Circuit 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

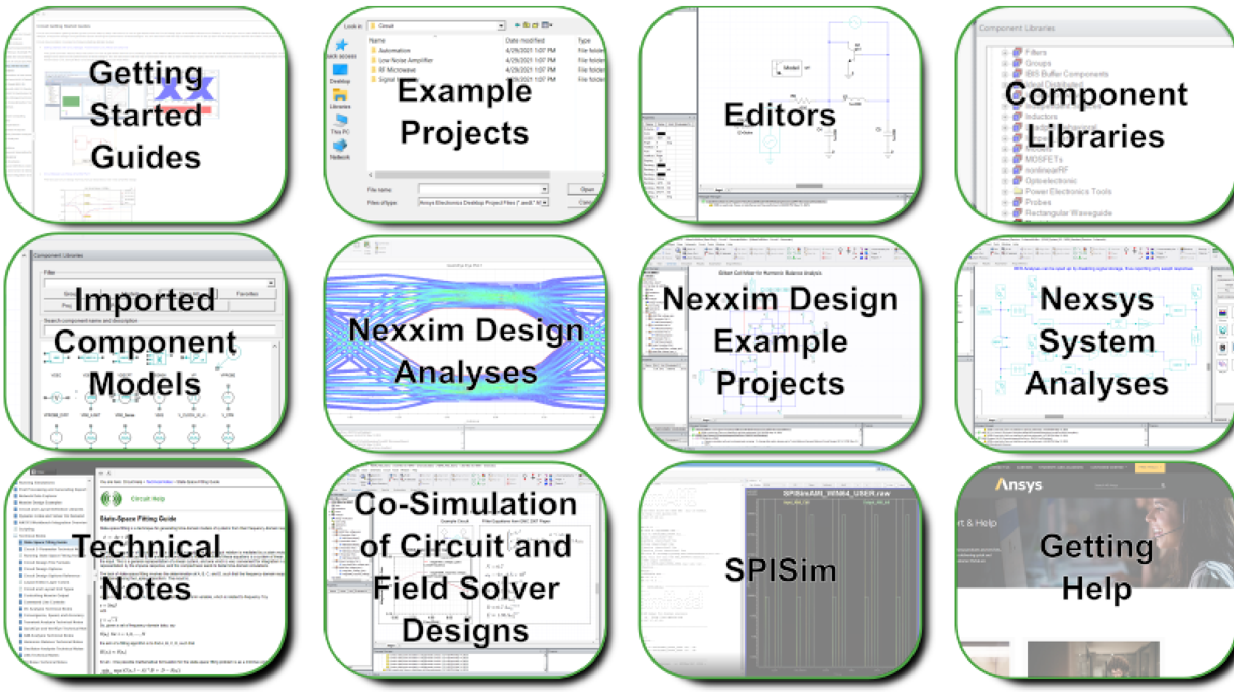
Release 2024 R2
July 2024

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001:2015 companies.

Table of Contents

1 - Welcome to Circuit Help

Ansys Electronics Desktop Circuit Designs allow you to create electronic circuits using components and models from a variety of sources including behavioral system-level elements, simulate in the time and frequency domains, and generate meaningful plots and reports.



Nexxim and Nexsys Component Help

The Circuit Design help includes documentation in CHM format for the complete set of built-in Nexxim and Nexsys components. To access the component on-line help, click the **Help** item in the top menu bar and select **Circuit Components** from the drop-down menu.

Finding Information in the Online Help

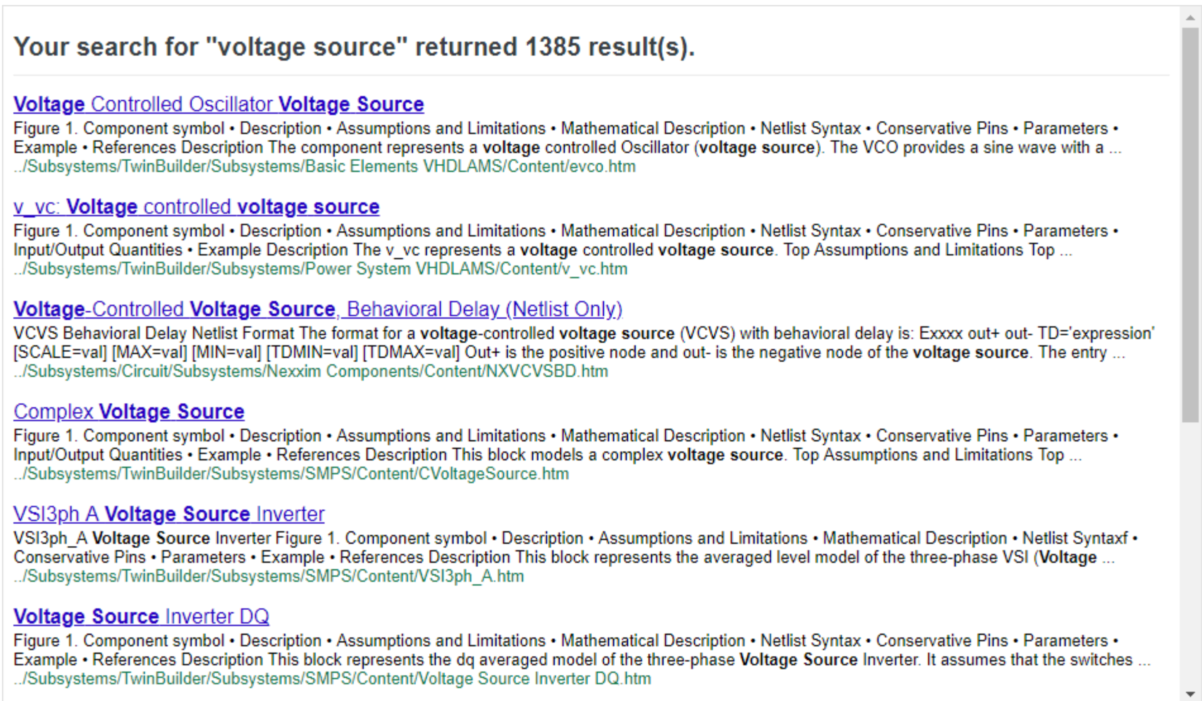
The help system provides different ways to find information and navigate quickly:

- 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

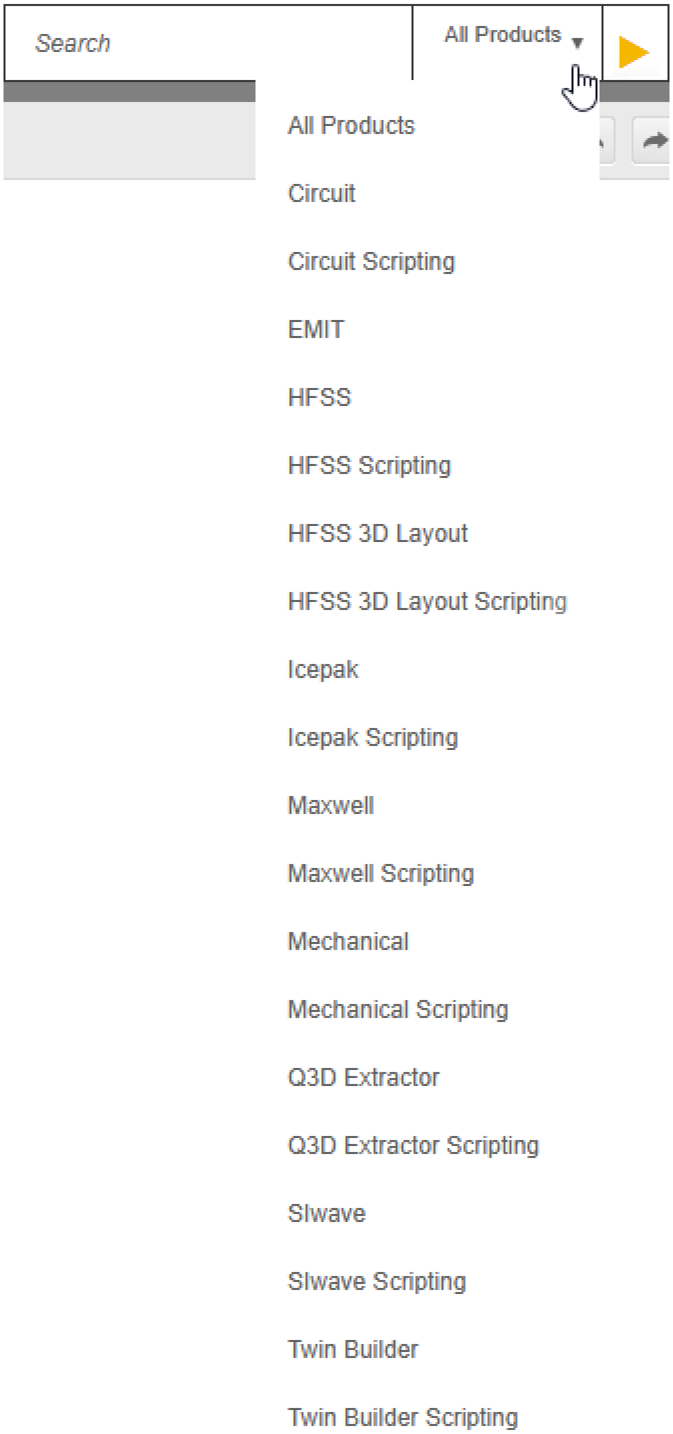
When you enter words or strings to search for in the help, the search engine lists all topics in which any of the words occur. For example, if you enter “voltage source” without the quotation marks, the results show all topics that contain “voltage” or “source.”



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.

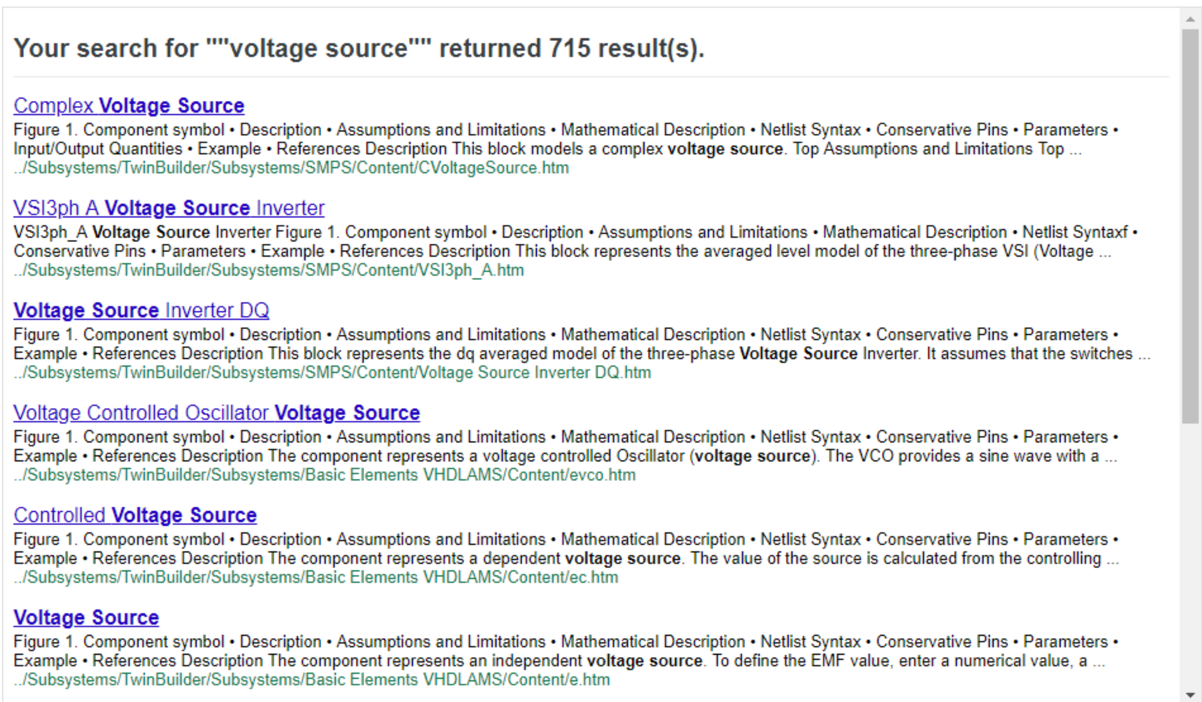


- If you accessed the help for a specific product by pressing **F1** in Electronics Desktop, the search box will have preselected that product. You can change the selection to a different product, a product's scripting guide, or all products.

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 ().

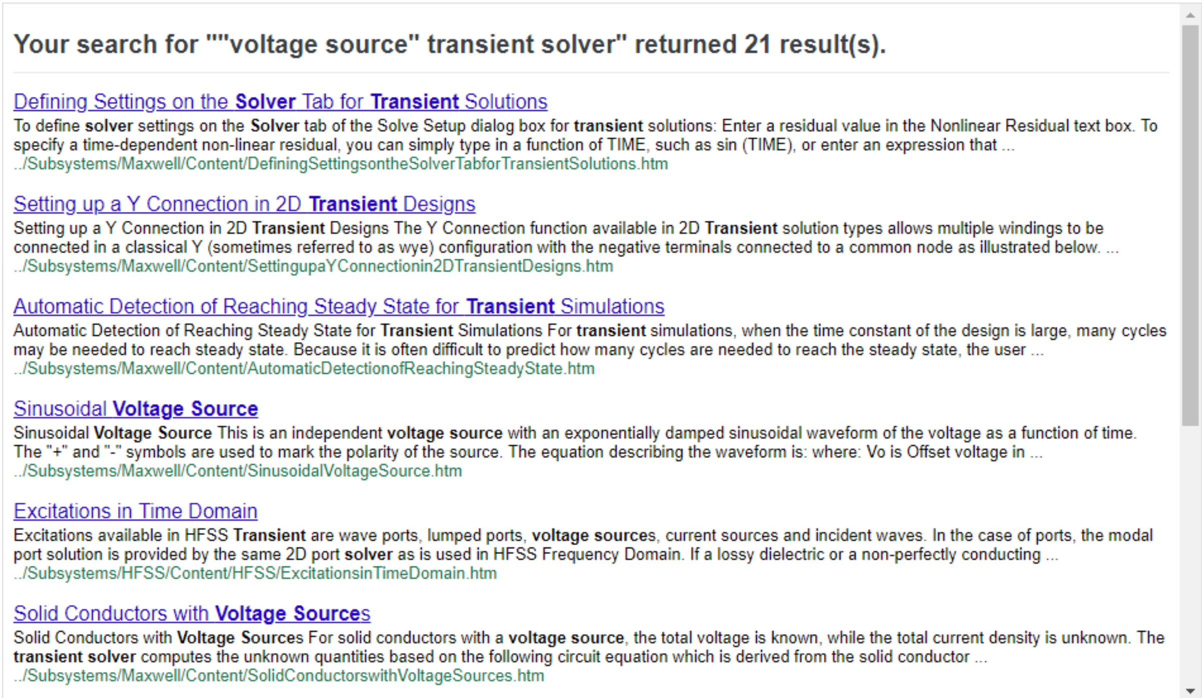
Searching with Quotation Marks

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.

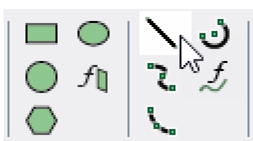
- Procedures are presented as numbered lists. A single bullet indicates that the procedure has only one step.
- Bold type is used for the following:
  - Keyboard entries that should be typed in their entirety exactly as shown. For example, “**copy file1**” means you must type the word **copy**, then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

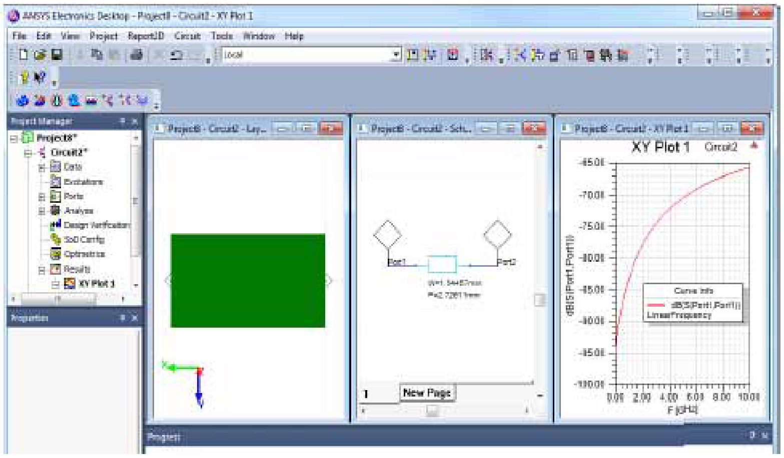
Circuit Getting Started Guides

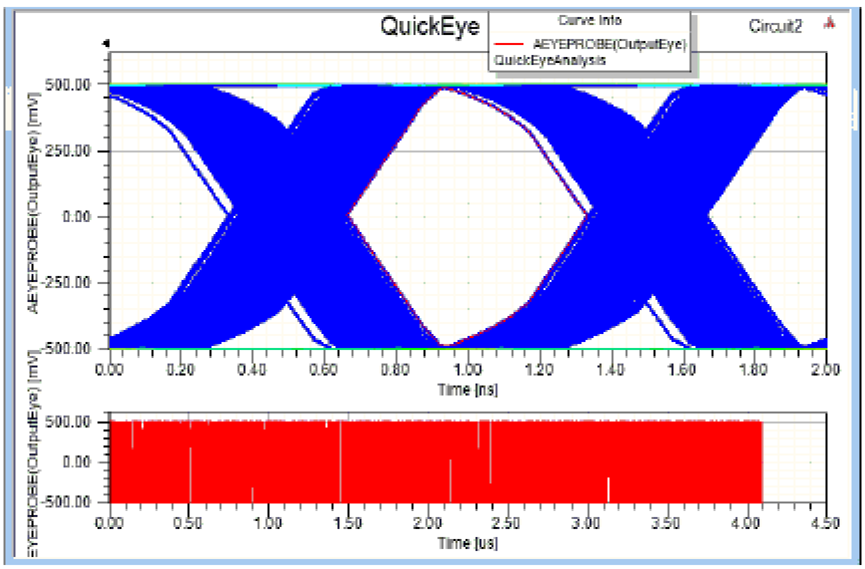
Circuit documentation getting started guides provide step-by-step instructions on how to get started with the Circuit Design type of the Ansys Electronics Desktop. You will learn how to start Ansys Electronics Desktop, build basic designs, analyze the designs and generate reports showing the performance of those designs. You will also learn with the help of examples how to set up each of the design types, start the simulation, and perform post processing.

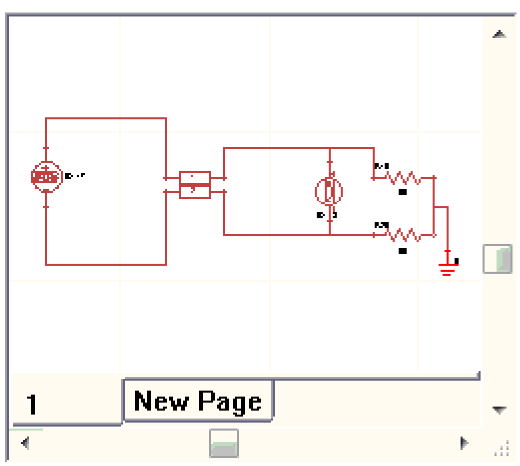
Circuit documentation includes the following Getting Started Guides:

- *Getting Started with Circuit Design: Transmission Line, Mixer and Channel*

  This guide provides step-by-step instructions on how to get started with the Circuit Design type of the Ansys Electronics Desktop. You will learn how to start Ansys Electronics Desktop, build basic designs, analyze the designs and report on the performance of those designs. You will also learn with the help of examples how to set up each of the design types, start the simulation, and perform post processing The examples include Basic Transmission Line, Simple Mixer and Simple Channel, and Low Pass Filter.

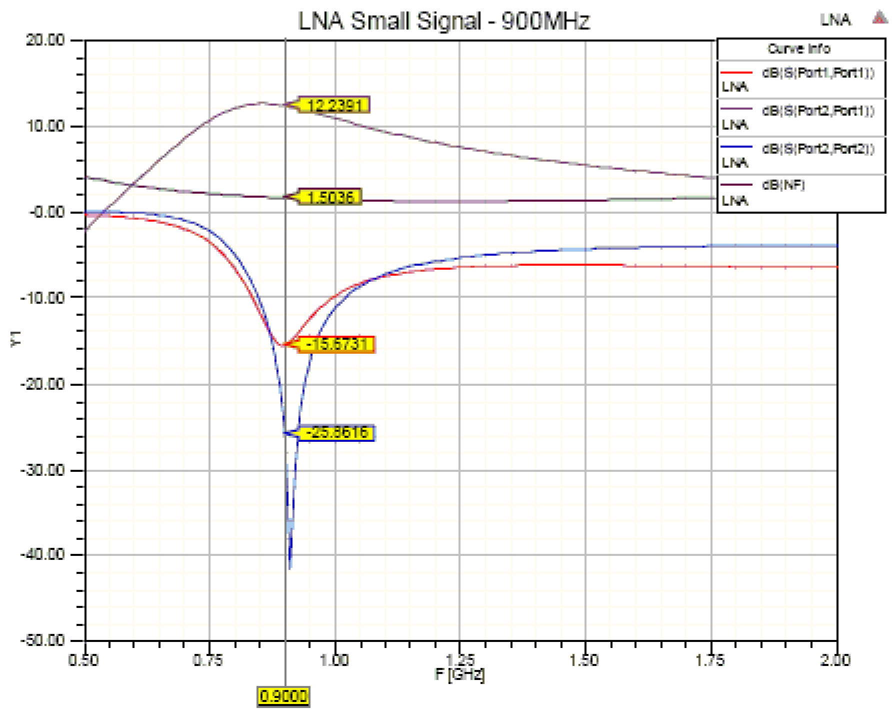



QuickEye
Curve Info
AEYEPROBE(OutputEye)
QuickEyeAnalysis
Circuit2
AEYEPROBE(OutputEye) [mV]
Time [ns]
Time [us]


1
New Page

- *Circuit Design Low Noise Amplifier Part 1*

  This two-part circuit design training manual describes a low noise amplifier design.



