derivative of its branch current i:

$$v = L\frac{di}{dt}$$

It's often convenient to instead separate this into two separate relationships. The magnetic flux ϕ is defined as:

$$\phi = Li$$

Accompanying this is the relationship between voltage and the time derivative of the flux:

$$v = \frac{d\phi}{dt}$$

For this example, force the voltages on two inductors to be equal to one another; in other words, $v_1 = v_2$, or:

$$\frac{d\phi_1}{dt} = \frac{d\phi_2}{dt}$$

Assuming there was some time in the past when both inductances were completely discharged, integrate this equation to arrive at the result:

$$\phi_1 = \phi_2$$

It is often more convenient to work with inductor fluxes than voltages, as this avoids carrying the time derivative operator.

Coupled Inductances

Two coupled inductances have their fluxes determined by the currents in both inductors:

$$\phi_1 = L_{11} i_1 + L_{12} i_2$$

$$\phi_2 = L_{12} i_1 + L_{22} i_2$$

The quantities L_{ij} are the entries of the inductance matrix L for these two conductors:

$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} \\ L_{12} & L_{22} \end{bmatrix}$$

The voltages across these coupled elements are computed from the fluxes in exactly the same way as for uncoupled elements:

$$v_1 = \frac{d\phi_1}{dt}, \quad v_2 = \frac{d\phi_2}{dt}$$

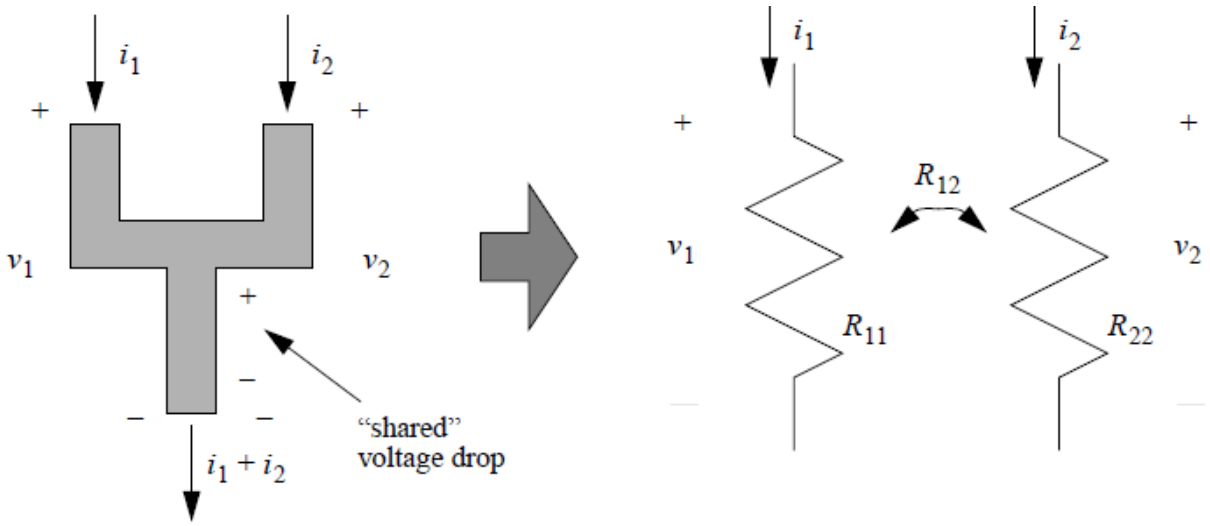
It is easy to expand these two relationships in terms of time-derivatives of the two currents. It is also straightforward to generalize from the case of two coupled inductances to an arbitrary number of couplings.

Resistance and Mutual Resistance

The relationship that describes a two-terminal resistor is the familiar Ohm's law, which relates branch voltage v to branch current i:

$$v = Ri$$

In certain situations, it is meaningful to consider *mutual resistance* effects. The simplest case to visualize is the multiterminal conductor ("net") shown below.



This figure demonstrates mutual resistance between two current-carrying paths on the same multiterminal conductor. In this example, a current flowing into one terminal produces voltage drops throughout the conductor. Part of this voltage drop is across the bottom leg of the Y shape. This portion of the voltage drop is shared between v_1 and v_2, and so it appears in both. The "shared" voltage drop creates mutual resistance. Consider:

$$v_1 = R_{11} i_1 + R_{12} i_2$$

$$v_2 = R_{12} i_1 + R_{22} i_2$$

Thus, the mutual resistance R_{12} is the voltage drop sensed by v_1 due to a one ampere current flowing in i_2. For linear, isotropic media, this is the same as the voltage drop induced in v_2 by a 1 amp current in i_1.

> **Note:**
>
> If the voltage drops were measured on two disjoint conductors and the frequency was zero, there could be non-shared voltage drop, and the mutual resistance in this case would be *zero*, not infinity. Zero-valued mutual resistance means that a current flowing in one conductor does not induce voltage drop on the other.
>
> At nonzero frequencies, a non-zero mutual resistance appears between disjoint conductors, caused by induced eddy currents.

When dealing with multiple conductors, it is natural to describe them with a resistance matrix. For the two-conductor example, the resistance matrix would consist of the self-resistances on

the main diagonal and the mutual resistance values between the conductors on the off-diagonals:

$$\begin{bmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{bmatrix}$$

Impedance (Z) Matrix

The resistance matrix and inductance matrix usually behave in the same way when the matrix reduction operations in Q3D Extractor are applied. The same formulas that yield the reduced ***L*** matrix will yield the reduced ***R*** matrix. However, for certain operations, the resistance and inductance matrices must be combined to work in terms of the impedance (***Z***) matrix. The entries of the impedance matrix are $Z_{ij} = R_{ij} + j\omega L_{ij}$ where $\omega = 2\pi f$ is the angular frequency.

At non-zero frequency, the voltage on a conductor will have resistive and inductive components. Therefore, the impedance matrix must be used when the matrix reduction operation enforces conditions on the voltage drop. These cases include: joining conductors in parallel, joining selected terminals, and grounding a conductor. For simplicity, the reduction formulas for these operations are derived as if the problem were lossless (having zero resistance).

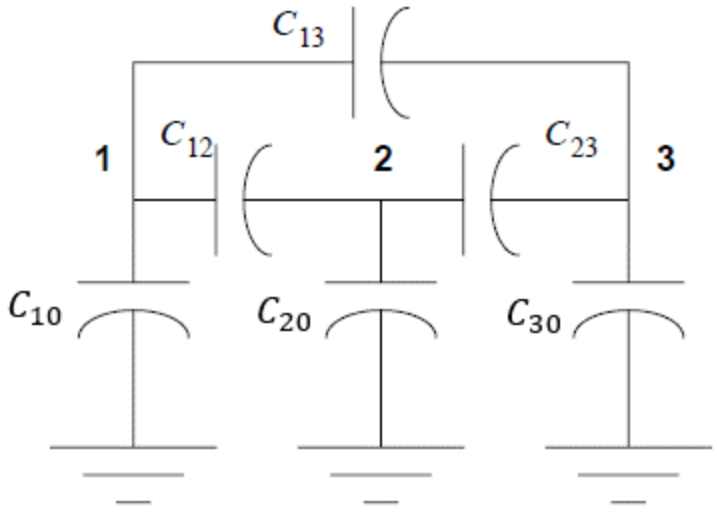
The results obtained will also apply to problems with non-zero resistance if L_{ij} is replaced by the impedance Z_{ij} in the formulas. As with the admittance matrix, the reduced impedance matrix, once computed, can be separated into the reduced ***R*** and ***L*** matrices by taking the real and imaginary parts of ***Z*** respectively and then dividing the imaginary part by ω to get ***L***.

Grounding and Floating Conductors

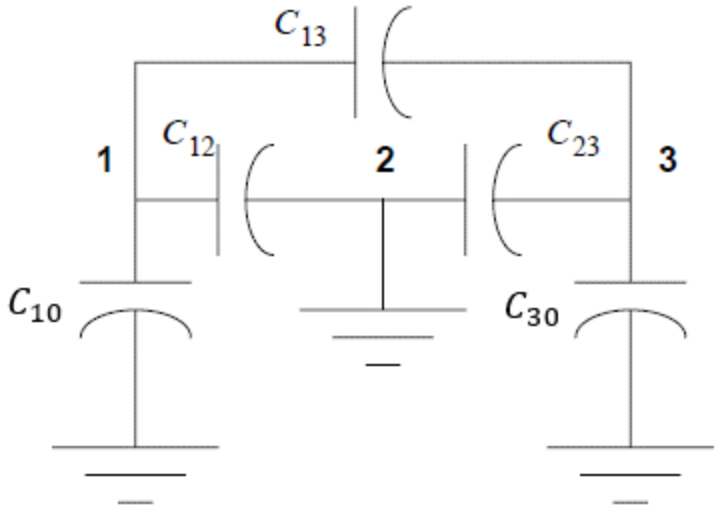
This topic describes the effects of grounding and floating conductors on the capacitance and inductance matrices.

Grounding a Conductor (Effect on the Capacitance Matrix)

Suppose a 3x3 Maxwell capacitance matrix. A collection of two-terminal capacitances can be synthesized using this same capacitance matrix. Recall the figure below from the section introduction:



Suppose it is necessary to ground node number 2. This would involve placing a short across the capacitor:



This eliminates C_{20} from the circuit, effectively leaving C_{10} in parallel with C_{12}, and C_{30} in parallel with C_{23}.The former node-to-node coupling capacitances C_{12} and C_{23} are now grounded.

The corresponding Maxwell capacitance matrix for the reduced system is:

$$\begin{bmatrix} C_{11}^M & C_{12}^M \\ C_{12}^M & C_{22}^M \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{13} \\ -C_{12} & C_{30}+C_{13}+C_{23} \end{bmatrix}$$

The diagonal entries that remain are unchanged because the total capacitance on each node is the same as before. The remaining off-diagonal entries are also unchanged. This reduced 2x2 capacitance matrix could also have been obtained simply by deleting row and column number 2 of the original 3x3 capacitance matrix.

> **Summary:**
>
> To compute the reduced capacitance matrix obtained by grounding the i^{th} conductor, delete row i and column i from the original Maxwell capacitance matrix.

The ***G*** matrix is affected in the same way as the capacitance matrix when a conductor is grounded: delete the corresponding row and column of the matrix.

Floating a Conductor (Effect on the Inductance Matrix)

Suppose three conductors and a corresponding 3x3 inductance matrix:

$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{23} \\ L_{13} & L_{23} & L_{33} \end{bmatrix}$$

The corresponding magnetic fluxes are:

$$\phi_1 = L_{11}i_1 + L_{12}i_2 + L_{13}i_3$$

$$\phi_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3$$

$$\phi_3 = L_{13}i_1 + L_{23}i_2 + L_{33}i_3$$

Open-circuiting one of the conductors forces its current to be zero.

For this example, open-circuit conductor 2.

$i_2 = 0$, and the fluxes associated with the remaining conductors are:

$$\phi_1 = L_{11}i_1 + L_{13}i_3$$

$$\phi_2 = L_{12}i_1 + L_{23}i_3$$

$$\phi_3 = L_{13}i_1 + L_{33}i_3$$

The remaining conductors can be characterized by a reduced 2x2 inductance matrix:

$$\begin{bmatrix} L_{11} & L_{13} \\ L_{13} & L_{33} \end{bmatrix}$$

The situation here is completely analogous to what happened to the capacitance matrix when grounding a conductor. To float a conductor, simply delete one row and column from the inductance matrix.

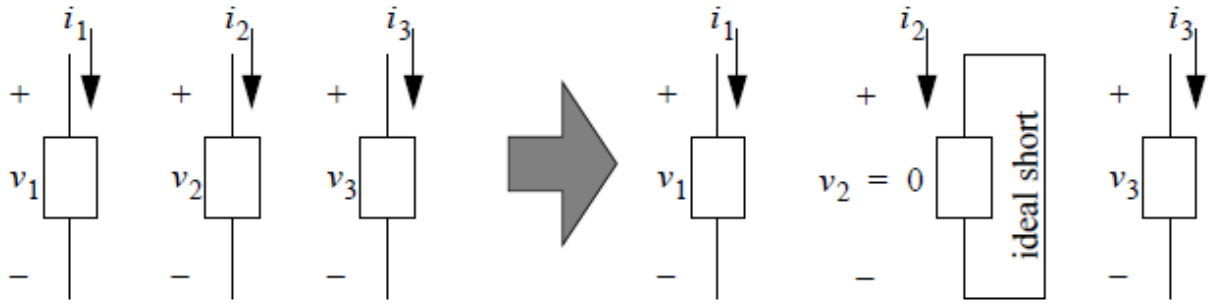
> **Summary:**
>
> To compute the reduced inductance matrix obtained by floating (open-circuiting) the i^{th} conductor, delete row i and column i from the original matrix

It's worth noting that there is still a voltage drop across conductor 2 (findable by taking the time derivative of its flux). This may seem counter-intuitive. How can there be a voltage drop if there's no current flowing through the conductor? This paradox is resolved by remembering that the voltage drop is induced by magnetic fields produced by the time-varying currents flowing in the surrounding conductors. The resistance matrix is affected in the same way as the inductance matrix when open-circuiting a conductor.

Grounding a Conductor (Effect on the Inductance Matrix)

When dealing with the inductance circuit element, the term "grounding" is a misnomer. It would be better described as "zeroing" the voltage across the element, as if by connecting an ideal short circuit across its terminals.

The figure below shows an example of "grounding" an inductance element by zeroing out its voltage drop with an ideal short circuit. Conductor 2 has been zeroed out here:



Zeroing an inductor's voltage is equivalent to zeroing its flux. If the example above is a set of 3 coupled inductances where conductor 2 has been grounded, then:

$$\phi_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3 = 0$$

Zeroing out the voltage does not simultaneously force the current in conductor 2 to be zero. It's possible for a current to flow in the loop formed by this conductor and the ideal short placed across it. In fact, this current can be computed by solving the above equation:

$$i_2 = -\frac{L_{12}}{L_{22}} i_1 - \frac{L_{23}}{L_{22}} i_3$$

In other words, i_2 is completely determined by the currents i_1 and i_3 in the surrounding conductors. All occurrences of i_2 can be eliminated from the expressions for the fluxes in these remaining conductors:

$$\begin{aligned}\phi_1 &= L_{11} i_1 + L_{12} i_2 + L_{13} i_3 \\ &= L_{11} i_1 + L_{12} \left(-\frac{L_{12}}{L_{22}} i_1 - \frac{L_{23}}{L_{22}} i_3 \right) + L_{13} i_3 \\ &= \left(L_{11} - \frac{L_{12}^2}{L_{22}} \right) i_1 + \left(L_{13} - \frac{L_{12} L_{13}}{L_{22}} \right) i_3\end{aligned}$$

In a similar way, find the expression for ϕ_3:

$$\phi_3 = \left(L_{13} - \frac{L_{12} L_{23}}{L_{22}} \right) i_1 + \left(L_{33} - \frac{L_{23}^2}{L_{22}} \right) i_3$$

The reduced 2x2 inductance matrix for the problem is:

$$\begin{bmatrix} L_{11} - \frac{L_{12}^2}{L_{22}} & L_{13} - \frac{L_{12} L_{13}}{L_{22}} \\ L_{13} - \frac{L_{12} L_{13}}{L_{22}} & L_{33} - \frac{L_{23}^2}{L_{22}} \end{bmatrix}$$

The same result could be arrived at by taking the original 3x3 matrix and performing a single step of Gaussian elimination on it using entry (2,2) as the pivot.

Notice that the self-inductances in the reduced matrix will be lower than the self-inductances of the original matrix, due to the subtraction of positive quantities like L_{12}^2 / L_{22}.

The discussion above is only rigorously true for lossless (zero-resistance) conductors. For conductors with finite resistances, work with the ***Z*** matrix, as described in the section introduction.

Floating a Conductor (Effect on the Capacitance Matrix)

Recall from circuit theory that the current flowing into the + terminal of a capacitor is given by the time derivative of the charge on its top plate:

$$i = \frac{dQ_+}{dt}$$

Thus, zeroing the current in a particular capacitor forces the charge on it to become a constant. If the open-circuited capacitor is discharged initially (i.e., it has zero charge), then it has zero charge forever. *Assume from this point on that all capacitors are initially discharged*. This simplifies the following discussion.

Consider a three-conductor system in which conductor 2 is open-circuited (floated). Now $Q_2 = 0$. The total charge on conductor 2 is given by the Maxwell capacitance matrix's second row:

$$Q_2 = C_{12}^M v_1 + C_{22}^M v_2 + C_{23}^M v_3 = 0$$

Notice that the voltage on conductor 2 is not necessarily equal to zero. In fact, it is now controlled entirely by the voltages on the surrounding conductors:

$$v_2 = -\frac{C_{12}^M}{C_{22}^M} v_1 - \frac{C_{23}^M}{C_{22}^M} v_3$$

The voltage on conductor 2 must be able to adjust itself in response to changes in the voltages on the surrounding conductors to ensure that its total charge remains equal to zero.

Eliminate any occurrence of the voltage v_2 in the expressions for the charges on the remaining conductors. For example, for conductor 1:

$$\begin{aligned} q_1 &= C_{11}^M v_1 + C_{12}^M v_2 + C_{13}^M v_3 \\ &= C_{11}^M v_1 + C_{12}^M \left(-\frac{C_{12}^M}{C_{22}^M} v_1 - \frac{C_{23}^M}{C_{22}^M} v_3 \right) + C_{13}^M v_3 \\ &= \left(C_{11}^M - \frac{\left(C_{12}^M\right)^2}{C_{22}^M} \right) v_1 + \left(C_{13}^M - \frac{C_{12}^M C_{23}^M}{C_{22}^M} \right) v_3 \end{aligned}$$

Following the same procedure for conductor 3 arrives at the reduced 2x2 capacitance matrix for the problem:

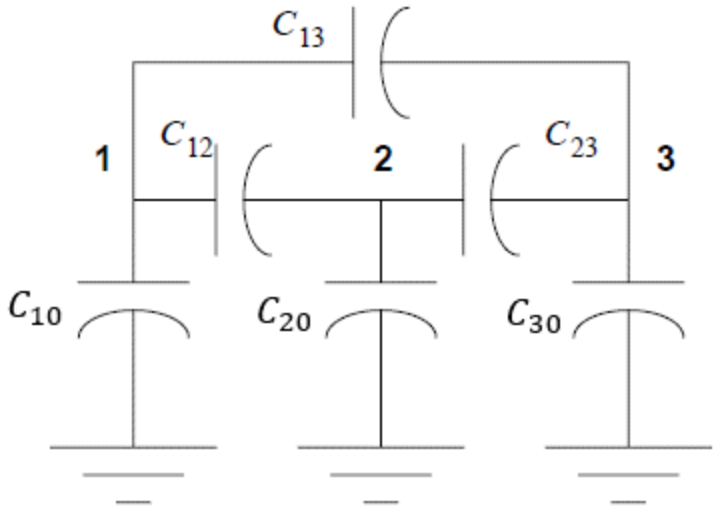
$$\begin{bmatrix} C_{11}^M - \frac{\left(C_{12}^M\right)^2}{C_{22}^M} & C_{13}^M - \frac{C_{12}^M C_{23}^M}{C_{22}^M} \\ C_{13}^M - \frac{C_{12}^M C_{23}^M}{C_{22}^M} & C_{33}^M - \frac{\left(C_{23}^M\right)^2}{C_{22}^M} \end{bmatrix}$$

Taking the original capacitance matrix and deleting row and column 2 would arrive at a similar result, but there is clearly an additional correction term that must then be subtracted from the remaining entries.

There are two implications of this:

- When floating a conductor, the self-capacitances of the remaining conductors decrease. This is because the original self capacitances are positive but the correction term is negative.
- Because the original coupling capacitance is negative and the correction term is also negative, the magnitude of the coupling capacitances increases.

To think of the effect on coupling capacitance in another way, return to this figure:



If node 2 is disconnected from any external voltage sources, there is now an additional capacitive path from node 1 to node 3 by way of capacitances C_{12} and C_{23} (C_{22} is involved as well, in a complicated way). This extra path is in parallel to the direct coupling capacitance C_{13}. Since capacitors connected in parallel add, the coupling between the two nodes is increased.

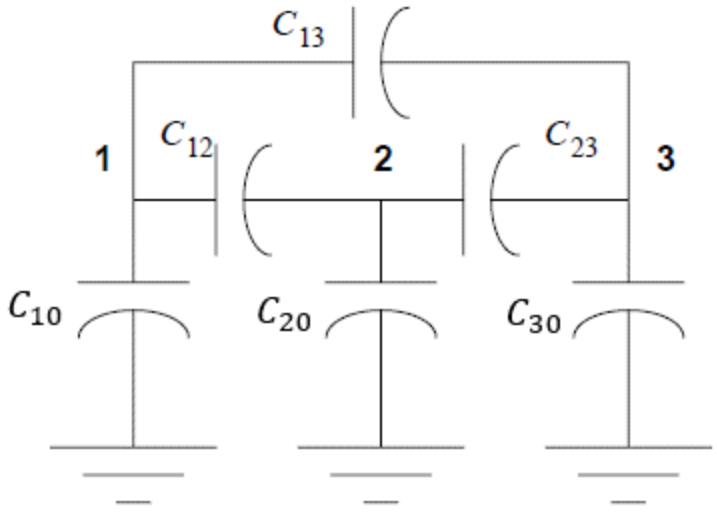
The Maxwell self capacitance on node 1 drops because the capacitance seen looking through this extra path consists of C_{12} in series with the parallel combination of C_{22} and C_{23}. Adding capacitors in series lowers the capacitance value, so the contribution of C_{12} to the Maxwell self-capacitance is decreased.

Connecting Elements

This topic describes the effects of connecting elements in parallel or in series.

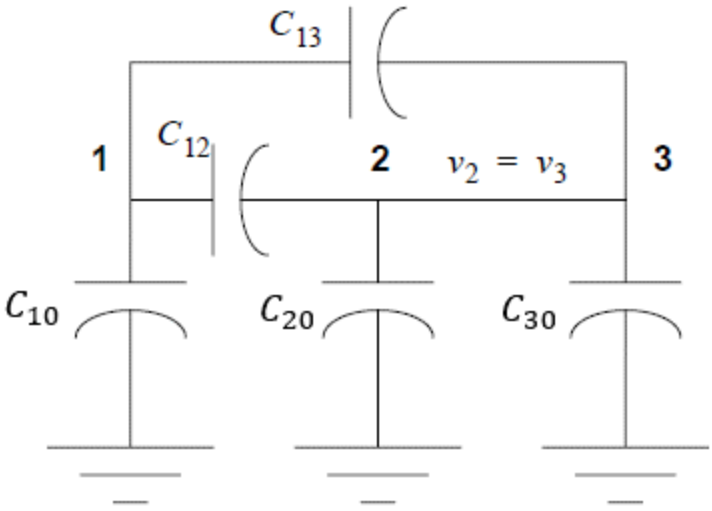
Connecting Elements in Parallel (Effect on the Capacitance Matrix)

A parallel connection in this context means forcing two conductor voltages to be the same. This could be accomplished by placing an ideal short circuit between the two conductors. Look again at the figure below.



Capacitances C_{20} and C_{30} could be connected in parallel by placing a short from nodes 2 to 3.

This results in the following:



Several things happen:

- The coupling capacitance C_{23} is shorted out and vanishes.
- Capacitances C_{12} and C_{13} are placed in parallel; they add to form a new coupling capacitance $C_{12} + C_{13}$ from node 1 to the combined nodes 2 and 3.
- Capacitances C_{20} and C_{30} also add in parallel to become $C_{22} + C_{33}$.

The net effect of this is to produce the reduced 2x2 Maxwell capacitance matrix given below:

$$\begin{bmatrix} C_{11}^M & C_{12}^M \\ C_{12}^M & C_{22}^M \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -(C_{12}+C_{13}) \\ -(C_{12}+C_{13}) & C_{20}+C_{30}+C_{12}+C_{13} \end{bmatrix}$$

Compare this with the original Maxwell capacitance matrix:

$$\begin{bmatrix} C_{11}^M & C_{12}^M & C_{13}^M \\ C_{12}^M & C_{22}^M & C_{23}^M \\ C_{13}^M & C_{23}^M & C_{33}^M \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix}$$

The same reduced 2x2 matrix could have been obtained by:

- Adding row 2 to row 3 of the original 3x3 Maxwell capacitance matrix, then deleting the original row 2 to obtain:

  $$\begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12}-C_{13} & C_{20}+C_{12} & C_{30}+C_{13} \end{bmatrix}$$

- Adding the resulting column 2 to column 3, then deleting the old column 2 to obtain:

  $$\begin{bmatrix} C_{10}+C_{12}+C_{13} & -(C_{12}+C_{13}) \\ -(C_{12}+C_{13}) & C_{20}+C_{30}+C_{12}+C_{13} \end{bmatrix}$$

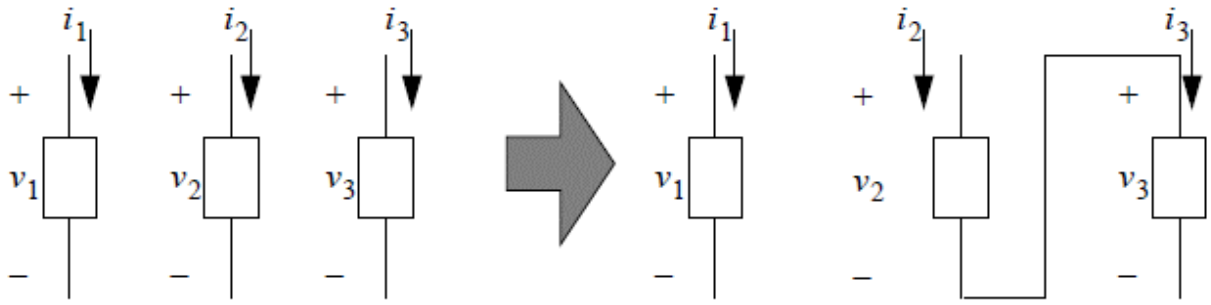
Summary:

To compute the reduced capacitance matrix obtained by connecting conductor i in parallel with conductor j (that is, short-circuiting the conductors together), add row i to row j of the capacitance matrix (and delete the old row i). Then add column i of the result to column j (and delete the old column i).

Connecting Elements in Series (Effect on the Inductance Matrix)

A series connection forces the current in one branch to flow into another.

The figure below shows an example of connecting two floating inductances (in this case, branches 2 and 3) in series. In this example, the – node of branch 2 is connected to the + node of branch 3.



Since the inductance element is floating (neither end necessarily grounded), the series connection can be accomplished in two different ways. For this example, assume it is desirable to connect the – reference node of one element to the + node of another. Performing the connection another way (for example, connecting two – nodes together) would involve first performing a source-sink exchange operation on one of the branches to redefine the +/– reference direction (See: Joining Elements in Parallel).

In the scenario pictured above, the series connection produces a single "macro" branch. Two features of the series connection are clear from the figure:

- By Kirchhoff's Current Law (KCL), $i_2 = i_3$.
- By Kirchhoff's Voltage Law (KVL), the net voltage across the combined elements is $v = v_2 + v_3$. If assuming zero initial conditions, the fluxes ϕ_2 and ϕ_3 also add.

Using these observations, obtain the branch relationship for the series combination of 2 and 3 as:

$$\phi_1 = L_{11}i_1 + L_{12}i_2 + L_{13}i_3$$

$$\phi_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3$$

$$\phi_3 = L_{13}i_1 + L_{23}i_2 + L_{33}i_3$$

However, $i_2 = i_3 = i$ implies:

$$\phi_1 = L_{11}i_1 + (L_{12} + L_{13})i$$

$$\phi_2 = L_{12}i_1 + (L_{22} + L_{23})i$$

$$\phi_3 = L_{13}i_1 + (L_{23} + L_{33})i$$

Since $\phi = \phi_2 + \phi_3$, add the last two equations together to obtain ϕ:

$$\phi = (L_{12} + L_{13})i_1 + (L_{22} + 2L_{23} + L_{33})i$$

Collecting together the results for the remaining branch fluxes ϕ_1 and ϕ obtains the reduced 2x2 inductance matrix for the problem:

$$\begin{bmatrix} L_{11} & L_{12}+L_{13} \\ L_{12}+L_{13} & L_{22}+2L_{23}+L_{33} \end{bmatrix}$$

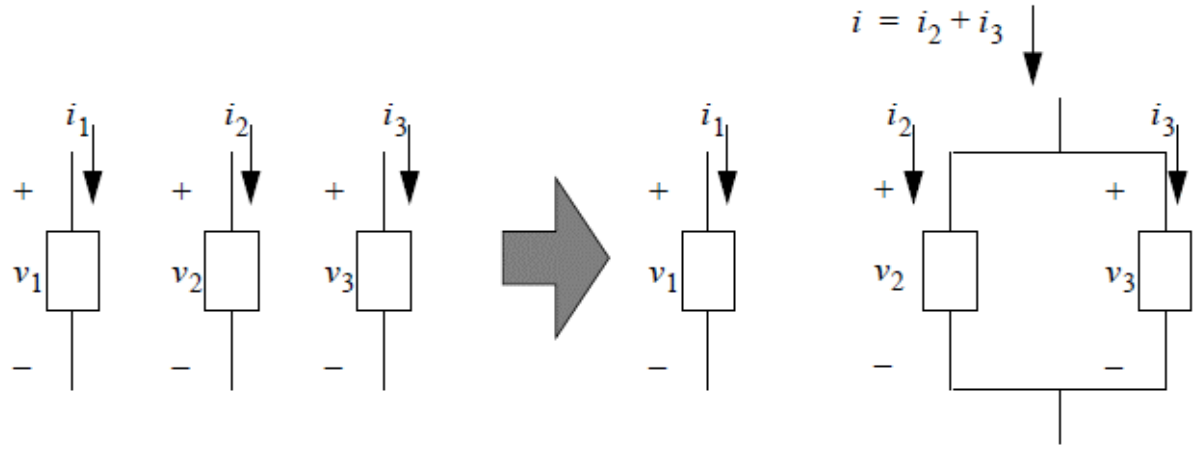
Comparing this with the original inductance matrix, note that the same result could have been obtained by adding rows 2 and 3, and then adding the resulting columns 2 and 3 together.

Summary:

To compute the reduced inductance matrix obtained by connecting conductor in series with conductor (that is, the – node of the first conductor to the + node of the second), add row to row of the inductance matrix (and delete the old row). Then add column of the result to column (and delete the old column).

Joining Elements in Parallel

The intent of this operation is to force the branch voltages on two elements to be the same. This can be accomplished by adding two ideal shorts to the circuit. When dealing with floating inductances, be careful to specify how the nodes of the elements are connected. Assume here that one connection is made between the two + nodes (source terminals), and another between the two – nodes (sink terminals), as in the figure below.



If a different connection is desired, a source-sink exchange operation can first be performed on one of the conductors.

Observe that the following relationships now hold:

- The two branch voltages are the same: $v_2 = v_3 = v$
- The fluxes are also the same: $\phi_2 = \phi_3 = \phi$
- The total current of parallel combination is simply the sum of the two currents: $i = i_2 + i_3$

The math of dealing with this connection is more complicated than that in previous topics.

The basic procedure for evaluating an inductance matrix is to perform a sequence of "experiments" in which one branch at a time is excited by a 1 amp current source. All other branches are left open-circuited. The fluxes on all the different circuit branches are measured, and this set of fluxes constitutes a single column of the desired inductance matrix.

For this particular problem, first excite i_1 with 1 amp while keeping $i = i_2 + i_3 = 0$. Then set $i_1 = 0$ and $i = i_2 + i_3 = 1\ \mathbf{amp}$.

Case of $i_1 = 1$ and $i = 0$ (calculating L_{11} and L_{12})

Because the total current i of the parallel branches is zero, $i_3 = -i_2$.

Looking at the equations for the fluxes of branches 2 and 3:

$$\phi_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3 = L_{12}i_1 + (L_{22} - L_{23})i_2$$

$$\phi_3 = L_{13}i_1 + L_{23}i_2 + L_{33}i_3 = L_{13}i_1 + (L_{23} - L_{33})i_2$$

Because branches 2 and 3 are connected in parallel, $\phi_2 = \phi_3$, implying:

$$L_{12}i_1 + (L_{22} - L_{23})i_2 = L_{13}i_1 + (L_{23} - L_{33})i_2$$

Solve for the current i_2 in terms of i_1:

$$(L_{22} - 2L_{23} + L_{33})i_2 = (L_{13} - L_{12})i_1$$

$$i_2 = \frac{L_{13} - L_{12}}{L_{22} - 2L_{23} + L_{33}} i_1 = -i_3$$

Return to the equation for ϕ_2 to find $\phi = \phi_2 = \phi_3$:

$$\phi_2 = L_{12}i_1 + (L_{22} - L_{23})i_2$$

$$= L_{12}i_1 + (L_{22} - L_{23})\frac{L_{13} - L_{12}}{L_{22} - 2L_{23} + L_{33}} i_1$$

$$= \left[L_{12} + \frac{(L_{22} - L_{23})(L_{13} - L_{12})}{L_{22} - 2L_{23} + L_{33}}\right] i_1$$

Using the equation for ϕ_3 would have obtained the same result. The quantity in brackets above is the mutual inductance between conductor 1 and the parallel combination of conductors 2 and 3.

To compute the self-inductance of conductor 1, use its flux equation:

$$\phi_1 = L_{11}i_1 + L_{12}i_2 + L_{13}i_3$$

$$= L_{11}i_1 + (L_{12} - L_{13})i_2$$

$$= \left[L_{11} - \frac{(L_{12}-L_{13})^2}{L_{22}-2L_{23}+L_{33}}\right]i_1$$

The new self-inductance of conductor 1 is always lower than the original self-inductance L_{11}. Why should placing conductors 2 and 3 in parallel affect the self-inductance of a different conductor? The answer can be reasoned as follows: before the parallel connection was made, a current flowing in conductor 1 would induce voltage drops in conductors 2 and 3. These drops are not necessarily the same. When the connection between conductors 2 and 3 is made parallel, their voltage drops must equalize. This is accomplished by inducing a current which flows in the closed loop formed by conductors 2 and 3 and the ideal shorts across them. The existence of this current in branches 2 and 3 in turn induces a voltage drop back on conductor 1. This back-EMF opposes the one induced by direct current flow.

Case of $i_1 = 0$ and $i = 1$ (calculating L_{22})

There is only one quantity left to compute: the self-inductance for the parallel combination of conductors 2 and 3. Because $i = i_2 + i_3 = 1$, $i_3 = i - i_2$.

From the flux equations, gather that:

$$\phi_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3 = 0 + (L_{22} - L_{23})i_2 + L_{23}i$$

$$\phi_3 = L_{13}i_1 + L_{23}i_2 + L_{33}i_3 = 0 + (L_{23} - L_{33})i_2 + L_{33}i$$

Since $\phi_2 = \phi_3$:

$$(L_{22} - L_{23})i_2 + L_{23}i = (L_{23} - L_{33})i_2 + L_{33}i$$

Solving for i_2 in terms of i yields:

$$i_2 = \frac{L_{33}-L_{23}}{L_{22}-2L_{23}+L_{33}}i$$

Finally, find the flux ϕ in the parallel conductors:

$$\phi = (L_{23} - L_{33})i_2 + L_{33}i$$

$$= \left[L_{33} - \frac{(L_{33}-L_{23})^2}{L_{22}-2L_{23}+L_{33}}\right]i$$

$$= \left[\frac{L_{22}L_{33}-L_{23}^2}{L_{22}-2L_{23}+L_{33}}\right]i$$

The quantity in brackets is the self-inductance of the parallel combination. The parallel combination has a self-inductance which is lower than either of the original self-inductances of the two branches that were connected together.

Summary

The reduced 2x2 inductance matrix for the parallel connection is given below:

$$\begin{bmatrix} L_{11} - \frac{(L_{12}-L_{13})^2}{L_{22}-2L_{23}+L_{33}} & L_{12} + \frac{(L_{22}-L_{23})(L_{13}-L_{12})}{L_{22}-2L_{23}+L_{33}} \\ L_{12} + \frac{(L_{22}-L_{23})(L_{13}-L_{12})}{L_{22}-2L_{23}+L_{33}} & \frac{L_{22}L_{33}-L_{23}^2}{L_{22}-2L_{23}+L_{33}} \end{bmatrix}$$

Join Selected Terminals Operation

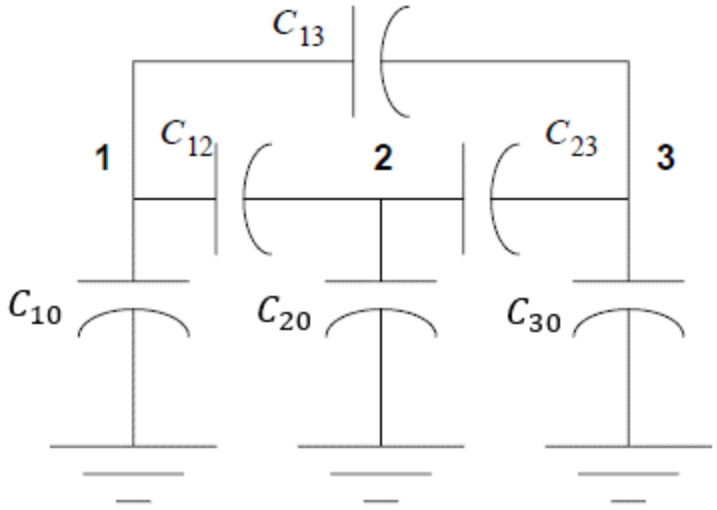
The Join Selected Terminals operation is a version of the Join in Parallel operation intended for nets with multiple sources. The voltages on the selected source terminals are forced to be equal, and a new combined source terminal is created. The sink terminal of the net is left unchanged by this operation, and the net's capacitance is unaltered. The same mathematics described above are used when joining terminals.

Floating at Infinity for Capacitance Matrices

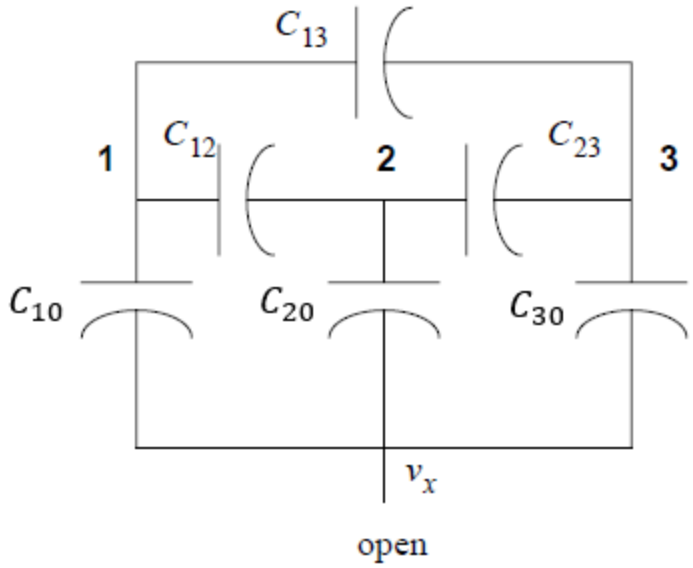
Q3D Extractor uses a boundary element technique to calculate the capacitance matrix for a 3-dimensional structure. During the solution process, it is assumed that the zero reference for potentials is a point infinitely distant from the problem region. Imagine putting the problem geometry inside a large, conducting balloon; this balloon is shorted to ground. If there is a non-zero total charge in the problem region, this gives rise to a "mirror" charge spread across the balloon. Next, suppose that the balloon is inflated until it becomes infinitely large. This enormous grounded conductor continues to mirror the charges in the problem region.

Sometimes this is desired; sometimes it's not. Sometimes what is desired is that the balloon have zero total charge, as if the connection from ground to the conducting balloon has been broken. If it's initially discharged, it will continue to have zero total charge. An interesting consequence of making this change is that the balloon may no longer have a potential of zero volts; the ground reference for the problem has "shifted" to a new location at a finite distance.

What is the effect of this operation on the capacitance matrix computed by Q3D? Consider the following figure.



If the ground connection is broken and the former ground node is left open-circuited, the result looks like the following:



Nodes 1, 2, and 3 are assumed to have known voltages.

The voltage v_x on the former ground node is unknown. This voltage can be determined by calculating the total charge on the node, and asking what voltage is required to ensure that this

total charge is zero. Summing over the bottom plates of C_{10}, C_{20}, and C_{30}, the total charge is found to be:

$$q = C_{10}\left(v_x - v_1\right) + C_{20}\left(v_x - v_2\right) + C_{30}\left(v_x - v_3\right) = 0$$

Demanding zero total charge allows computation of the voltage v_x on the ground node:

$$v_x = \frac{C_{10}v_1 + C_{20}v_2 + C_{30}v_3}{C_{10} + C_{20} + C_{30}}$$

With this result, the new Maxwell capacitance matrix for the problem can be determined. Recall that the capacitance matrix can be calculated by zeroing all of the node voltages (1,2,3) except for one, which is set to 1 Volt. The total charges on nodes 1—3 are then determined. This set of charges constitutes one column of the Maxwell capacitance matrix.

Setting $v_1 = 1$ and $v_2 = v_3 = 0$ results in:

$$v_x = \frac{C_{10}v_1}{C_{10} + C_{20} + C_{30}}$$

The charge on node 1 is then:

$$q_1 = C_{10}\left(v_1 - v_x\right) + C_{12}\left(v_1 - v_2\right) + C_{13}\left(v_1 - v_3\right)$$

$$= C_{10}\left(v_1 - \frac{C_{10}v_1}{C_{10} + C_{20} + C_{30}}\right) + C_{12}v_1 + C_{13}v_1$$

$$= \left[\frac{C_{10}\left(C_{20} + C_{30}\right)}{C_{10} + C_{20} + C_{30}} + C_{12} + C_{13}\right]v_1$$

The quantity in brackets is the new (Maxwell) self-capacitance term for conductor 1. The old self-capacitance was $C_{10} + C_{12} + C_{13}$, and so the self-capacitance has been decreased by this operation.

Using a similar approach, the charges on conductors 2 and 3 can be computed. The results are:

$$q_2 = -\left[C_{12} + \frac{C_{10}C_{20}}{C_{10} + C_{20} + C_{30}}\right]v_1$$

$$q_3 = -\left[C_{13} + \frac{C_{10}C_{30}}{C_{10} + C_{20} + C_{30}}\right]v_1$$

The quantities in brackets are the new coupling capacitances from node 1 to nodes 2 and 3, respectively. The original coupling capacitances were C_{12} and C_{13}, so the magnitude of the coupling has increased.

Using the procedure outlined above, similar experiments can be performed, setting $v_2 = 1$, $v_1 = v_3 = 0$, and then $v_3 = 1$, $v_1 = v_2 = 0$. The new Maxwell capacitance matrix summarizes the results:

$$\begin{bmatrix} \frac{C_{10}(C_{20}+C_{30})}{C_{10}+C_{20}+C_{30}} + C_{12} + C_{13} & -\left(C_{12} + \frac{C_{10}C_{20}}{C_{10}+C_{20}+C_{30}}\right) & -\left(C_{13} + \frac{C_{10}C_{30}}{C_{10}+C_{20}+C_{30}}\right) \\ -\left(C_{12} + \frac{C_{10}C_{20}}{C_{10}+C_{20}+C_{30}}\right) & \frac{C_{20}(C_{10}+C_{30})}{C_{10}+C_{20}+C_{30}} + C_{12} + C_{23} & -\left(C_{23} + \frac{C_{20}C_{30}}{C_{10}+C_{20}+C_{30}}\right) \\ -\left(C_{13} + \frac{C_{10}C_{30}}{C_{10}+C_{20}+C_{30}}\right) & -\left(C_{23} + \frac{C_{20}C_{30}}{C_{10}+C_{20}+C_{30}}\right) & \frac{C_{30}(C_{10}+C_{20})}{C_{10}+C_{20}+C_{30}} + C_{13} + C_{23} \end{bmatrix}$$

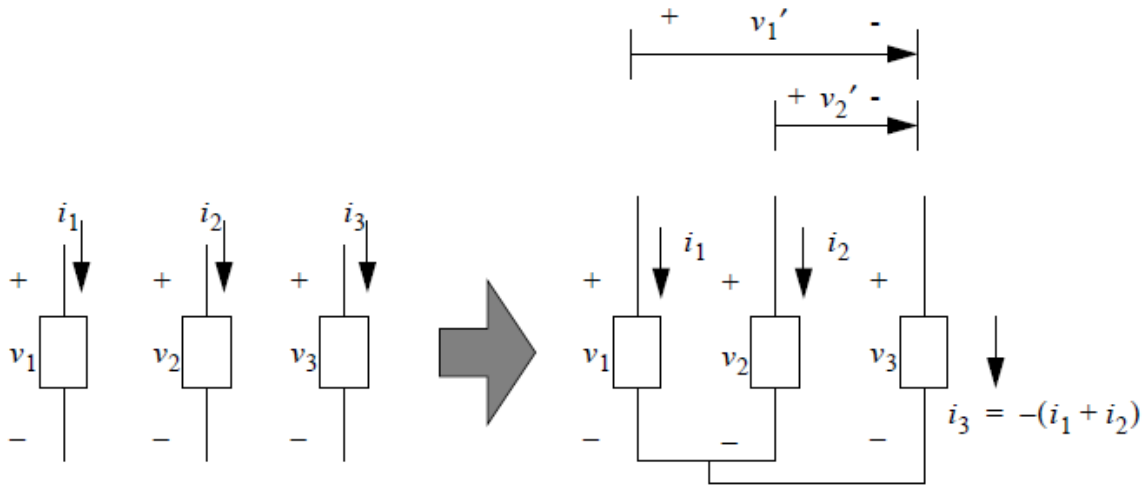
This is the first example of a matrix reduction operation that leaves the size of the matrix unchanged.

One thing to note about this operation is that the resulting capacitance matrix is indefinite. That is, the sum of the matrix rows (or columns) is zero. Thus, if trying to determine the values of the two-terminal capacitors to ground for the reduced matrix, they will be zero; the grounded capacitors disappear.

When there is a non-zero G matrix, the total current at the ground node (instead of total charge) must be equal to zero. Therefore, the Y matrix (not the capacitance matrix) must be used to derive the matrix reduction formula. The result shown above still applies, with C_{ij} replaced by Y_{ij}.

Return Path Reduction for Inductance Matrices

Return Path Reduction is an operation performed to ensure that the total current passing through a plane is equal to zero. From a set of conductors, one is selected to act as the "return." This conductor will gather up the current from the remaining conductors and send it back in the opposite direction. In the figure below, conductor 3 is being taken as the return path for the other conductors. All of the currents leaving through the – reference nodes of the other conductors are forced into the – node of the return path conductor. They pass through this conductor and re-emerge at its + node.



When performing this operation, no other connections can be made to the bottom node (the one to which all – references are connected). If connections were made, then current could flow through these external connections and disrupt the balance of currents shown in the figure.

For this reason, it is convenient to redefine the – reference for the branch voltages in the problem as the + node of the return path conductor. The new branch voltages are named v_1' and v_2'. When making this change, it is imperative that calculations incorporate the voltage drop across the return path conductor in calculations.

This is done by applying KVL:

$$v_1' = v_1 - v_3$$

$$v_2' = v_2 - v_3$$

To calculate these new branch voltages from the two remaining independent currents i_1 and i_2, switch to the element fluxes and observe that:

$$\phi_1' = \phi_1 - \phi_3$$

$$= (L_{11}i_1 + L_{12}i_2 + L_{13}i_3) - (L_{13}i_1 + L_{23}i_2 + L_{33}i_3)$$

$$= (L_{11} - L_{13})i_1 + (L_{12} - L_{23})i_2 + (L_{13} - L_{33})i_3$$

$$\phi_2' = \phi_2 - \phi_3$$

$$= (L_{12}i_1 + L_{22}i_2 + L_{23}i_3) - (L_{13}i_1 + L_{23}i_2 + L_{33}i_3)$$

$$= (L_{12} - L_{13})i_1 + (L_{22} - L_{23})i_2 + (L_{23} - L_{33})i_3$$

Since KCL requires that $i_3 = -i_1 - i_2$, this can be simplified further to:

$$\phi_1^{'} = (L_{11} - 2L_{13} + L_{33})i_1 + (L_{12} - L_{13} - L_{23} + L_{33})i_2$$

$$\phi_2^{'} = (L_{12} - L_{13} - L_{23} + L_{33})i_1 + (L_{22} - 2L_{23} + L_{33})i_2$$

From these relationships, read out the entries of the 2x2 reduced inductance matrix:

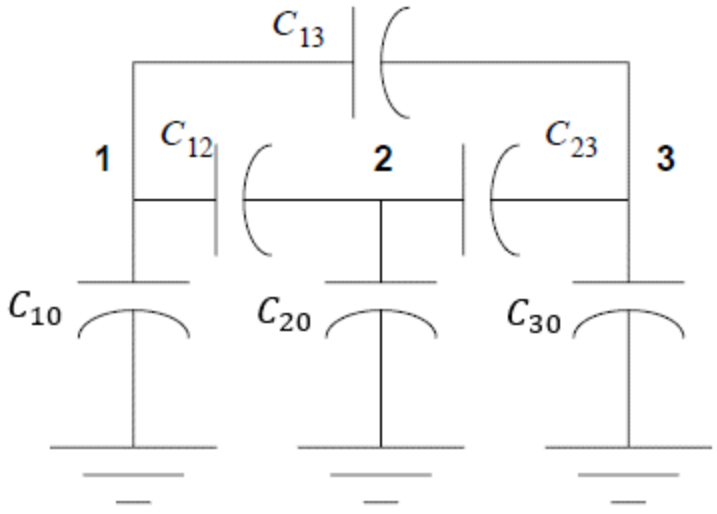
$$\begin{bmatrix} L_{11} - 2L_{13} + L_{33} & L_{12} - L_{13} - L_{23} + L_{33} \\ L_{12} - L_{13} - L_{23} + L_{33} & L_{22} - 2L_{23} + L_{33} \end{bmatrix}$$

Notice that the self-inductance of the return path conductor adds to every element of the matrix. For mutual inductances, subtract the mutuals between both of the coupled conductors and the return path conductor.

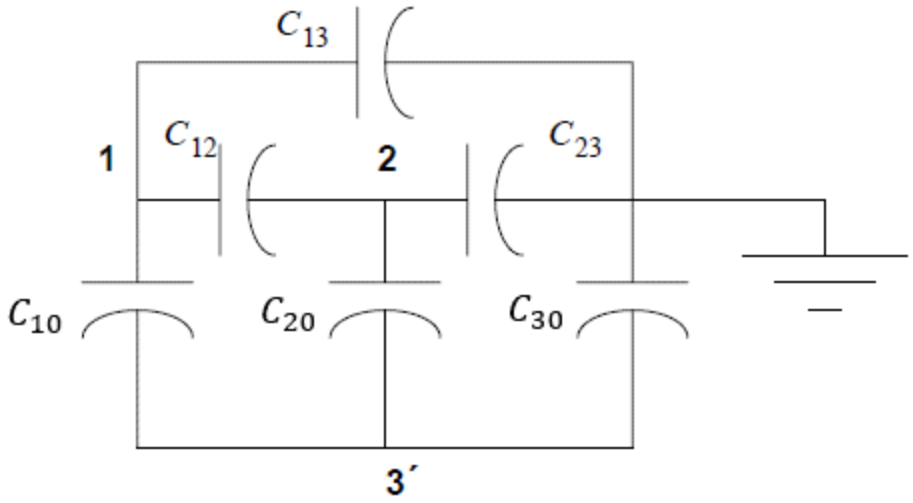
Redefining the Ground Reference for the Capacitance Matrix

Using an already computed capacitance matrix, it is possible to determine what the capacitance matrix would have been if a different conductor were selected as the ground reference.

Consider the two-terminal capacitance network of the figure below.



Assume that nodes 1, 2, and 3 have known voltages, and suppose that it is desirable to make node 3 the new ground reference. The new situation is shown below.



A few features of this new situation are worth noting:

- The remaining nodes 1 and 2 still have the same capacitors connected to them, and the self-capacitance terms in the Maxwell capacitance matrix remain unchanged.
- The coupling capacitances to the new node 3´ are the former capacitances to ground.

By inspection of this network, determine the new Maxwell capacitance matrix:

$$\begin{bmatrix} C_{11}^M & C_{12}^M & C_{13}^M \\ C_{12}^M & C_{22}^M & C_{23}^M \\ C_{13}^M & C_{23}^M & C_{33}^M \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{10} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{20} \\ -C_{10} & -C_{20} & C_{10}+C_{20}+C_{30} \end{bmatrix}$$

Comparing with the original Maxwell capacitance matrix, the redefinition of ground can be accomplished by replacing row 3 and column 3 with new entries. The off-diagonal entries are the former capacitances to ground, and the diagonal entry is the sum of the capacitances to ground from the original system.

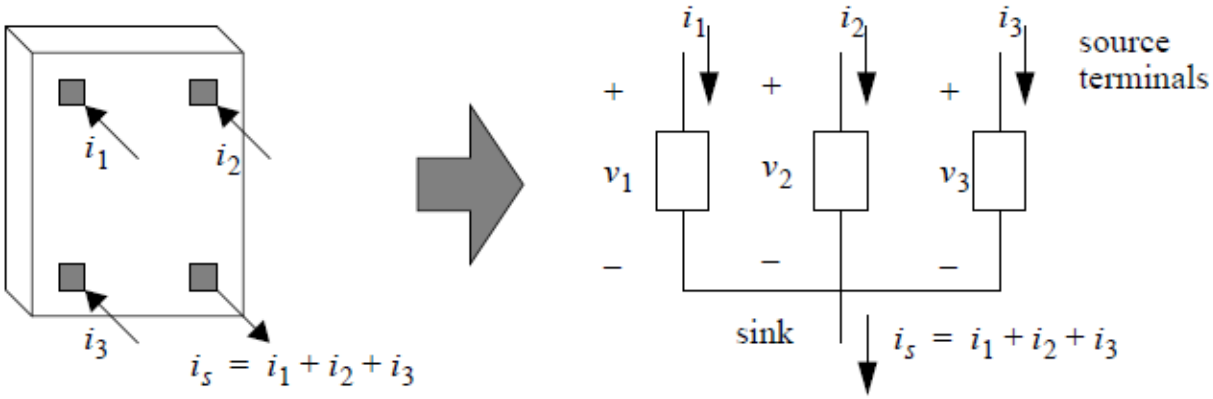
Operations for Multiterminal Conductors

In Q3D Extractor, a conductor with $n+1$terminals is modeled with a single sink terminal and n source terminals. All currents entering through the source terminals re-emerge from the sink terminal to satisfy Kirchhoff's current law.