  Part 1 of the guide shows how to use Circuit Design in the Ansys Electronics Desktop to design a small signal 900 MHz low noise amplifier (LNA) using an s-parameter model of NEC BJT NE68133, including noise parameters. The document also shows how to synthesize matching networks using the built-in Smith Tool. Tuned circuits are connected to the input and output to provide matching, essential to finalizing the design of the low noise amplifier.

- *Circuit Design Low Noise Amplifier Part 2*

  Part 2 covers the following topics:

  - Circuit Simulation
  - Schematic Capture
  - DC Analysis
  - Linear (S-parameter) Analysis
  - Nonlinear Analysis
  - Single Tone
  - Compression

  - Multi Tone
  - Intermodulation
- *Getting Started Guide: Cable Modeling Solutions*

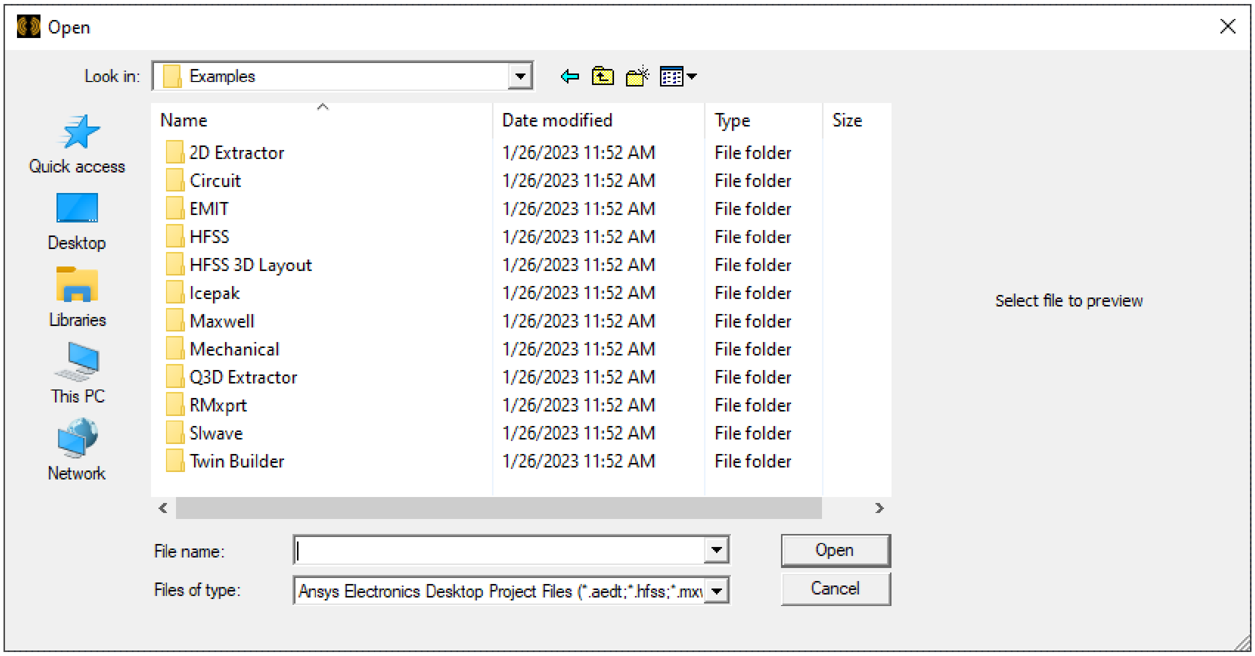
  The Ansys Electronics Desktop cable modeling solution is implemented using dynamic/data links between HFSS, 2D Extractor, and Circuit. A cable harness in HFSS is modeled as a single external field source based on quasi-static simulation of each cable cross-section in 2D Extractor and an analysis of the 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.

  The solution include the following steps and data transfers:

  - 2D Extractor
    - solve the cable cross-section
    - send a transmission line model for the cable network solution
    - send transmission line modes and fields for the 3D cable solution
  - Circuit
    - define a step voltage on the appropriate ports on the circuit schematic
    - solve the cable network
    - send v and i at the ends of the cable for the 3D cable solution
  - HFSS
    - map fields onto the 3D cable, and solve 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. The examples under Circuit, HFSS, and HFSS 3D Layout include subfolders, organizing the projects according to different applications:

- Circuit has subfolders for Automation, High Speed Digital, Low Noise Amplifier, RF Microwave, and Signal Integrity examples. The High Speed Digital folder has further subfolders for PCIe and USB projects. And the RF Microwave folder has further subfolders for Amplifiers, DC-IV, Filters, Misc, Mixers, Oscillators, and System projects.
- HFSS has subfolders for ADAS, Antennas, EMI EMC, Filters, Metamaterial, RCS, RF Microwave, Signal Integrity, and Transmission Lines.
- HFSS 3D Layout has subfolders for Antennas, Component, Filters, FSS, and Signal Integrity.

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

Many other projects include descriptions in the help.

- Circuit Example Projects

Circuit Design Example Projects

The following example project subdirectories can be found in the C:\Program Files\AnsysEM\v###\Win64\Examples\Circuit folder, where v### represents your download version, such as v241 for version 24.1.

- **Automation** - contains example scripts to create models of the user's choice from within a directory. Additionally the user may choose to enable passivity and causality enforcement.
- **High Speed Digital** - USB and PCIe examples. A technical note PDF will guide you through the methodology of creating circuit schematic and simulating time/frequency domain waveforms for the chosen emitter model.
- **Low Noise Amplifier** - Contains a two-part guide to show you how to use Circuit to design a small signal 900 MHz low noise amplifier (LNA) using an s-parameter model of NEC BJT NE68133, including noise parameters, and how to synthesize matching networks using the built-in Smith Tool.
- **Power Electronics** -
  - 3-Phase Inverter Conducted Emissions Example: demonstrates the use of the space vector modulation block and how to set-up a complete simulation environment for power application purposes.
  - Semiconductor Characterization: Double Pulse Test and UARK Model Extraction Examples. Relevant PDF guides are found within their subdirectories.
- **RF Microwave** - Amplifiers, DC-IV, Filters, Misc, Mixers, Oscillators, and System projects.
- **Signal Integrity** - Contains various signal integrity example projects. Notes for each project can be accessed after launching the model. In the **Project Manager** window, expand the project components and select **Notes**.

Circuit Design Editors

The Electronics Desktop utilizes both a **Schematic Editor** and a Netlist Editor that allow you to create and view designs using industry-standard tools, components, and models.

Schematic Editor

Most circuit designs in **Electronics Desktop** use the Schematic Editor. The **Schematic Editor** allows you to create multi-level designs that combine built-in components, static models, and dynamically linked projects. The editor is an industry-standard tool for creating and viewing designs under the Nexxim circuit simulator.

The Schematic editor allows you to insert components and models from a library using “drag-and-drop,” adjust component instance and model parameter values, and wire the components together. The editor provides built-in support for ports and grounds. Wire names (node, bundle, and bus names) and component names (reference designators) can be generated automatically or manually in a variety of ways. The **Schematic Editor** has built-in support for hierarchical designs with multiple subcircuits and multiple schematic pages.

Netlist Editor

Alternatively, **Electronics Desktop** supports designs created with the Netlist Editor or saved as netlists from schematic designs. automatically generates a netlist from any schematic (Nexxim Circuit) design. Simulation is actually run on the netlist description using simulation controls that are set up on the Schematic editor.

The Netlist editor allows you to view the netlist that has been generated for a schematic design. The Nexxim simulator also allows direct entry of a netlist. The Nexxim netlist simulation uses simulation controls that are written directly into the netlist. This feature allows for the import of netlists in HSPICE™ format from third-party design capture tools for simulation by Nexxim. The editor uses a color-coding scheme to highlight the various elements of the netlist, such as components, options, and numerical values. The Nexxim netlist format is HSPICE™-compatible, so any industry-standard system-design netlist can be easily imported for analysis.

Circuit Design Component Libraries

Built-in Circuit components include a wide range of active, passive, and distributed device models from transistors to transmission lines, including sources and probes. To support system-level mixed simulations, Circuit designs can include built-in system-level behavioral models including mixer and amplifier models. All the built-in components are easily accessible from in the Schematic Editor.

Imported Circuit Design Component Models

A Circuit design can also import models of components in common formats, and place multiple instances of the modeled components in the schematic. Imported models are “static,” in that they do not change in response to circuit changes. Importable Circuit models include SPICE/PSPICE libraries, Verilog libraries, IBIS and IBIS-AMI libraries, SYZ-parameter models, W-elements, and X-parameter models.

Nexxim Circuit Design Analyses

Nexxim® is the Ansys state-of-the-art circuit simulator for RF, analog, and mixed-signal designs. Nexxim combines transistor-level simulation accuracy, high execution speed, and robust capability to handle circuits with thousands of active and passive elements. Nexxim simulates in all three domains — DC, time, and frequency — on the same set of device models.

- In addition to the classic variable sweeps of temperature, voltage, and current, Nexxim allows you to perform sweeps of any model or parameter in a circuit.
- Circuit designs can be simulated using a variety of time domain and frequency domain solvers, including system-level analysis with combinations of electrical and behavioral components.

Nexxim Time Domain analyses include DC analysis, transient analysis, statistical eye analysis with VerifEye and QuickEye, and IBIS-AMI analysis. Support is available for HSPICE simulation when that system is installed. The focus of the time-domain solvers is on signal integrity issues.

- DC analysis calculates the circuit's DC operating point and is the basis for other analyses.
- Transient analysis calculates the time-domain behavior of the circuit. Nexxim transient analysis incorporates a new, Ansys-proprietary variable time-step integration formula and advanced error estimation and control. The analysis can include S-parameter input data files for accurate analysis in the time domain.

Circuit time domain analyses can yield time-voltage plots, eye diagrams, and tabular data arrays of the time domain simulation results. Information on Circuit design reports is included in the analysis documentation.

Nexxim Frequency Domain analyses include harmonic balance, oscillator analysis, envelope analysis, time-varying noise analysis, and linear network analysis.

- Harmonic balance analysis calculates the frequency-domain responses of the circuit. Nexxim harmonic balance performs single-tone and multi-tone analysis that combines Krylov subspace methods with Ansys-proprietary preconditioning algorithms to achieve fast convergence.
- Small-signal analyses (AC, linear network, DC noise) are based on the DC solution. The Nexxim proprietary linear system solver is optimized for circuit simulation, and produces accurate answers without compromising speed. Nexxim also supports small-signal frequency analysis based on harmonic balance.

Frequency domain analyses can generate spectral plots, constellation diagrams, and Smith charts. Information on Circuit design reports is included in the analysis documentation.

Nexxim Circuit Design Example Projects

The Electronics Desktop comes with Circuit examples to demonstrate Nexxim time-domain analyses, Nexxim frequency-domain analyses, Nexsys analyses, and selected components on the Nexxim Component Library. For more information, see Nexxim Design Examples.

Nexsys System Analyses

In a Nexsys System design, functional and electrical components and sub-designs may be connected arbitrarily. Nexsys analysis seamlessly integrates with Nexxim Transient analysis and Envelope analysis to solve mixed-mode design problems at differing levels of abstraction.

Nexsys System simulation include both time-domain and frequency-domain analyses. Nexsys Discrete Time-Domain Analysis allows the simulation of arbitrary wired/wireless communications system topologies and other system-level applications. System frequency domain analysis allows you to evaluate and troubleshoot RF/high-frequency designs for wireless and microwave applications.

Technical Notes for Circuit Design

Technical notes provide background and reference material for the Circuit features. Technical notes that apply to one solver or component are placed in the relevant topic. Notes that apply more generally are located in Circuit Design Technical Notes.

Co-simulation of Circuit and Field Solver Designs

The Electronics Desktop features Dynamic Links and Solver on Demand™ technology to allow flexible and accurate modeling of unrestricted structures used in high-frequency system designs of components and circuits. Dynamic Links allows the Circuit project to incorporate projects from HFSS, Q3D, 2D Extractor, and SIwave as active components. Use Solver on Demand technology to simulate structures via circuit, system, and field solver models at the touch of a button. Solver on Demand incorporates electromagnetic solver results directly into larger network topologies and allows you to maximize both speed and accuracy. Solution data is cached and used in place of re-simulating unless you change structure parameters.

For more information see Dynamic Links and Solver On Demand.

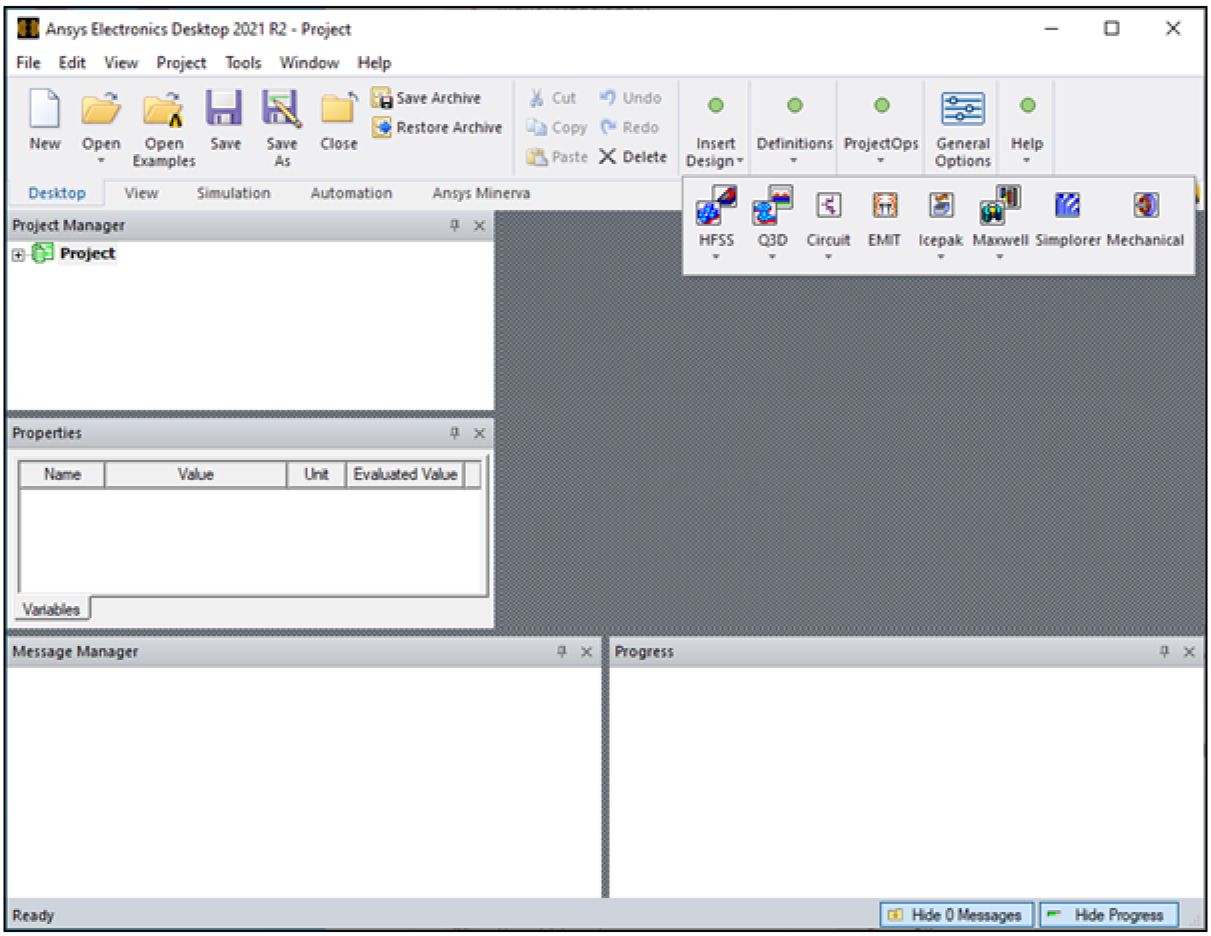
Getting Help from Ansys Technical Support

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:

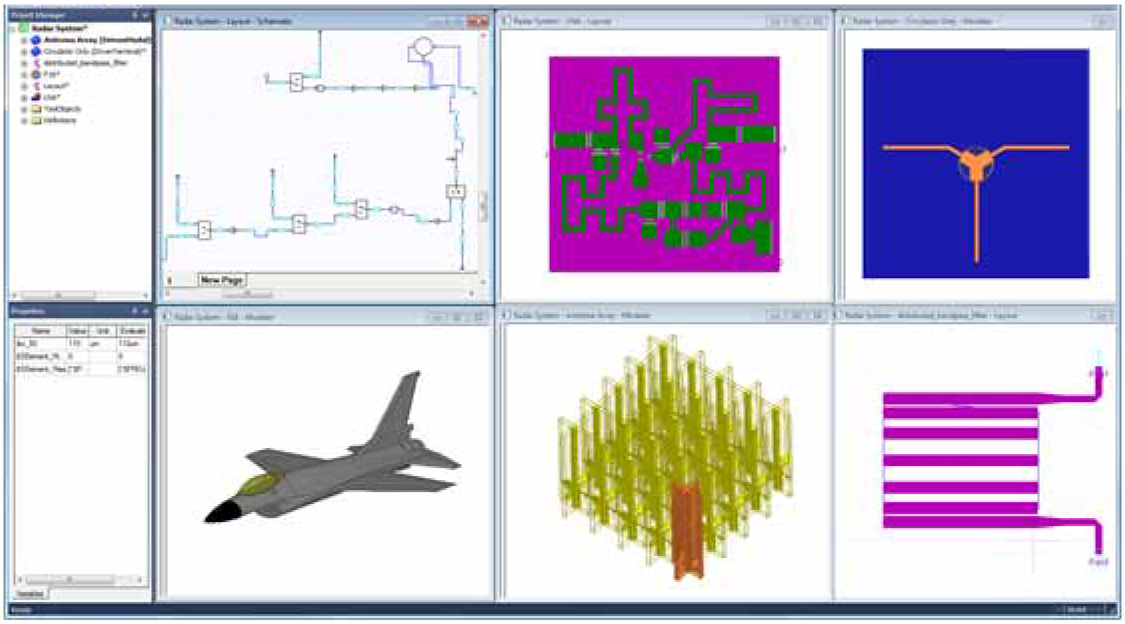
- **HFSS** – a general purpose 3D interface for the design, analysis, and simulation of electromagnetic components.
- **HFSS 3D Layout** – a full-wave, layout-based electromagnetic simulator with a specialized interface for geometries created in layout.

- **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 2024 R2 **- 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:

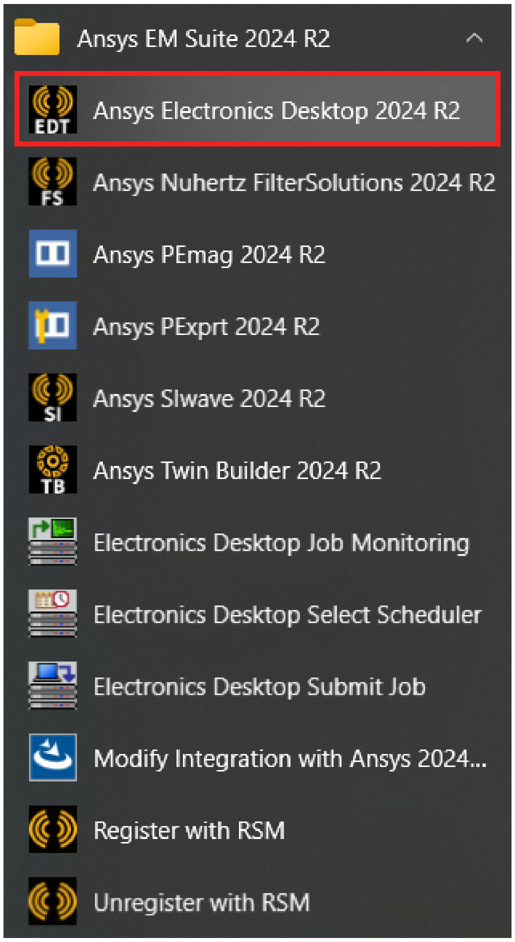
Windows vs. Linux Installations

Launching Ansys Electronics Desktop

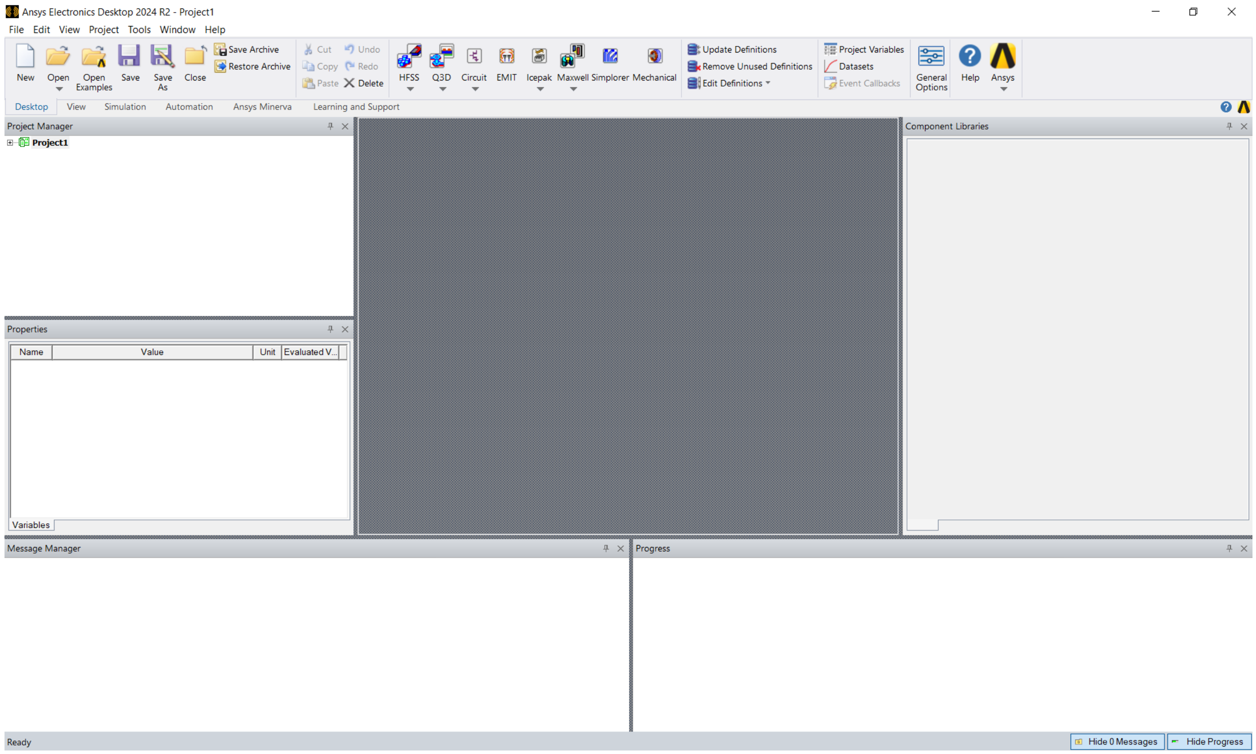
Once you have installed Ansys Electronics Desktop, start the program using one of the following methods:

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

- From the Windows Start menu, select **Ansys EM Suite 2024 R2** > **Ansys Electronics Desktop 2024 R2**.



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 the Ansys Electronics Installation Guide. If you installed Ansys Electronics Desktop in the C:\ drive of a Windows machine, the installation guide is available at C:\AnsysEM\v242\Win64\Help\.
- 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 ElectronicsDesktop2024.2 -
> 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?*

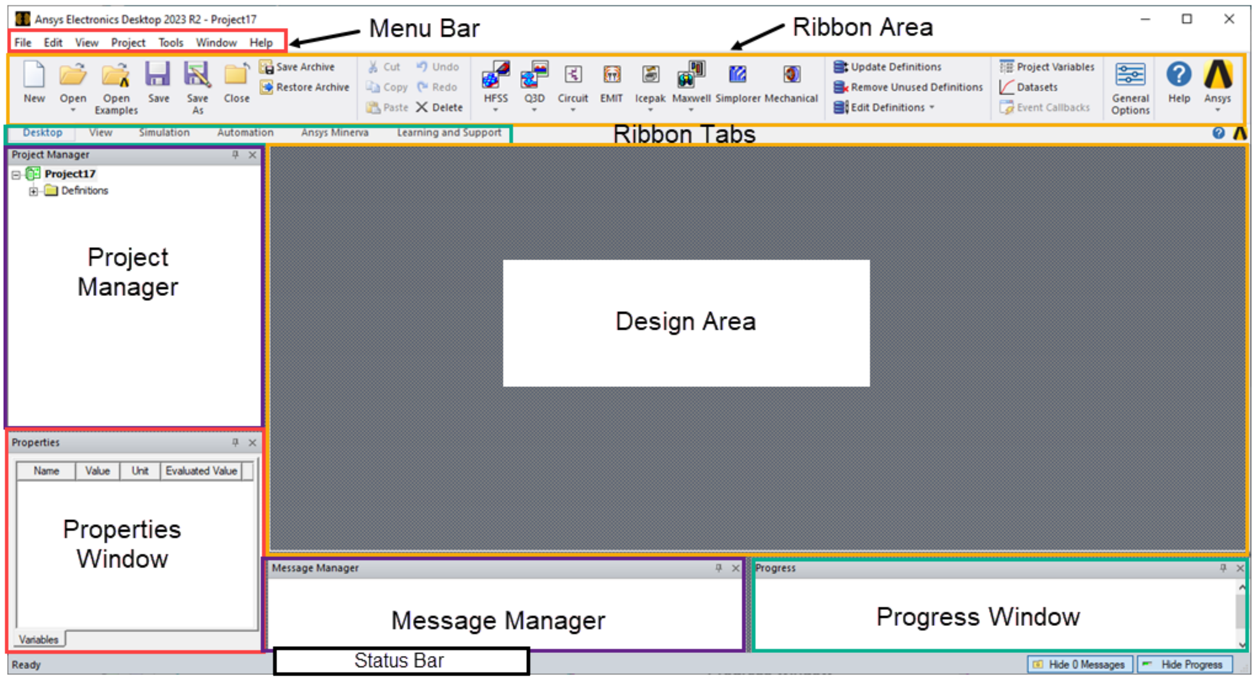
   Yes. Your enrollment choice applies to all Ansys products you use on the same machine. Similarly, if you end your enrollment in the program for one product, you end your enrollment for all Ansys products on that machine.

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.

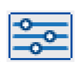
Ansys Electronics Desktop Overview

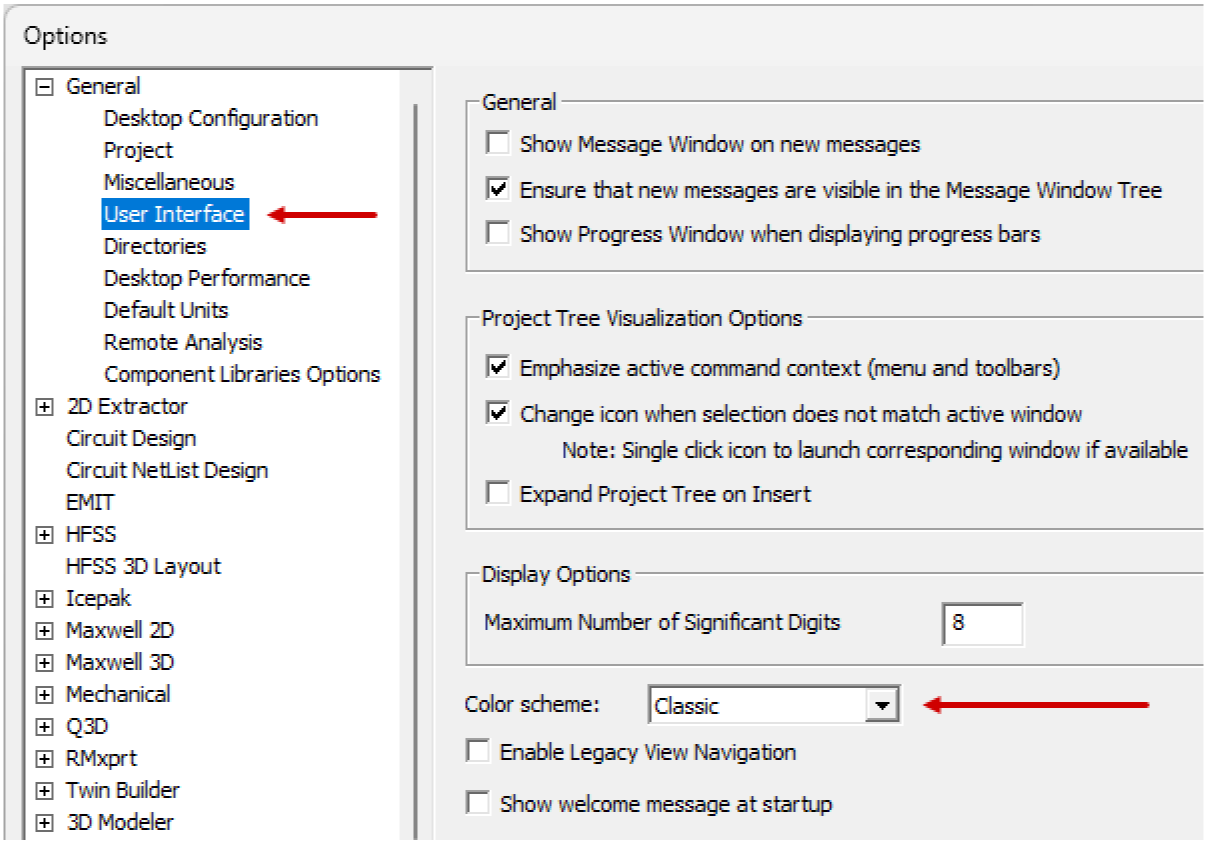
The Ansys Electronics Desktop consists of several windows, a menu bar, multitab ribbon, and a status bar. You can customize the appearance of the desktop by moving, hiding, or showing windows. Some menus, and other features, change depending on the type of project that is loaded and the editor that is active in the Design Area.



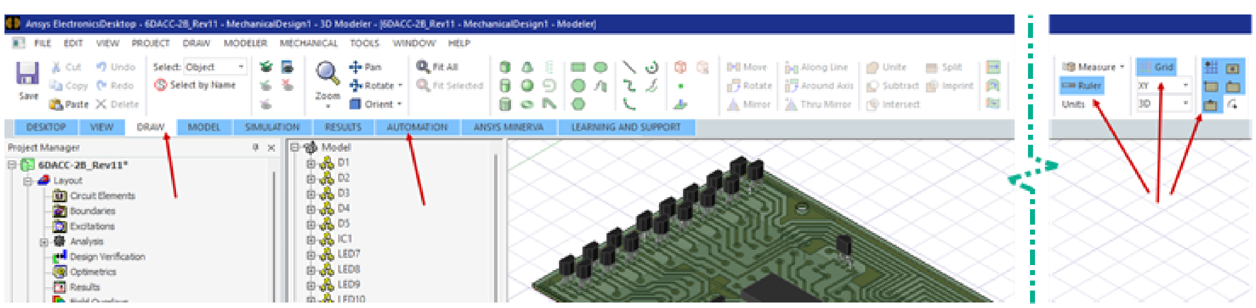
Choosing a Color Scheme [Beta]

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:

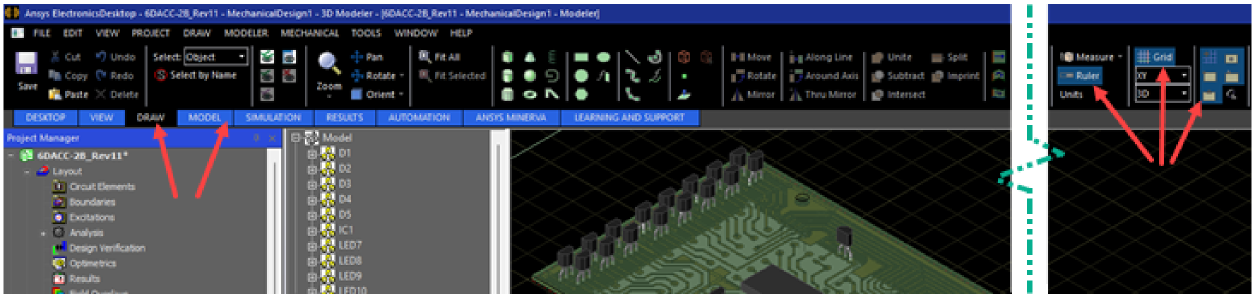
1. Access the *Options* dialog box using one of the following two methods:
   - On the **Desktop** ribbon tab, click  **General Options**.
   - From the menu bar, click **Tools** > **Options** > **General Options**.
2. In the tree at the left side of the dialog box, expand **Desktop Configuration** and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



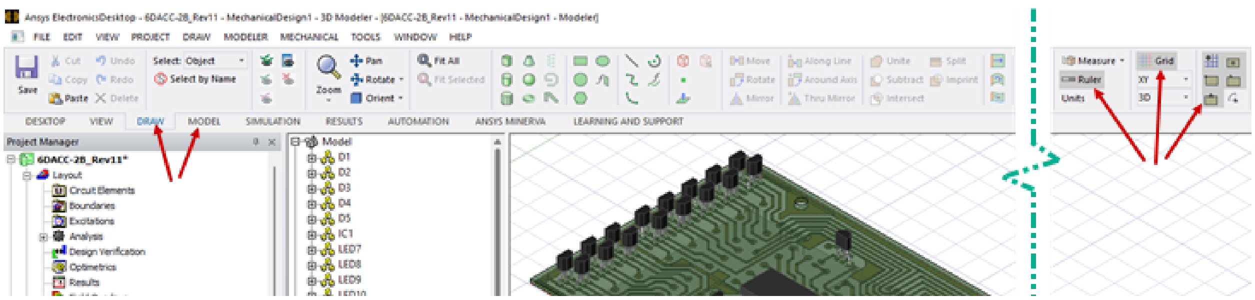
3. From the **Color scheme** drop-down menu, choose one of the following three options:
   - **Light (beta):** A light gray and white scheme with light blue highlighting for selected ribbon icons and for inactive ribbon tabs. The active ribbon has a white background (including the tab):



   - **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):



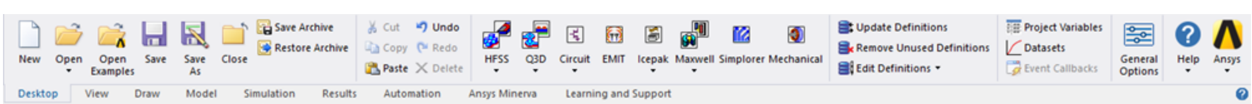
- **Classic:** A light gray and white scheme with medium gray highlighting for selected ribbon icons. The ribbon background is light gray. The name of the active ribbon tab has a blue font. A dark gray font is used for the inactive ribbon tab names:



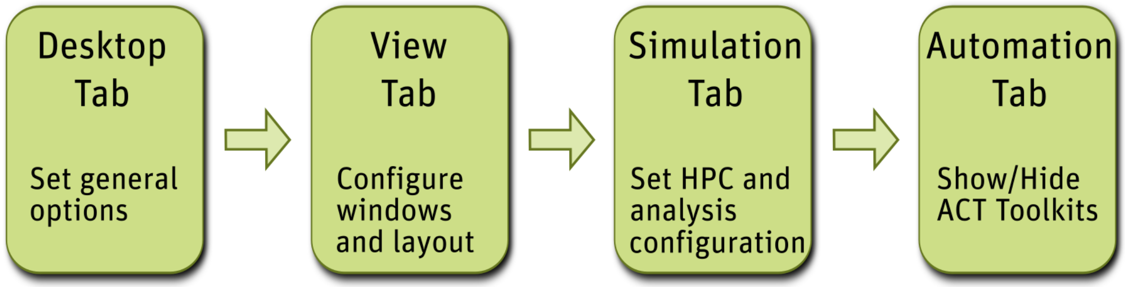
4. Click **OK** to close the *Options* dialog box.

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.



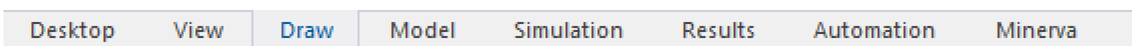
The initial workflow for local configuration and personal customization is:



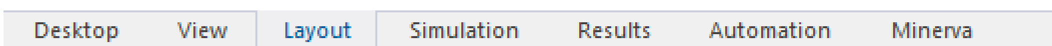
Available ribbon tabs depend on the design type. After you open or add a project and insert a design, you see additional ribbon tabs appear. The visible tabs and features are those that are appropriate for the design type and solver.

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

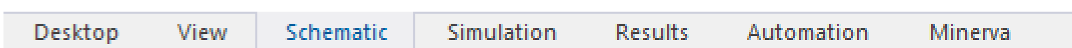
- For HFSS, Icepak, Maxwell, Mechanical, and Q3D designs, the **Draw**, **Model**, and **Results** tabs also appear:



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



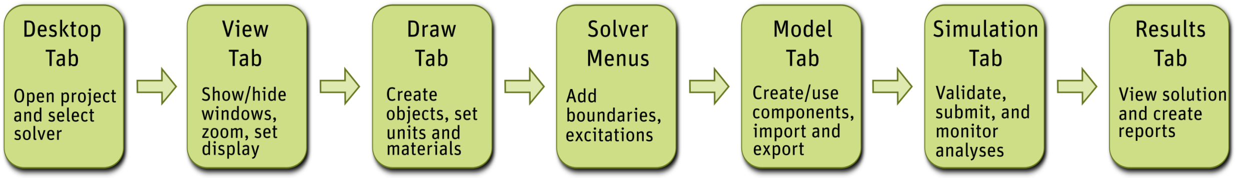
- For Circuit and Simplorer designs, the **Schematic** and **Results** tabs also appear:



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

- **Schematic Tab** – contains tools that allow you to place components, ports, connectors, and wires.
- **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), install PyAEDT (Beta), and access toolkits.

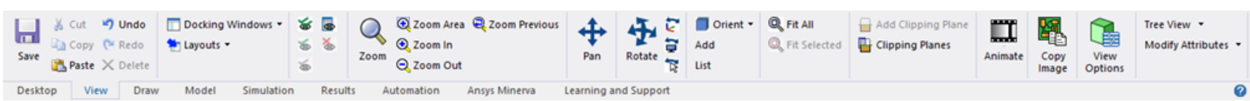
Each tab contains features specific to the design type, and the general workflow is from left to right.



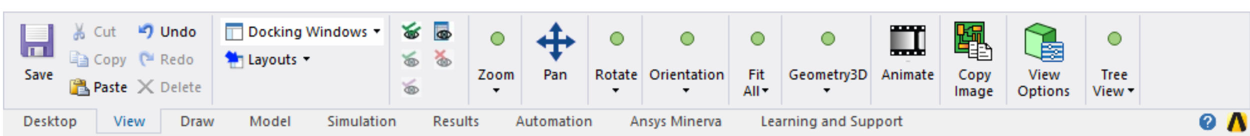
Note that workflow can vary depending on tasks at hand. For example, if you open a completed model in HFSS or use 3D components or the ACT Antenna Design toolkit (Windows only), you may not need the features of the **Draw** tab. If Validation on the **Simulation** tab identifies problems in a model, you may need to use the **Modeler** menu commands for Analysis and Heal, or use the SpaceClaim 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:

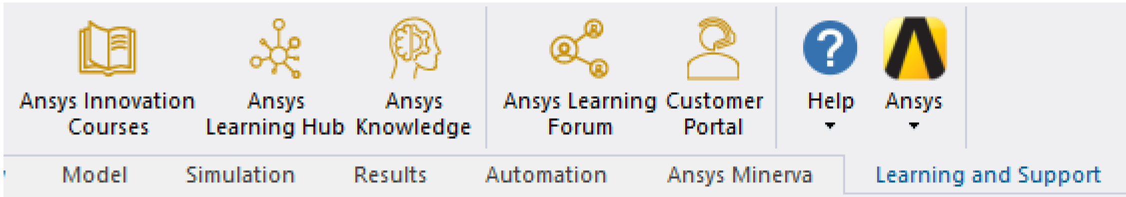


If you reduce the size of the Desktop window, the ribbon tabs become compressed. Fewer icons are shown, and available features are moved into drop-down menus rather than shown separately. The following example shows the **View** tab in a compressed state:



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.

Ansys Electronics Desktop Windows

Ansys Electronics Desktop contains the following windows:

- Project Manager
- Message Manager
- Progress
- Properties
- Layers
- Nets
- Components

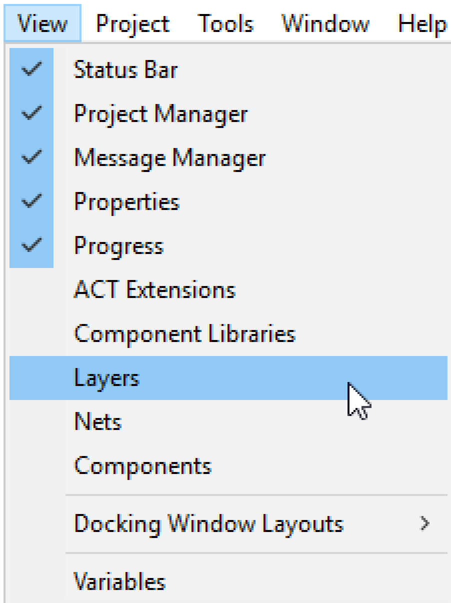
You can toggle the display of each window, auto-hide windows, or move and resize windows. Additionally, you can toggle the status bar or add ACT Extensions (Windows only).

Showing and Hiding Windows

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

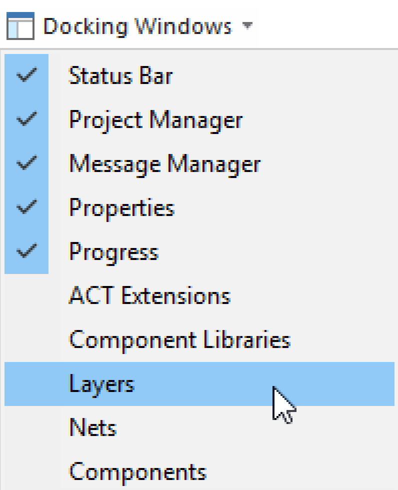
From the **View** menu:

- Click a window to show or hide it.



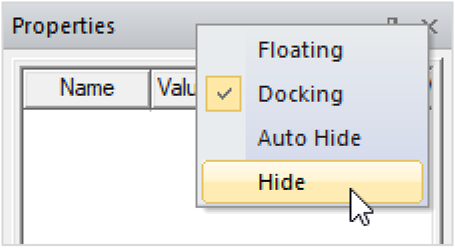
From the **View** tab:

- 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

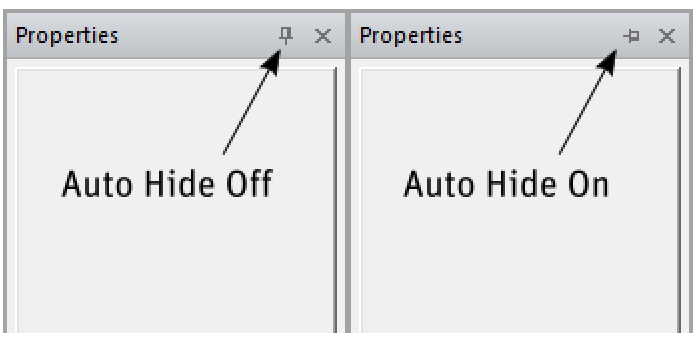
Windows can be moved out of the way so that only their title bars show. When you hover over this title bar, the window appears. This mode is called Auto Hide.