The currents entering the source terminals may be considered to be independent of one another; the current leaving the sink terminal is clearly not independent of the others because it is just the sum of the source terminal currents.

Q3D Extractor also defines a set of n independent branch voltages for the multiterminal conductor, measured by taking the difference in potential from the source terminal to the sink terminal. The sink terminal is the common reference point for all of the branch voltages.

The relationship between the branch voltages and the source terminal currents is described by $n \times n$ inductance and resistance matrices (or equivalently, an impedance matrix). The figure below shows a 4-terminal conductor with 3 source terminals and 1 sink terminal. It is characterized by a set of 3 independent currents which are defined as entering the source terminals. A set of 3 independent voltages is also defined. These voltages share a common – reference in the sink terminal.

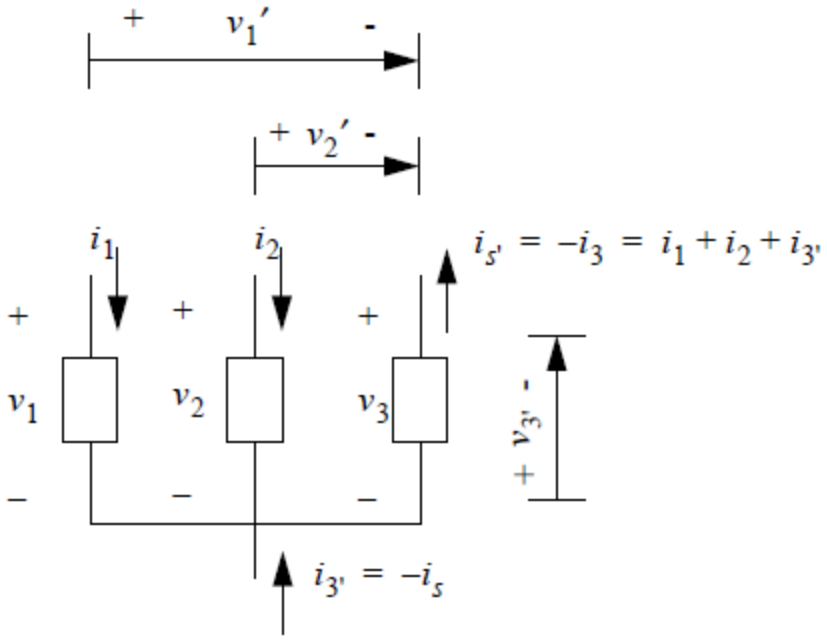


Exchanging (Moving) Source and Sink Points

The selection of the sink terminal is essentially arbitrary; any terminal could have been designated as the sink prior to solving the field problem. If the designated sink needs to be changed, circuit theory can be used to recompute the inductance matrix without having to re-run the field solver.

The approach is similar to the one we used earlier in performing return path reduction.

The figure below shows an example in which the sink point is moved to terminal 3. The former sink current i_s must now be treated as one of the three independent currents i_3' in the problem. Also, the branch voltages must be redefined with their common – reference as terminal 3.



Observe the following:

- By KVL, the new branch voltages can be calculated from the old ones as:

  $v_1' = v_1 - v_3$

  $v_2' = v_2 - v_3$

  $v_3' = -v_3$

  The new fluxes will obey analogous relationships.

- Most of the currents are unchanged. However, current 3 is more interesting. Per KCL:

  $i_3' = -i_1 - i_2 - i_3$

The KCL relationship is key, because it allows us to eliminate i_3 from the problem:

$$i_3 = -i_1 - i_2 - i_3'$$

Returning to the flux equations for the various branches:

$$\phi_1 = L_{11} i_1 + L_{12} i_2 + L_{13} i_3$$

$$= L_{11} i_1 + L_{12} i_2 - L_{13} \left(i_1 + i_2 + i_3' \right)$$

$$= (L_{11} - L_{13})i_1 + (L_{12} - L_{13})i_2 - L_{13}i_3^{'}$$

$$\phi_2 = L_{12}i_1 + L_{22}i_2 + L_{23}i_3$$

$$= L_{12}i_1 + L_{22}i_2 - L_{23}\left(i_1 + i_2 + i_3^{'}\right)$$

$$= (L_{12} - L_{23})i_1 + (L_{22} - L_{23})i_2 - L_{23}i_3^{'}$$

$$\phi_3 = L_{13}i_1 + L_{23}i_2 + L_{33}i_3$$

$$= L_{13}i_1 + L_{23}i_2 - L_{33}\left(i_1 + i_2 + i_3^{'}\right)$$

$$= (L_{13} - L_{33})i_1 + (L_{23} - L_{33})i_2 - L_{33}i_3^{'}$$

With these results, compute the new set of fluxes for the problem:

$$\phi_1^{'} = \phi_1 - \phi_3$$

$$= (L_{11} - 2L_{13} + L_{33})i_1 + (L_{12} - L_{13} - L_{23} + L_{33})i_2 + (L_{33} - L_{13})i_3^{'}$$

$$\phi_2^{'} = \phi_2 - \phi_3$$

$$= (L_{12} - L_{13} - L_{23} + L_{33})i_1 + (L_{22} - 2L_{23} + L_{33})i_2 + (L_{33} - L_{23})i_3^{'}$$

$$\phi_3^{'} = -\phi_3$$

$$= (L_{33} - L_{13})i_1 + (L_{33} - L_{23})i_2 + L_{33}i_3^{'}$$

Thus, the reduced inductance matrix is:

$$\begin{bmatrix} L_{11} - 2L_{13} + L_{33} & L_{12} - L_{13} - L_{23} + L_{33} & L_{33} - L_{13} \\ L_{12} - L_{13} - L_{23} + L_{33} & L_{22} - 2L_{23} + L_{33} & L_{33} - L_{23} \\ L_{33} - L_{13} & L_{33} - L_{23} & L_{33} \end{bmatrix}$$

Adding sink points

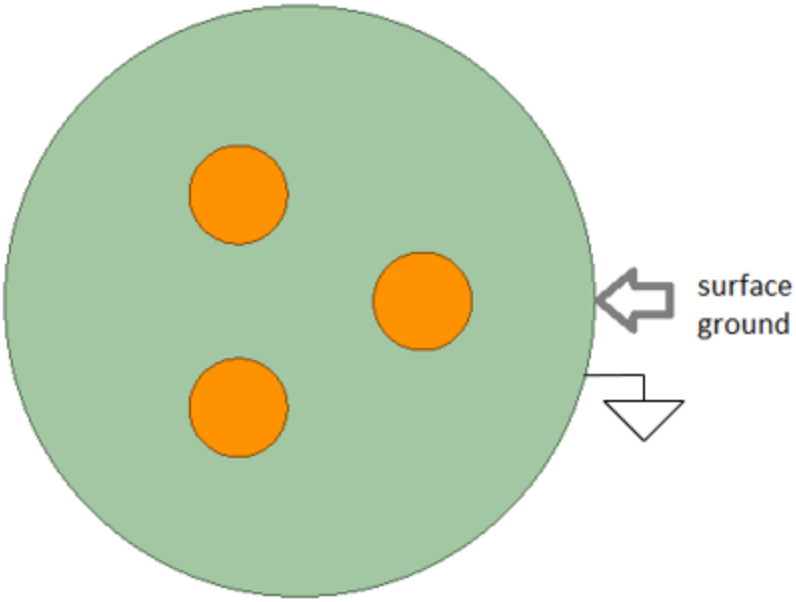
Adding an additional sink point to the multiterminal conductor means shorting one of the source terminals to the existing sink or sinks. This corresponds to the grounding operation for inductance matrices.

Set Ground vs. Add Ground in 2D Extractor

2D Extractor provides two different grounding operations:

- **Add Ground** – the same as the grounding operations previously described. Add Ground forces the potential on one capacitor node to zero volts and ensures that the voltage drop across one inductor is zeroed. In effect, the conductor is connected to the ground reference everywhere.
- **Set Ground** – In a closed problem, a non-conducting object must be assigned to act as a "surface ground." The outer boundary of this object is treated as the ground reference for capacitance calculations and as the current return path for inductance computations. Using the Set Ground operation in a closed problem, redefines the ground reference to be the user-selected signal or non-ideal ground conductor. The surface ground is converted to a floating conductor.

The figure below shows a typical situation in which the Set Ground operation is employed in 2D Extractor.



Using a Set Ground operation causes a sequence of 2 matrix reduction operations to be performed on the capacitance matrix. First, the surface ground conductor is floated. The effect on the capacitance matrix is identical to what occurs in 3D when performing a Float at Infinity operation to open-circuit the ground node. After the floating operation, a second matrix reduction is performed. The newly designated ground conductor is grounded in the traditional sense (that is, its row and column are deleted from the capacitance matrix).

The effect of Set Ground on the inductance matrix is easier to understand. The newly designated ground is taken as the return path for all currents. The matrix operations needed to do this are identical to the ones described in Return Path Reduction for Inductance Matrices.

The current in the bounding conductor always remains equal to the negative of the sum of the currents in the original signal conductors. Forcing one of the signal conductors to act as a return path ensures that the sum of these currents is zero. Therefore, this operation effectively open-circuits the bounding conductor. Thus, it is consistent with the operations performed on the capacitance matrix.

Differential Pair Matrix Reduction in 2D Extractor

Differential pair matrix reduction is a special type of operation that converts the single-ended signal voltages of a pair of conductors into a differential and a common mode voltage (collectively called mixed-mode voltages). Suppose conductors 1 and 2 have been designated as a differential pair, with conductor 1 as the positive reference and conductor 2 as the negative reference. Then the differential voltage v_d is defined as the voltage difference between them:

$$v_d = v_1 - v_2$$

The common-mode voltage v_{cm} is the average of the two conductor voltages:

$$v_{cm} = \frac{1}{2}\left(v_1 + v_2\right)$$

It is easy to go back and forth between the single-ended and mixed-mode voltages. To illustrate:

$$v_1 = \frac{1}{2}v_d + v_{cm}$$

$$v_2 = -\frac{1}{2}v_d + v_{cm}$$

Define mixed-mode currents as follows:

$$i_d = \frac{1}{2}\left(i_1 - i_2\right)$$

$$i_{cm} = i_1 + i_2$$

The single-ended currents can also be expressed in terms of mixed-mode currents:

$$i_1 = i_d + \frac{1}{2}i_{cm}$$

$$i_2 = -i_d + \frac{1}{2}i_{cm}$$

A 2x2 capacitance matrix can be converted to mixed-mode form. To begin, take the Maxwell capacitance matrix and rewrite the single-ended voltages in terms of the mixed-mode voltages:

$$q_1 = C_{11}^M v_1 + C_{12}^M v_2$$

$$= C_{11}^M \left(\frac{1}{2}v_d + v_{cm}\right) + C_{12}^M \left(-\frac{1}{2}v_d + v_{cm}\right)$$

$$= \tfrac{1}{2}\left(C_{11}^M - C_{12}^M\right)v_d + \left(C_{11}^M + C_{12}^M\right)v_{cm}$$

$$q_2 = C_{12}^M v_1 + C_{22}^M v_2$$

$$= C_{12}^M\left(\tfrac{1}{2}v_d + v_{cm}\right) + C_{22}^M\left(-\tfrac{1}{2}v_d + v_{cm}\right)$$

$$= \tfrac{1}{2}\left(C_{12}^M - C_{22}^M\right)v_d + \left(C_{12}^M + C_{22}^M\right)v_{cm}$$

Note that because a capacitor's current is the time-derivative of its charge, the charges q_1 and q_2 must be combined in the same way as the single-ended currents to find the mixed-mode charges (and currents):

$$q_d = \tfrac{1}{2}\left(q_1 - q_2\right)$$

$$= \tfrac{1}{4}\left(C_{11}^M - 2C_{12}^M + C_{22}^M\right)v_d + \tfrac{1}{2}\left(C_{11}^M - C_{22}^M\right)v_{cm}$$

$$q_{cm} = q_1 + q_2$$

$$= \tfrac{1}{2}\left(C_{11}^M - C_{22}^M\right)v_d + \left(C_{11}^M + 2C_{12}^M + C_{22}^M\right)v_{cm}$$

Thus, the mixed-mode capacitance matrix will be:

$$\begin{bmatrix} \tfrac{1}{4}\left(C_{11}^M - 2C_{12}^M + C_{22}^M\right) & \tfrac{1}{2}\left(C_{11}^M - C_{22}^M\right) \\ \tfrac{1}{2}\left(C_{11}^M - C_{22}^M\right) & C_{11}^M + 2C_{12}^M + C_{22}^M \end{bmatrix}$$

Notice that in a symmetrical structure $C_{11}^M = C_{22}^M$, and so the capacitance coupling between the differential and common mode will be zero in that case.

For the mixed-mode inductance matrix, start with the single-ended inductance matrix relationship and rewrite the single-ended fluxes in terms of mixed-mode currents. Thus:

$$\phi_1 = L_{11}i_1 + L_{12}i_2$$

$$= L_{11}\left(i_d + \tfrac{1}{2}i_{cm}\right) + L_{12}\left(-i_d + \tfrac{1}{2}i_{cm}\right)$$

$$= (L_{11} - L_{12})i_d + \tfrac{1}{2}\left(L_{11} + L_{12}\right)i_{cm}$$

$$\phi_2 = L_{12}i_1 + L_{22}i_2$$

$$= L_{12}\left(i_d + \tfrac{1}{2}i_{cm}\right) + L_{22}\left(-i_d + \tfrac{1}{2}i_{cm}\right)$$

$$= (L_{12} - L_{22})i_d + \frac{1}{2}(L_{12} + L_{22})i_{cm}$$

The fluxes must be combined in the same way as the single-ended voltages to yield the differential fluxes:

$$\phi_d = \phi_1 - \phi_2$$

$$= (L_{11} - 2L_{12} + L_{22})i_d + \frac{1}{2}(L_{11} - L_{22})i_{cm}$$

$$\phi_{cm} = \frac{1}{2}(\phi_1 + \phi_2)$$

$$= \frac{1}{2}(L_{11} - L_{22})i_d + \frac{1}{4}(L_{11} + 2L_{12} + L_{22})i_{cm}$$

Therefore, the differential inductance matrix is:

$$\begin{bmatrix} L_{11} - 2L_{12} + L_{22} & \frac{1}{2}(L_{11} - L_{22}) \\ \frac{1}{2}(L_{11} - L_{22}) & \frac{1}{4}(L_{11} + 2L_{12} + L_{22}) \end{bmatrix}$$

If the structure is symmetrical, $L_{11} = L_{22}$ and the inductive coupling between the differential and common-mode signals will be zero. This is why differential pairs are laid out in a symmetric fashion.

S-Parameter Calculations in Q3D Extractor

The fundamental solution quantities in Q3D are the lumped RLGC parameters for a 3D structure. It is important to understand that in the low frequency limit where lumped parameters are defined, many different circuit arrangements lead to the same RLGC values. For example, a single-section T network or a single-section Pi network are both valid representations of the same RLGC values. Even a multi-section ladder model could reduce to the same lumped RLGC values. Each of these networks would have different S-parameters, but the same RLGC values.

Therefore, if Q3D is asked to compute S-parameters from its RLGC solution, there is no unique way to accomplish that. Q3D must pick one circuit topology from the infinite universe of possible circuits, and then compute the S-parameters from it.

When computing S-parameters in Q3D, you can choose from two options: the **Equivalent transmission line model** and the **Lumped RLGC model**. The differences in the two approaches are described below.

Equivalent Transmission Line Model Approach

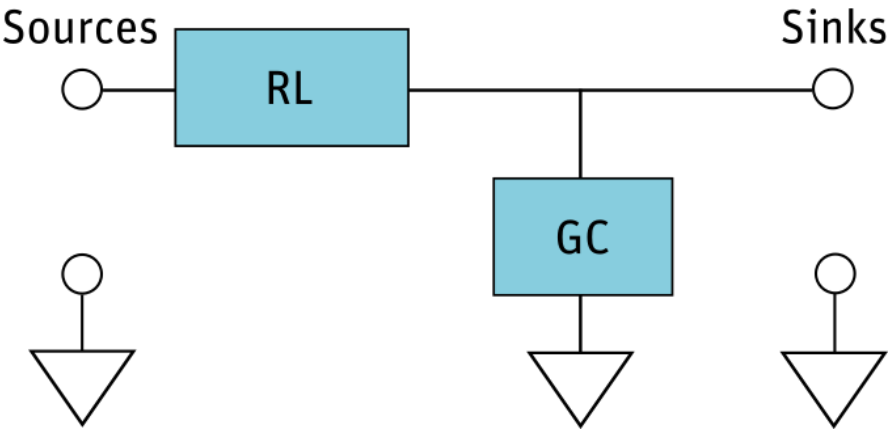
In this approach, Q3D assumes that the structure being analyzed is a collection of parallel interconnects of similar length. It then takes the lumped RLGC parameters for this structure and

distributes them uniformly across a system of coupled transmission lines. Distributed transmission line theory (similar to what 2D Extractor uses) is then used to calculate the S-parameters.

This approach works well when the 3D structure being analyzed matches the above assumptions: a collection of parallel wires of similar length. However, it can lead to non-physical results when the structure deviates substantially from this situation. Examples of structures that might cause problems are ones that involve wires with very different lengths, such as a 50 cm long wire next to a 5 mm long wire. For such structures, the new **Lumped RLGC model** approach is more appropriate.

Lumped RLGC Model Approach

In the lumped circuit approach, Q3D simply lumps all the capacitance and conductance (C and G) at the sink end of the model, and places the inductance and resistance (L and R) between the sources and sinks. See the following figure:



The S-parameters are then calculated from this network. This approach is recommended when dealing with arbitrary 3D structures.

Solution Process

Basic Assumptions in Q3D Extractor

All electromagnetic phenomena, including capacitance and inductance effects, are described by Maxwell 's equations. These relate the electric field $\vec{\mathbf{E}}$ to the magnetic field $\vec{\mathbf{H}}$ in the presence of materials described by electric permittivity **ε**, conductivity **σ** and magnetic permeability **μ**.

In differential frequency-domain form, Maxwell 's equations are:

$$\nabla \times \vec{E} = -j\omega\mu\vec{H}$$

$$\nabla \times \vec{H} = (\sigma + j\omega\varepsilon)\vec{E}$$

$$\nabla \cdot \varepsilon\vec{E} = \rho$$

and

$$\nabla \cdot \mu\vec{H} = 0$$

ρ denotes the density of free charges, ***j*** is the imaginary unit, and $\omega = 2\pi f$.

Q3D is based upon a simplification of Maxwell 's equations called the quasistatic approximation: it is assumed that the size of the structure to be analyzed is small compared to the wavelength at the maximum frequency of interest. A general rule of thumb is that the size of the structure should be less than one tenth of a wavelength. The wavelength is inversely proportional to frequency. In most printed circuit, IC and packaging materials it is approximately 15 cm at 1 GHz, and 1.5 cm at 10 GHz. This assumption allows Q3D to solve very complicated structures in an efficient way.

$$j\omega\varepsilon\vec{E}$$

$$j\omega\mu\vec{H}$$

For structures that are larger than one tenth of a wavelength, a full-wave field solver (one that considers the two-way coupling between the electric and magnetic fields, such as HFSS) should be used. The full-wave solver will ensure that all electromagnetic phenomena are modeled, but may be more time-consuming to run than Q3D.

Capacitance and Conductance Solution

The capacitance (C) and conductance (G) matrices for a system of conductors and dielectric objects are calculated together in the field solver.

Q3D takes the following steps:

- Divides the surfaces of conductors and dielectric objects into a triangular mesh (It is not necessary to mesh the interior of objects for this solution).
- Applies a 1 Volt excitation to one of the signal nets, and 0 volts to the others.
- Uses the fast multipole solver to compute the distribution of charges on the conductor surfaces, and bound charges on the dielectric surfaces. These charges are constrained by certain boundary conditions described below.
- Sums up the free charges on the conductor surfaces to evaluate one column of the capacitance and conductance matrices. These charges are in general complex quantities. The real part determines the capacitance and imaginary part determines the conductance.

- The above process is repeated, with each signal net being set to 1 Volt in turn while the other nets are maintained at 0 volts, until solutions have been obtained for each net.

Formulation of the Capacitance/Conductance Solution

The electrostatic potential Φ produced by a distribution of charges ρ on a surface *S* is given by the integral equation

$$\Phi\left(\vec{x}\right) = \int_S \frac{\rho\left(\vec{x}\,'\right)}{4\pi\varepsilon||\vec{x}-\vec{x}\,'||} dS$$

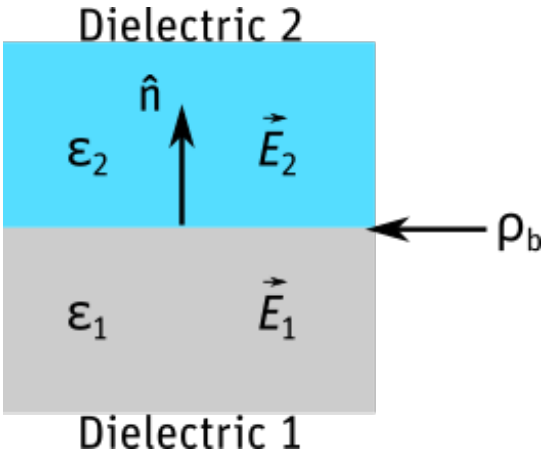
Here the operator || || denotes the Euclidean length of a vector. The equation is based upon an implicit assumption that the potential will go to zero as the distance between the source and observation points goes to infinity. This assumption can be relaxed by applying the **Floating at Infinity** matrix reduction operation as a post-processing step.

$\vec{x}$

If dielectric objects are present in the problem, they must also be taken into account. The electromagnetic boundary condition at the interface between two dissimilar dielectrics is the continuity of the normal electric flux:

$$\hat{\mathbf{n}} \cdot \left(\varepsilon_2 \vec{E}_2 - \varepsilon_1 \vec{E}_1\right) = 0$$

Here n̂ is the unit vector normal to the dielectric interface. The dielectric constants ε_1 and ε_2 on opposite sides of the interface are real-valued for lossless dielectrics and complex-valued (and possibly frequency-dependent) for lossy dielectrics. $\vec{E}_1$ and $\vec{E}_2$ are the corresponding electric field vectors.



ρ_b

$$\hat{\mathbf{n}} \cdot \left(\varepsilon_0 \vec{E}_2 - \varepsilon_0 \vec{E}_1 \right) = \rho_b$$

Because the electric field appearing in this equation is the negative of the gradient of the electrostatic potential Φ, it can be found by carefully differentiating the equation. The bound charge density ρ_b appearing in the equation constitutes an additional set of field sources that must be computed by the solver. Q3D creates additional mesh triangles on the dielectric interface and solves for the bound charge densities in these triangles, iteratively updating them until all conditions are satisfied.

Extracting Capacitance and Conductance from the Charge Solution

From circuit theory, it is known that the current *I* across the parallel combination of a conductance *G* and a capacitance *C* excited by a voltage *V* is given by

$$I = (G + j\omega C)V = j\omega Q$$

Here ω is the angular frequency, and we have made use of the fact that the current is the time derivative of the charge on the circuit node, *Q*. This implies that we can extract the conductance and capacitance if we know the total charge on the conductor and the applied voltage using the formula:

$$C - j\frac{G}{\omega} = \frac{Q}{V}$$

By matching the real and imaginary parts on both sides of this equation, we get separate expressions for the capacitance and conductance:

$$C = \mathbf{Re}\left(\frac{Q}{V}\right)$$

and

$$G = -\omega \mathbf{Im}\left(\frac{Q}{V}\right)$$

Q_{mn}

$$C_{mn} = \mathbf{Re}\left(\frac{Q_{mn}}{V_n}\right)$$

and

$$G_{mn} = -\omega \mathbf{Im}\left(\frac{Q_{mn}}{V_n}\right)$$

DC Resistance and Inductance Solution

To calculate the DC resistance and inductance matrices, Q3D Extractor does the following:

- Divides the interior of the conductors into a finite element mesh.
- Applies 1 amp to one of the source terminals while open-circuiting the remaining source terminals.
- Computes the electrostatic potential and volume current density everywhere within the excited conductors using the finite element conduction solver.
- Finds the resulting voltages on the source terminals. These give the entries for one column of the DC resistance matrix.
- Uses the IE solver to evaluate the magnetic fields produced by the DC current distribution, and finds the resulting stored energy. This gives the DC inductance values.
- Repeats the above process for every source terminal, setting each source in turn to 1 Amp while open-circuiting the others, until solutions have been obtained for each source.

Formulation of the DC Conduction Solution

Any object that has been identified as a conductor is included in the conduction simulation. The current density $\vec{J}$ in each conductor is, by Ohm 's law, proportional to the electric field that results from gradients in the electrostatic potential **Φ**:

$$\vec{J} = \sigma\vec{E} = -\sigma\nabla\Phi$$

where σ is the conductivity of the material.

Under steady state DC conditions, the amount of charge leaving any infinitesimally small region must equal the charge flowing into that region. That is, the charge density, ρ (x,y,z), in any region will not change with time:

$$\nabla \cdot \vec{J} = \frac{\partial \rho}{\partial t} = 0$$

Therefore Q3D must find a solution to the equation:

$$\nabla \cdot \sigma\nabla\Phi = 0$$

This is one form of Laplace 's equation. The conduction solver uses the finite element method with quadratic basis functions to compute the potential at each vertex and edge midpoint in the mesh. The finite element matrix is solved using a direct solver that exploits the sparsity of the linear system of equations. The boundary conditions that must be applied are:

- The normal component of the current density must be zero at an interface between the conductor and an insulating region, such as a dielectric or free space.

- The electrostatic potential must be continuous across an interface between two conductors.
- The electrostatic potential on a source or sink terminal is constant (equipotential) over the terminal surface, and the total current is equal to the externally applied source.