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**.

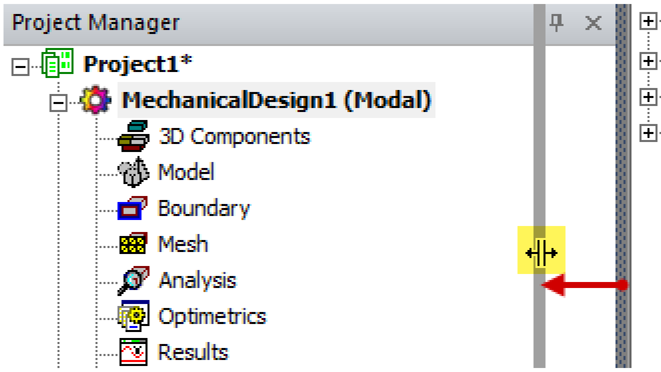
> **Note:** If a window shares vertical or horizontal space with another window, the first click will expand it to take up the entire area while hiding all other windows. Click the pin again to hide the window.

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

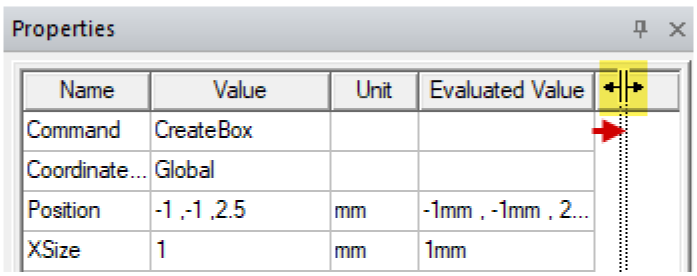


Moving and Resizing Windows

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):

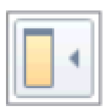


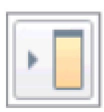
You can also click and drag the borders of columns within certain windows containing tabular data, such as the **Properties** window:

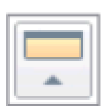


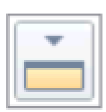
Major Ansys Electronics Desktop windows can be docked in various locations or can float over other windows. To move a window, click and drag the window's title bar. While dragging a window's location, several docking position icons appear on the screen.

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:

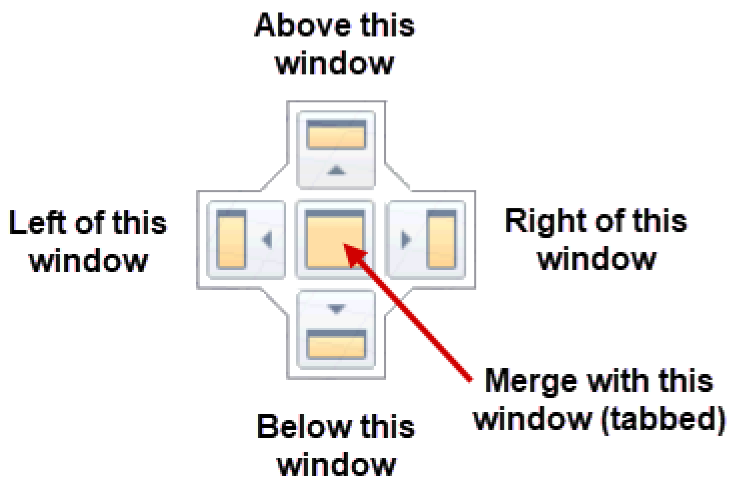
Left edge of user interface: 

Right edge of user interface: 

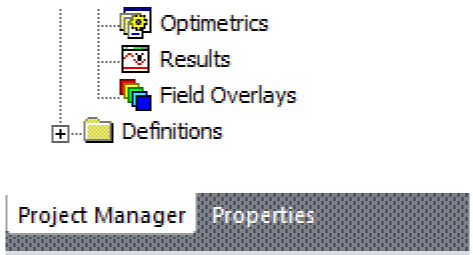
Top of user interface: 

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:

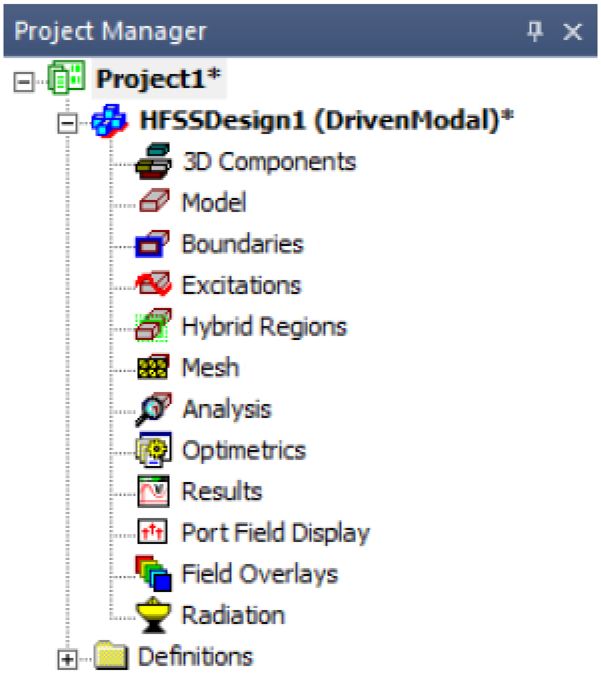


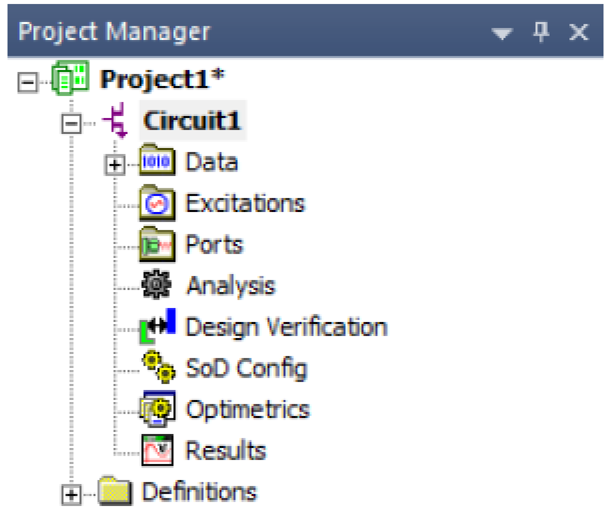
If the middle icon does not appear, it's because tabbed windows are not supported by one of the windows. For example, you cannot merge any window with the Modeler window.

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 a Circuit design, the project includes data, excitations, ports, analysis setup, design verification, SoD configuration, Optimetrics, results, and relevant definitions.

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.

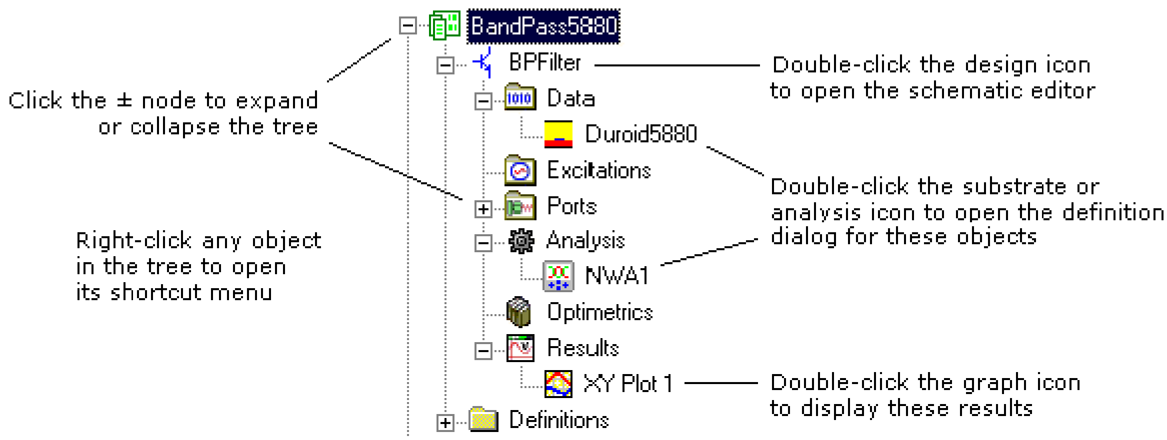
- Right-click in the toolbars area on the desktop, and then click **Project Manager** on the shortcut menu.

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

Project Manager Window for Circuit Designs

The Project Manager window of the Ansys Electronics Desktop for circuit designs shows the projects loaded into the desktop. Each project may consist of one or more product designs.

Each design's model data, ports, excitations, solution setup, and results are displayed as entries in a separate subtree on the **Project** tab in the **Project Manager** window of the Ansys Electronics Desktop.



To help facilitate the easy copying of design materials, Ansys Electronics Desktop allows you to drag and drop designs and components from the **Project Manager** window to the **Schematic Editor**.

- You can drag and drop one or more copies of a component from the Components folder in the project tree to a schematic. Alternately, you can select Copy from the **Project** tree right-click menu and then select Paste from the schematic right-click menu (or use the Ctrl+C and Ctrl+V keyboard commands).
- If **Multiple Placement** is turned on for components in Ansys Electronics Desktop's Schematic Options dialog box, you can place multiple instances of a component in the schematic by clicking at multiple locations. To stop placing components during multiple placement, do either of the following:

  — Press **Enter**, the spacebar, **Backspace**, or **Esc** on your keyboard.

  — Right-click, and then select **Place and Finish**, **Finish**, **Cancel**, or **Back**.

Project Manager Tree Organization

Information contained in the design is organized into folders in the Project manager tree:

- **Data**—Substrate data, library references, netlist fragments, and so on
- **Excitations**—Sources in the circuit
- **Ports**—Ports in the circuit
- **Analysis**—Setup information to perform various analyses
- **Optimetrics**—Setup information to perform optimization, statistical analysis, and so on
- **Results**—Collection of graphs and tables for the design
- **Definitions**—The collection of components, symbols, footprints, and so on, accessible from Ansys Electronics Desktop

Double-clicking an item in these folders typically opens a dialog used to set the properties of that item.

Right-click Shortcut Menu

Right-clicking a folder or item pops up a shortcut menu that allows you to perform various operations. Virtually all project editing and management can be done from the **Project Manager** window using the right mouse button shortcut menus. Many of these menus are also available from the main menu bar in the various Ansys Electronics Desktop **product** menus.

Dockable Project Manager Window

The **Project Manager** window is a dockable window. It can be moved and sized any way you want it and it can attach itself to any edge of the Ansys Electronics Desktop.

Dragging and Dropping

To help facilitate the easy copying of design materials, Ansys Electronics Desktop allows you to drag and drop (or copy and paste) designs and components from the **Project Manager** window to the **Schematic Editor**. In particular, in the **Project** tab of the **Project Manager** window, you can select and drag:

- A *design* from the Project Tree to a schematic
- A *component* from the Definitions folder to a schematic

You can also drag and drop (or copy and paste) designs in the Project Window onto a Circuit Design icon in the Project window.

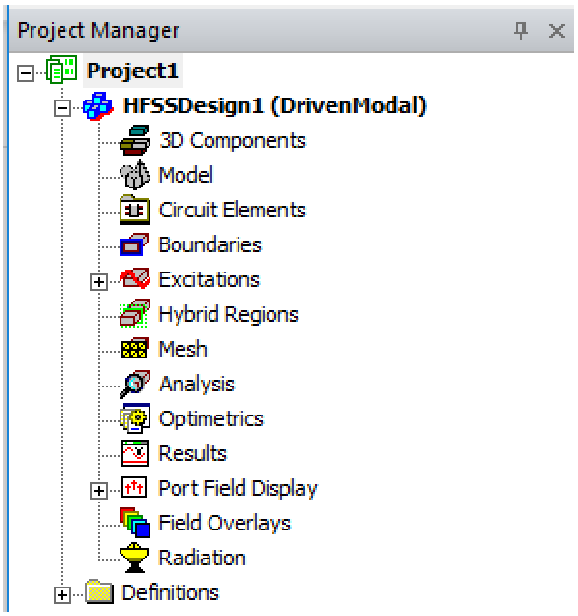
Dragging and dropping (or copying and pasting) an HFSS, Q3D, 2D Extractor, or SIwave design onto a Circuit Design or schematic creates a new dynamic link. Note If you hold the Ctrl key down while dropping or pasting, the appropriate Dynamic Link dialog will open, which allows you to make choices such as using an HFSS dynamic link in Transmission Line Model mode.

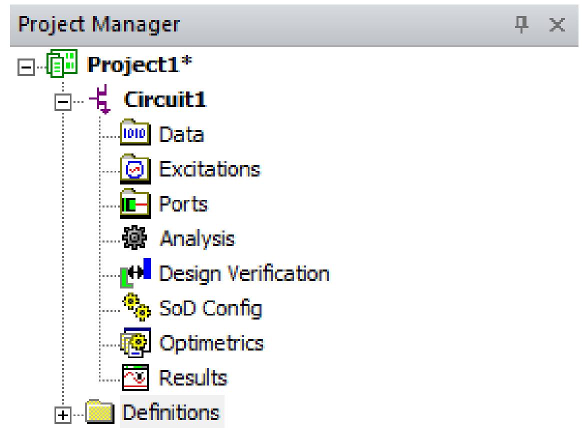
Closing the Project Manager Window

You can turn off the **Project Manager** window display by deselecting the **Project Manager** check box on the **View** menu in the Top Menu Bar. For more information, see the View Menu.

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).

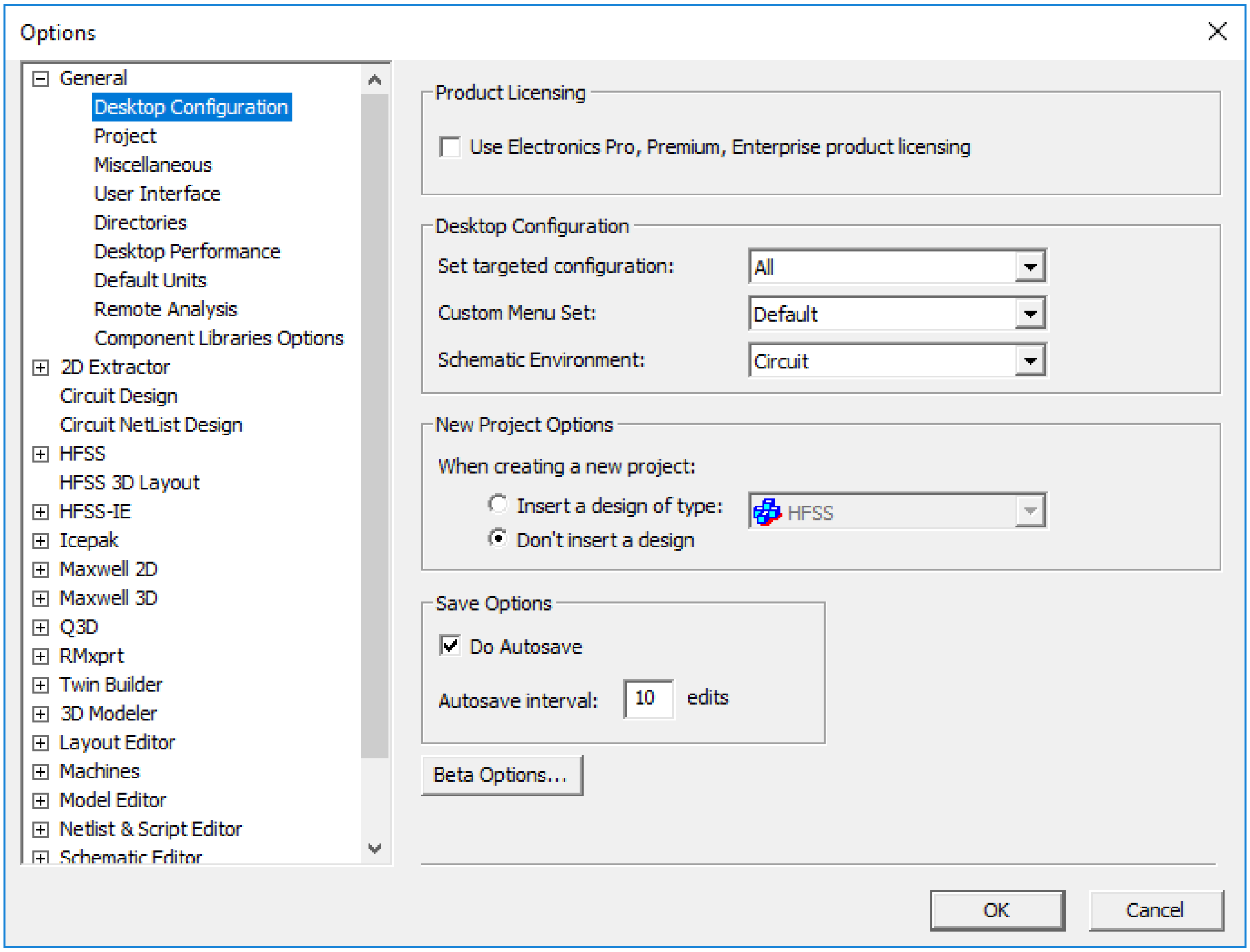
By default, the Project tree icon for the active window is highlighted. See General Options: Miscellaneous for options.

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**.

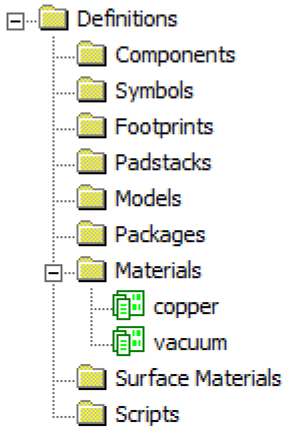
   The **Options** window appears.



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.
Excitations	Displays the excitations assigned to an Ansys Electronics Desktop design, which are used to specify the sources of electromagnetic fields and charges, currents, or voltages on objects or surfaces in the design.
Hybrid Regions	Displays the hybrid regions defined for an HFSS design, which can be FE-BI, IE Region, PO Region, SBR+ Region, and Dielectric Cavity.
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.
Port Field Display	Displays all port fields in the active model.
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.
Radiation	Displays far- and near-field setups added to an Ansys Electronics Desktop design.
Documentation	Displays files you have added as documentation.
Definitions	Displays definitions for Circuit, 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.

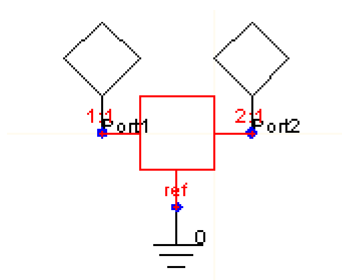
2. Expand **General** and click **User Interface**.
3. In the **General** area, check the **Ensure that new messages are visible in the Message Window Tree** check box.
4. Click **OK**.

Action Messages

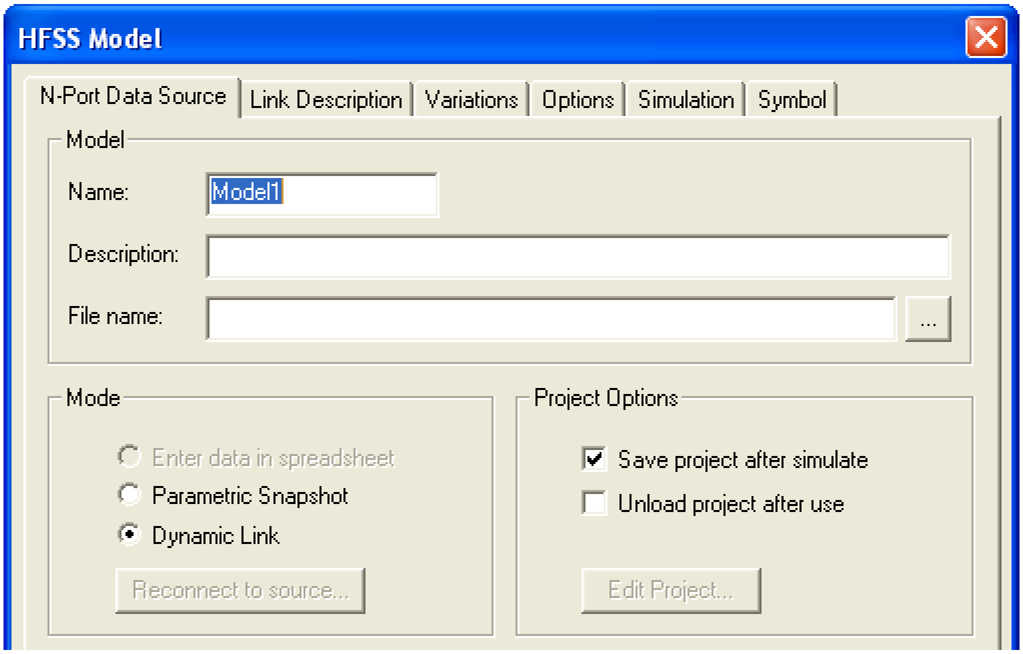
Messages displayed in the **Message Manager** can be associated with actions that can be invoked in order to address a condition you are alerted to by the message. If a message has an associated action, a magnifying-glass icon will be present to the left of the message icon. When the cursor moves over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

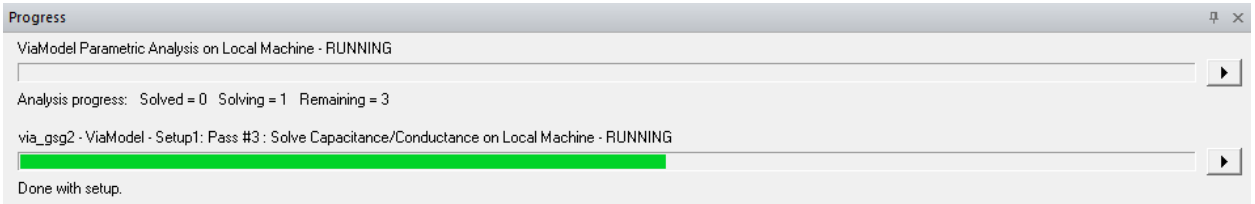
If you prefer to hide the Message Manager until a message is added:

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.

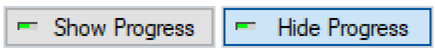
Progress Window

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 :

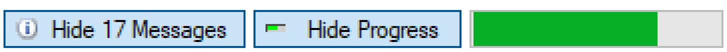
- Click **Show Progress** or **Hide Progress** on the status bar:



- 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.

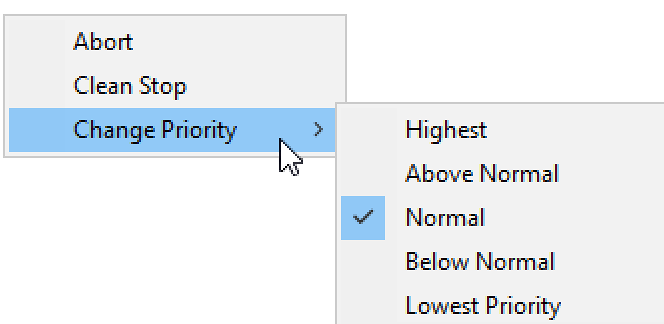
When more than one progress bar is active, the top progress bar is represented on the status bar with a progress indicator.



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

Stopping or Aborting Simulation Progress

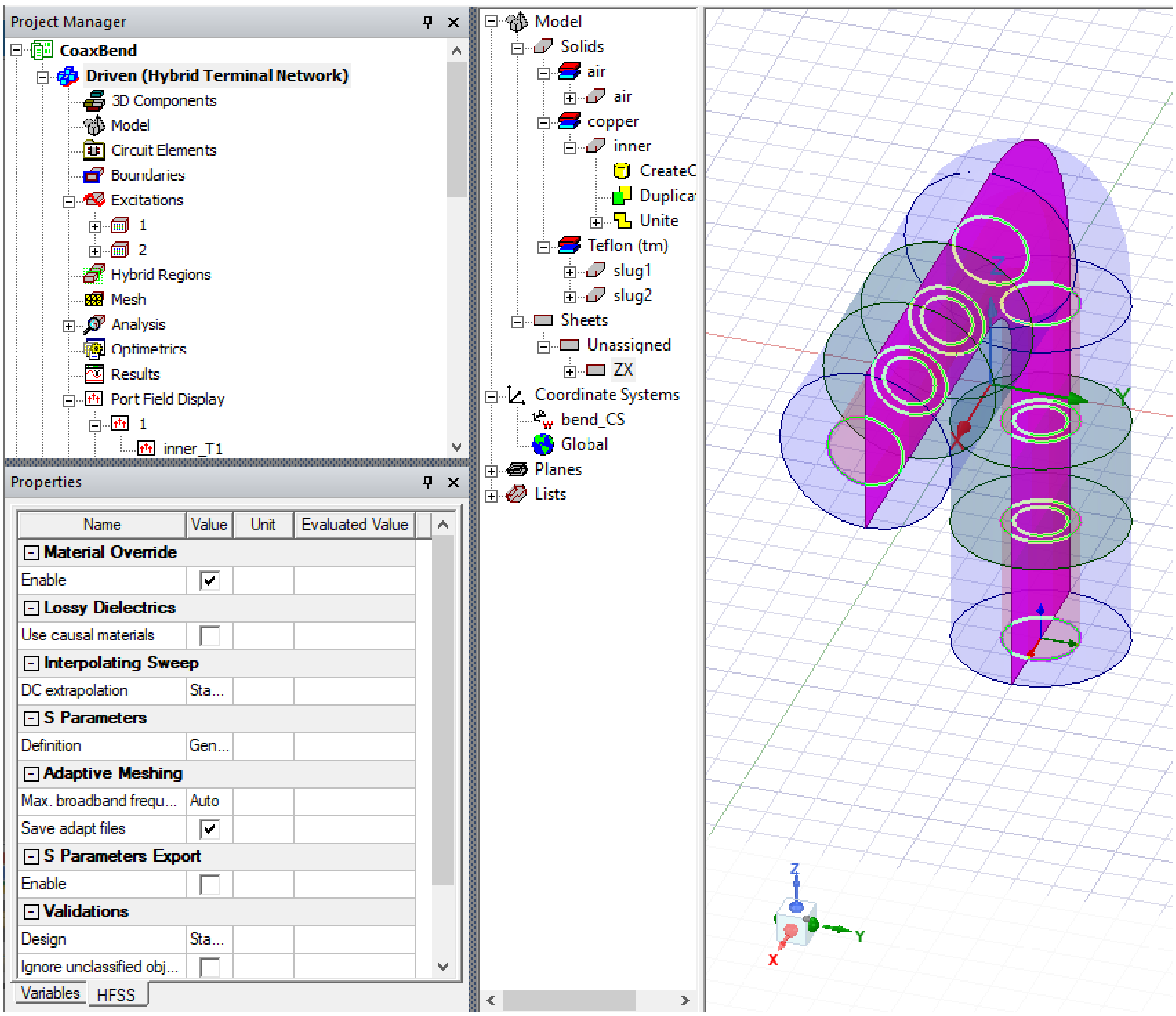
- To abort progress, right-click in the **Progress** window and select **Abort**.
- To stop the simulation cleanly between time steps, right-click in the **Progress** window and select **Clean Stop**.
- To change priority, right-click. From the **Change Priority** submenu, select from Highest, Above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



Properties Window

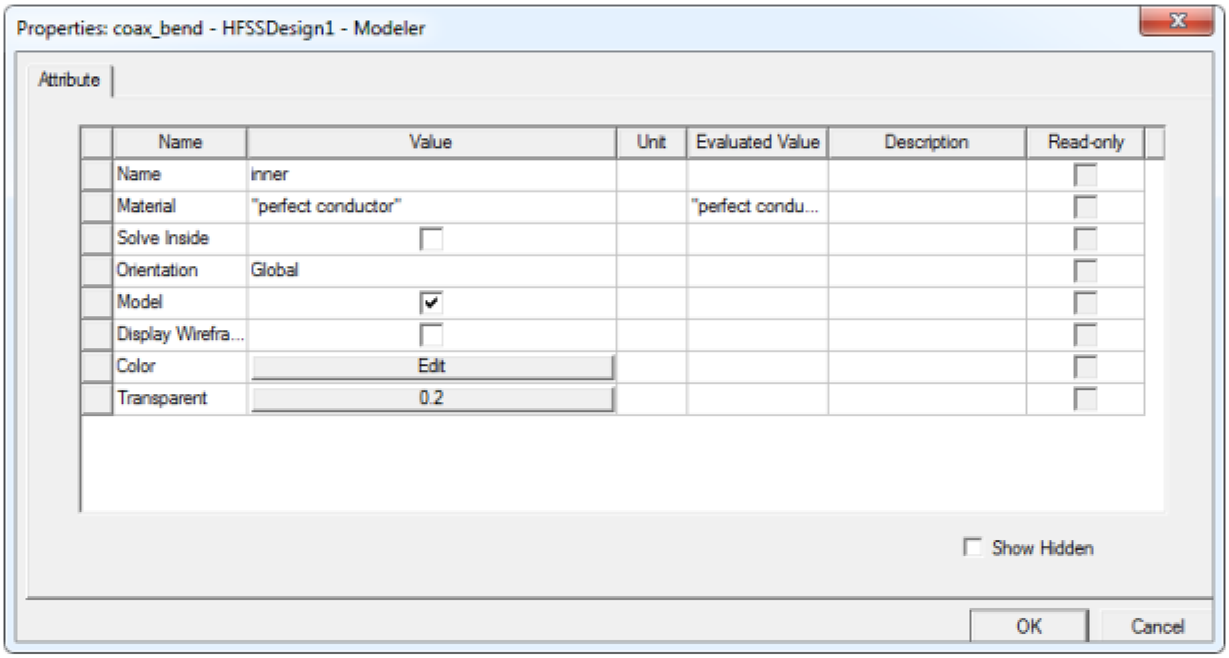
The **Properties** window displays and allows you to edit the properties of an item selected in the project tree, the history tree, or the **3D Modeler** window. The properties shown and their editability vary depending on the type of item selected.

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.

Showing and Hiding the Properties Window

To show or hide the **Properties** window on the desktop, do one of the following:

- Click **View**> **Properties**.

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

- Right-click in the toolbars area at the top of the desktop, and then click **Properties** on the shortcut menu.

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

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.

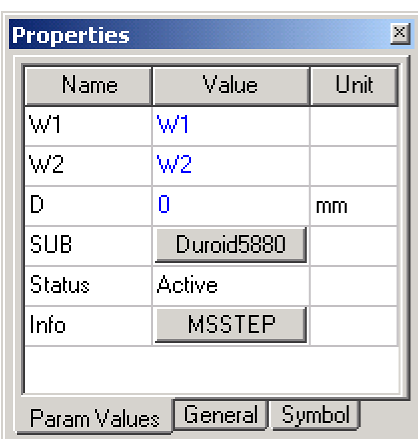
Property Window for Schematic and Layout Editors

The **Property** window shows the properties or parameters of objects selected in the schematic and layout editors, and objects in the **Project Manager** window. It allows you to modify property values. The **Property** window has three tabs:

- The Param Values Tab
- The General Tab
- The Symbol Tab

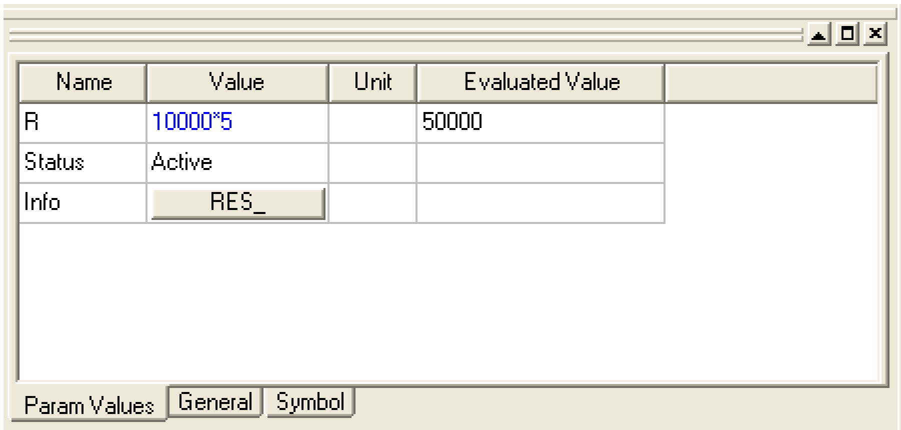
Param Values Tab

The **Param Values** tab in the **Property** window of the Ansys Electronics Desktop lists the simulation parameters of the component or components selected.



To set or change the value of a component parameter, click on the **Value** field and enter the new value. If the value requires a multiplier unit (such as **K** for 1000) click on the **Unit** field to select the multiplier unit. The Evaluated value shows the resulting number.

You can specify a parameter value using an expression that evaluates to a constant. The expression is retained in the **Value** field, while the **Evaluated Value** field shows the constant resulting from evaluation the expression. This allows you to identify and modify the expression in the future.



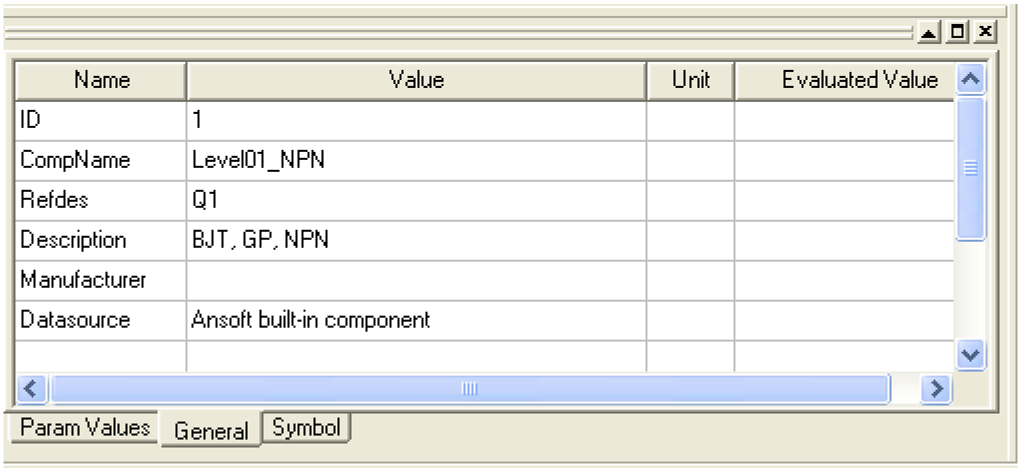
Launching Help from the Property Window

You can launch the online help for a component from the **Property** window. To do this:

1. Click the **Param Values** tab.
2. Scroll down to the **Info** parameter entry, and then click the button in its **Value** cell.

General Tab

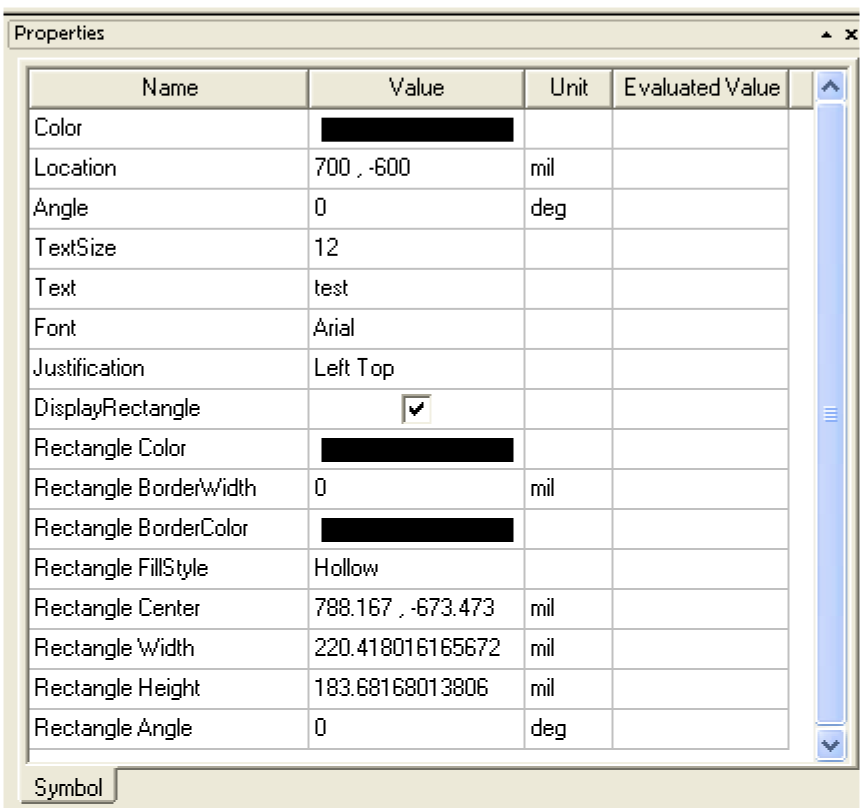
The **General** tab in the **Property** window of the Ansys Electronics Desktop lists the selection's name, symbol name, reference designator, and other data. These are generally not editable.



Name	Value	Unit	Evaluated Value
ID	1		
CompName	Level01_NPN		
Refdes	Q1		
Description	BJT, GP, NPN		
Manufacturer			
Datasource	Ansoft built-in component		

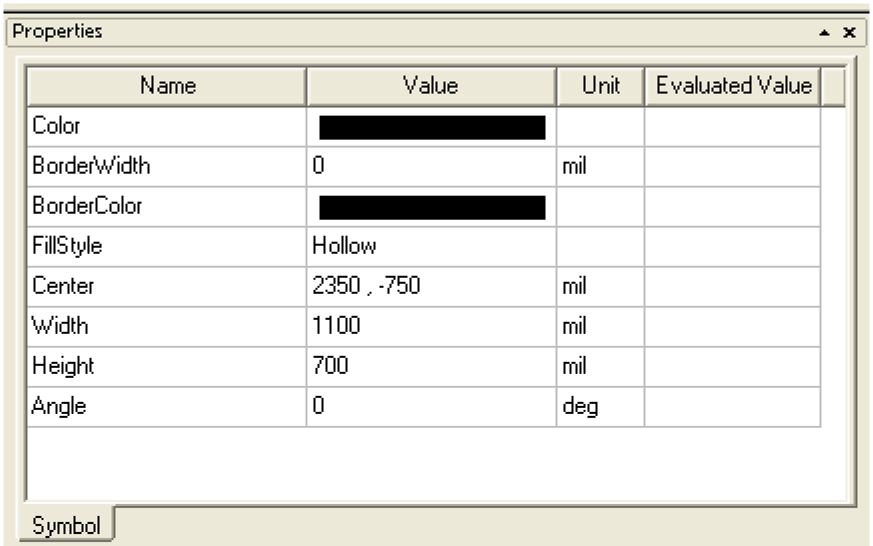
Symbol Tab

The **Symbol** tab in the **Property** window of the Ansys Electronics Desktop provides information on a number of modifiable attributes of components and symbols displayed in the schematic.



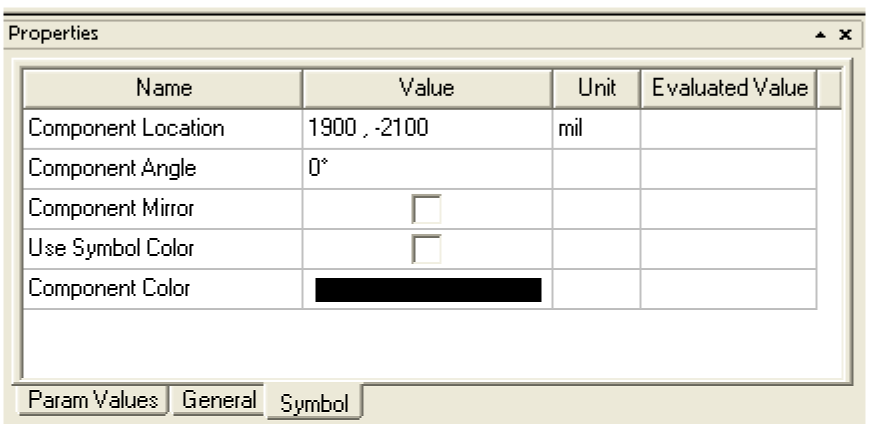
Name	Value	Unit	Evaluated Value
Color			
Location	700 , -600	mil	
Angle	0	deg	
TextSize	12		
Text	test		
Font	Arial		
Justification	Left Top		
DisplayRectangle	✓		
Rectangle Color			
Rectangle BorderWidth	0	mil	
Rectangle BorderColor			
Rectangle FillStyle	Hollow		
Rectangle Center	788.167 , -673.473	mil	
Rectangle Width	220.418016165672	mil	
Rectangle Height	183.68168013806	mil	
Rectangle Angle	0	deg	

The contents of the **Symbol** tab varies depending upon the number and type of components or symbols selected in the schematic.



Name	Value	Unit	Evaluated Value
Color			
BorderWidth	0	mil	
BorderColor			
FillStyle	Hollow		
Center	2350 , -750	mil	
Width	1100	mil	
Height	700	mil	
Angle	0	deg	

But whatever components or symbols are selected, each field displayed in the **Symbol** tab is modifiable.



Name	Value	Unit	Evaluated Value
Component Location	1900 , -2100	mil	
Component Angle	0°		
Component Mirror			
Use Symbol Color			
Component Color			

Symbol tab Value fields can be modified using the following guidelines:

- Click in the **Value** field for **Component Location** to enter a new set of X,Y coordinates for the symbol. Press **Return** to move the symbol to the new location.
- Click in the **Value** field for **Component Angle**, and select an angle from the drop-down menu (choices are 0, 90, 180, and 270 degrees). The symbol rotates as you select an angle.
- Click in the check box in the **Value** field for **Component Mirror** to flip the component left-to-right. The mirror operation is performed as soon as you check the box. Uncheck the box to return the symbol to its original orientation.

- Click in the **Value** field to enter a new value for whatever symbol attribute you wish to change. Press **Return** to move the symbol to the new location.
- Click the colored bar displayed in the **Value** field to open a palette from which to select a new color for the symbol. The new color is applied when the symbol is unselected.

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:

- Parameter Values
- General
- Symbol
- Property Displays

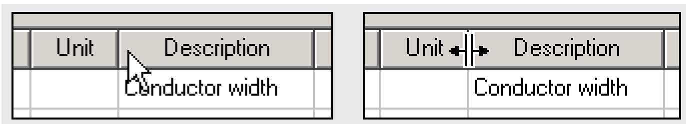
Opening the Properties Dialog Box

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**.

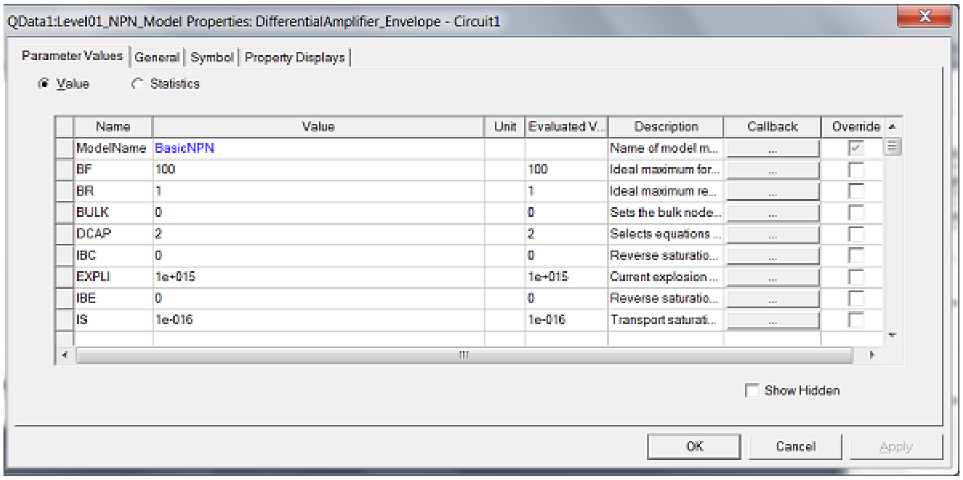
You can manipulate the **Properties** dialog box in the following ways:

- 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.