Extracting Resistance from the DC Conduction Solution

Resistance can be computed by finding the power loss in the conductors. Suppose that the n^{th} source terminal is excited with 1 Amp, producing a volume distribution of currents $\boldsymbol{J_n}$. The resulting Ohmic power loss $\boldsymbol{P}$ is given by the integral:

$$P = \int_V \frac{1}{\sigma} \left(\overrightarrow{J_n} \left(\overrightarrow{\mathbf{x}'} \right) \cdot \overrightarrow{J_n} \left(\overrightarrow{\mathbf{x}'} \right) \right) dV$$

From circuit theory, the DC power loss in a resistor $\boldsymbol{R_{nn}}$ is given by:

$$P = R_{nn} {I_n}^2$$

Since $\boldsymbol{I_n}$,= 1 Amp, the equations above imply that the self resistance is given by:

$$R_{nn} = \int_V \frac{1}{\sigma} \left(\overrightarrow{J_n} \left(\overrightarrow{\mathbf{x}'} \right) \cdot \overrightarrow{J_n} \left(\overrightarrow{\mathbf{x}'} \right) \right) dV$$

If multiple conduction paths exist, then the mutual resistance $\boldsymbol{R_{mn}}$ between paths ***m*** and ***n*** can be calculated from the formula:

$$R_{mn} = \int_V \frac{1}{\sigma} \left(\overrightarrow{J_m} \left(\overrightarrow{\mathbf{x}'} \right) \cdot \overrightarrow{J_n} \left(\overrightarrow{\mathbf{x}'} \right) \right) dV$$

Formulation of the DC Inductance Solution

The DC inductance solver uses the volume current density solutions computed by the DC conduction solver as its starting point. Suppose that the n^{th} source has been excited. Given the DC current density $\overrightarrow{J_n}$, the resulting magnetic vector potential $\overrightarrow{A_n}$ can be calculated from the volume integral equation:

$$\overrightarrow{A_n}(\vec{x}) = \int_V \frac{\mu_0 \overrightarrow{J_n} \left(\overrightarrow{\mathbf{x}'} \right)}{4\pi ||\overrightarrow{\mathbf{x}} - \overrightarrow{\mathbf{x}'}||} dV$$

Here ***V*** denotes the volume of the conducting objects. The calculation of the vector potential $\vec{A}$ can be time-consuming if direct integration from every source $\overrightarrow{\mathbf{x}'}$ to every observation point $\vec{x}$ is used. To make the calculation efficient, Q3D uses the fast multipole method.

If magnetic materials are present, Q3D uses Adaptive Cross Approximation (ACA) with the help of the fast multipole method.

Extracting DC Inductance from the Field Solution

W_M

$$W_M = \frac{1}{2}\int_V \overrightarrow{A_n}(\vec{x}) \cdot \overrightarrow{J_n}(\vec{x}) dV$$

L_{nn}

$$W_M = \frac{1}{2} L_{nn} {I_n}^2$$

Because the applied current I_n = 1 Amp, the equations above imply that the self inductance of source n is:

$$L_{nn} = \int_V \overrightarrow{A_n}(\vec{x}) \cdot \overrightarrow{J_n}(\vec{x}) dV$$

If there are multiple sources in the problem, then the above procedure must be modified slightly.

To compute the mutual inductance L_{mn} between two different current paths ***m*** and ***n***, the formula is:

$$L_{mn} = \int_V \overrightarrow{A_m}(\vec{x}) \cdot \overrightarrow{J_n}(\vec{x}) dV$$

AC Resistance and Inductance Solution

The AC resistance/inductance calculation in Q3D is derived using the assumption that the working frequency is sufficiently high enough that the skin effect and current crowding effects are well developed. This is tantamount to assuming the inductive reactance dominates the resistance, and correspondingly, the surface currents are primarily determined by inductance.

To calculate the AC resistance and inductance matrices, Q3D Extractor does the following:

- Divides the surfaces of the conductors into a finite element mesh.
- Applies one volt to one of the source terminals while short-circuiting the remaining terminals.

- Solves the boundary element equation using the high frequency "lossless" assumption that tangential component of the E-field vanishes.
- Repeat the previous two steps for each source to generate the high frequency or AC inductance matrix.
- Use the "lossless" currents in conjunction with a skin effect-based surface resistance matrix to determine the perturbation of the inductance matrix caused by the resistance.

Formulation of the AC Inductance Solution

The AC inductance solver seeks a surface current distribution such that:

- The tangential component electric field vanishes on the conductors.
- The total current entering the conductors from the source terminal(s) equals the total current exiting the conductors from the sink(s).

The first condition is enforced using the quasi-static version of the electric field integral equation:

$$\vec{E}_{\tan}(\vec{r}) = j\omega\mu_0 \iint \vec{J}\left(\overrightarrow{r'}\right) \frac{1}{4\pi||\overrightarrow{r} - \overrightarrow{r'}||} \overrightarrow{dr'}$$

This equation relates the incident tangential electric field to the surface current density. This integral equation is solved using divergence conforming basis functions. To prevent charge build up at the terminals, and also to enforce the second condition, these basis functions are then projected into a solenoidal subspace.

$$\dot{j} \times \dot{r}$$

Since total tangential electric field is enforced to be zero, the resulting impedance matrix is lossless (pure inductive.) Therefore, the imaginary part of Z determines the inductance matrix L where:

$$\omega L = \mathrm{Im}\{Z\}$$

To get the resistance matrix, the power losses due to the surface current density are computed using the following integral over the metal surfaces:

$$R_{mn} = \int_S Z_s \overrightarrow{J_m}(\vec{r}) \overrightarrow{J_n}(\vec{r}) dS$$

Here, Z_s is the surface impedance of the metal:

$$Z_s = \mathrm{Re}\left\{\sqrt{\frac{j\omega\mu}{\sigma}}\right\}$$

Extracting AC Resistance from the Field Solution

The AC resistance can be computed by finding the power loss on the conductor surface. Suppose that the n^{th} source terminal is excited with 1 Amp, producing a surface current distribution $\overrightarrow{K_n}$.

The resulting AC self resistance is given by the integral:

$$R_{nn} = \mathrm{Re}\left[\int_S Z_s(\omega)\left(\overrightarrow{K_n}\left(\vec{\mathbf{x}}'\right) * \cdot \overrightarrow{K_n}\left(\vec{\mathbf{x}}'\right)\right) dS\right]$$

Z_s

$$Z_s(\omega) = \sqrt{\frac{j\omega\mu_0}{\sigma}}$$

If multiple conduction paths exist, then the AC mutual resistance R_{mn} between paths ***m*** and ***n*** can be calculated from the formula:

$$R_{mn} = \mathrm{Re}\left[\int_S Z_s\left(\overrightarrow{K_m}\left(\vec{\mathbf{x}}'\right) * \cdot \overrightarrow{K_n}\left(\vec{\mathbf{x}}'\right)\right) dS\right]$$

Extracting AC Inductance from the Field Solution

W_M

$$L_{nn} = \int_S \overrightarrow{A_n}(\vec{x}) \cdot \overrightarrow{K_n}(\vec{x}) dS$$

If there are multiple sources in the problem, then the above procedure must be modified slightly.

To compute the mutual inductance L_{mn} between two different current paths ***m*** and ***n***, the formula is:

$$L_{mn} = \int_S \overrightarrow{A_m}(\vec{x}) \cdot \overrightarrow{K_n}(\vec{x}) dS$$

Solving for Fields

Q3D invokes various different field solvers depending upon which circuit quantities have been requested.

Circuit Parameter	Solver(s) Used	Sources	Field Computed	Derived Field Quantities
Capacitance/Conductance	capacitance extractor (caxtr)	Charge	Q	**E**, **D**
DC Resistance	conduction (cnq3d)	DC Current	V	**J, E**
DC Inductance	IE (ens_solver) DC conduction (cnq3d)	DC Current	A V	**H, B, E** **J**
AC Resistance	IE (ens_solver) AC conduction	AC Surface Current	K_s V	**J_s, H**
AC Inductance	IE (ens_solver) AC conduction	AC Surface Current	A H K_s	**B**

H Fields

In order to visualize H fields in Ensight after AC/RL extraction, H fields must be computed at all required observed points in the space or model surface based on the solution of surface current density. Q3D Extractor uses radiation integrals, which in quasistatic case collapses to:

$$\vec{H}(x,y,z) = \int_S \frac{\vec{J} \times \left(\vec{r} - \vec{r}'\right)}{4\pi \cdot \left|\vec{r} - \vec{r}'\right|^3} dS \approx \sum_{i=1}^{N} \frac{\vec{I}_i \times}{4\pi} \int_{S_i} \frac{\vec{r} - \vec{r}'}{\left|\vec{r} - \vec{r}'\right|^3} dS$$

Where N is the total number of elements. Q3D Extractor also assumes that current density on each element is constant.

23 - 2D Extractor Technical Notes

At low clock rates, voltage pulses that propagate along cables, through connectors, and along circuit board traces can be treated as logical signals that are transmitted with no distortion and no delays. However, as clock rates increase, these interconnects can no longer be treated as ideal channels. Instead, capacitance and inductance introduce time delays and distortion that can adversely affect the signal.

2D Extractor computes RLGC circuit parameter matrices for any arbitrary multi-conductor transmission line. These circuit parameters depend only upon the geometry of the structure and the characteristics of the materials that make up the structure. Once computed, these circuit parameters can then be transformed into a SPICE subcircuit that can be incorporated within a larger circuit or schematic, allowing you to analyze its impact on signal integrity.

This section outlines the technical aspects of the general procedure used by 2D Extractor to generate an equivalent subcircuit for a transmission line. Although much of this procedure is transparent, a general knowledge of the underlying theory can help you operate 2D Extractor more efficiently.

Basic Assumptions

The following basic assumptions are made in 2D Extractor:

- You are only trying to model quasi-TEM wave effects (structures with two or more conductors).
- The transmission line has a uniform cross-section along its length, and can therefore be represented by a single cross section. Discontinuities associated with bends, junctions, and vias are not taken into account by the software. To model the effect of such discontinuities, use Q3D Extractor (for low to middle frequencies) or HFSS (for high frequencies), which allow you to draw and analyze a full three-dimensional model of a structure.
- All electric and magnetic fields lie in the XY cross-section being modeled. No component of the E-field or B-field lies in the direction of propagation, except possibly a small electric field component due to conductor losses.

Required Input

To generate a solution, 2D Extractor requires you to:

- Draw the cross-section of the structure being modeled.
- Identify which objects are signal lines, ground lines, and floating conductors.
- Specify object materials.
- Specify boundary conditions.

Output

2D Extractor provides (as requested):

- RLGC parameter matrices for a lossless transmission line.
- A SPICE equivalent subcircuit of the structure.
- Forward and backward crosstalk coefficients.
- Characteristic impedance matrices.
- Propagation velocity, delay and attenuation.

Additional Technical Notes

Balloon Boundary

Matrix Generation for 2D Extractor

Phasor Notation

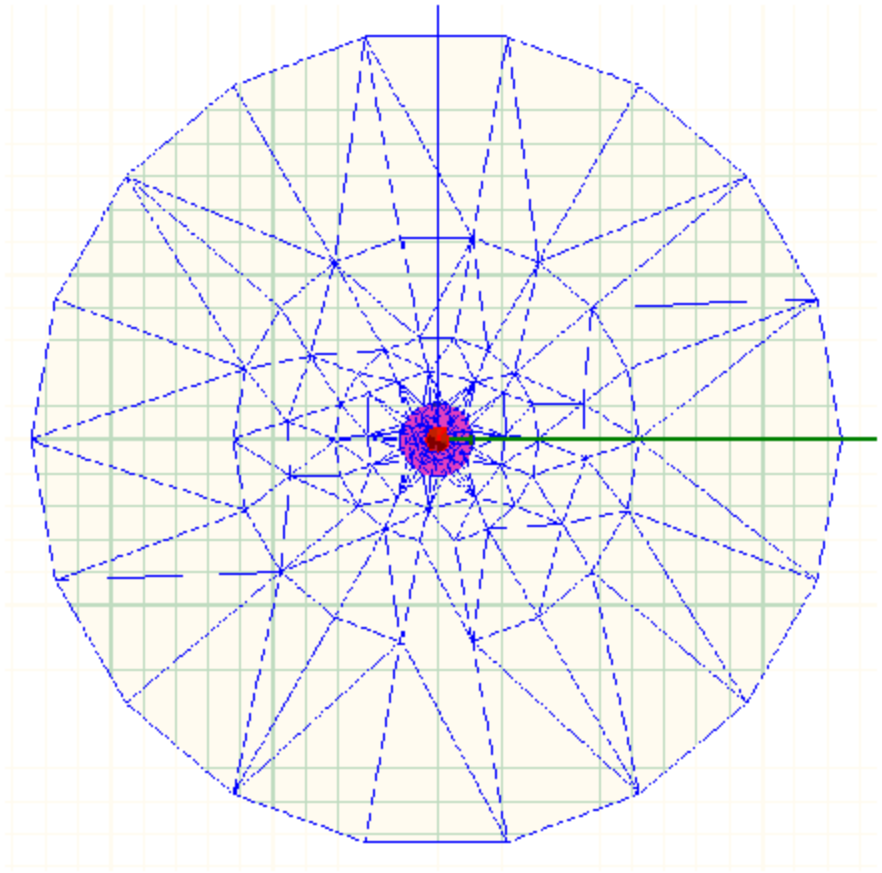
Solution Process

Solving for Fields in 2D Extractor

Transmission Line Circuit Models

Balloon Boundary

Inside 2D Extractor, three layers of increasingly large triangles surrounding the problem region are created, as shown below:



On the outer surface of this "balloon" region, a zero-total-charge boundary condition is imposed for CG solutions. That is, the integral (over the outer surface) of the normal component of the electric field is required to be zero. This integral is expressed in terms of the finite element DOFs and imposed as an additional constraint equation on the solution.

This prevents any net field lines from escaping from the drawn region to the outer boundary. Thus the outer boundary will not act like an additional ground.

For RL solutions, 2D Extractor imposes the condition Az = 0 (where Az is the z-component of the magnetic vector potential). This ensures there is no net current on the boundary, so it will not act as an additional current return path.

Matrix Generation for 2D Extractor

For a transmission line, 2D Extractor gives you the option of extracting any combination of four parameters: capacitance (C), inductance (L), resistance (R), and conductance (G).

Keep in mind that:

- For a lossless transmission line, both capacitance and inductance matrices are required to generate a SPICE equivalent circuit.
- For a lossy transmission line, one of the following combinations is required to generate a SPICE equivalent circuit:
  - Resistance, capacitance, and inductance
  - Conductance, capacitance, and inductance

To compute all elements in a matrix for an ***n*** conductor transmission line, 2D Extractor performs a sequence of field simulations, then computes the energy stored in the simulated field. In each field simulation, 1 volt (for capacitance and conductance matrices) or 1 amp (for inductance and resistance matrices) is applied to a single conductor and 0 volts or 0 amps is applied to all other conductors. Therefore, for an ***n*** -conductor system, ***n*** field simulations are performed.

The CG matrices generated from field simulator outputs are in a Maxwell matrix format. Since a standard SPICE component only has two terminals, the software derives a SPICE matrix format from the Maxwell capacitance and conductance elements. This is only necessary for the capacitance and conductance matrices, because both formats yield the same results for the inductance and resistance matrices.

Capacitance Matrices

At the simplest level, capacitance is proportional to the amount of energy stored in the electric field in and around a structure. In a circuit, this is the energy stored in the electric field resulting from the voltage differential across a dielectric:

$$U_e = \frac{1}{2}Cv^2$$

where:

- U_e is the energy stored in the electric field
- C is the capacitance
- v is the voltage across the dielectric

Q3D Extractor uses AC conduction to compute the capacitance between two lines by simulating the electric field that arises when a voltage differential is applied and then computing the energy stored in the field. The solver uses the following equation to solve for capacitance in terms of U_e:

$$C = \frac{2U_e}{v^2}$$

To compute the elements in the capacitance matrix for an *n* conductor transmission line, the Q3D Extractor performs a sequence of field simulations. In each field simulation, 1 volt is applied to a single conductor and 0 volts are applied to all other conductors. The energy stored in the electric field is given by the following relationship:

$$U_{ij} = \frac{1}{2}\int_{\Omega} D_i \cdot E_j d\Omega$$

where:

- U_{ij} is the energy stored in the electric field linking conductor *i* with conductor *j*.
- D_i is the electric flux density produced when 1 volt is applied to conductor *i* and 0 volts are applied to all other conductors.
- E_j is the electric field produced when 1 volt is applied to conductor *j* and 0 volts are applied to all other conductors.

The capacitance coupling *i* and *j* is therefore:

$$C_{ij} = \frac{2U_{ij}}{v^2} = \int_{\Omega} D_i \cdot E_j d\Omega$$

Since a standard SPICE component only has two terminals, the software derives a SPICE matrix format from the Maxwell matrix capacitance elements.

Capacitance in Terms of Static Charge

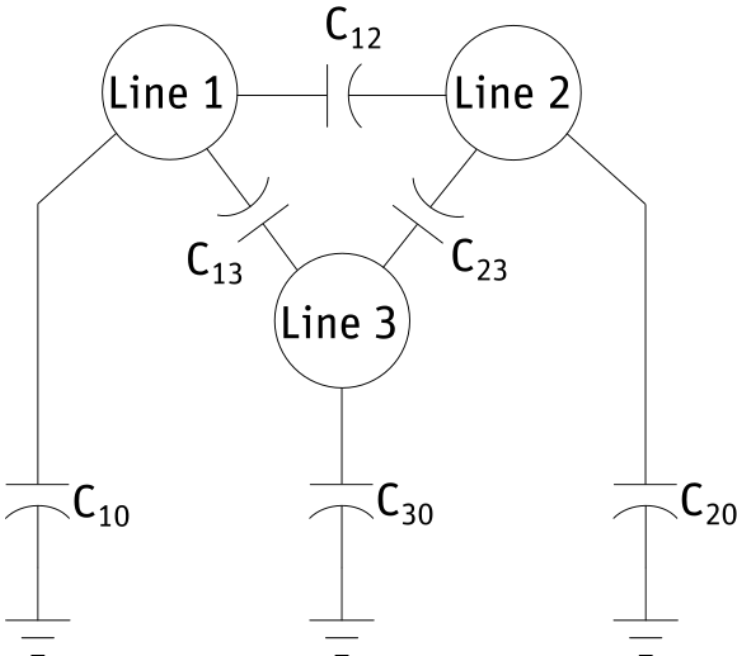
The capacitance matrix represents the charge coupling within a group of conductors—that is, the relationship between charges and voltages for the conductors.

For example, given the three conductors in the following figure, with the **Background** object used as a grounded reference, the net charge on each object is:

$$Q_1 = C_{10}V_1 + C_{12}(V_1 - V_2) + C_{13}(V_1 - V_3)$$

$$Q_2 = C_{20}V_2 + C_{12}(V_2 - V_1) + C_{23}(V_2 - V_3)$$

$$Q_3 = C_{30}V_3 + C_{13}(V_3 - V_1) + C_{23}(V_3 - V_2)$$



This results in the following matrix equation, which gives the relationship between *Q* and *V* for the three conductors:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

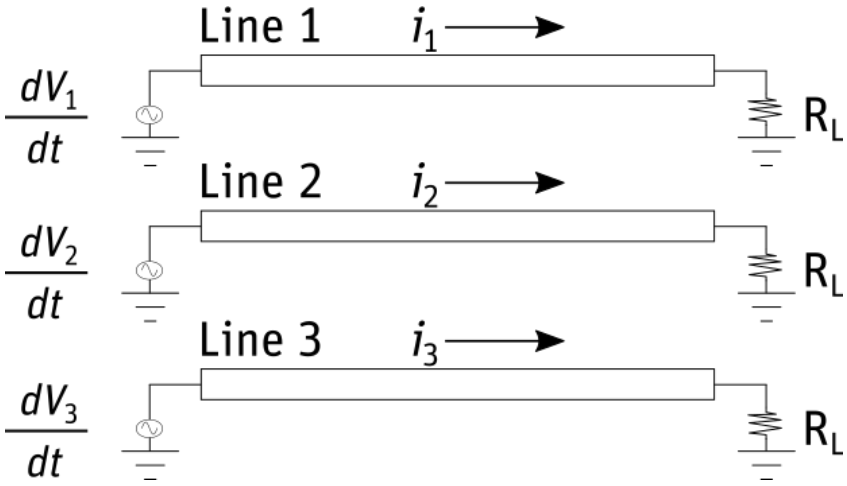
In a device with *n* conductors, this relationship would be expressed by an *n* x *n* capacitance matrix (Capacitance matrix values are in farads).

Capacitance in Terms of Time-Varying Voltage

A capacitance matrix also represents the relationship between currents and time-varying voltages in a system of conductors.

Given the three transmission lines shown below, the current changes 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:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = [C] \begin{bmatrix} \frac{dV_1}{dt} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C_{10} + C_{12} + C_{13} \\ -C_{12} \\ -C_{13} \end{bmatrix} \frac{dV_1}{dt}$$

This capacitance matrix gives the currents that are induced on lines 2 and 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 (Capacitance matrix values are in farads).

Maxwell Matrices

The matrices generated by the software from the field simulators are in Maxwell matrix format. Since a standard SPICE component only has two terminals, the software derives a SPICE matrix format from the Maxwell matrix capacitance and conductance elements. You can then use the elements of the SPICE matrix directly in a circuit design (as an exported subcircuit), or to create a SPICE deck. This is only necessary for the capacitance and conductance matrices, because both formats yield the same results for the inductance and resistance matrices. Thus, using the definitions:

$$U_e = \frac{1}{2} C v^2$$

and

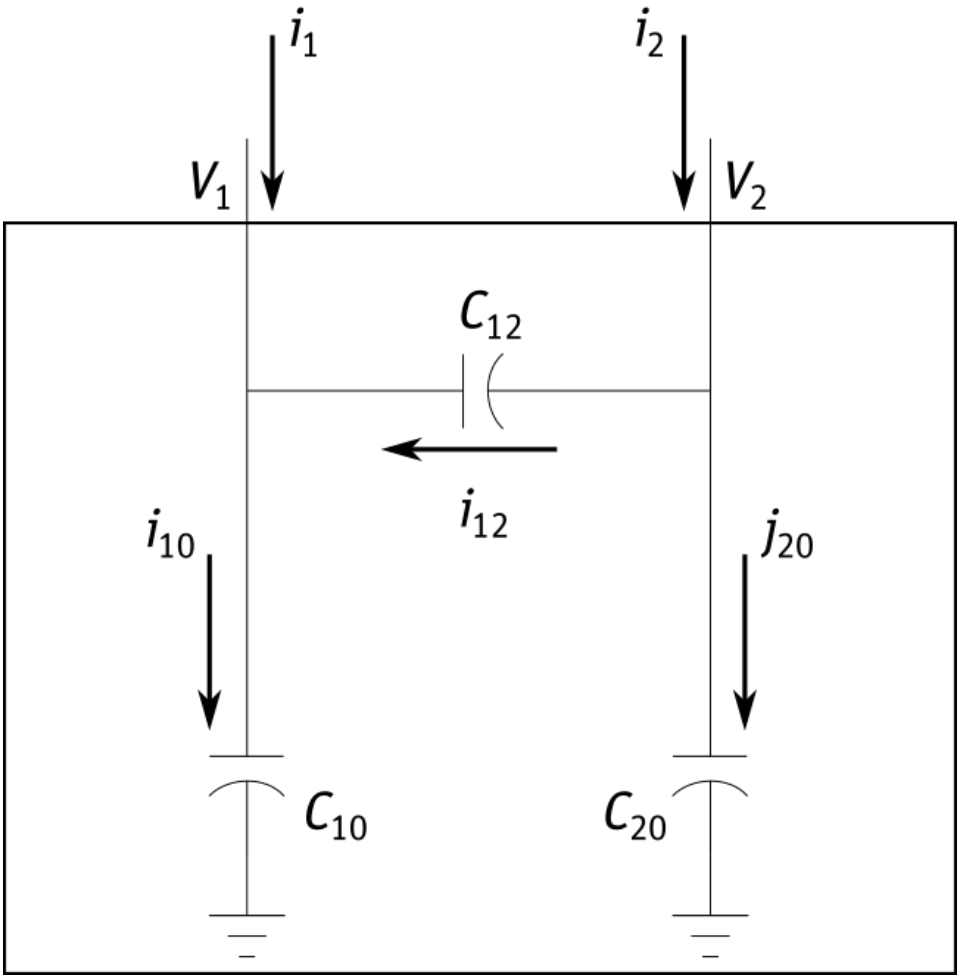
$$Q = \frac{2U_e}{v}$$

The Maxwell capacitance matrix equation, calculated from the energy (U_e) of the field solution for two conductors, is:

$$\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

The self-capacitances (the diagonal matrix elements) are greater than zero, and the mutual capacitances (the off-diagonal matrix elements) are less than or equal to zero. This is because physically, placing a positive voltage on a conductor while leaving the other conductors grounded causes positive charge to accumulate on the excited conductor, while a corresponding negative charge develops on surrounding conductors.

In a SPICE matrix, the elements are the values of the ideal two-terminal components that must be connected between nodes to create the equivalent of the Maxwell matrix. The SPICE circuit equivalent for previous two-conductor matrix appears as:



where:

- C_{10} (of the SPICE matrix) = C_{11} + C_{12} (of the Maxwell matrix)
- C_{20} (of the SPICE matrix) = C_{22} + C_{12} (of the Maxwell matrix)

Thus, the SPICE matrix equation has the values:

$$\begin{matrix} C_{10} & -C_{12} \\ -C_{12} & C_{20} \end{matrix}$$

This matrix is symmetric. Its diagonal entries are greater than or equal to zero (they may be exactly zero if the conductors are shielded from each other), and its off-diagonal entries are also greater than or equal to zero. This is because the SPICE matrix represents actual circuit values, rather than one where each conductor is included separately.

Modeling conductance to generate the conductance matrix is similar, except that charge is replaced by current.

Matrix Reduction Operations in Q3D Extractor and 2D Extractor

Matrix reduction operations provide a convenient way to explore different types of connections and grounding configurations without additional runs of the field solver.

This section of technical notes explains the meanings of various matrix reduction operations available in Q3D Extractor and 2D Extractor, as well as the assumptions behind them. It is divided into the following topics:

- **Introduction** – explains basic concepts from electromagnetics and circuit theory that are essential for understanding this section.
- **Grounding and Floating Conductors** – describes the effects of grounding and floating conductors on the capacitance and inductance matrices.
- **Connecting Elements** – describes the effects of connecting elements in parallel or in series on the capacitance and inductance matrices.
- **Joining Elements in Parallel** – describes the effects of joining elements in parallel on the inductance matrix. The Join Selected Terminals operation is also covered.
- **Floating at Infinity** – describes the Float at Infinity operation and its effect on the capacitance matrix.
- **Return Path Reduction** – describes the Return Path Reduction operation and its effect on the inductance matrix.
- **Redefining the Ground Reference** – describes redefining the ground reference and the effect of doing so on the capacitance matrix.
- **Operations for Multiterminal Conductors** – describes operations for multiterminal conductors (such as moving source and sink points or adding sink points) and their effect on the inductance and capacitance matrices.
- **Set Ground vs Add Ground in 2D Extractor** – describes the two different grounding options available in 2D Extractor and the effects of each.

- **Differential Pair Matrix Reduction in 2D Extractor** – describes the differential pair matrix reduction operation and its effects.