Parameter Values Tab

The **Parameter Values** tab in the Properties Dialog Box lists the simulation parameters of the component or components selected.



To set or change the value of a component parameter, click on the **Value** field and enter the new value. You can enter the units in the value field, or you can click on the **Unit** field to enter the units.

You can specify a parameter value using an expression that evaluates to a constant (as shown in the picture above). The expression is retained in the **Value** field, while the **Evaluated Value** field shows the constant resulting from evaluation the expression. This allows you to identify and modify the expression in the future. The Property Displays Tab allows you to display the expression, the evaluated value, or both.

When the **Properties** dialog box is opened for a design (**Design Properties**), the **Parameter Values** tab is initially empty. You can add properties to the design by clicking the **Add** button. Note that the **Add** button does not appear on the **Properties** dialog box for a component. See: Project Variables.

> **Note:**
>
> Clicking the **Show Hidden** check box allows you to view hidden properties of a component, and may result in the display of complex-value computations upon which some calculated values are based. Hidden properties contain system-defined values and rules for interpreting predefined component parameters. Modifying hidden properties requires specialized knowledge of the component, and is not needed for normal operation.

General and Symbol Tabs

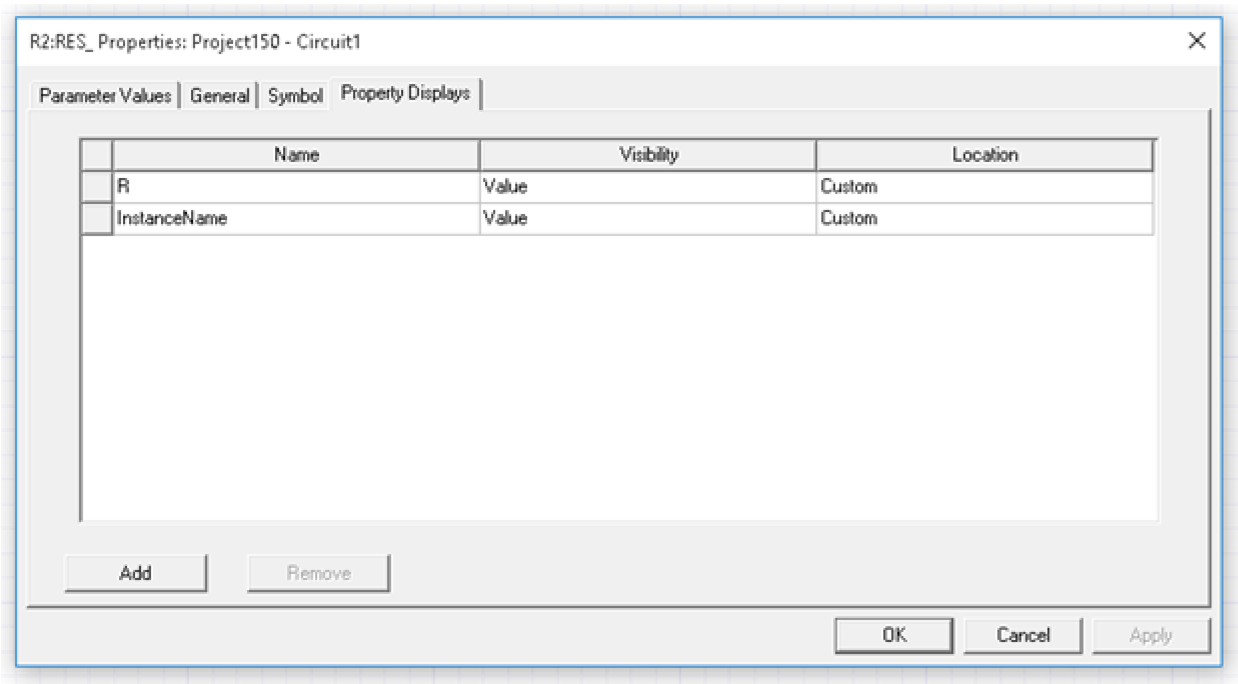
The **General** tab in the Properties Dialog Box lists the selection's name, symbol name, reference designator, and so on. These are generally not editable. The information is identical to that on the General tab of the Property Window.

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:

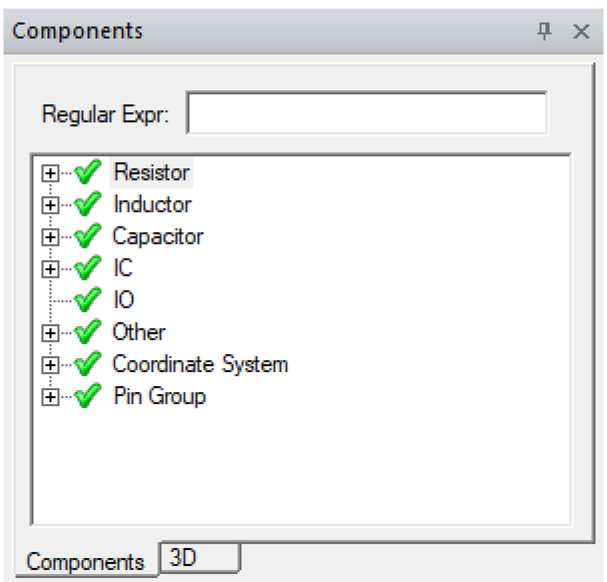
- **None** – results in no label being shown.
- **Name** – displays the parameter name (e.g., R).
- **Value** – displays the component value, which can be a single value (e.g., 10000) or an expression (e.g., 10000*5).
- **Both** – displays the parameter name and its value (e.g., R = 10000, R=10000*5).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

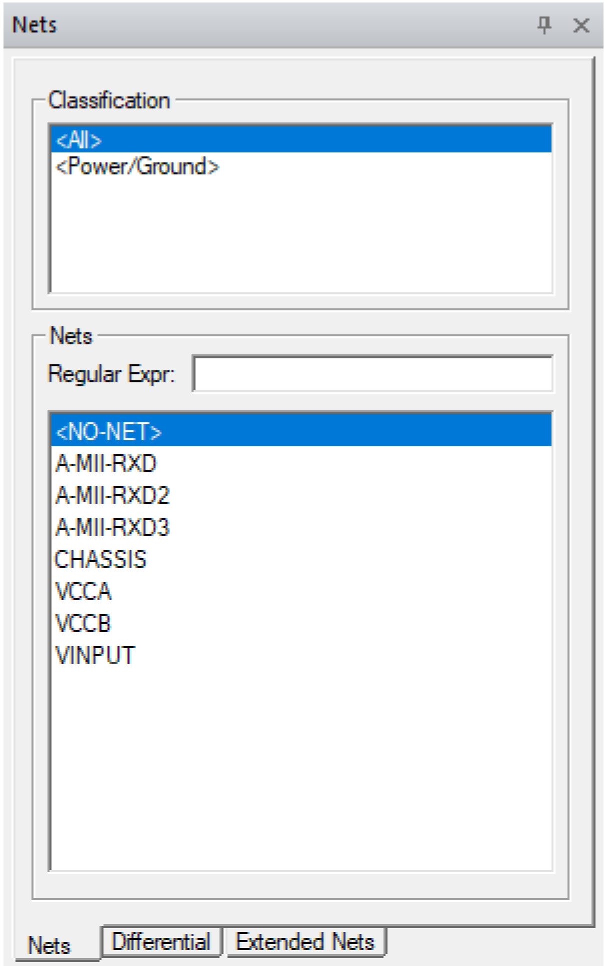
For more information, see Layout-based Component Encapsulation.

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.

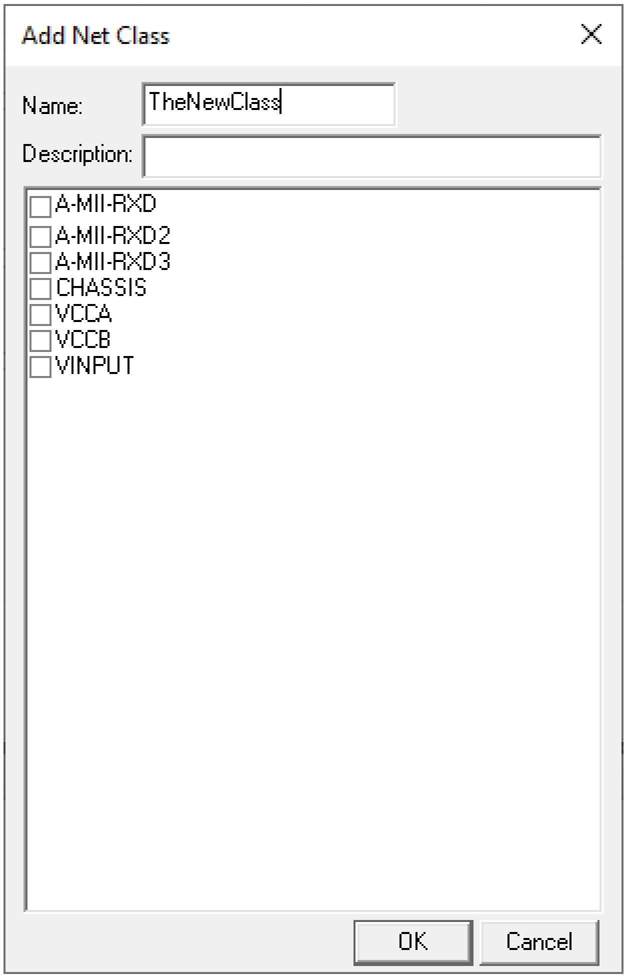
Nets Pane

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).

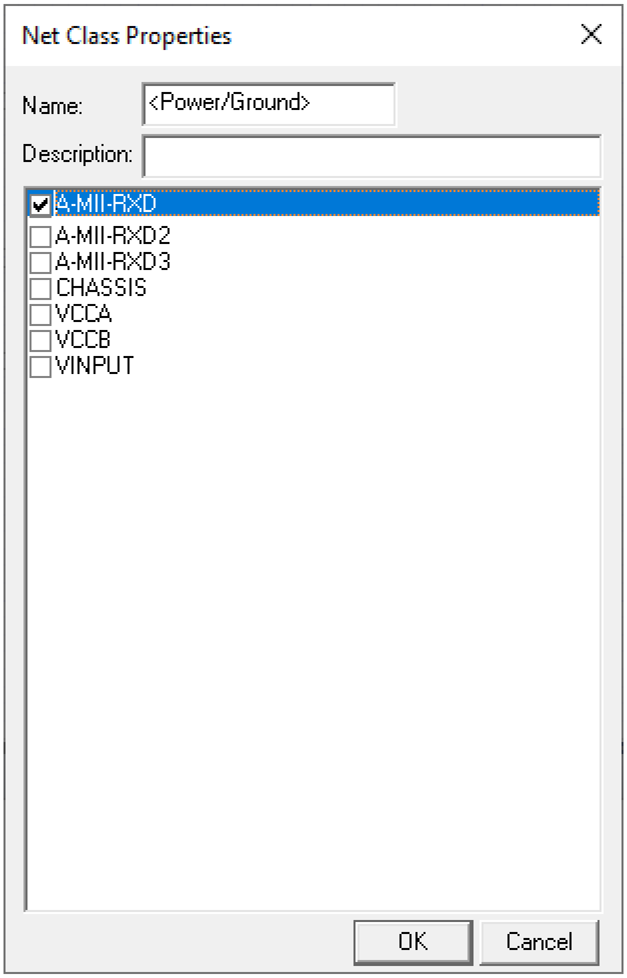
Add Net Class

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.

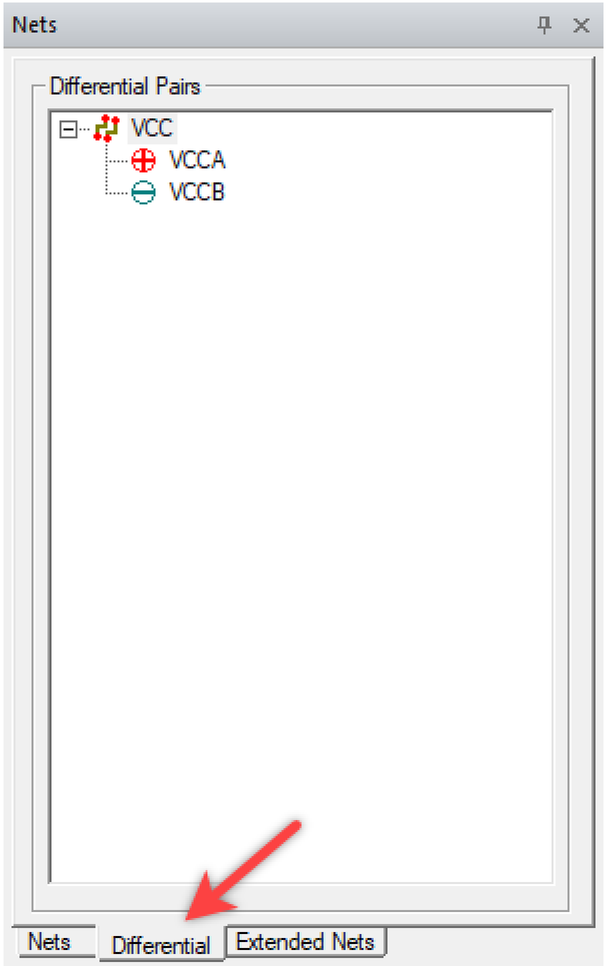


Differential Pairs

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

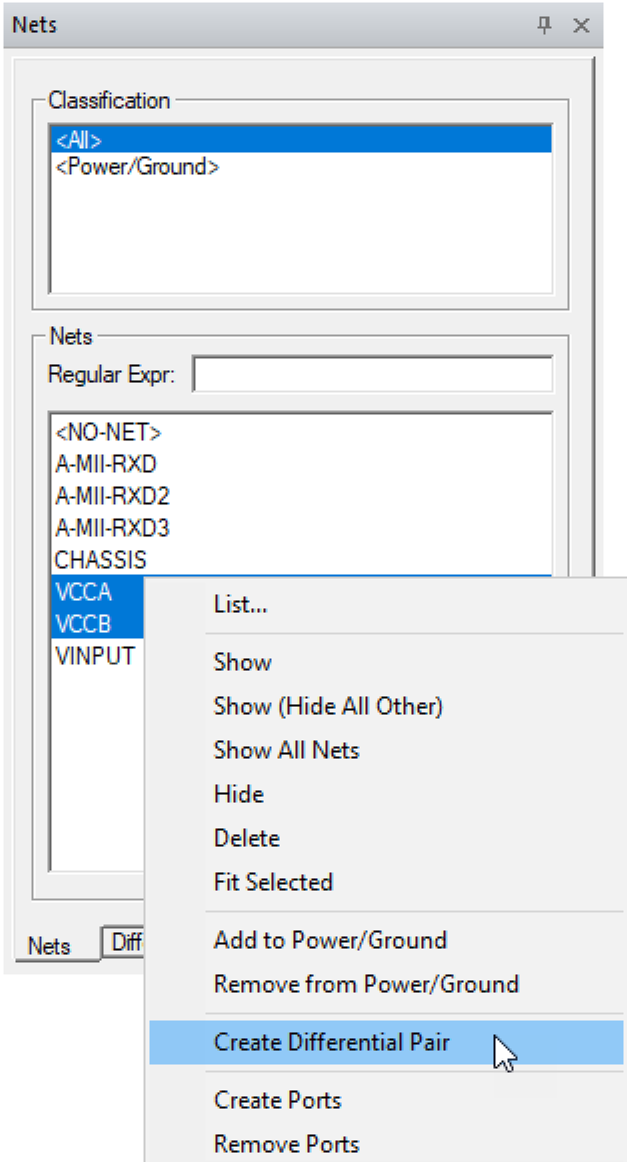
Viewing Differential Pairs

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



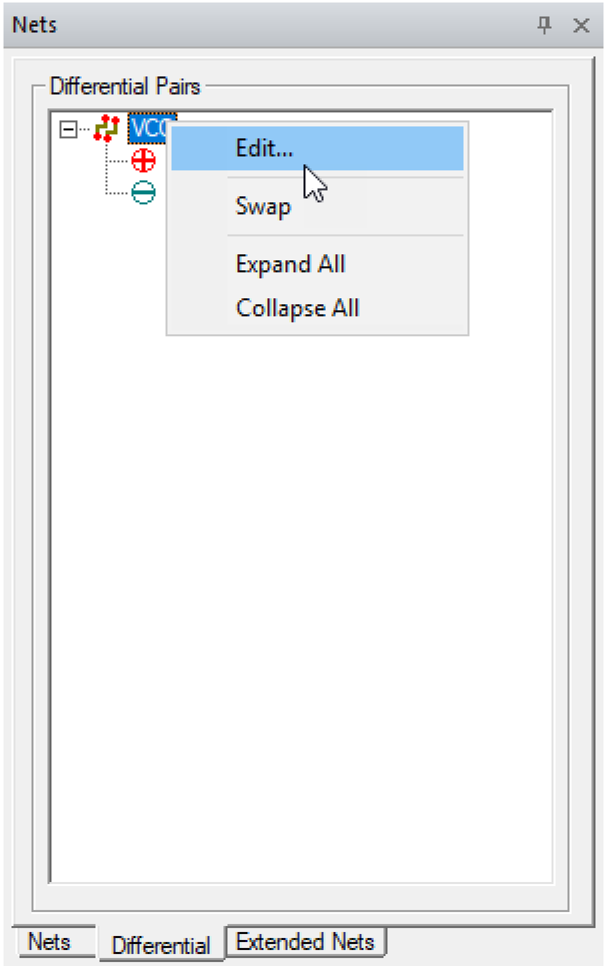
Creating Differential Pairs

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



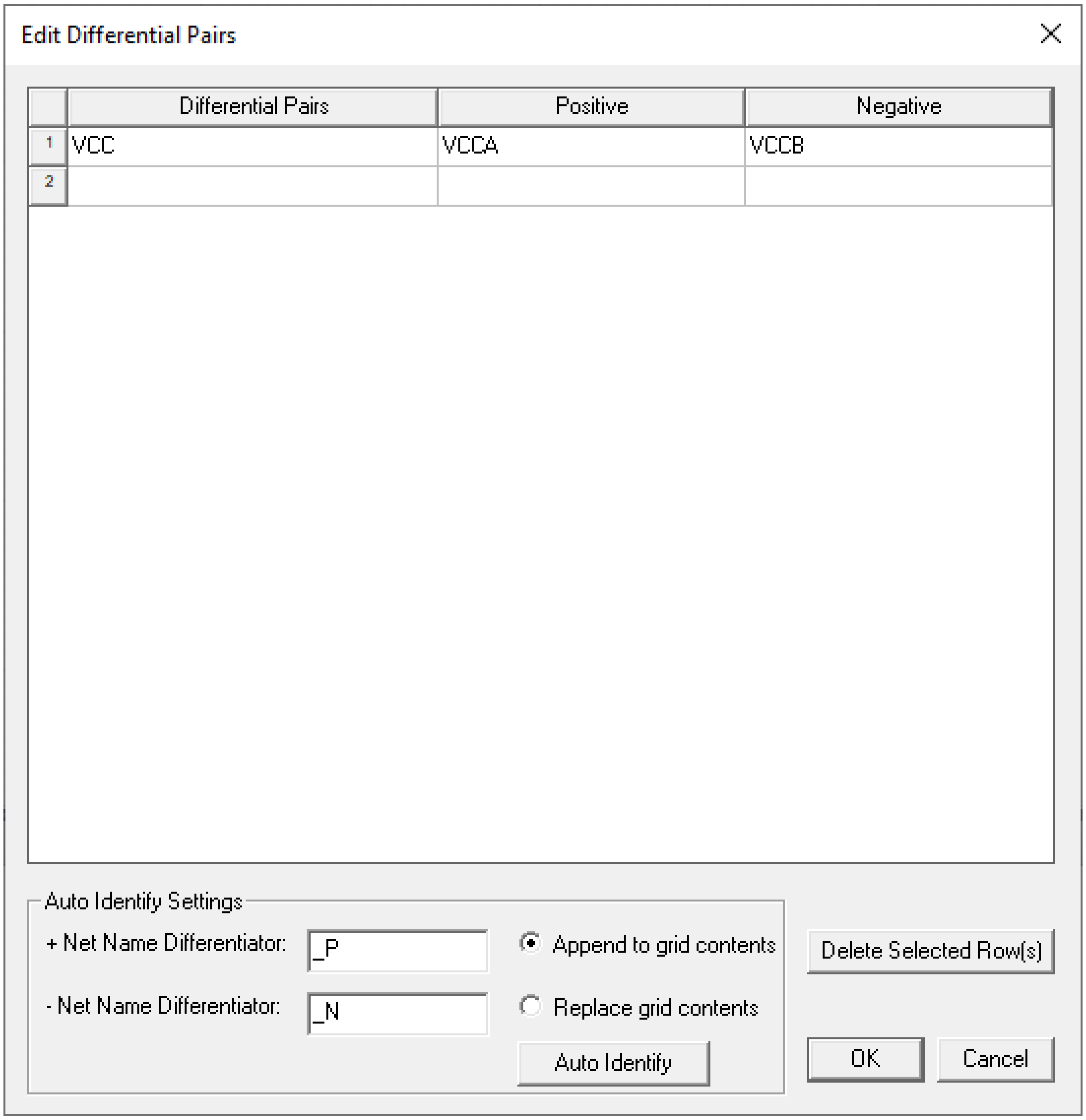
From the **Differential Pairs** list, you can select a pair, open the right-click menu and select:

- **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.



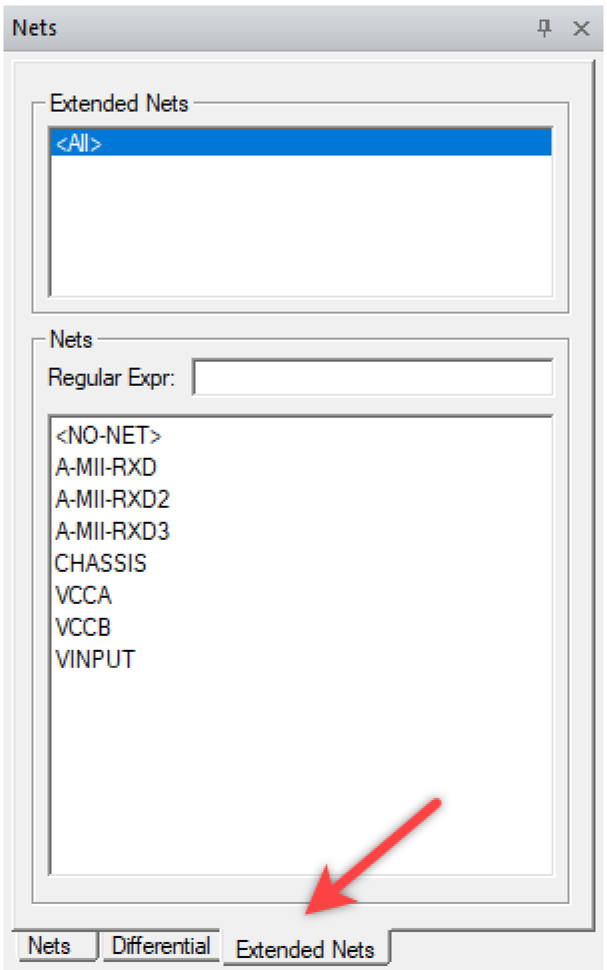
Extended Nets Tab

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.



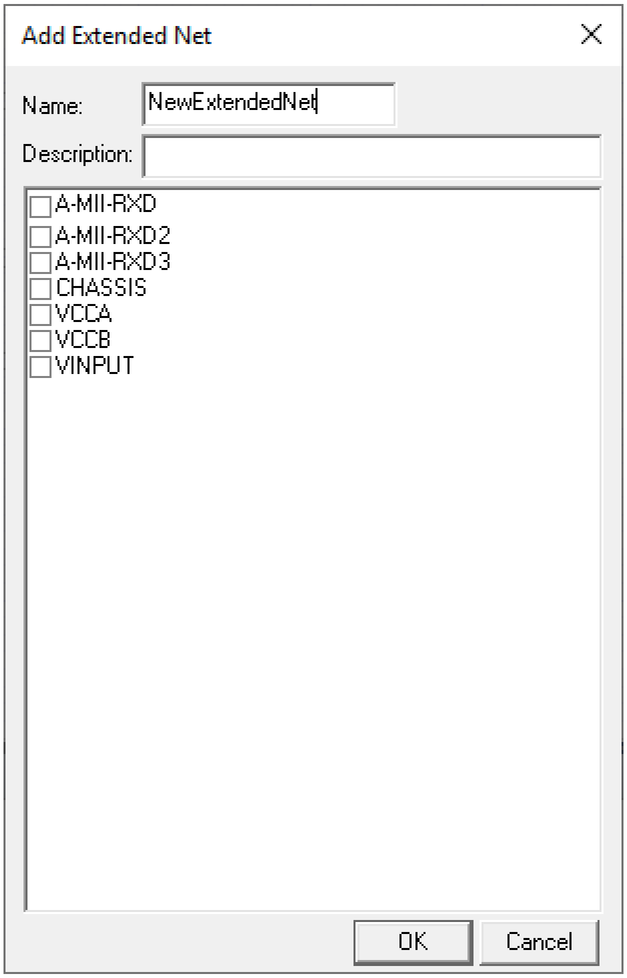
Extended Nets Pane

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

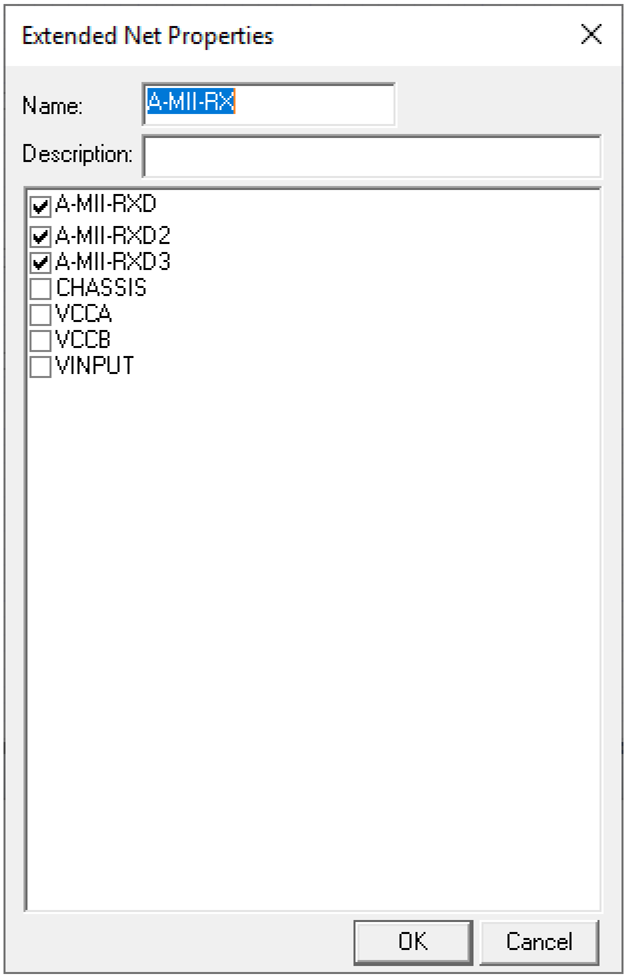
1. In the **Extended Nets** pane, right click and click **New** to open the **Add Extended Net** window.



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

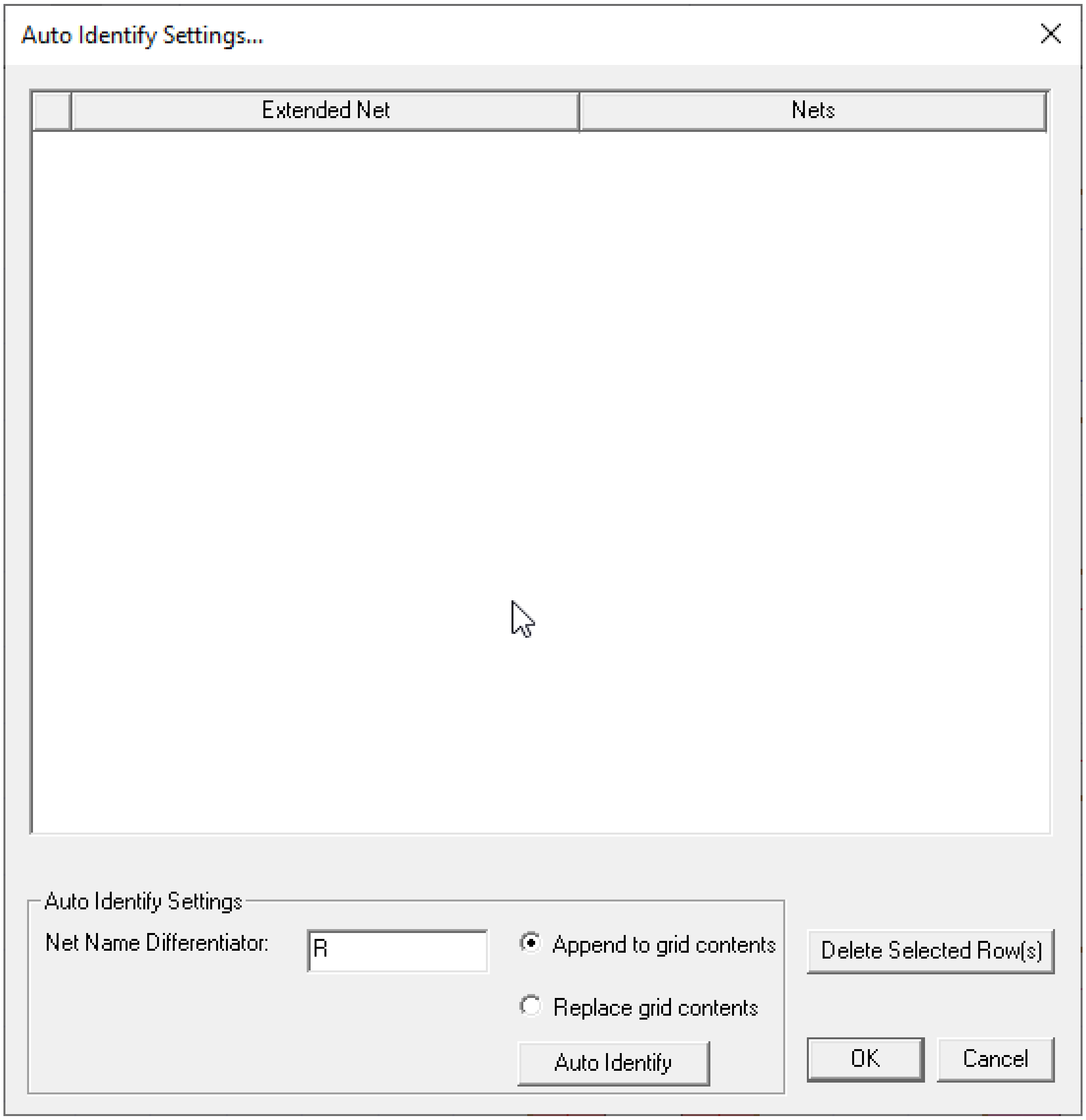
1. In the **Extended Nets** pane, right click and select **Edit** to open the **Extended Net Properties** window.



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

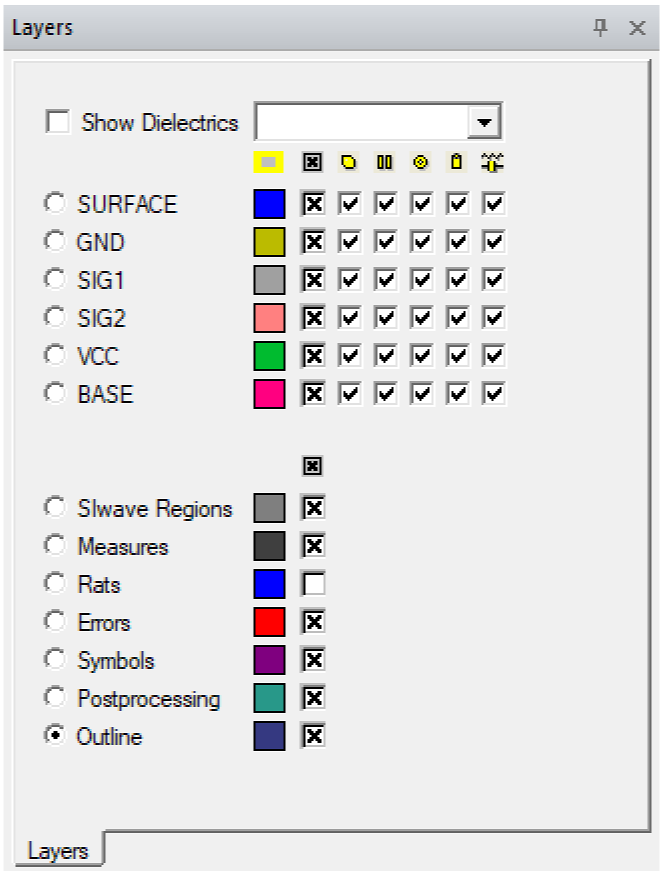
The **Nets** pane displays the members of the extended nets that are selected in the **Extended Nets** pane. These nets can be filtered using regular expressions. In the **Nets** pane, the short menu offers these actions:

- **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.

Working with the Layers Window

In the **Layers** window, you can select the working layer and the visibility of layout objects on specific layers, objects such as shapes, pads, components, nets, and the layers themselves. The **Layers** window is divided into three sections: stackup layers, non-stackup layers, and a pre-process mesh section. This dockable window can be resized and relocated.



- To show or hide the **Layers** window on the desktop, click **View** > **Layers**. A check box appears next to this command if the **Layers** window is visible.
- To select a working layer, click the radio button of the layer you want to work on.
- To view specific items on different layers, use **Show Dielectrics**, **Zone Filter**, the layers check boxes, and View Name drop-down menu.

If the stackup is a multizone stackup, select a zone from the **Zone Filter** to see the available layers.

The **Show Dielectrics** check box controls whether dielectric stackup layers are shown. If unchecked, dielectric layers are hidden and holes (vias and pins) are rendered continuously visible from the top-most visible signal layer to the bottom most visible layer. If checked, dielectric layers appear and hole visibility can be finely controlled by each layer (but not necessarily continuously visible).

The View Name drop-down menu lists user-defined views and their controls. You can configure how layers and nets are displayed and then save this visibility as a view. Later, select that view from the drop-down menu to reset the **Layout** window to it. The view controls and default options are:

- **<Surface>** shows the top and bottom conducting layers of the layer stackup and hides all other layers. This is the default view.
- **Save Current View** opens a dialog that allows you to save the current setting of layers and nets to a new layout view.
- **Delete View** opens a dialog that allows you to delete saved layout views.

Check boxes control what is displayed in a layer. Icons at the top of a column control the entire column. Individual check boxes control the item visibility for the layer corresponding to the check box's row. Hover over an icon to see what the column controls:

Name	Icon	Controls the visibility of	Appears
Fill/Unfill All		Color in the object	Always
Show/Hide All		All objects	Always
Shapes		Shapes	Always
Lines		Lines and paths	Always
Pads		Pads	Always
Holes		Holes	Always
Components		Components	Always
Mesh		Mesh of signal layers	When a pre-process mesh is displayed

Name	Icon	Controls the visibility of	Appears
Background Dielectric Mesh		Mesh of dielectric layers	When a pre-process mesh is displayed with a laminate stackup

A pre-process mesh section may appear below the non-stackup layer section. It controls design-wide visibility of:

- Airbox non-mesh and mesh
- Hole, solderball, and bondwire mesh

The visibility of these objects is not restricted to their layers. Also, layer visibility check boxes cannot control these mesh visibilities. Non-mesh visibilities are controlled by the layer check boxes.

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

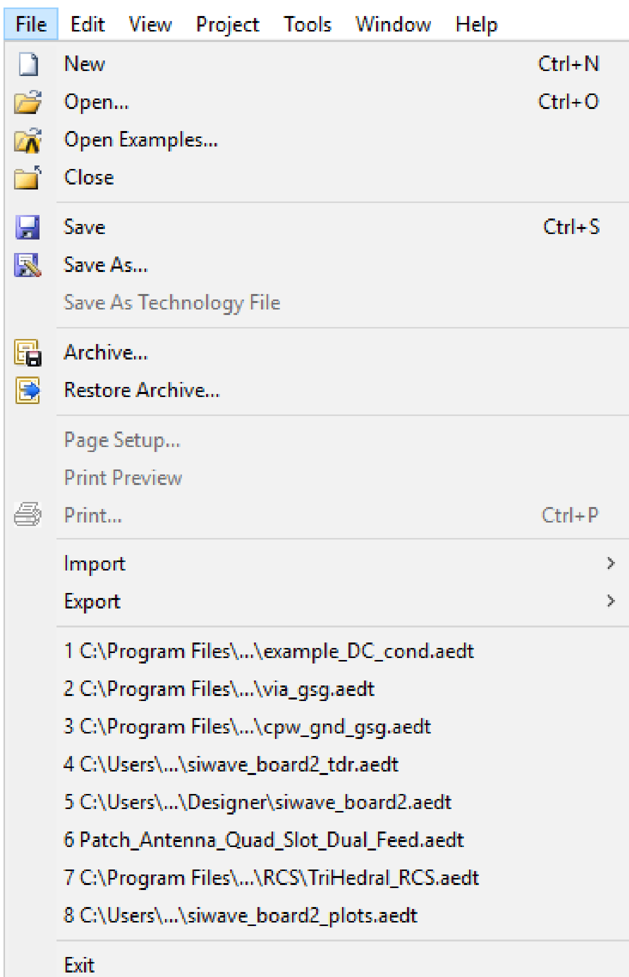
Use the shortcut menu in the toolbars area of the desktop to toggle the show/hide settings for various desktop windows and toolbars. To access the shortcut menu, right-click in the toolbars area at the top of the desktop.

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

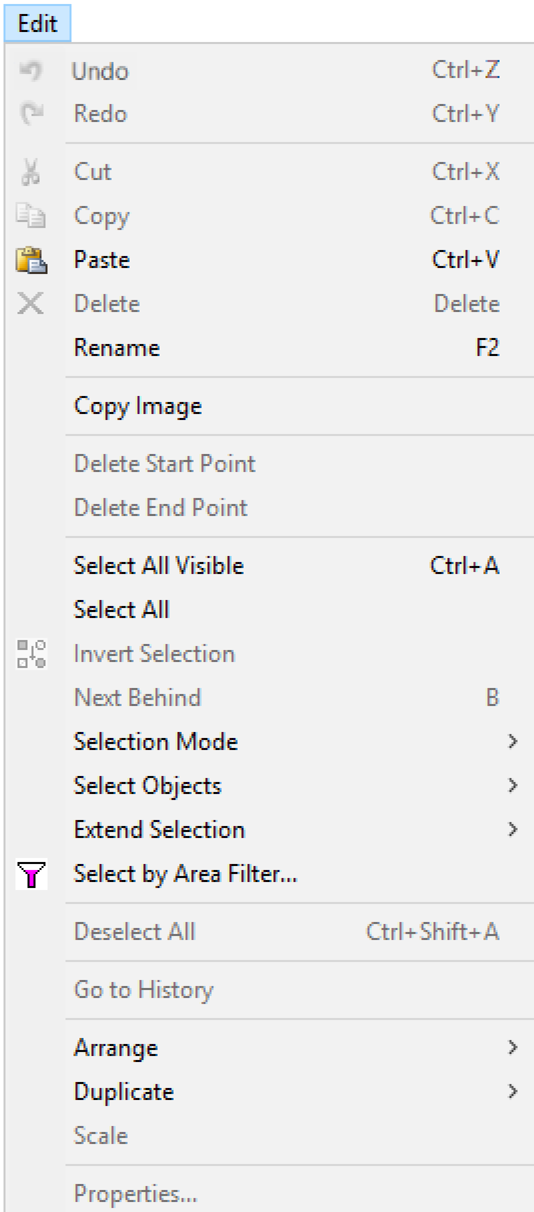
The **Edit** menu changes, depending on the active window in the Design Area.

These menus include:

- Schematic Editor
- Layout Editor
- 3D Modeler
- Netlist Editor
- Report Window

Edit Menu for 3D Modeler

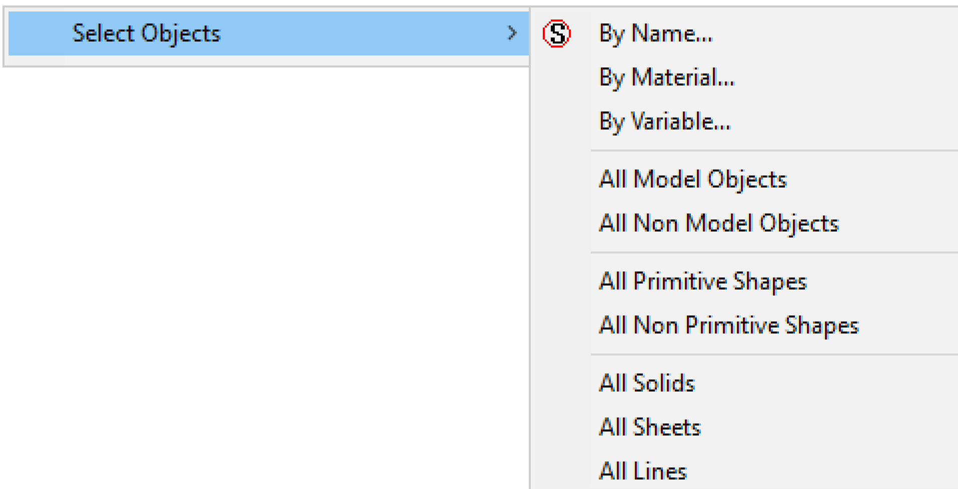
When a Modeler window is active in the Design Area, the **Edit** menu is similar to the following image:



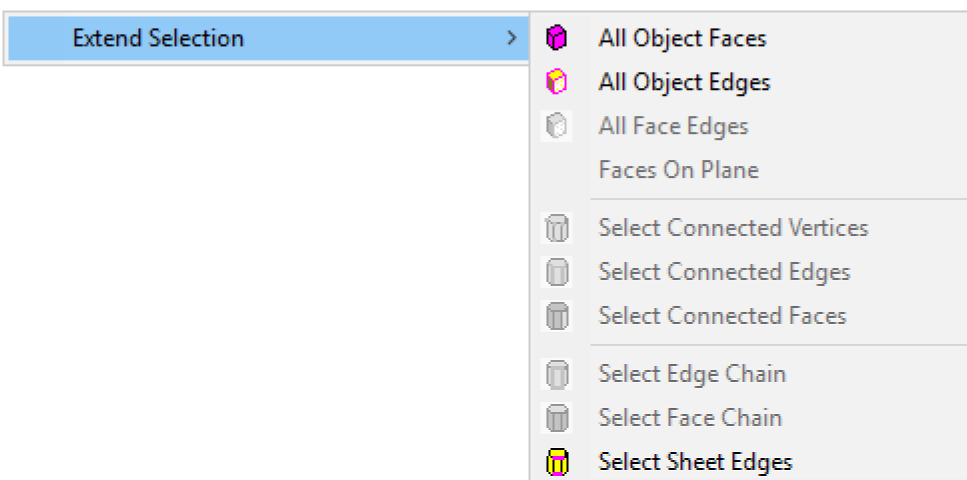
Comands will be available or not available for clicking depending on what is currently selected or on whether any operations have been performed, undone, or redone. The 3D Modeler's Edit menu contains the following options:

- **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.