Circuit-oriented approaches will be used to explain the concepts involved, and the examples will distinguish between the cases of inductance and capacitance matrices because many (if not all) matrix reduction operations have very different effects on the two types of matrices.

Phasor Notation

Time varying quantities that have the form $F(t) = F_m cos(\omega t + \theta)$ can be represented as rotating phasors in the complex plane.

Using Euler 's formula:

$e^{j\alpha} = cos\alpha + jsin\alpha$

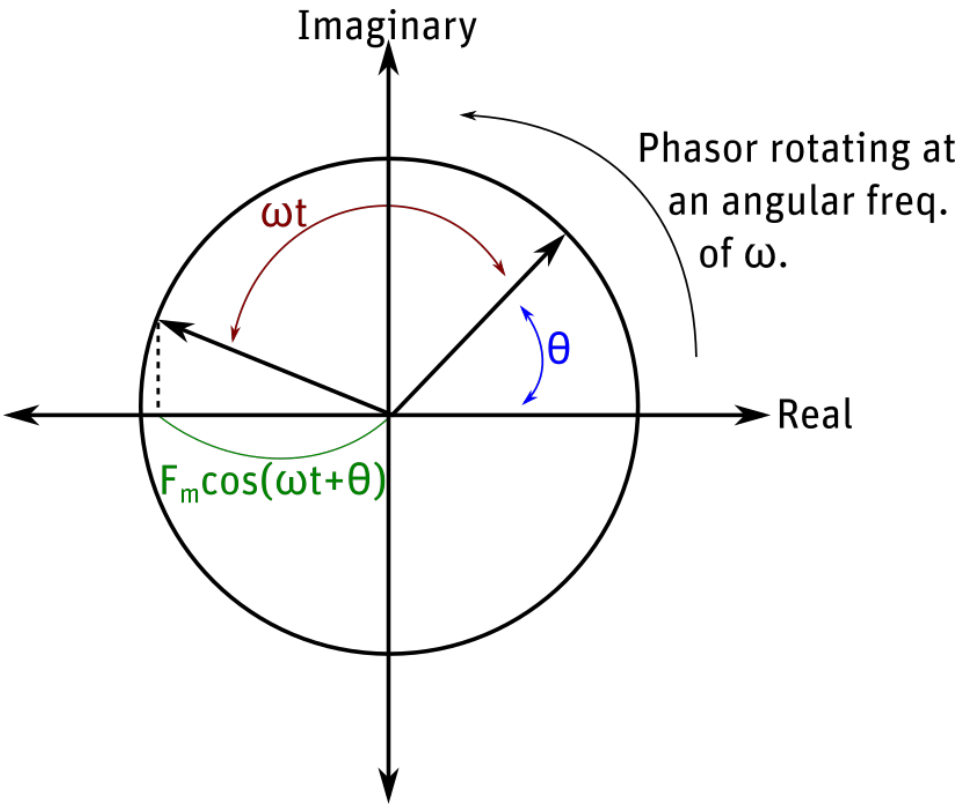
Setting $\omega t + \theta = \alpha$, $F(t)$ equals the real portion of $e^{j(\omega t + \theta)}$:

$F(t) = F_m cos(\omega t + \theta)$

$= \Re [F_m(cos(\omega t + \theta) + jsin(\omega t + \theta))]$

$= \Re [F_m e^{j(\omega t + \theta)}]$

Each time-varying quantity now has the form $F_m e^{j\omega t} e^{j\theta}$. The $F_m e^{j\theta}$ component is a complex constant that can be represented by a stationary phasor in the complex plane. The $(F_m e^{j\theta}) e^{j\omega t}$ component is a complex number that depends on t, and can be represented as a rotating phasor in the complex plane, as shown in the following figure:



The phasor 's projection on the real axis oscillates sinusoidally. It reaches a peak when parallel with the real axis, and is zero when parallel with the imaginary axis. Therefore, a phasor with **θ = 90°** represents a quantity that peaks 90 degrees after a phasor with **θ = 0°**.

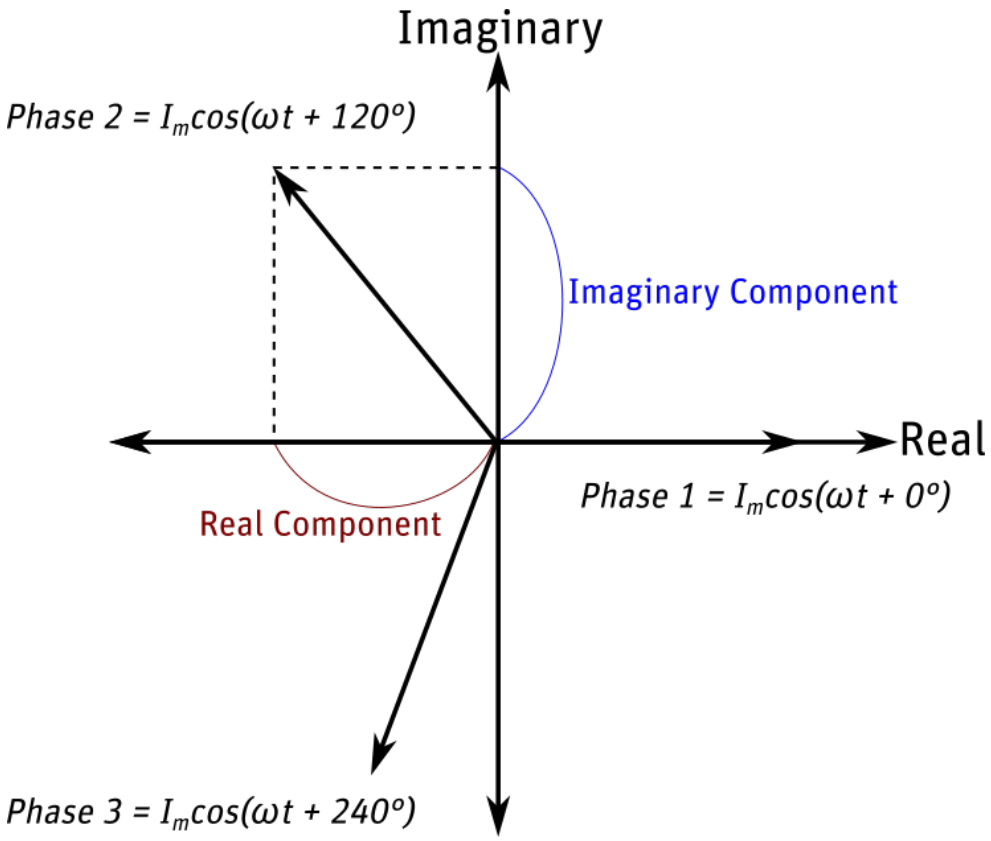
You need to enter magnitudes and phase angles when you specify voltages, currents and other boundary or source quantities in models where time-varying fields are computed (that is, eddy current and AC conduction models).

Magnitude and phase angle are more readily handled in calculations when they are expressed in rectangular form (using the real and imaginary form ***x + jy***) than in polar form (using magnitude and phase angle). Therefore, to specify currents and boundary conditions as functions you must describe them in terms of real and imaginary components.

The ***x + jy*** (rectangular) description of a phasor — $[F_m(\cos(\omega t + \theta) + j\sin(\omega t + \theta))]$— indicates that the phasor is the sum of two components: a sinusoid that peaks at **ωt = 0°**and a sinusoid that peaks at **ωt = 90°**.

- The ***x*** component of the phasor is the real component and can be 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 the following three-phase system are related to the magnitude and phase of a sinusoid in this manner:



Solution Process

As Q3D Extractor solves for the matrices you have requested, it goes through the following steps:

1. Uses the appropriate field simulators to generate a field solution. Different field simulators (field solvers) are used to compute the various matrices. For example, the electrostatic field simulator is used to simulate the electric fields from which capacitances are computed. Similarly, the magnetostatic field simulator is used to simulate the magnetic fields from which inductances are computed.

> **Note:**
>
> Because circuit parameters are computed using the final mesh generated during the field solution, adequate refinement of the mesh during the field solution increases the accuracy of circuit parameter solutions.

2. After the field solutions are complete, the simulator performs an error analysis in each triangle in the mesh. The triangles with the highest energy error are broken down into smaller triangles — producing a more accurate solution in these areas.
3. Individually applies one volt (for CG parameters) or one amp (for RL parameter problems) to each conductor you have defined — objects made of conducting materials may or may not be defined conductors for your problem. The rest of the conductors are temporarily set to zero (volts or amps).

- As each conductor is excited, the field solution is used to compute a capacitance, inductance, or resistance for that conductor relative to the reference ground and to the other conductors.
- The resulting parameter is entered in the matrix (the desired end result). For instance, in the capacitance matrix, the capacitance of the first conductor relative to ground is element C_{10}. The diagonal elements of the matrix would represent self-capacitances, and the off-diagonal elements would represent mutual capacitances (mutual elements).

The resulting matrix is in Maxwell matrix format. These cannot be used to generate SPICE equivalent subcircuits in capacitance and conductance problems, so a SPICE matrix is generated.

4. If both capacitance and inductance matrices are requested, the Q3D Extractor automatically generates a modal solution of voltage and current modes once both matrices have been computed. The software uses per unit length capacitances and inductances to compute:

- The modal transformation matrix that converts voltage signals into voltage modes and current signals into current modes. This matrix is used to generate a distributed SPICE equivalent subcircuit that models voltage and current signals carried on the transmission line as decoupled modal lines. This provides a more accurate representation of the line than modeling it using lumped circuits.
- The velocity of each mode.
- The characteristic resistance matrix (Z0) of the physical transmission line structure.

Solving for Fields in 2D Extractor

A different field simulator is used to compute each circuit parameter. The software selects the solvers that are needed to solve for the parameters that you have requested.

Circuit Parameter	Field Solver	Sources	Field Computed	Derived Field Quantities
Capacitance, Conductance	AC Conduction	AC Voltage, Charge	Φ	**E**, **D**
Inductance, Resistance	Eddy Current	AC Current	$\mathbf{A}_Z$ or $\mathbf{A}_\Phi$	**H**, **B, J**

AC Conduction Solver

To generate an conductance matrix associated with a lossy transmission line, the software uses the AC conduction solver to compute the energy stored in the electric fields and energy losses in dielectrics (the electric scalar potential), Φ, at all points in the problem region. It then derives the E-field, D-field, and the current density ***j*** from Φ.

The solver uses the following equation to calculate the fields:

$$\nabla \bullet (\sigma + j\omega\varepsilon)\nabla\Phi = 0$$

where:

- ω is the angular frequency at which the potential is oscillating.
- σ is conductivity.
- ε is permittivity.
- $-\nabla\Phi$ is the electric field, ***E***.

The equation above is derived from the current continuity equation:

$$\nabla \bullet (J) = 0$$

where ***J*** is the total current density given by:

$$J = (\sigma + j\omega\varepsilon)E$$

Assumptions

The AC Conduction field solver makes the following assumptions about the field quantities it solves for:

- All time-varying electromagnetic quantities have the periodic waveform:

  $$F(t) = F_m \mathrm{Cos}(\omega t + \theta)$$

- 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 **E**, due to time-varying magnetic fields caused by conduction currents, can be neglected.

Eddy Current Solver

To generate the resistance matrix associated with a lossy transmission line, the software uses the eddy current field simulator to compute the energy stored in the magnetic fields and energy losses in the line (the electric scalar potential), f, at all points in the problem region. From f it first derives the magnetic vector potential, Az, and then derives the B-field, H-field, and the current density J from Az.

> **Note:**
>
> - Use the AC conduction solver to compute conduction currents resulting from time-varying electric fields in lossy dielectrics.
> - Eddy solver uses magnetic loss tangent and ignores the dielectric loss tangent.
> - In Q3D and 2D Extractor, displacement current is assumed to be negligible during Eddy current computations, and is therefore neglected in the RL solution.

The following two equations are solved by the eddy current field simulator using the finite element method:

$$\nabla \times \frac{1}{\mu} (\nabla \times A) = (\sigma + j\omega\varepsilon) (-j\omega A - \nabla\Phi)$$

and

$$I_T = \int_{\Omega} \sigma(-j\omega A - \nabla\Phi) d\Omega$$

where:

- $\boldsymbol{A}$ is the magnetic vector potential.
- Φ is the electric scalar potential.
- μ is the magnetic permeability.
- ω is the angular frequency.
- σ is the conductivity.
- I_T is the total current flowing in conductors.
- Ω is the area of the conductor cross-section.

These are derived from Maxwell 's equations, and use the total current you specify in conductors connected to an external source. Phasor notation is used to represent complex quantities.

The eddy current solver makes the following assumptions about the field quantities for which it solves:

- All quantities must have the same value of , but can have different phase angles and amplitudes (Fm). If a current is not a pure sinusoid, it is decomposed into sinusoidal harmonics, and solved separately at each frequency.
- The time-varying electromagnetic quantities B and D are assumed to have the periodic waveform F(t) = Fm cos(t + q) .
- All currents (source and eddy) are assumed to flow in the z-direction. Therefore, the magnetic fields associated with these currents lie within the xy-plane. As a result, the magnetic vector potential, A, 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 f is constant over the cross-section of each conductor in the problem.

Magnetic and Electric Fields (First Equation)

To solve for the magnetic vector potential A_Z, the eddy current solver uses the definition:

$$\nabla \times \boldsymbol{A} = \boldsymbol{B}$$

Substituting this into the first of Maxwell 's equations, and neglecting the displacement current term,

$$\nabla \times \boldsymbol{H} = \boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t} = \boldsymbol{J}$$

The result is:

$$\nabla \times \frac{1}{\mu}(\nabla \times \boldsymbol{A}) = \boldsymbol{J}$$

$$\boldsymbol{J} = \sigma \boldsymbol{E}$$

$$\nabla \times \frac{1}{\mu}(\nabla \times \boldsymbol{A}) = \sigma \boldsymbol{E}$$

Substituting it into the second of Maxwell 's equations:

$$\nabla \times \boldsymbol{E} = \frac{\partial \boldsymbol{B}}{\partial t}$$

And using the definition of ***E***, we find that

$$\boldsymbol{E} = (-j\omega \boldsymbol{A} - \nabla \Phi)$$

So:

$$\nabla \times \frac{1}{\mu}(\nabla \times A) = (-j\omega A - \nabla\Phi)\sigma$$

Current and Current Density (Second Equation)

Notice that the first equation is given in the form of conductivity multiplied by the complex value of ***E***:

$$E = (-j\omega A - \nabla\Phi)$$

Hence, the total complex current density (J_{total}) is defined as:

J_{total} + J_{source} +J_{eddy}

where J_{source} = $-\sigma\nabla\Phi$

and J_{eddy} = $-\sigma j\omega A$

The $j\omega$ term in J_{eddy} indicates that eddy current becomes increasingly significant as frequency increases.

The integral of this expression over the cross-section of a conductor is must equal the total current that you specify in that conductor when you set up a problem.

The total current, I_{total}, is defined as:

I_{total} = I_{source} +I_{eddy}

$$= \int_{\Omega} J_{total}\, d\Omega$$

Where:

- I_{source} is the source current, and J_{source} is the source current density, due to the potential difference $-\sigma\nabla\Phi$ 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 eddy current in the conductor.
- I_{eddy} is the eddy current $-\int j\omega\sigma A\, d\Omega$, and J_{eddy} is the eddy current density $-j\omega\sigma A$, induced in the conductor by time-varying magnetic fields.

Since ***A*** only has a z-component in the eddy current solver, Φ is constant for each cross-section of a conductor. Therefore, the field simulator does not have to solve for Φ at every node.

Eddy Current and Skin Depth

Induced currents allow magnetic fields to penetrate conductors only to a certain depth, which is approximated by the formula:

$$\delta = \sqrt{\frac{2}{\omega \sigma \mu_0 \mu_r}}$$

Where:

- ω is the angular frequency, $2\pi f$, where *f* is the frequency at which source currents and voltages oscillate during the solution.
- σ is the conductor 's conductivity.
- μ is the conductor 's relative permeability.
- μ_0 is the permeability of free space, $4\pi \times 10^{-7}$ henries.

Currents are concentrated near the surface of the conductor, decaying rapidly past the skin depth. As the formula above indicates, skin depth gets smaller as the frequency increases.

2D Extractor 's eddy current solver can model the development of the skin effects.

Transmission Line Circuit Models

When you export a transmission line equivalent subcircuit, the software uses the information from the RLC matrices to construct the subcircuit.

There are two basic types of circuit models:

- **Lumped Models** – consist of RLC and K elements. They approximate the Telegrapher's equation with a finite number of circuit elements.
- **Distributed Models** – such as the W elements, exactly represent the underlying Telegrapher's equations.

Lumped Circuit Models

To export a transmission line model that is *frequency independent* and lumped:

1. Follow the general procedure for exporting a circuit equivalent.
2. Select the adaptive solution.
3. Select **Capacitance** or **Conductance**.
4. Select **Inductance** or **Resistance**.
5. Choose **Lumped** to model the transmission line as a lumped SPICE equivalent. Increasing the number of cells allows you to accurately model faster rise times using lumped elements.

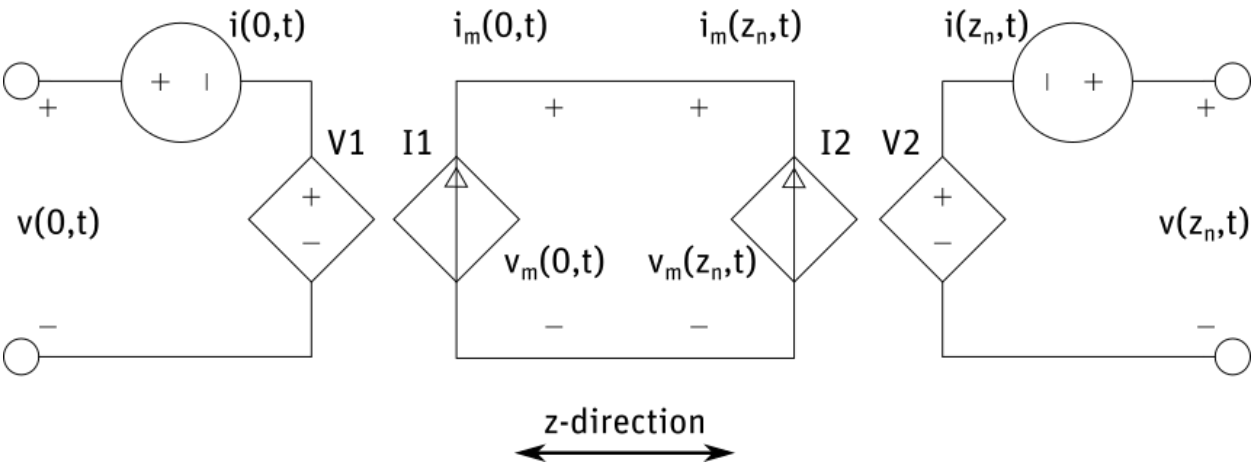
Distributed Circuit Models

To export a transmission line model that is frequency independent and distributed:

1. Follow the general procedure for exporting a circuit equivalent.
2. Select the adaptive solution.
3. Select **Capacitance**.
4. Select **Inductance**.

> **Note:**
>
> You cannot select **Resistance** or **Conductance** (the model must be lossless).

Q3D Extractor uses modal analysis to decouple the transmission line voltages and currents, and to create the distributed model. The following figure shows the network used for a lossy, distributed transmission line model:



The equivalent circuit transforms *n* voltage signals arriving at one end of the line into a set of *n* voltage patterns or modes. After a delay of *t* seconds, the modes are assumed to arrive at the other end of the line, and are transformed back into a collection of *n* individual signals.

Distributed Models for Transmission Lines

Distributed models for transmission lines (for example, the W element model in Nexxim and HSPICE) are based on **Telegrapher 's equations** without any approximations to the partial derivatives.

$$\frac{\partial V(z,t)}{\partial z} = -Ri\,(z,t) - L\left(\frac{\partial I(z,t)}{\partial t}\right)$$

where:

- $V(z,t) = [v_1, v_2, v_3, \ldots, v_n]$ are vectors of the signal voltages.
  - v_n is the voltage between the n^{th} conductor and ground.

- *t* is time, in seconds.
- *L* is the *nxn* matrix of self- and mutual- inductances between lines.
- *R* is the *nxn* matrix of losses resulting from series resistance.

and:

$$\frac{\partial I(z,t)}{\partial z} = -Gv(z,t) - C\left(\frac{\partial V(z,t)}{\partial t}\right)$$

where:

- $I(z,t) = [v_1, v_2, v_3, \ldots, v_n]$ are vectors of the signal voltages.
  - i_n is the current flowing in the n^{th} conductor
- *z* is the position along the length of the line.
- *C* is the nxn matrix of self- and mutual- capacitances between lines.
- *G* is the nxn matrix of losses resulting from conductance between lines.

The equations are used to derive frequency domain wave equations for lossy and lossless transmission lines. These equations and the wave equations are fully coupled. To make them easier to solve in the time domain, the software uses modal analysis to decouple the equations. This allows a single, coupled, n-conductor transmission line to be modeled using a set of n uncoupled single transmission lines.

Typically, ports are defined at the ends of the lines and the Telegrapher's equations are then solved analytically to compute the port responses.

If you solve for a frequency sweep, 2D Extractor can produce a frequency dependent distributed model. Otherwise, the distributed model is independent of frequency.

Frequency Independent Lumped Circuit Model

Lumped circuit models attempt to represent the behavior of a transmission line by approximating the Telegrapher's equations used to solve Distributed Models. A finite difference expression replaces the partial derivative with respect to the spatial coordinate. The resulting equations can be realized using lumped RLC circuit elements.

Lumped circuit models are an approximation to the Telegrapher's equations. In the frequency domain, the equations for a single line are:

$$\frac{\partial v}{\partial z} = -(R + sL)i(z) = 0$$

$$\frac{\partial i}{\partial z} = -(G + sC)v(z) = 0$$

The lumped approximation makes use of a finite difference formula to eliminate the partial derivative with respect to ***z***. We approximate the partial derivative of the current at ***z*** as:

$$\frac{\partial i}{\partial z} \cong \frac{i(z+\Delta z/2)-i(z-\Delta z/2)}{\Delta z}$$

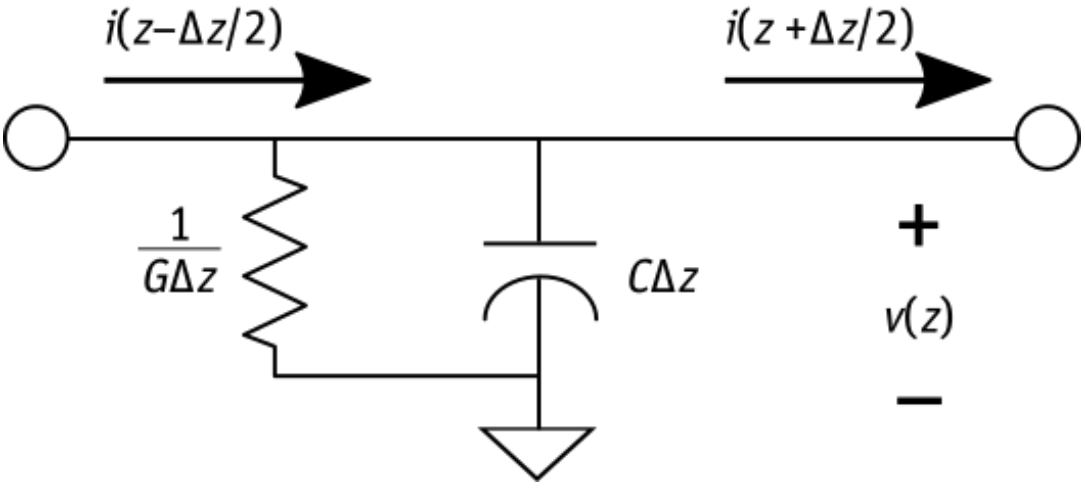
Δz

$$\frac{i(z+\Delta z/2)-i(z-\Delta z/2)}{\Delta z} = -(G+sC)v(z)$$

This can be rewritten as:

$$i(z+\Delta z/2) = i(z-\Delta z/2) - (G\Delta z + sC\Delta z)v(z)$$

Δz



$z + \Delta z/2$

$$\frac{\partial v}{\partial z} \cong \frac{v(z+\Delta z)-v(z)}{\Delta z}$$

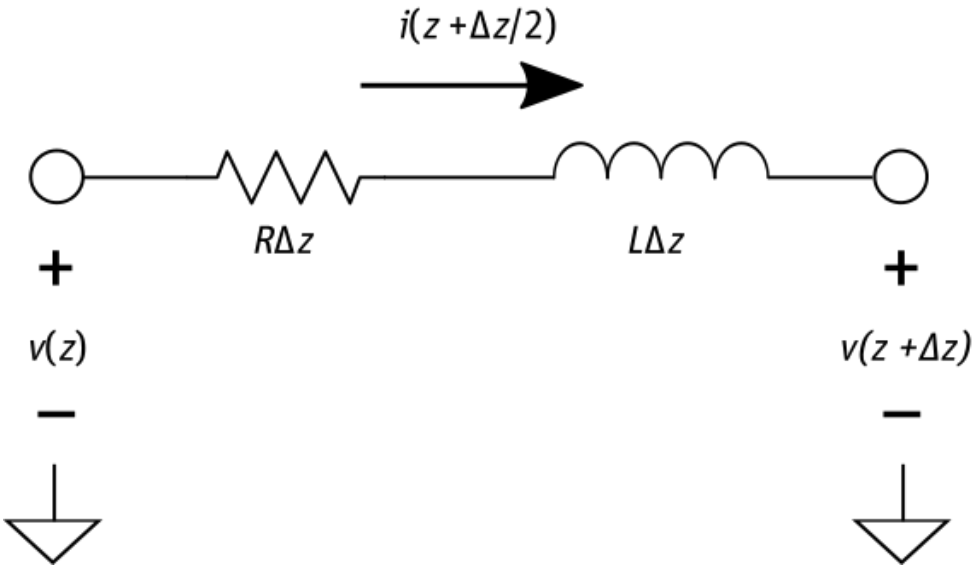
Inserting this into the first Telegrapher's equation yields:

$$\frac{v(z+\Delta z)-v(z)}{\Delta z} = -(R+sL)i(z+\Delta z/2)$$

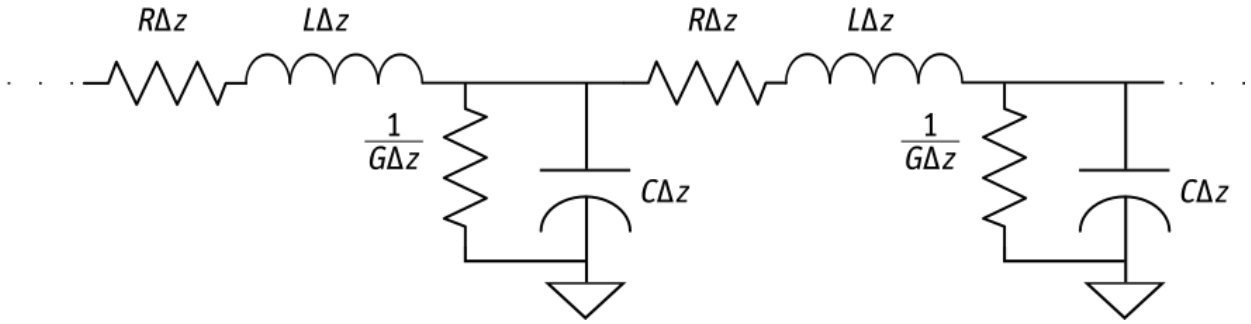
This is equivalent to the following Kirchhoff's Voltage Law (KVL) equation:

$$v(z+\Delta z) = v(z) - (R\Delta z + sL\Delta z)i(z+\Delta z/2)$$

This equation can be realized using the following lumped equivalent circuit:



The two circuit models can then be combined into an RLC "ladder" equivalent network to model the entire transmission line:



The ladder model above can be generalized to handle coupled systems of transmission lines. Each line will have its own ladder network, and in addition to the "self" RLC elements there will be coupling capacitances and mutual inductances between them.

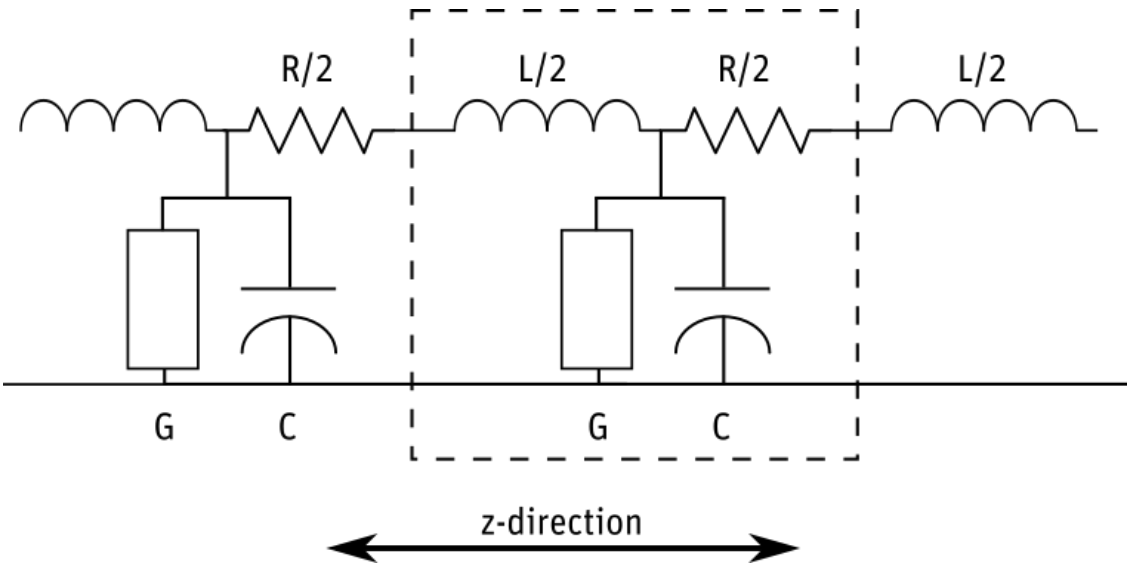
Δz

To export a circuit model that is frequency independent and lumped:

1. Follow the general procedure for exporting a circuit equivalent.
2. Select **Capacitance** and/or **Conductance**.
3. Select **Inductance** and/or **Resistance**.

4. Choose **Lumped** to model the transmission line as a lumped SPICE equivalent. Increasing the number of cells allows you to more accurately model faster rise times using additional lumped elements.

The following figure shows the ladder network for a frequency independent lumped model:



Lossy Transmission Lines

Lossy transmission lines are those where either:

- Some of the conductors are not perfect, or
- Some of the dielectric material (insulators) have either non-zero conductivity or non-zero loss tangent.

You can export either a lumped, frequency independent equivalent circuit, or a distributed transmission line model.

For a lossy transmission line, the following frequency domain wave equations are derived from the Telegrapher's equations:

$$\frac{d^2V(z)}{dz^2} = ZYV(z)$$

$$\frac{d^2I(z)}{dz^2} = YZV(z)$$

where:

- $Z = R(\omega) + j\omega L(\omega)$
- $Y = G(\omega) + j\omega C(\omega)$

Modal Analysis of Lossy Transmission Lines

A system of ***N*** lossy, coupled transmission lines is defined by its length ***l*** and by its set of per-unit-length circuit parameter matrices R,L,G, and C. These matrices are *N x N* and are typically frequency dependent. It is convenient to also define per-unit-length impedance ***Z(jω)*** and admittance matrices ***Y(jω)*** as follows:

$$Z(j\omega) = R(j\omega) + j\omega L(j\omega)$$

$$Y(j\omega) = G(j\omega) + j\omega C(j\omega)$$

The frequency-domain telegrapher equations for the transmission lines are then defined as:

$$\frac{\partial v}{\partial x} + Z(j\omega)i = 0$$

$$\frac{\partial i}{\partial x} + Y(j\omega)v = 0$$

Here ***v*** and ***i*** are *N x 1* vectors representing respectively the voltages and currents on the transmission lines. Both are functions of the frequency ***ω*** and the position variable ***x***, which ranges from *0* to *l*.

The telegrapher equations can be combined into a second-order differential equation by differentiating the first equation with respect to the position variable ***x*** and then using the second to eliminate the derivative of the current:

$$\frac{\partial^2 v}{\partial x^2} - ZYv = 0$$

This is a set of coupled differential equations. To make it possible to solve them, we can decouple the equations using the eigenvectors of the matrix ***ZY***.

Let the eigenvector decomposition be:

$$(ZY)M_v = M_v\Gamma^2$$

where M_V is a matrix whose columns represent the eigenvectors (voltage modes) and

$$\Gamma^2 = \mathbf{diag}\left(\gamma^2{}_\mathbf{n}\right)$$

is a diagonal matrix whose entries $\gamma^2{}_n$ are the eigenvalues of ***ZY***.

Define a vector of modal voltages $e = [e_1 \;\; e_2 \;\; \cdots \;\; e_N]^T$ related to the total voltages ***v*** by the transformation $v = M_v e$

Using this definition we can transform the coupled second-order differential equations into a system of uncoupled equations:

$$\frac{\partial^2 e}{\partial x^2} - \Gamma^2 e = 0$$

$$\frac{\partial^2 e_n}{\partial x^2} - \gamma^2{}_n e_n = 0,\ n = 1, 2, \ldots, N$$

These equations have simple analytic solutions:

$$e_n(x) = A_n \exp(\gamma_n x) + B_n \exp(-\gamma_n x),\ n = 1, 2, \ldots, N$$

Here the coefficients A_n and B_n are determined by the voltages at the ends of the transmission line. The first term corresponds to a backward-traveling modal wave, and the second corresponds to a forward-traveling wave.

The propagation constants γ_n are complex numbers with real part α_n and imaginary part β_n.

$$\gamma_n = \alpha_n + j\beta_n$$

It can be seen that α_n represents the attenuation per unit length that the mode experiences as it moves down the transmission line. The imaginary part β_n represents the change in phase per unit length.

A change in phase of 2π constitutes one wavelength; therefore the effective wavelength λ_n of mode n satisfies $\beta_n \alpha_n = 2\pi$

So:

$$\lambda_n = \frac{2\pi}{\beta_n}$$

The phase velocity c_n of the *n*-th mode satisfies $c_n = \lambda_n f$

So the velocity is:

$$c_n = \frac{2\pi f}{\beta_n} = \frac{\omega}{\beta_n}$$

ε_n

$$c_n = c_0 / \sqrt{\varepsilon_n}$$

Therefore we can define an effective permittivity for the n-th mode:

$$\varepsilon_n = \left(\frac{c_0}{c_n}\right)^2 = \left(\frac{c_0 \beta_n}{\omega}\right)^2$$

Finally we note that it is possible to define a characteristic impedance matrix $\boldsymbol{Z_0}$ for the system of transmission lines. This is a matrix that describes the ideal, reflection-free termination for the system of transmission lines. It is possible to express it as:

$$Z_0 = M_v \Gamma^{-1} M_v^{-1} Z$$

Lossless Transmission Lines

Some circuit simulators cannot model lossy transmission lines.

To produce circuit models for these simulators, the following time domain wave equations are derived from the telegrapher's equations:

$$\frac{\partial^2 V(z,t)}{\partial z^2} = LC \cdot \frac{\partial^2}{\partial t^2} V(z,t)$$

Here R and G terms have been ignored. To produce an equivalent circuit model from these equations, Q3D Extractor uses Modal Analysis.

Glossary

This section defines the terminology used in the Ansys desktop help topics. Terms are listed in alphabetical order.

Glossary: A

A/D

See Analog-to-Digital.

ACPR

Adjacent channel power ratio.

Active Substrate

A hybrid or multichip module substrate formed from a semiconductor. Termed active because components such as transistors can be fabricated directly into the substrate.

Active Trimming

The process of trimming components such as resistors while the circuit is under power. Such components are fabricated directly onto the substrate of a hybrid or multichip module, and the trimming is usually performed using a laser beam.

Active-High

A signal whose active state is considered to be a logic 1.

Active-Low

A signal whose active state is considered to be a logic 0.

Actuator

A transducer that converts an electronic signal into a physical equivalent. For example, a loudspeaker is an actuator which converts electronic signals into corresponding sounds.

Adaptive Hardware

Refers to devices which allow new design variations to be "compiles" in real-time, which may be thought of as dynamically creating subroutines in hardware (see also Virtual Hardware and Cache Logic).

Additive Process

A process in which conducting material is added to specific areas of a substrate. Groups of tracks, individual tracks, or portions of tracks can be built up to precise thicknesses by iterating the process multiple times with selective masking.

Address Bus

A unidirectional set of signals used by a computer to point to memory locations in which it is interested.

Analog

A continuous value that most closely resembles the real world and can be as precise as the measuring technique allows.

Analog Circuit

A collection of components used to generate or process analog signals.

Analog-to-Digital (A/D)

The process of converting an analog value into its digital equivalent.

Anisotropic Adhesive

Special adhesives which contain minute particles of conductive material. These adhesives find particular application with the flipped-chip techniques used to mount bare die on the substrates of hybrids, multichip modules, or circuit boards. The conducting particles are only brought in contact with each other at the sites where the raised pads on the die are pressed down over their corresponding pads on the substrate, thereby forming good electrical connections between the pads.

Anti-Fuse Technology

A programmable logic device technology in which conducting paths (anti-fuses) are grown by applying signals of relatively high voltage and current to the device's inputs.

Anti-Pad

The area of copper etched away around a via or a plated through-hole on a power or negative signal plane, thereby preventing an electrical connection being made to that plane.

Application-Specific Integrated Circuit (ASIC)

A device whose function is determined for a particular application or group of applications.

Application-Specific Standard Part (ASSP)

Refers to an integrated circuit created by a device manufacturer using ASIC technologies, and for these components to be sold as standard parts to anybody who wants to buy them.

ASIC

See Application-Specific Integrated Circuit.

ASIC Cell

A logic function in the cell library defined by the manufacturer of an application-specific integrated circuit.

Assertion-Level Logic

Special symbols which are used to more precisely indicate the function of gates with active-low inputs.

ASSP

See Application-Specific Standard Part.

Asynchronous

A signal whose data is acknowledged or acted upon immediately, irrespective of any clock signal.

Atto

Unit qualifier (symbol = a) representing one millionth of one millionth of one millionth, or 10-18. For example, 3aS stands for 3 x 10-18 seconds.

Attenuator

A passive device used to reduce signal strength while maintaining proper input and output impedance.

Glossary: B

Backplane

The medium used to interconnect a number of circuit boards. Typically refers to a special, heavy-duty printed or discrete wired circuit board.

Ball Grid Array (BGA)

A packaging technology similar to a pad grid array, in which a device's external connections are arranged as an array of conducting pads on the base of the package. However, in the case of a ball grid array, small balls of solder are attached to the conducting pads.

Bandpass

The frequency limits between half-power points of a signal or filter.

Bare Die

An unpackaged integrated circuit.

Base

Refers to the number of digits in a numbering system. For example, the decimal numbering system is said to be base-10. May also be referred to as the "radix".

Basic Cell

A pre-defined group of unconnected components that is replicated across the surface of a gate array.

BER

Bit error rate.

BGA

See Ball Grid Array.

BiCMOS

A technology in which the function of each logic gate is implemented using low-power CMOS, while the output stage is implemented using high-drive bipolar transistors

Binary Encoding

A form of state assignment for state machines that requires the minimum number of state variables.

BiNMOS

A relatively new low-voltage integrated circuit technology in which complex combinations of bipolar and NMOS transistors are used to form sophisticated output stages providing both high speed and low static power dissipation.

Bi-quinary

A system which utilizes two bases, base-2 and base-5, to represent decimal numbers. Each decimal digit is represented by the sum of two parts, one of which has the value of decimal zero or five, and the other the values of zero through four. The abacus is one practical example of the use of a bi-quinary system.

Bit

Abbreviation of binary digit.

Bipolar Junction Transistor (BJT)

A family of transistors.

Blackbox

Same as N-Port.

Blind Via

A via that is only visible from one side of the substrate.

Bobble

A small circle used on the inputs to a logic gate symbol to indicate an active-low input or control, or on the outputs to indicate a negation (inversion) or a complementary signal. Some engineers prefer to use the term bubble.

Bounce Pad

A special pattern etched onto the power or negative signal plane of a microwire circuit board to be used in conjunction with a laser beam which is employed to create blind vias. The laser beam evaporates the epoxy forming the outer layers of the board and continues down to the bounce pad which reflects, or bounces, it back up, thereby terminating the via.

Braze

To unite or fuse two pieces of metal by heating, or with a hard solder with a high melting point.

Bulk Storage

Refers to some form of media, typically magnetic, such as tape or a disk drive which can be used to store large quantities of information relatively inexpensively.

Buried Via

A via used to link conducting layers internal to a substrate. Such a via is not visible from either side of the substrate.

Bundle

A set of signals related in some way that makes it appropriate to group them together for ease of representation or manipulation. May contain both scalar and vector elements; for example, {a,b,c,d[5:0]}.

Bus

A set of signals performing a common function and carrying similar data. Typically represented using vector notation; for example, address[7:0].

Byte

A group of eight binary digits, or bits.

Glossary: C

Capacitance

A measure of the ability of two adjacent conductors separated by an insulator to hold a charge when a voltage differential is applied between them. Capacitance is measured in units of Farads.

CDMA

Code division multiple access.

Cell Library

The collective name for the set of logic functions defined by the manufacturer of an application-specific integrated circuit. The designer decides which types of cells should be realized and connected together to make the device perform its desired function.

Ceramic

An inorganic, nonmetallic material, such as alumina, beryllia, steatite, or forsterite, which is fired at a high temperature and is often used in electronics as a substrate or to create component packages.

CGA

See Column Grid Array.

Channel

(1)The area between two arrays of basic cells in a channeled gate array.

(2)The gap between the source and drain regions in a MOS transistor.

Channeled Gate Array

Application-specific integrated circuit organized as arrays of basic cells. The areas between the arrays are known as channels.

Channel-Less Gate Array

Application-specific integrated circuit organized as a single large array of basic cells. May also be referred to as a "sea of cells" or a "sea of gates" device.

Checksum

The final cyclic-redundancy-check value stored in a linear feedback shift register (or software equivalent). Also known as a "signature" in the guided-probe variant of functional test.

Chip

Popular name for an integrated circuit.

Chip-On-Board (COB)

A process in which unpackaged integrated circuits are physically and electrically attached to a circuit board, and are then encapsulated with a "glob" of protective material such as epoxy.

Chip-On-Chip (COC)

A process in which unpackaged integrated circuits are mounted on top of each other. Each die is very thin and it is possible to have over a hundred dies forming a 3D cube.

Chip-On-Flex (COF)

Similar to chip-on-board (COC), except that the unpackaged integrated circuits are attached to a flexible printed circuit.

Circuit Board

The generic name for a wide variety of interconnection techniques, which include rigid, flexible, and rigid-flex boards in single-sided, double-sided, multilayer, and discrete wired configurations.

CMOS

Logic gates constructed using both NMOS and PMOS transistors connected in a complementary manner.

Coaxial Cable

A conductor in the form of a central wire surrounded first by a dielectric (insulating) layer, and then by a conducting tube which serves to shield the central wire from external interference.

Coefficient of Thermal Expansion

Defines the amount a material expands and contracts due to changes in temperature. If materials with different coefficients of thermal expansion are bonded together, changes in temperature will cause shear forces at the interface between them.

Cofired Ceramic

A substrate formed from multiple layers of "green" ceramic that are bonded together and fired at the same time.

Column Grid Array (CGA)

A packaging technology similar to a pad grid array, in which a device's external connections are arranged as an array of conducting pads on the base of the package. However, in the case of a column grid array, small columns of solder are attached to the conducting pads.

Combinatorial

A digital function whose output value is directly related to the current combination of values on its inputs. Also known as combinational.

Compiled Cell Technology

A technique used to create portions of a standard cell application-specific integrated circuit. The masks used to create components and interconnections are directly generated from Boolean representations using a silicon compiler. May also be used to create data-path functions and memory functions.

Complementary Output

Refers to a function with two outputs carrying complementary logical values. One output is referred to as the true output and the other as the complementary output.

Complex Programmable Logic Device (CPLD)

A device that contains a number of PLA or PAL functions sharing a common programmable interconnection matrix.

Component

Components are items placed on schematics and layouts to represent electrical elements and sub circuits. The component information defines a signal-processing function or source, or secondarily a data source, data channel, or similar entity. Components have pins for connections, bitmaps in the project tree, and properties for simulation. The information that constitutes a component includes its component chooser bitmap, schematic symbol, layout footprint, pin properties, parameters and parameter values, and netlist string definition. A component can be associated with more than one simulation if it can be analyzed in more than one simulator.

Computer-Generated Hologram (CGH)

Refers to a slice of quartz or similar material into which three-dimensional patterns are cut using a laser. The angles of the patterns cut into the quartz are precisely calculated for use in the optical communication strategy known as holographic interconnect. All of these calculations are performed by a computer, and the laser used to cut the three-dimensional patterns into the

quartz is also controlled by a computer. Thus, the slice of quartz is referred to a computer-generated hologram.

Conductive Ink Technology

A technique in which tracks are screen printed directly onto the surface of a circuit board using a conductive ink.

Configurable Hardware

A product whose function may be customized once or a very few times (see also Reconfigurable Hardware, Remotely Reconfigurable Hardware, Dynamically Reconfigurable Hardware, and Virtual Hardware).

Conjunction

Propositions combined with an AND operator; for example, "You have a parrot on your head AND you have a fish in your ear." The result of a conjunction is true if all the propositions comprising that conjunction are true.

Conversion Loss

The ratio in dB of the IF output of a mixer to the RF input power. All conversion loss measurements and specification are normally based on the mixer being terminated on all ports and a stated LO signal power level being applied.

Cost Function

In an optimization setup, a cost function is based on goal values specified for at least one solution quantity. Optimetrics changes the design parameter values to fulfill the cost function. In Q3D, the cost function can be based on any solution quantity that Q3D Extractor can compute, such as resistance, inductance, and capacitance.

CPLD

See Complex Programmable Logic Device.

CRC

See Cyclic Redundancy Check.

CSIC

See Customer-Specific Integrated Circuit.

Customer-Specific Integrated Circuit (CSIC)

An alternative and possibly more accurate name for an ASIC, but this term is rarely used in the industry and shows little indication of finding favor with the masses.

Chemical Vapor Deposition (CVD)

A process for growing thin films on a substrate, in which a gas containing the required molecules is converted into a plasma by heating it to extremely high temperatures using microwaves. The plasma carries atoms to the surface of the substrate where they are attracted to the crystalline structure of the substrate. This underlying structure acts as a template. The new atoms continue to develop the structure to build up a layer on the substrate's surface.

Chemical Vapor Infiltration (CVI)

A process similar to chemical vapor deposition (CVD) but, in this case, the process commences by placing a crystalline powder of the required substance in a mold. Additionally, thin posts, or columns, can be pre-formed in the mold, and the powder can be deposited around them. When exposed to the same plasma as used in the CVD technique, the powder coalesces into a polycrystalline mass. After the CVI process has been performed, the posts can be dissolved leaving holes through the crystal for use in creating vias. CVI processes can produce layers twice the thickness of those obtained using CVD techniques at a fraction of the cost.

CW

Continuous wave; refers to an unmodulated sine-wave signal.

Cyclic Redundancy Check (CRC)

A calculation used to detect errors in data communications, typically performed using a linear feedback shift register. Similar calculations may be used for a variety of other purposes such as data compression.

Glossary: D

D/A

See Digital-to-Analog.

Data Bus

A bi-directional set of signals used by a computer to convey information from a memory location to the central processing unit and vice versa. More generally, a set of signals used to convey data between digital functions.

Data-Path Function

A well-defined function such as an adder, counter, or multiplier used to process digital data.

dBm

Power level relative to 1mV rms.

DC Balance

Stream of data encoded to ensure an equal balance of 1 or 0 . 8b10b encoding has been developed to ensure DC balancing.

DC Coupling

A method of coupling two different circuits together, allowing them to share both the static DC and varying AC characteristics of a signal.

Decoder (digital)

A logic function that uses a binary value, or address, to select between a number of outputs and to assert the selected output by placing it in its active state.

Deep Sub-Micron

Typically taken to refer to integrated circuits containing structures which are smaller than 0.5 microns.

DeMorgan Transformation

The transformation of a Boolean expression into an alternate, and often more convenient, form.

Design

Designs are the building blocks of projects, and can be Circuit designs or 3D planar EM models. Designs consist of schematics or geometrical models, model data, solution setup information, output graphs and tables, and other pieces of information that go into describing simulation of electrical circuits. A design is the largest single simulatable entity in a project.

Design Variation

A single combination of variable values that is solved during a parametric or optimization setup.

Device

A discrete, separate electrical entity such as a diode, a capacitor or a packaged transistor.

Die

(1)An unpackaged integrated circuit. In this case, the plural of die is also die (in much the same way that "a shoal of herring" is the plural of "herring"). (2)A piece of metal with a design engraved or embossed on it for stamping onto another material, upon which the design appears in relief.

Die Separation

The process of separating individual die from the wafer by marking the wafer with a diamond scribe and fracturing it along the scribed lines.

Die Stacking

A technique used in specialist applications in which several bare die are stacked on top of each other to form a sandwich. The die are connected together and then packaged as a single entity.

Dielectric Layer

(1) An insulating layer used to separate two signal layers.

(2) An insulating layer used to modify the electrical characteristics of an MCM-D substrate.

Diffusion Layer

The surface layer of a piece of semiconductor into which impurities are diffused to form P-type and N-type material. In addition to forming components, the diffusion layer may also be used to create embedded traces.

Digital

A value represented as being in one of a finite number of discrete states called quanta. The accuracy of a digital value is dependent on the number of quanta used to represent it.

Digital Circuit

A collection of logic gates used to process or generate digital signals.

Digital Signal Processor (DSP)

A primarily digital component used to process either digital or analog signals. In the case of the latter, the signal may first be conditioned, then converted into a digital equivalent using an analog-to-digital (A/D) converter function. The signal conditioning and A/D functions may either be external to the DSP or resident in the device. A typical DSP application might be the compression/decompression of video data.

Digital-to-Analog (D/A)

The process of converting a digital value into its analog equivalent.

Diode

A two-terminal device that only conducts electricity in one direction; in the other direction it behaves like an open switch. The term diode is typically taken to refer to a semiconductor device, although alternative implementations such as vacuum tubes are available.

Diode-Transistor Logic (DTL)

Logic gates implemented using particular configurations of diodes and bipolar junction transistors. For the majority of today's designers, diode-transistor logic is of historical interest only.

Discrete Device

Typically taken to refer to an electronic component such as a resistor, capacitor, diode, or transistor that is presented in an individual package. More rarely, the term may be used in connection with a simple integrated circuit containing a small number of primitive gates.

Discrete Wire Board (DWB)

A form of circuit board in which a special computer-controlled wiring machine ultrasonically bonds extremely fine insulated wires into the surface layer of the board. This discipline has enjoyed only limited recognition, but may be poised to emerge as the technology-of-choice for high-speed designers.

Discrete Wire Technology

The technology used to fabricate discrete wire boards.

Doping

The process of inserting selected impurities into a semiconductor to create P-type or N-type material.

Double-Sided

A printed circuit board with tracks on both sides

DPSK

Differential phase-shift keying.

DQPSK

Differential quadrature phase-shift keying.

DRAM

See Dynamic RAM.

DSP

See Digital Signal Processor.

DTL

See Diode-Transistor Logic.

DUT

Device under test.

Dynamic Flex

A type of flexible printed circuit which is used in applications that are required to undergo constant flexing such as ribbon cables in printers.

Dynamic RAM (DRAM)

A memory device in which each cell is formed from a transistor-capacitor pair. Called dynamic because the capacitor loses its charge over time, and each cell must be periodically recharged if it is to retain its data.

Glossary: E

E2PROM

See electrically Erasable Programmable Read-Only Memory.

EBE

See Electron Beam Epitaxy.

ECL

See Emitter-Coupled Logic.

Edge port

A place in a layout or footprint geometry through which excitation signals enter and leave the structure.

Edge-Sensitive

An input that only affects a function when it transitions from one logic value to another.

EEPROM

See electrically Erasable Programmable Read-Only Memory.

electrically Erasable Programmable Read-Only Memory (EEPROM or E2PROM)

A memory device whose contents can be electrically programmed by the designer. Additionally, the contents can be electrically erased allowing the device to be reprogrammed. Also known as EEPROM and E2PROM.

Electromigration

(1)A process in which structures on an integrated circuit's substrate are eroded by the flow of electrons in much the same way as land is eroded by a river (also known as subatomic erosion). (2)The process of forming transistor-like regions in a semiconductor using an intense magnetic field.

Electron Beam Epitaxy (EBE)

A technique for creating thin films on substrates in precise patterns, in which the substrate is first coated with a layer of dopant material before being placed in a high vacuum. A guided beam of electrons is fired at the substrate causing the dopant to be driven into it, effectively allowing molecular-thin layers to be "painted" onto the substrate where required.

Electron Beam Lithography

An integrated circuit fabrication process in which fine beams of electrons are used to draw extremely high-resolution patterns directly into the resist without the use of a mask.

Electro-Static Discharge (ESD)

The process of moving around can generate static electricity. The term electro-static discharge refers to a charged person, or object, discharging static electricity. Although the current associated with such a static charge is low, the electric potential can be in the millions of volts and can severely damage electronic components. CMOS devices are particularly prone to damage from static electricity.

EM

Electromagnetic.

Emitter-Coupled Logic (ECL)

Logic gates implemented using particular configurations of bipolar junction transistors.

Enzyme

One of numerous complex proteins which are produced by living cells and catalyze biochemical reactions at body temperatures.

EPROM

See Erasable Programmable Read-Only Memory.

Equivalent Gate

A concept in which each type of logic function is assigned an equivalent gate value for the purposes of comparing functions and devices. However, the definition of an equivalent gate varies depending on who you're talking to.

Equivalent Integrated Circuit

A concept used to compare the component density supported by diverse interconnection technologies such as circuit boards, hybrids, and multichip modules

Erasable Programmable Read-Only Memory (EPROM)

A memory device whose contents can be electrically programmed by the designer. Additionally, the contents can be erased by exposing the die to ultraviolet light through a quartz window mounted in the top of the component's package.

ESD

See Electro-Static Discharge.

Etching

The process of selectively removing any material not protected by a resist using an appropriate solvent or acid. In some cases the unwanted material is removed using an electrolytic process.

Eutectic Bond

A bond formed when two pieces of metal, or metal-coated materials, are pressed together and vibrated at ultrasonic frequencies.

Euler Angles

Euler angles are used in Ansoft software to carry out a coordinate transformation from one coordinate system to another. The Swiss mathematician and physicist Leonhard Euler first developed the classical rotation theorem to describe rotations in 3D space. The angles used are Euler angles and can be used to describe any 3D rotation. These angles, given by (ö, è, ø) represent a series of sequential rotations about two axis of the coordinate system. The first rotation (ö) represents a rotation about the Z-axis of the source coordinate system (X, Y, Z) which results in an intermediate coordinate system denoted by (X'', Y'', Z''). The second rotation (è) represents a rotation of the intermediate coordinate system about the X''-axis, again resulting in an intermediate coordinate system denoted by (X', Y', Z'). The third and final rotation (ø) represents a rotation about the Z'-axis of the intermediate coordinate system. The final rotation completes the rotation and results in the "target" coordinate system denoted (X, Y, Z).

For further information see, Eric W. Weisstein, "Euler Angles." From *MathWorld* - A Wolfram Web Resource.

http://mathworld.wolfram.com/EulerAngles.html .

Eye Diagram

Eye diagrams are commonly used to analyze signal integrity issues with communications channels. The bits are superimposed at unit intervals representing the duration of each bit.

Eye Mask

The size of the eye opening in the center of an eye diagram indicates the amount of voltage and timing margin available to sample this signal. Thus, for a particular electrical interface, a fixed reticule or window could be placed over the eye diagram showing how the actual signal compares to minimum criteria window, know as the eye mask. If a margin rectangle with width equal to the required timing margin and height equal to the required voltage margin fits into the opening, then the signal has adequate margins. Voltage margin can often be traded off for timing margin.

Glossary: F

Fan-Out Via

In the case of surface mount devices attached to double-sided or multilayer boards, each component pad is usually connected by a short length of track to a via which forms a link to other conducting layers, and this via is known as a fan-out via. The term fan-out via is generally also taken to include any vias that fall inside the device's footprint (under the body of the device). Some designers attempt to differentiate these vias from those that fall outside the device's footprint by referring to them as fan-in vias, but this is not an industry-standard term.

Falling-Edge

A transition from a logic 1 to a logic 0. Also known as a negative edge.

Falltime

The time it takes for a waveform to transition from the high logic state to the low logic state. Falltime is usually measured from 90% of the total signal swing to 10% of the signal swing.

Fan-In and Fan-Out Vias

In the case of surface mount devices attached to double-sided or multilayer boards, each component pad is usually connected by a short length of track to a via which forms a link to other conducting layers, and this via is known as a fan-out via. The term fan-out via is generally also taken to include any vias that fall inside the device’s footprint (under the body of the device). Some designers attempt to differentiate these vias from those that fall outside the device’s footprint by referring to them as fan-in vias, but this is not an industry-standard term.

Femto

Unit qualifier (symbol = f) representing one thousandth of one millionth of one millionth, or 10-15. For example, 3fS stands for 3 x 10-15 seconds.

FET

See Field-Effect Transistor.

Field-Effect Transistor (FET)

A transistor whose control, or gate, signal creates an electro-magnetic field which turns the transistor ON or OFF.

Field-Programmable Gate Array (FPGA)

A programmable logic device which is more versatile than traditional programmable devices such as PALs and PLAs, but less versatile than an application-specific integrated circuit. Some

field-programmable gate arrays use fuses such as those found in programmable logic devices, but others are based on SRAM equivalents.

Field-Programmable Interconnect Chip (FPIC)

An alternative, proprietary name for a field-programmable interconnect device (FPID).

Field-Programmable Interconnect Device (FPID)

A device which is used to connect logic devices together, and which can be dynamically reconfigured in the same way as standard SRAM-based FPGAs. Because each FPID may have around 1,000 pins, only a few such devices are typically required on a circuit board.

Filter

Filters are used to block out undesired frequencies. There are two types of filters: band pass and rejection. A band pass filter permits only the desired range to pass through, while the rejection filter attenuates an undesired range of frequencies.

Firmware

Refers to programs, or sequences of instructions, that are hard-coded into non-volatile memory devices.

First-In First Out (FIFO)

A memory device in which data is read out in the same order that it was written in.

Flash (e.g., Gold Flash)

An extremely thin layer of gold with a thickness measured on the molecular level which is either electroplated or chemically plated onto a surface.

FLASH Memory

An evolutionary technology that combines the best features of the EPROM and E2PROM technologies. The name FLASH is derived from the technology's fast reprogramming time compared to EPROM.

Flex

See Flexible Printed Circuit.

Flexible Printed Circuit (FPC or Flex)

A specialist circuit board technology, often abbreviated to "flex", in which tracks are printed onto flexible materials. There are a number of flavors of flex, including static flex, dynamic flex, and rigid flex.

Flipped Chip

A generic name for processes in which unpackaged integrated circuits are mounted directly onto a substrate with their component-sides facing the substrate.

Flipped TAB

A combination of flipped chip and tape automated bonding.

Footprint

The area occupied by a device mounted on a substrate.

Fourier Analysis

A mathematical procedure used to determine the collection of sine waves (differing in frequency and amplitude) that is necessary to make up the square-wave pattern under consideration.

FPC

See Flexible Printed Circuit.

FPGA

See Field-Programmable Gate Array.

FPIC

See Field-Programmable Interconnect Chip.

FPID

See Field-Programmable Interconnect Device.

FR4

The most commonly used insulating base material for circuit boards. FR4 is made from woven glass fibers which are bonded together with an epoxy. The board is cured using a combination of temperature and pressure which causes the glass fibers to melt and bond together, thereby giving the board strength and rigidity. The first two characters stand for "Flame Retardant". FR4 is technically a form of fiberglass, and some people do refer to these composites as fiberglass boards or fiberglass substrates, but not often.

Free-Space Optical Interconnect

A form of optical interconnect in which laser-diode transmitters communicate directly with photo-transistor receivers without employing optical fibers or optical waveguides.

Full Custom

An application-specific integrated circuit in which the designer has complete control over every mask layer used to fabricate the device. The manufacturer does not provide a cell library or pre-fabricate any components on the substrate.

Functional Latency

Refers to the fact that, at any given time, only a portion of the logic functions in a device or system are typically active (doing anything useful).

Functional Test

A test strategy in which signals are applied to a circuit's inputs, and the resulting signals which are observed on the circuit's outputs are compared to known good values.

Fuse

See Fusible-Link Technology.

Fusible-Link Technology (Fuse)

A programmable logic device technology which employs links called fuses. Individual fuses can be removed by applying pulses of relatively high voltage and current to the device's inputs.

Fuzz-Button

A small ball of fibrous gold used in one technique for attaching components such as multichip modules to circuit boards. Fuzz-buttons are inserted between the pads on the base of the package and their corresponding pads on the board. When the package is forced against the board, the fuzz-buttons compress to form good electrical connections. Even when the pressure is removed, the fuzz-buttons act in a similar manner to Velcro and continue to hold the component in place. One of the main advantages of the fuzz-button approach is that it allows broken devices to be quickly removed and replaced. Even though fuzz-button technology would appear to be inherently unreliable, it is used in such devices as missiles, so one can only assume that it is fairly robust.

Glossary: G

Gain

The ratio of the power output to the power input of the amplifier in dB. The gain is specified in the linear operating range of the amplifier where a 1 dB increase in input power gives rise to a 1 dB increase in output power. Gain = 20*log(S21)

Gate Array

An application-specific integrated circuit in which the manufacturer pre-fabricates devices containing arrays of unconnected components organized in groups called basic cells. The designer specifies the function of the device in terms of cells from the cell library and the connections between them, and the manufacturer then generates the masks used to create the metallization layers.

Glue Logic

Simple logic gates used to interface more complex functions together.

Gold Flash

An extremely thin layer of gold with a thickness measured on the molecular level which is either electroplated or chemically plated onto a surface.

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.

Gray Code

A sequence of binary values in which each pair of adjacent values differs by only a single bit; for example, 00, 01, 11, 10.

Green Ceramic

Unfired, malleable ceramic.

Ground Bounce

Momentary noise on the device negative signal plane causing a 0 signal to erroneously be seen as a 1. Ground bounce is caused by simultaneously switching outputs (SSO).

Guard Condition

A Boolean expression associated with a state transition in a state diagram or state table. The expression must be satisfied for that state transition to be executed.

Guided Probe

A form of functional test in which the operator is guided in the probing of a circuit to isolate a faulty component or track.

Guided-Wave

A form of optical interconnect, in which optical waveguides are fabricated directly on the substrate of a multichip module. These waveguides can be created using variations on standard opto-lithographic thin-film processes.

Glossary: H

Hard Macro (Macro Cell)

A logic function defined by the manufacturer of an application-specific integrated circuit. The function is described in terms of the simple functions provided in the cell library and the connections between them. The manufacturer also defines how the cells forming the macro will be assigned to basic cells and the routing of tracks between the basic cells.

Hardware

Generally understood to refer to any of the physical portions constituting an electronic system, including components, circuit boards, power supplies, cabinets, and monitors.

Harmonic

Integer multiples of the fundamental frequency of interest commonly produced by a non-linear amplifier.

Harmonic Balance

A frequency domain analysis technique for simulating nonlinear circuits and systems. This method assumes the input stimulus consists of a relatively few steady state sinusoids. Therefore the solution can be expressed as a sum of steady state sinusoids that includes the input frequencies in addition to any significant harmonics or mixing terms. A circuit with a single input source will require a single tone HB simulation. The harmonic balance simulation is ideal for situations where transient simulation methods are problematic, such as:

- Components modeled in frequency domain, for instance (dispersive) transmission lines
- Circuit time constants large compared to period of simulation frequency
- Circuits with lots of reactive components

Harmonic balance methods, therefore, are the best choice for most microwave circuits excited with sinusoidal signals (e.g., mixers, power amplifiers).

Harmonic Tuning

Impedance-matching at the harmonic frequencies for enhanced performance and efficiency.

Hertz (Hz)

Unit of frequency. One Hertz equals one cycle, or one oscillation, per second.

Heterojunction

The interface between two regions of dissimilar semiconductor materials. The interface of a hetrojunction has naturally occurring electric fields which can be used to accelerate electrons, and transistors created using hetrojunctions can switch much faster than their counterparts of the same size.

Hexadecimal

Base-16 numbering system. Each hexadecimal digit can be directly mapped onto four binary digits, or bits.

High Impedance State

The state on a signal that is not being driven by any value. A high-impedance state is indicated by the character Z.

Holographic Interconnect

A form of optical interconnect based on a thin slice of quartz, into which three-dimensional images are cut using a laser beam. Thus, the quartz is referred to as a computer-generated hologram, and this interconnection strategy is referred to as holographic.

Homojunction

An interface between two regions of semiconductor having the same basic composition but opposing types of doping. Homojunctions dominate current processes because they are easier to fabricate than heterojunctions.

Hybrid

An electronic sub-system in which a number of integrated circuits (packaged and/or unpackaged) and discrete components are attached directly to a common substrate. Connections between the components are formed on the surface of the substrate, and some components such as resistors and inductors may be fabricated directly onto the substrate.

Hydrogen Bond

The electrons in a water molecule are not distributed equally, because the oxygen atom is a bigger, more robust fellow which grabs more than its fair share. The end result is that the oxygen atom has an overall negative charge, while the two hydrogen atoms are left feeling somewhat on the positive side. This unequal distribution of charge means that the hydrogen atoms are attracted to anything with a negative bias; for example, the oxygen atom of another water molecule. The resulting bond is known as a hydrogen bond.

Hz

See Hertz.

Glossary: I

IBIS

IBIS (I/O Buffer Information Specification) is a standard for electronic behavioral specifications of integrated circuit input/output analog characteristics. The core of an IBIS model is a table of current versus voltage and I/O switching timing information. Xilinx IBIS models contain tables for typical, slow/MIN (weak transistors, high temperature, low voltage) and fast/MAX (strong

transistors, low temperature, high voltage) process corners. IBIS models are derived from SPICE simulation results and/or lab measurements. The benefit for IBIS model user is fast and accurate simulation while preserving IC vendors intellectual property (information about circuit and process details).

IC

See Integrated Circuit.

ICR

See In-Circuit Reconfigurable.

Impedance

The resistance to the flow of current caused by resistive, capacitive, or inductive devices (or undesired elements) in a circuit.

Impedance Matching

Function of ensuring that the impedance of the transmitter, the receiver, and the transmission line are identical. Mismatched impedances could result in signal reflections, ringing, overshoot, undershoot, and stairstep waveforms.

Incident Voltage

The user specified voltage at an input.

In-Circuit Reconfigurable (ICR)

An SRAM-based, or similar component which can be dynamically reprogrammed on-the-fly while remaining resident in the system.

Inductance

A property of a conductor that allows is to store energy in a magnetic field which is induced by a current flowing through it. Inductance is measured in units of Henries (the base unit is a Henry).

Insertion Loss

Insertion Loss (dB) is defined as the drop in power as a signal enters an RF component. This value not only includes the reflected incoming signal, but also the attenuation of the component.

In-System Programmable (ISP)

An E2-based, FLASH-based, or similar component which can be reprogrammed while remaining resident on the circuit board.

Integrated Circuit (IC)

A device in which components such as resistors, capacitors, diodes, and transistors are formed on the surface of a single piece of semiconductor.

Ion

A particle formed when an electron is added to, or subtracted from, a neutral atom or group of atoms.

Ion Implantation

A process in which beams of ions are directed at a semiconductor to alter its type and conductivity in certain regions.

Isolation

The ratio (expressed in dB) of the power level at one port compared to the resulting power level of the output port.

ISP

See In-System Programmable.

Glossary: J

JEDEC

See Joint Electronic Device Engineering Council.

Jitter

The jitter of a periodic signal is the delay between the expected transition of the signal and the actual transition. Jitter is a zero mean random variable. When worst case analysis is undertaken the maximum value of this random variable is used.

Jitter Tolerance

Jitter tolerance is defined as the peak-to-peak amplitude of sinusoidal jitter applied on the input that causes a predefined, acceptable loss at the output. For example jitter applied to the input of an OC-N equipment interface that causes an equivalent 1dB optical power penalty.

Jitter Transfer

Jitter transfer is defined as the ratio of jitter on the output of a device to the jitter applied on the input of the device, versus frequency. Jitter transfer is important in applications where the system is utilized in a loop-timed mode, where the recovered clock is used as the source of the transmit clock.

Joint Electronic Device Engineering Council (JEDEC)

A council which creates, approves, arbitrates, and oversees industry standards for electronic devices. In programmable logic, the term JEDEC refers to a textual file containing information used to program a device. The file format is a JEDEC approved standard and is commonly referred to as a JEDEC file.

Jumper

A small piece of wire used to link two tracks on a circuit board.

Glossary: K

Karnaugh Map

A graphical technique for representing a logical function. Karnaugh maps are often useful for the purposes of minimization.

Kelvin Scale of Temperature

A scale of temperature which was invented by the British mathematician and physicist William Thomas, first Baron of Kelvin. Under the Kelvin, or absolute, scale of temperature, 0 K (corresponding to -273oC) is the coldest possible temperature and is known as absolute zero.

Kilo

Unit qualifier (symbol = K) representing one thousand, or 103. For example, 3KHz stands for 3 x 103 Hertz.

Kirchhoff's current law

The sum of all currents entering a node is equal to the sum of all currents leaving the node.

Kirchhoff's voltage law

The directed sum of the electrical potential differences around a circuit must be zero.

Glossary: L

Laminate

A material constructed from thin layers or sheets. Often used in the substrate of circuit boards.

Large-Scale Integration (LSI)

Refers to the number of logic gates in a device. By one convention, large-scale integration represents a device containing 100 to 999 gates.

Laser Diode

A special semiconductor diode which emits a beam of coherent light.

Last-In First-Out (LIFO)

A memory device in which data is read out in the reverse order to which it was written in.

Latch-Up Condition

A condition in which a circuit draws uncontrolled amounts of current, and certain voltages are forced, or "latched-up", to some level. Particularly relevant in the case of CMOS devices which can latch-up if their operating conditions are violated.

Lateral Thermal Conductivity

Good lateral thermal conductivity means that the heat generated by components mounted on a substrate can be conducted horizontally across the substrate and out through its leads.

Layers (and Stackup)

Layers are used in the layout editor to organize and isolate sets of geometry or other visual indicators. Signal, Negative Signal, and Dielectric are common physical layers, while Symbol (to show component symbols in layout), Error, and Ratsnest (to show connectivity) are non-physical layers. The stackup contains additional properties of the physical layers, such as material, thickness, and elevation. Geometrical information on these layers is used to generate masks for manufacturing.

LDMOS

Laterally diffused metal oxide semiconductor.

Lead

(1) A metallic element (chemical symbol Pb).

(2) A metal conductor used to provide a connection from the inside of a device package to the outside world for soldering or other mounting techniques. Leads are also commonly called pins.

Lead Frame

A metallic frame containing leads and a base to which an unpackaged integrated circuit is attached. After encapsulation, the outer part of the frame is cut away and the leads are bent into the required shapes.

Lead Through-Hole (LTH)

A technique for populating circuit boards in which component leads are inserted into plated through-holes. Often abbreviated to "through-hole" or "thru-hole". When all of the components have been inserted, they are soldered to the board, usually using a wave soldering technique.

Level-Sensitive

An input whose effect on a function depends only on its current logic value or level, and is not directly related to it transitioning from one logic value to another.

LFSR

See Linear Feedback Shift Register.

Library

A library is a collection of one or more components or component dependencies (materials, symbols, footprints, or padstacks) stored in a container file. A library must be configured to a circuit before use, either by the user (manually) or by loading technology files (automatically). User libraries and Personal libraries are used to add foundry support, user defined models, and any custom set of components or simulation models. See the Library Overview topic for more information.

LIFO

See Last-In First-Out.

Limiting Level

The input power level when the output power is goes into compression and no longer becomes linear.

Line

Used to refer to the width of a track; for example, "This track has a line-width of 0.12mm."

Linear Feedback Shift Register (LFSR)

A shift register whose data input is generated as an XOR or XNOR of two or more elements in the register chain.

Linearity

Describes how closely an output signal is to a perfectly scaled multiple of a corresponding input signal.

Load Pull

Automated measurement of RF performance of a device under test (at a constant frequency) by varying the source and load impedance presented to the device

Logic Function

A mathematical function that performs a digital operation on digital data and returns a digital value.

Logic Gate

The physical implementation of a logic function.

Logic Synthesis

A process in which a program is used to optimize the logic used to implement a design.

Low-Fired Cofired

Similar in principle to standard cofired ceramic substrate techniques. However, low-fired cofired uses modern ceramic materials with compositions that allow them to be fired at temperatures as low as 650oC to 750oC. Firing at these temperatures in an inert atmosphere such as nitrogen allows non-refractory metals such as copper to be used to create tracks.

LSI

See Large-Scale Integration.

LTH

See Lead Through-Hole.

Glossary: M

Mask Programmable

A device such as a read-only memory which is programmed during its construction using a unique set of masks.

Maximal Displacement

A linear feedback shift register whose taps are selected such that changing a single bit in the input data stream will cause the maximum possible disruption to the register's contents.

Maximal Length

A linear feedback shift register that sequences through (2n - 1) states before returning to its original value.

Maxterm

The logical OR of the inverted variables associated with an input combination to a logical function.

MBE

See Molecular Beam Epitaxy.

MCM

See Multichip Module.

Medium-Scale Integration (MSI)

Refers to the number of logic gates in a device. By one convention, medium-scale integration represents a device containing 13 to 99 gates.

Meg

Unit qualifier (symbol = M) representing one million, or 106. For example, 3MHz stands for 3 x 106 Hertz.

Metallization Layer

A layer of conducting material on an integrated circuit that is selectively deposited or etched to form connections between logic gates. There may be several metallization layers separated by dielectric (insulating) layers.

Metal-Oxide Semiconductor (MOS)

A family of transistors where the controlling terminal is connected to a plate that is separated from the semiconduxtor by an insulating layer. This plate was originally made out metal (we now use polysilicon, or poly) and the insulator is an oxide -- hence the "metal-oxide" appellation.

Meta-Stable

A condition where the outputs of a logic function are oscillating uncontrollably between undefined values.

Micro

Unit qualifier (symbol = u) representing one millionth, or 10-6. For example, 3uS stands for 3 x 10-6 Seconds.

Microwave

The range in the electromagnetic spectrum from 300 MHz to 30 GHz (with corresponding wavelengths from 100 cm to 1 cm).

Microwire

A trade name for one incarnation of discrete wire technology. Microwire augments the main attributes of multiwire with laser-drilled blind vias, allowing these boards to support the maximum number of tracks and components.

Millman's method

The voltage on the ends of branches in parallel is equal to the sum of the currents flowing in every branch divided by the total equivalent conductance.

Minterm

The logical AND of the variables associated with an input combination to a logical function.

MMIC

Monolithic Microwave Integrated Circuit

Mod or Modulus

Refers to the number of states that a function such as a counter will pass through before returning to its original value. For example, a function that counts from 00002 to 11112 has a modulus of 16 and would be called a modulo-16 or mod-16 counter.

Molecular Beam Epitaxy (MBE)

A technique for creating thin films on substrates in precise patterns, in which the substrate is placed in a high vacuum, and a guided beam of ionized molecules is fired at it, effectively allowing molecular-thin layers to be "painted" onto the substrate where required.

MOS

See Metal-Oxide Semiconductor.

MOSFET

Metal-oxide semiconductor field-effect transistor.

MSI

See Medium-Scale Integration.

Multichip Module (MCM)

A generic name for a group of advanced interconnection and packaging technologies featuring unpackaged integrated circuits mounted directly onto a common substrate.

Multilayer

A printed circuit board constructed from a number of very thin single-sided and/or double-sided boards which are bonded together using a combination of temperature and pressure.

Multiplexer (digital)

A logic function that uses a binary value, or address, to select between a number of inputs and conveys the data from the selected input to the output.

Multiwire

A trade name for one incarnation of discrete wire technology.

Multizone

A stackup that contains zones or areas, each of which contains a subset of the layers in the stackup.

Mutual Capacitance

The capacitance between two conductors (one considered aggressor, the other victim) when all other conductors are connected together and then regarded as an ignored ground. It describes the amount of coupling due to the electric field. The mutual capacitance will inject an often undesired current into the victim line proportional to the rate of change of voltage on the aggressor line. Mutual Capacitance it a cause of crosstalk.

Mutual Inductance

The inductance between two conductors (one considered aggressor, the other victim) placed close enough that the magnetic field induced by a current flowing into the aggressor line encompasses the victim. The mutual inductance will inject an often undesired voltage noise onto the victim proportional to the rate of change of the current on the aggressor line.

Glossary: N

Nano

Unit qualifier (symbol = n) representing one thousandth of one millionth, or 10^{-9}. For example, 3nS stands for 3 x 10^{-9} Seconds.

Nanobot

A molecular-sized robot (see Nanotechnology below)

Nanophase Materials

A form of matter which was only recently discovered, in which small clusters of atoms form the building blocks of a larger structure. These structures differ from those of naturally occurring crystals, in which individual atoms arrange themselves into a lattice.

Nanotechnology

Nanotechnology is an elusive term that is used by different research-and-development teams to refer to whatever it is that they're working on at the time. However, irrespective of their particular area of interest, nanotechnology always refers to something extremely small. One of the more exciting branches of nanotechnology that has been suggested as having potential in the future is that of micro-miniature electronic products that assemble themselves.

N-channel MOS (NMOS)

Refers to the order in which the semiconductor is doped in a MOS device. That is, which structures are constructed as N-type versus P-type material.

Negative-Edge

A transition from a logic 1 to a logic 0. Also known as a falling edge.

Negative Ion

An atom or group of atoms with an extra electron.

Negative Logic

A convention which dictates the relationship between logical values and the physical voltages used to represent them. The more negative potential is considered to represent TRUE and the more positive potential is considered to represent FALSE. Also known as negative true logic.

Negative Resist

A process where ultraviolet radiation passing through the transparent areas of a mask causes the resist to be cured. The uncured areas are then removed using an appropriate solvent.

Negative Signal Plane

A conducting layer in, or on, a substrate providing a grounding, or reference, point for components. There may be several negative signal planes separated by insulating layers.

Negative-True

A convention which dictates the relationship between logical values and the physical voltages used to represent them. The more negative potential is considered to represent TRUE and the more positive potential is considered to represent FALSE. Also known as negative logic.

Nibble

See Nybble.

NMOS

See N-channel MOS.

Noise Figure / Noise Factor

The Noise Factor of a transducer at a specified input frequency is the ratio of (a/b) where “a and b” are:

> (a) the available Signal to Noise Ratio (SNR) at the signal generator terminals per unit bandwidth when the temperature of the input termination (generator or source) is 290 K and the bandwidth is limited by the transducer, to
>
> (b) the available SNR per unit bandwidth at the output terminals of the transducer.

Traditionally:

Noise Figure NF = 10 log(noise factor F)

Noise Temperature (T_e) = T_o(F - 1)

Where:

- T_e is the noise temperature
- T_o is standard temperature 290 K
- F is noise factor

Noise Floor

This is defined as the lowest possible input to a chain or a component, that will produce a detectable output.

Noise Temperature

This is the amount of thermal noise in a chain or a component. Noise Factor and Noise Temperature (T_e) are related as follows:

Noise Temperature (T_e) = (F - 1)T_o

Where:

- T_e is the noise temperature
- T_o is standard temperature 290 K
- F is noise factor

For example, a noise figure of 2.0 dB is equivalent to a Noise Temperature of 170 K.

Nominal Design

The original model on which Optimetrics analyses are based.

Non-Volatile

A memory device which does not lose its data when power is removed from the system.

Non-Volatile RAM

A device which is generally formed from an SRAM die mounted in a package with a very small battery, or as a mixture of SRAM and EEPROM cells fabricated on the same die.

Norton's theorem

Any two-terminal collection of voltage sources and resistors is electrically equivalent to an ideal current source in parallel with a single resistor.

NPN (N-type - P-type - N-type)

Refers to the order in which the semiconductor is doped in a bipolar junction transistor.

N-Port

An N-port component is typically characterized by network parameter data contained in the N-port itself in spreadsheet form or (more usually) by network parameter data contained in an external file.

N-type

A piece of semiconductor doped with impurities that make it amenable to donating electrons.

Nybble

A group of four binary digits, or bits (also called a nibble).

Glossary: O

Octal

Base-8 numbering system. Each octal digit can be directly mapped onto three binary digits, or bits.

Ohm

Unit of resistance. The Greek letter omega, *, is often used to represent ohms; for example, 1M* indicates one million ohms.

Ohm's law

The voltage across a resistor is the product of its resistance and the current flowing through it.

One-Hot Encoding

A form of state assignment for state machines in which each state is represented by an individual state variable.

One-Time Programmable

A device such as a PAL, PLA, or PROM that can only be programmed a single time and whose contents cannot be subsequently erased.

Optical Interconnect

The generic name for interconnection strategies based on opto-electronic systems, including fiber-optics, free-space, guided-wave, and holographic techniques.

Optical Lithography

A process in which radiation at optical wavelengths (usually in the ultraviolet range) is passed through a mask, and the resulting patterns are projected onto a layer of resist coating the substrate material.

Optical Mask

A sheet of material carrying patterns that are either transparent or opaque to the wavelengths used in an optical-lithographic process. Such a mask can carry hundreds of thousands of fine lines and geometric shapes.

Opto-Electronic

Refers to a system which combines optical and electronic components.

Organic Resist

A material which is used to coat a substrate and is then selectively cured to form an impervious layer. These materials are called organic because they are based on carbon compounds as are living creatures.

Organic Solvent

A solvent for organic materials such as those used to form organic resists.

Organic Substrate

Substrate materials such as FR4, in which woven glass fibers are bonded together with an epoxy. These materials are called organic because epoxies are based on carbon compounds as are living creatures.

Overglassing

One of the final stages in the integrated circuit fabrication process in which the entire surface of the wafer is coated with a layer of silicon dioxide or silicon nitride. This layer may also be referred to as the barrier layer or the passivation layer. An additional lithographic step is required to pattern holes in this layer to allow connections to be made to the pads.

Glossary: P

Package

Leaded assembly (inside of which one or more dies are mounted and connected) for use in larger circuits.

Pad

An area of metallization on a substrate used for probing or to connect to a via, plated through-hole, or an external interconnect.

Pad Grid Array (PGA)

A packaging technology in which a device's external connections are arranged as an array of conducting pads on the base of the package.

Padstack

Refers to any pads, anti-pads, and thermal relief pads associated with a via or a plated through-hole as it passes through the layers forming the substrate.

Padcap

A special flavor of circuit board used for high-reliability military applications. Distinguished by the fact that the outer surfaces of the board have pads but no tracks. Signal layers are only created on the inner planes, and tracks are connected to the surface pads by vias.

Parallel-In Serial-Out (PISO)

Refers to a shift register in which the data is loaded in parallel and read out serially.

Parasitic Effects

The effects caused by undesired resistance, capacitance, or inductance inherent in the material or topology of a track or component.

Passive Trimming

A process in which a laser beam is used to trim components such as thick-film and thin-film resistors on an otherwise unpopulated and unpowered hybrid or multichip module substrate. Probes are placed at each end of a component to monitor its value while the laser evaporates some of the material forming the component.

Pass-Transistor Logic

A technique for connecting MOS transistors such that data signals pass between their source and drain terminals. Pass-transistor logic minimizes the number of transistors required to implement a function, and is typically employed by designers of cell libraries or full-custom integrated circuits.

PGA

See either Pad Grid Array or Pin Grid Array.

Photo-Transistor

A special transistor which converts an optical input in the form of light into an equivalent electronic signal in the form of a voltage or current.

Pico

Unit qualifier (symbol = p) representing one millionth of one millionth, or 10^{-12}. For example, 3pS stands for 3 x 10^{-12} Seconds.

PIN Diode

A diode where a thin layer exists between the N and P regions. Rectification with pin diodes is limited. They actually behave more like a variable resistor that changes based upon the DC bias.

Pin Grid Array (PGA)

A packaging technology in which a device's external connections are arranged as an array of conducting leads, or pins, on the base of the package.

PISO

See Parallel-In Serial-Out.

Place-Value

Refers to a numbering system in which the value of a particular digit depends both on the digit itself and its position in the number.

PlasmA gaseous state in which the atoms or molecules are dissociated to form ions.

Plated Through-Hole (PTH)

(1) A hole in a double-sided or multilayer board that is used to accommodate a through-hole component lead and is plated with copper.

(2) An alternative name for the lead through-hole technique for populating circuit boards in which component leads are inserted into plated through-holes.

PMOS (P-channel MOS)

Refers to the order in which the semiconductor is doped in a MOS device. That is, which structures are constructed as P-type versus N-type material.

PNP (P-type - N-type - P-type)

Refers to the order in which the semiconductor is doped in a bipolar junction transistor.

Polysilicon Layer

An internal layer in an integrated circuit used to create the gate electrodes of MOS transistors. In addition to forming gate electrodes, the polysilicon layer can also be used to interconnect components. There may be several polysilicon layers separated by dielectric (insulating) layers.

Populating

The act of attaching components to a substrate.

Positive Logic

A convention which dictates the relationship between logical values and the physical voltages used to represent them. The more positive potential is considered to represent TRUE and the more negative potential is considered to represent FALSE. Also known as positive true logic.

Positive Resist

A process where radiation passing through the transparent areas of a mask causes previously cured resist to be degraded. The degraded areas are then removed using an appropriate solvent.

Positive-Edge

A transition from a logic 0 to a logic 1. Also known as a rising edge.

Positive-True

A convention which dictates the relationship between logical values and the physical voltages used to represent them. The more positive potential is considered to represent TRUE and the more negative potential is considered to represent FALSE. Also known as positive logic.

Power Amplifier

A class of amplifier with the primary purpose of delivering high output power (usually accompanied by significant dissipated power).

Power Plane

A conducting layer in or on the substrate providing power to the components. There may be several power planes separated by insulating layers.

Prepreg

Non-conducting semi-cured layers of FR4 used to separate conducting layers in a multilayer circuit board.

Primitives

Simple logic functions such as BUF, NOT, AND, NAND, OR, NOR, XOR, and XNOR may be referred to as primitive logic gates or primitives.

Product Term

A set of literals linked by an AND operator.

Programmable Array Logic (PAL)

A programmable logic device in which the AND array is programmable but the OR array is pre-defined.

Programmable Logic Array (PLA)

The most user-configurable of the traditional programmable logic devices, because both the AND and OR arrays are programmable.

Programmable Logic Device (PLD)

The generic name for a device constructed in such a way that the designer can configure, or "program" it to perform a specific function.

Programmable Read-Only Memory (PROM)

A programmable logic device in which the OR array is programmable but the AND array is pre-defined. Usually considered to be a memory device whose contents can be electrically programmed (once) by the designer.

Project

A container that groups designs and their associated settings, including report definitions, in a file with a .adsn extension. To the fullest extent possible, projects are portable in that they include, rather than merely refer to, the library elements (graphical symbols, materials, footprints, and so on) of the components and models they contain. Multiple projects can be open simultaneously.

PROM

See Programmable Read-Only Memory.

Pseudo-Random

An artificial sequence of values that give the appearance of being random.

PTH

See Plated Through-Hole.

P-type

A piece of semiconductor doped with impurities that make it amenable to accepting electrons.

Pulling

The difference between the maximum frequency of a VCO when the phase angle of the load impedance reflection coefficient varies through 360 degrees.

Pulsed Radar

A transmit and receive system used for ranging and detection that transmits a train of short bursts of high microwave signals and receives return signals reflected from a target.

Pushing

The change in frequency when the supply voltage changes, expressed in MHz/V.

Glossary: Q

Q

Quality factor; a measure of stored energy/dissipated energy. Also a measure of bandwidth.

QAM

Quadrature amplitude modulation.

QFP

See Quad Flat Pack.

QPSK

Quadrature phase-shift keying.

Quad Flat Pack (QFP)

The most commonly used package in surface mount technology to achieve a high lead count in a small area. Leads are presented on all four sides of a thin square package.

Quinary

Base-5 numbering system.

Glossary: R

Radio Frequency (RF)

The range in the electromagnetic spectrum loosely defined from 30 MHz to 3 GHz (with corresponding wavelengths from 1000 cm to 10 cm).

Radix

Refers to the number of digits in a numbering system. For example, the decimal numbering system is said to be radix-10. May also be referred to as the "base".

Rats Layer

A non-physical, default layer in the stackup that displays a drawing of logical connections between different components, circuit elements, and net connections. A single connection is called a rat and all of the connections on a rat layers is a rat's nest.

Reed-Müller Logic

Logic functions implemented using only XOR and XNOR gates.

Refractory Metal

Metals such as tungsten, titanium, and molybdenum which are capable of withstanding extremely high, or refractory, temperatures.

Remotely Reconfigurable Hardware

A product whose function may be customized remotely, by telephone or radio, while remaining resident in the system (see also Configurable Hardware, Reconfigurable Hardware, Dynamically Reconfigurable Hardware, and Virtual Hardware).

Reflection

The appearance of a previously transmitted signal on the transmission line causing interference with the current signal. Reflections are caused by a poorly terminated or discontinuous transmission line, where the signal energy is not fully absorbed within the receiver and is therefore transmitted back toward the transmitter.

Resist

A material which is used to coat the substrate and is then selectively cured to form an impervious layer.

Resistor-Transistor Logic (RTL)

Logic gates implemented using particular configurations of resistors and bipolar junction transistors. For the majority of today's designers, resistor-transistor logic is of historical interest only.

Return Loss

Return Loss (dB) is defined as a ratio of the incoming signal to the same reflected signal as it enters a component.

Return Loss (dB) = 10 * LOG10(Reflected Power/Incident Power)

RF

See Radio Frequency.

RF Power

A class of engineering. Circuits and signals primarily concerned with power levels ranging from a few watts to tens of thousands of watts in the RF spectrum.

RF Power Transmitter

A discrete packaged transistor used in the amplification of RF power.

Rigid Flex

Hybrid constructions which combine standard rigid circuit boards with flexible printed circuits, thereby reducing the component count, weight, and susceptibility to vibration of the circuit, and greatly increasing its reliability.

Ringing

Common name for the waveform that is seen when a transmission line ends at a high impedance discontinuity. The signal first overshoots, then dips down below the target value, and continues this with decreasing amplitude until it converges on the target voltage.

Risetime

The time it takes for a signal to rise from 10% of its total logic swing to 90% of its total logic swing.

Rising-Edge

A transition from a logic 0 to a logic 1. Also known as a positive edge.

RTL

See Resistor-Transistor Logic.

Glossary: S

Sample Rate

Time increment of analysis. Sometimes referred to as sampling rate.

Sampling

The process of converting an analog signal into a series of digital values.

Scalar Notation

A notation in which each signal is assigned a unique name; for example, a3, a2, a1, and a0.

Scaling

A technique for making transistors switch faster by reducing their size. This strategy is known as scaling, because all of the transistors features are typically reduced by the same proportion.

Schematic

Common name for a circuit diagram.

Scrubbing

The process of vibrating two pieces of metal, or metal coated materials, at ultrasonic frequencies to create a friction weld.

Seed Value

An initial value loaded into a linear feedback shift register or random number generator.

Sensor

A transducer that detects a physical quantity and converts it into a form suitable for processing. For example, a microphone is a sensor which detects sound and converts it into a corresponding voltage or current.

Sequential

A function whose output value depends not only on its current input values, but also on previous input values. That is, the output value depends on a sequence of input values.

Side-Emitting Laser Diode

A laser diode constricted at the edge of an integrated circuit's substrate such that, when power is applied, the resulting laser beam is emitted horizontally; that is, parallel to the surface of the substrate.

Sign Bit

The most significant binary digit, or bit, of a signed binary number. If set to a logic 1, this bit represents a negative quantity.

Signal Conditioning

Amplifying, filtering, or otherwise processing a signal.

Signal Layer

A layer carrying tracks in a circuit board, hybrid, or multichip module. See also wiring layer.

Signature

Refers to the checksum value from a cyclic-redundancy-check when used in the guided-probe form of functional test.

Signature Analysis

A guided-probe functional-test technique based on signatures.

Signed Binary Number

A binary number in which the most-significant bit is used to represent a negative quantity. Thus, a signed binary number can be used to represent both positive and negative values.

Sign-Magnitude

Negative numbers in standard arithmetic are typically represented in sign-magnitude form by prefixing the value with a minus sign; for example, -27. For reasons of efficiency, computers rarely employ the sign-magnitude form. Instead, they use signed binary numbers to represent negative values.

Silicon Bumping

The process of depositing additional metallization on a die's pads to raise them fractionally above the level of the Barrier Layer.

Silicon Chip

Although a variety of semiconductor materials are available, the most commonly used is silicon and integrated circuits are popularly known as silicon chips, or simply chips.

Silicon Compiler

The program used in compiled cell technology to generate the masks used to create components and interconnections. May also be used to create data-path functions and memory functions.

Single-Sided

A printed circuit board with tracks on only one side.

Sintering

A process in which ultra-fine metal powders weld together at temperatures much lower than those required for larger pieces of the same materials.

SIPO (Serial-In Parallel-Out)

Refers to a shift register in which the data is loaded in serially and read out in parallel.

SISO (Serial-In Serial-Out)

Refers to a shift register in which the data is both loaded in and read out serially.

Skew

Time delay between different bits transmitted at the same time, measured at the receiver.

Skin Effect

In the case of high frequency signals, electrons are only conducted on the outer surface, or skin, of a conductor. This phenomenon is known as the skin effect.

Small-Scale Integration (SSI)

Refers to the number of logic gates in a device. By one convention, small-scale integration represents a device containing 1 to 12 gates.

SMD

See Surface Mount Device.

SMOBC

See Solder Mask Over Bare Copper.

SMT

See Surface Mount Technology.

SNR

Signal-to-noise ratio.

Soft Macro (Macro Function)

A logic function defined by the manufacturer of an application-specific integrated circuit. The function is described in terms of the simple functions provided in the cell library and connections between them. The assignment of cells to basic cells and the routing of the tracks is determined at the same time, and using the same tools, as for the other cells specified by the designer.

Solder Bumping

A flipped chip technique in which spheres of solder are formed on the die's pads. The die is flipped and the solder bumps are brought into contact with corresponding pads on the substrate. When all the chips have been mounted on the substrate, the solder bumps are melted using reflow soldering or vapor-phase soldering.

Solder Mask

A layer applied to the surface of the substrate that prevents solder from sticking to any metallization except where holes are patterned into the mask.

Solder Mask Over Bare Copper (SMOBC)

A technique in which the solder mask is applied in advance of the tin-lead plating. This results in lighter circuit boards because the tin-lead alloy is only used to plate the pads.

Solution

A solution is the successful result of an analysis, or imported results available for plotting.

Space

Used to refer to the width of the gap between adjacent tracks.

SPICE

Simulated Program for Integrated Circuit Emulation

SRAM

See Static RAM.

SS-CDMA

Spread-spectrum code-division multiple access.

SSI

See Small-Scale Integration.

Stackup

An arrangement of physical signal and dielectric layers that is used in the design of circuit boards. In Circuit, the stackup editor lists conceptual non-stackup layers with the physical stackup layers.

Standard Cell

An application-specific integrated circuit which, unlike a gate array, does not use the concept of a basic cell and does not have any pre-fabricated components. The manufacturer creates custom masks for every stage of the device's fabrication allowing each logic function to be created using the minimum number of transistors.

State Assignment

The process by which the states in a state machine are assigned to the binary patterns that are to be stored in the state variables.

State Diagram

A graphical representation of the operation of a state machine.

State Machine

The actual implementation (in hardware or software) of a function that can be considered to consist of a set of states through which it sequences.

State Table

A tabular representation of the operation of a state machine. Similar to a truth table, but also includes the current state as an input and the next state as an output.

State Transition

An arc connecting two states in a state diagram.

State Variable

One of a set of registers whose values represent the current state occupied by a state Static Flex.

Statement

A sentence that asserts or denies an attribute about an object or group of objects.

Static Flex

A type of flexible printed circuit which can be manipulated into permanent three-dimensional shapes for applications such as calculators and high-tech cameras which require efficient use of volume and not just area.

Static RAM (SRAM)

A memory device in which each cell is formed from four or six transistors configured as a latch or a flip-flop. The term static is used because, once a value has been loaded into an SRAM cell, it will remain unchanged until it is explicitly altered or until power is removed from the device.

Steady State

A condition in which nothing is changing or happening.

Subatomic Erosion

A process in which structures on an integrated circuit's substrate are eroded by the flow of electrons in much the same way as land is eroded by a river (also known as electromigration)

Substrate

Generic name for the base layer of an integrated circuit, hybrid, multichip module, or circuit board. Substrates may be formed from a wide variety of materials, including semiconductors, ceramics, FR4 (fiberglass), glass, sapphire, or diamond depending on the application. Note that the term substrate has traditionally not been widely used in the circuit board world, at least not by the people who manufacture the boards. However, there is an increasing tendency to refer to a circuit board as a substrate by the people who populate the boards. The main reason for this is that circuit boards are often used as substrates in hybrids and multichip modules, and there is a trend toward a standard terminology across all forms of interconnection technology.

Subtractive Process

A process in which a substrate is first covered with conducting material, then any unwanted material is subsequently removed, or subtracted.

Superconductor

A material with zero resistance to the flow of electric current.

Surface Mount Device (SMD)

A component whose packaging is designed for use with surface mount technology.

Surface Mount Technology (SMT)

A technique for populating hybrids, multichip modules, and circuit boards, in which packaged components are mounted directly onto the surface of the substrate. A layer of solder paste is screen printed onto the pads and the components are attached by pushing their leads into the

paste. When all of the components have been attached, the solder paste is melted using either reflow soldering or vapor-phase soldering.

Surface-Emitting Laser Diode

A laser diode constricted on an integrated circuit's substrate such that, when power is applied, the resulting laser beam is emitted directly away from the surface of the substrate.

Sweep Definition

Also called *variable sweep definition*. A set of variable values within a range that Optimetrics drives the Electronics Desktop to solve when a parametric setup is analyzed. A parametric setup can include one or more sweep definitions.

Synchronous

(1)A signal whose data is not acknowledged or acted upon until the next active edge of a clock signal. (2)A system whose operation is synchronized by a clock signal.

System Gain

The net loss of a system as a measure of reliability with respect to system parameters. It measures the difference between the output power and minimum input power required for satisfactory performance. It represents a system net loss, and is represented as a negative dB value that is larger than or equal to the summed gains and losses of a signal propagating in a system from transmitter to receiver.

Glossary: T

Tap

A register output which is used to generate the next data input to a linear feedback shift register.

Tape Automated Bonding (TAB)

A process in which transparent flexible tape has tracks created on its surface. The pads on unpackaged integrated circuits are attached to corresponding pads on the tape which is then stored in a reel. Silver-loaded epoxy is screen printed on the substrate at the site where the device is to be located and onto the pads to which the device's leads are to be connected. The reel of TAB tape is fed through an automatic machine which pushes the device and the TAB leads into the epoxy. When the silver-loaded epoxy is cured using reflow soldering or vapor-phase soldering, it forms electrical connections between the TAB leads and the pads on the substrate.

TDM

Time-division multiplexing.

TDMA

Time-division multiple access.

Technology File

A technology file is a collection of information, specifiable by name at design-creation time using the Choose Layout Technology dialog box, that specifies material stackup properties for layout, substrates for simulation, and component libraries for design creation. A technology (.asty) file may be saved from the current project via the File menu. A technology file initializes a design with a set of data to avoid repeated entry of commonly used data. This data can consist of layers and stackup information for layout, configured libraries of components, and substrate definition (s) for circuit analysis. Users and foundries can customize Technology files for their own manufacturing process and simulation models.

Tertiary Logic

An experimental technology in which logic gates are based on three distinct voltage levels. The three voltages are used to represent the tertiary digits 0, 1, and 2, and their logical equivalents FALSE, TRUE, and MAYBE.

Thermal Impedance

Relates the temperature rise for a given dissipated power (which employs an analogy to the voltage-current relationship of impedance).

Thermal Relief Pad

A special pattern etched around a via or a plated through-hole to connect it into a power or ground plane. A thermal relief pad is necessary to prevent too much heat being absorbed into the power or ground plane when the board is being soldered.

Thermal Tracking

Typically used to refer to the problems associated with optical interconnection systems whose alignment may be disturbed by changes in temperature.

Thevenin's theorem

Any two-terminal combination of voltage sources and resistors is electrically equivalent to a single voltage source in series with a single resistor.

Thick-Film Process

A process used in the manufacture of hybrids and, to a lesser extent, multichip modules in which signal and dielectric (insulating) layers are screen-printed onto the substrate.

Thin-Film Process

A process used in the manufacture of hybrids and multichip modules in which signal layers and dielectric (insulating) layers are created using opto-lithographic techniques.

Time-Of-Flight

The time taken for a signal to propagate from one logic gate or opto-electronic component to another.

Tin-Lead Plating

An electroless plating process in which exposed areas of copper on a circuit board are coated with a layer of tin-lead alloy. The alloy is used to prevent the copper from oxidizing and provides protection against contamination.

Tinning

An abbreviation of tin-lead plating, which is an electroless plating process in which exposed areas of copper on a circuit board are coated with a layer of tin-lead alloy. The alloy is used to prevent the copper from oxidizing and provides protection against contamination.

Toggle

Refers to the contents or outputs of a logic function switching to the inverse of their previous logic values.

TOI

Third-order intercept point.

Total Voltage

Total voltage = Incident voltage + reflected voltage, that is $V^T = V^a + V^b$.

Trace

A conducting connection between electronic components. May also be called a track or a signal. In the case of integrated circuits, such interconnections are often referred to collectively as metallization.

Tracks

A conducting connection between electronic components. May also be called a trace or a signal. In the case of integrated circuits, such interconnections are often referred to collectively as metallization.

Transducer

A device that converts input energy of one form into output energy of another.

Transistor

A three-terminal semiconductor device that, in the digital world, can be considered to operate like a switch.

Tri-State Function

A function whose output can adopt three states: 0, 1, and Z (high-impedance) The function does not drive any value in the Z state and, in many respects, may be considered to be disconnected from the rest of the circuit.

Truth Table

A convenient way to represent the operation of a digital circuit as columns of input values and their corresponding output responses.

TTL (Transistor-Transistor Logic)

Logic gates implemented using particular configurations of bipolar junction transistors.

Glossary: U

ULA

See Uncommitted Logic Array.

Ultra-Large-Scale Integration (ULSI)

Refers to the number of logic gates in a device. By one convention, ultra-large-scale integration represents a device containing a million or more gates.

ULSI

See Ultra-Large-Scale Integration.

Uncommitted Logic Array (ULA)

One of the original names used to refer to gate array devices. This term has largely fallen into disuse.

Unsigned Binary Number

A binary number in which all the bits are used to represent positive quantities. Thus, an unsigned binary number can only be used to represent positive values.

Undershoot

The percentage a waveform falls below its lowest determined value before setting at the correct value.

Glossary: V

Vapor-Phase Soldering

A surface mount process in which a substrate carrying components attached by solder paste is lowered into the vapor-cloud of a tank containing boiling hydrocarbons. This melts the solder paste thereby forming good electrical connections. However, vapor-phase soldering is becoming increasingly less popular due to environmental concerns.

Vaporware

Refers to either hardware or software that exist only in the minds of the people who are trying to sell them to you.

Variable Sweep Definition

Also called *sweep definition*. A set of variable values within a range that Optimetrics drives HFSS or Q3D to solve when a parametric setup is analyzed. A parametric setup can include one or more sweep definitions.

Vector Notation

A notation used in logic simulation and synthesis in which a single name is used to reference a group of signals, and individual signals within the group are referenced by means of an index; for example, a[3:0] = a[3], a[2], a[1], and a[0].

Very-Large-Scale Integration (VLSI)

Refers to the number of logic gates in a device. By one convention, very-large-scale integration represents a device containing 1,000 to 999,999 gates.

Via

A hole filled or lined with a conducting material which is used to link two or more conducting layers in a substrate.

Virtual Hardware or Virtual Logic

An extension of dynamically configurable hardware based on a new generation of FPGAs which were introduced around the beginning of 1994. In addition to supporting the dynamic reconfiguration of selected portions of the internal logic, these devices also feature: no disruption to the device's inputs and outputs; no disruption to the system-level clocking; the continued operation of any portions of the device that are not undergoing reconfiguration; and no disruption to the contents of internal registers during reconfiguration, even in the area being reconfigured (see also Configurable Hardware, Reconfigurable Hardware, Remotely Reconfigurable Hardware, and Dynamically Reconfigurable Hardware).

Virtual Memory

A trick used by a computer's operating system to pretend that it has access to more memory than is actually available. For example, a program running on the computer may require ten mega-bytes to store its data, but the computer may have only five mega-bytes of memory available. To get around this problem, whenever the program attempts to access a memory location that does not physically exist, the operating system performs a slight-of-hand and exchanges some of the contents in the memory with data on the hard disk.

VLSI

See Very-Large-Scale Integration.

Volatile

Refers to a memory device which loses any data it contains when power is removed from the system; for example, random-access memory in the form of SRAM or DRAM

Glossary: W

Wafer Probing

The process of testing individual integrated circuits while they still form part of a wafer. An automated tester places probes on the device's pads, applies power to the power pads, injects a series of signals into the input pads, and monitors the corresponding signals returned from the output pads.

Wave Soldering

A process used to solder circuit boards populated with through-hole components. A wave generating mechanism maintains a wave of hot, liquid solder traveling back and forth across the surface of a tank. The populated circuit boards are passed over the wave soldering machine on a conveyor belt. The velocity of the convener belt is carefully controlled and synchronized such that the solder wave brushes across the bottom of the board only once.

Waveguide

A transparent path bounded by non-transparent, reflective areas, which is fabricated directly onto the surface of a substrate. Used in the optical interconnection strategy known as guided-wave.

W-CDMA

Wideband code-division multiple access. Typically defined with 5 MHz channels and 3.84 MHz carrier signals.

Wire Bonding

The process of connecting the pads on an unpackaged integrated circuit to corresponding pads on a substrate using wires that are finer than a human hair. Wire bonding may also be used to connect the pads on an unpackaged integrated circuit, hybrid, or multichip module to the leads of the component package.

Wiring Layer

A layer carrying wires in a discrete wired board. See also signal layer.

Word

A group of signals or logic functions performing a common task and carrying or storing similar data; for example, a value on the data bus could be referred to as a data word.

Glossary: X

X-Ray Lithography

Similar in principle to optical lithography, but capable of constructing much finer features due to the shorter wavelengths involved. However, X-ray lithography requires an intense source of X-rays, is more difficult to use, and is considerably more expensive than optical lithography.

Glossary: Y

Yield

The number of devices that work as planned, specified as a percentage of the total number actually fabricated.

Glossary: Z

Zepto (z)

The symbol used to represent the high-impedance state in tri-state logic.

Zone

A spatial area on a printed circuit board that may contain a subset or all of the layers in the board’s stackup.

Index

B

Q

T

W

Z