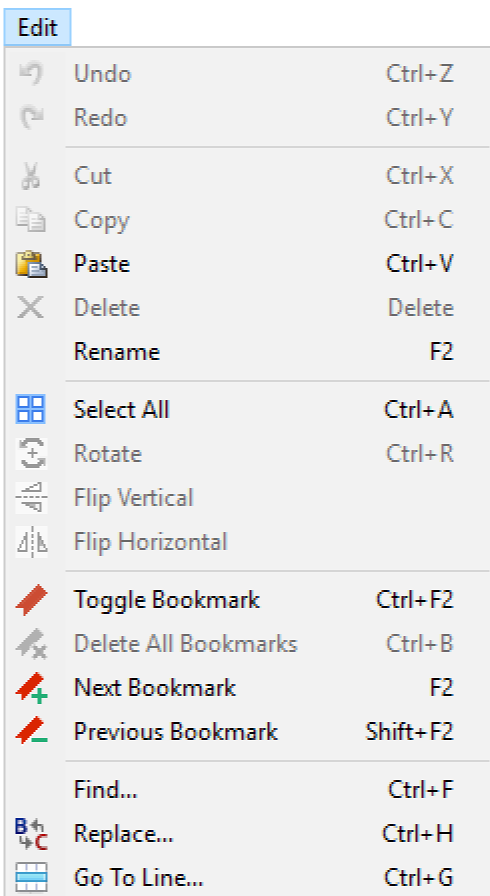


- **Select by Area Filter** – opens the **Select By Area Filters** window.
- **Deselect All** – deselects all selected objects.

- **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:



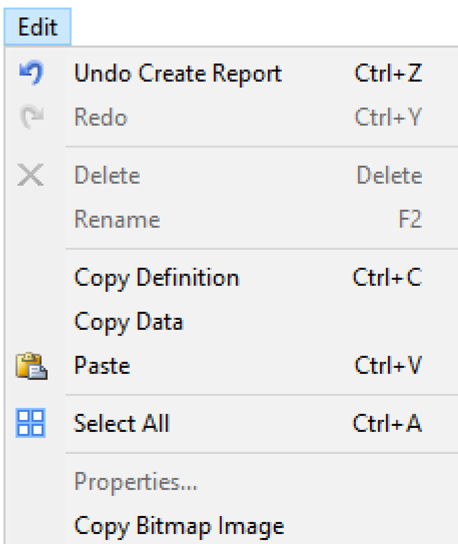
The Netlist Editor Edit menu contains the following options:

- **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:



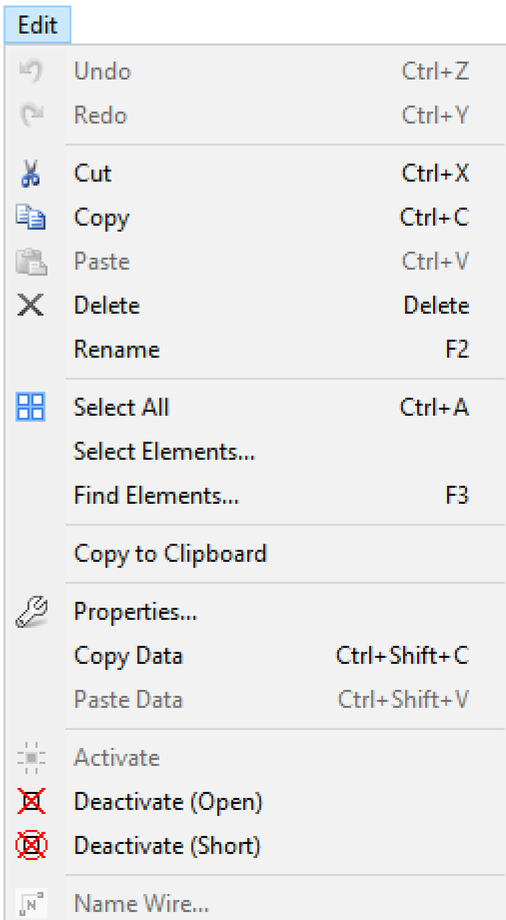
The Report Window Edit menu contains the following options:

- **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.

Edit Menu for Schematic Editor

When the Schematic Editor is active in the Design Area, the **Edit** menu appears similar to the following:



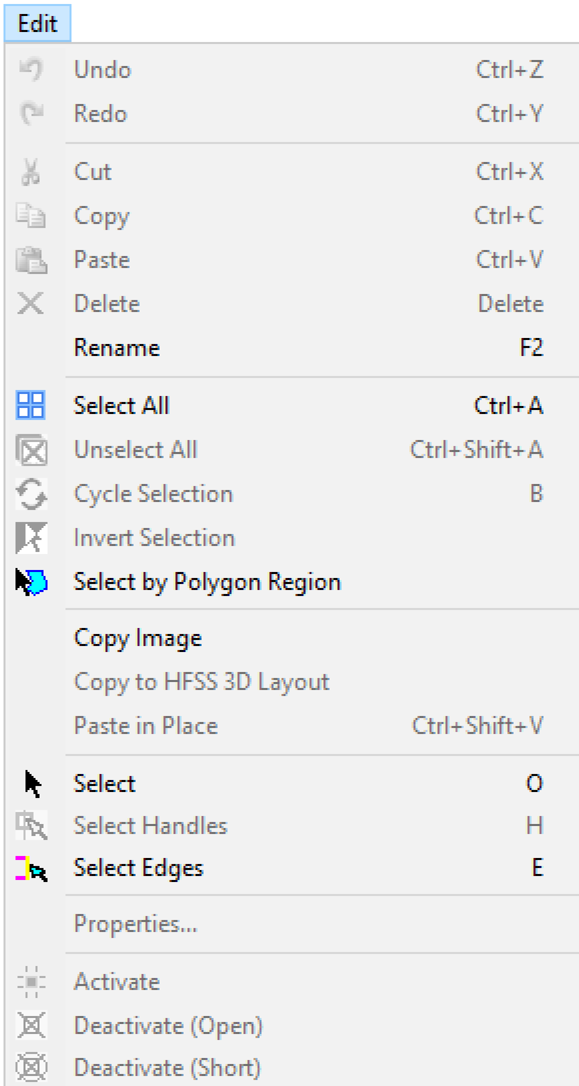
The Schematic Editor Edit menu contains the following options:

- **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 schematic elements.
- **Delete** – deletes a selected element.
- **Rename** – renames a selected element.
- **Select All** – selects everything in the design area.

- **Select Elements** – allows you to select certain nets or components.
- **Find Elements** – opens the **Find Elements** window, where you can search for elements by property name and value.
- **Copy to Clipboard** – copies the visible design area to the clipboard in bitmap format, for pasting into another program or document.
- **Properties** – displays the relevant properties window for the selected object.
- **Copy Data** – copies properties data for the selected object.
- **Paste Data** – pastes properties data into a selected object.
- **Activate** – causes the activation of one or more selected components.
- **Deactivate (Open)** – temporarily converts the component into an open circuit.
- **Deactivate (Short)** – temporarily converts the component into a short circuit.
- **Name Wire** – allows you to rename a selected wire.

Edit Menu for Layout Editor

When the Layout Editor is active in the Design Area, the **Edit** menu appears similar to the following:



The Layout 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 layout elements.
- **Delete** – deletes a selected element.
- **Rename** – renames a selected element.
- **Select All / Unselect All** – selects or deselects the entire layout.
- **Cycle Selection** – rotates through a selection of objects.
- **Invert Selection** – selects the opposite of the current selection.

- **Select By Polygon Region** – allows you to draw a polygon in the design area and selects all elements within that polygon.
- **Copy Image** – copies the design area display in bitmap format, for pasting into another program or document.
- **Copy to HFSS 3D Layout** – copies the design into an HFSS 3D Layout and adds that layout to the current project.
- **Paste in Place** – duplicates an object without any X / Y offset or displacement.
- **Select** – puts the cursor in "Select" mode (the default).
- **Select Handles** – puts the cursor in "Select Handles" mode; the cursor will only select handles.
- **Select Edges** – puts the cursor in "Select Edges" mode; the cursor will only select edges.
- **Properties** – opens the relevant Properties window for the selected object.
- **Activate** – causes the activation of one or more selected components.
- **Deactivate (Open)** – temporarily converts the component into an open circuit.
- **Deactivate (Short)** – temporarily converts the component into a short circuit.

View Menu

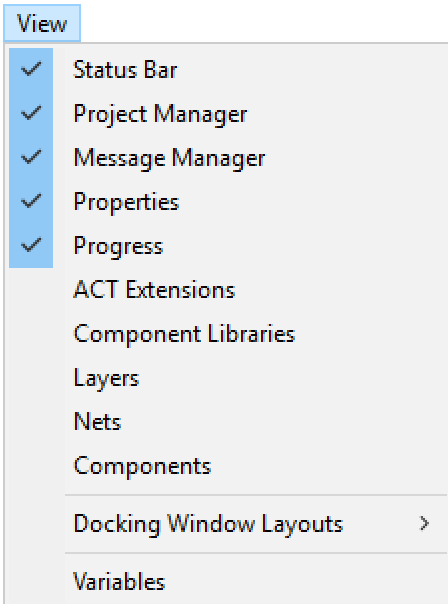
The **View** menu changes, depending on the active window in the Design Area.

These menus include:

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

Basic View Menu

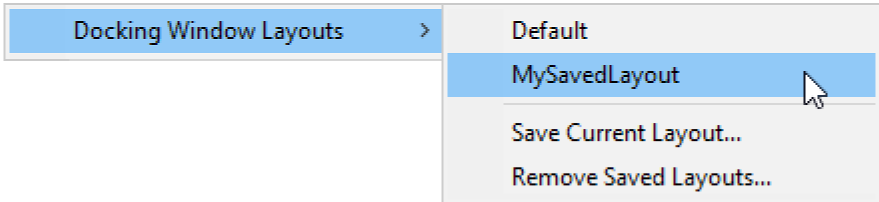
When no editor is open in the design area, or when the Netlist Editor or Report Window is active, the **View** menu appears similar to the following:



- 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.

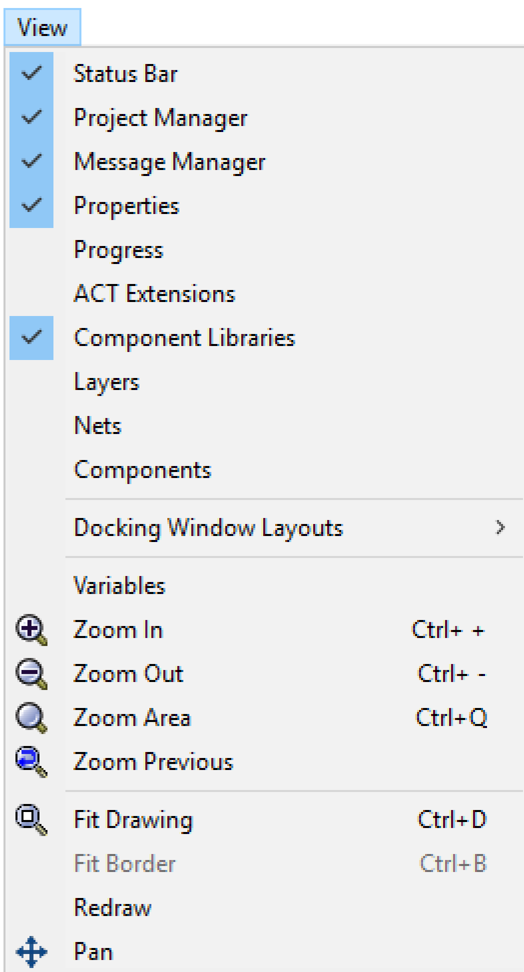
- The **Docking Window Layouts** submenu lets you select from the **Default** and any saved window layouts, **Save Current Layout**, or **Remove Saved Layouts** by selecting them from a list.



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

Schematic Editor View Menu

When the Schematic Editor is active, the **View** menu appears similar to the following:



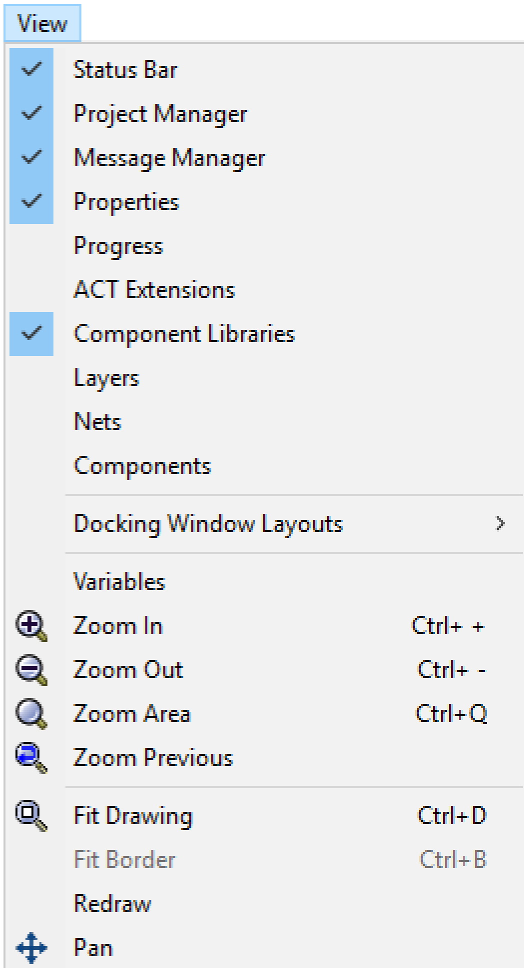
In addition to the options available on the Basic View Menu, the Schematic Editor View Menu contains the following:

- **Zoom In** / **Zoom Out** / **Zoom Area** / **Zoom Previous** – control the portion of the design displayed in the design area.
- **Fit Drawing** – changes the display so that the entire drawing fits to the edges of the design area.
- **Fit Border** – when a border has been added to a drawing, scales the bordered contents to fit within the bounds of the window.
- **Redraw** – redraws the current design.
- **Pan** – allows you to pan across the drawing.

See: Schematic Editor.

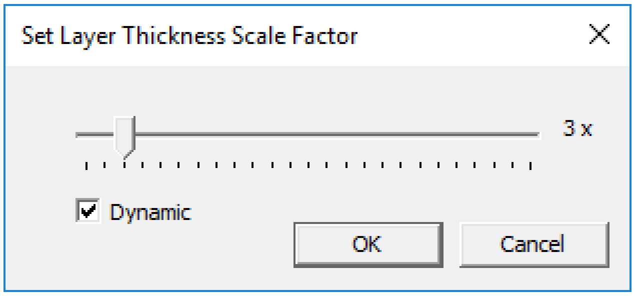
Layout Editor View Menu

When the Layout Editor is active in the design area, the **View** menu appears similar to the following:

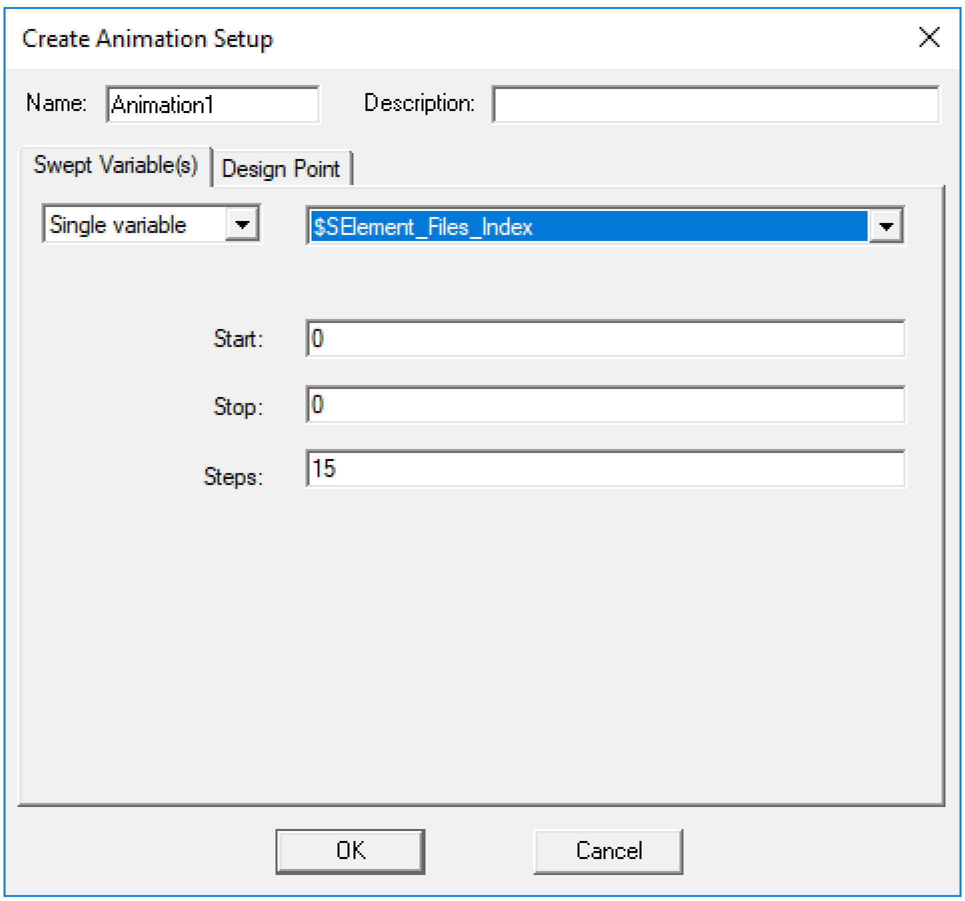


In addition to the options available on the Basic View Menu, the Layout Editor View Menu contains the following:

- **Zoom In** / **Zoom Out** / **Zoom Area** / **Zoom Previous** – control the portion of the design displayed in the design area.
- **Fit Drawing** – changes the display so that the entire drawing fits to the edges of the design area.
- **Fit Border** – when a border has been added to a drawing, scales the bordered contents to fit within the bounds of the window.
- **Redraw** – redraws the current design.
- **Pan** – allows you to pan across the drawing.
- **Stretch Z** – opens the **Set Layer Thickness Scale Factor** window, allowing you to change layer thickness.



- **Reset Orientation** – restores the design area to its default view.
- **Faster Transformations** – offers a simplified model during rotate, zoom, and pan operations to speed up performance.
- **Animate** – opens the **Create Animation Setup** window, so that you can begin creating animations.

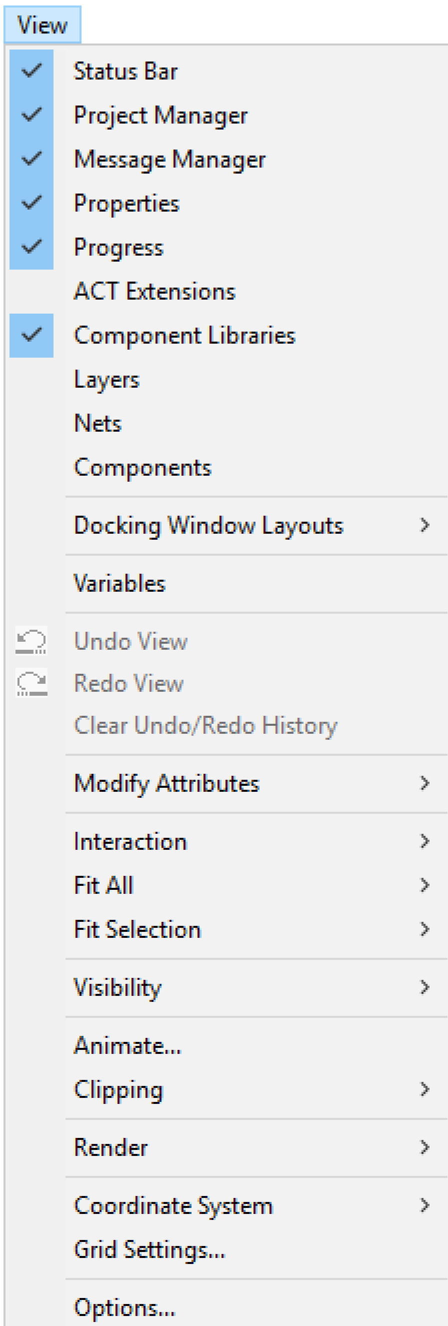


- **Clip Plane** – places a 3D coordinate system manipulator in the design area so that you can define clip planes.

See: Layout Editor.

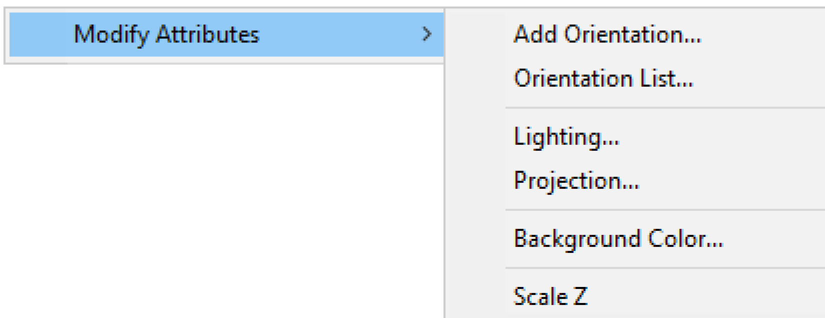
3D Modeler View Menu

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

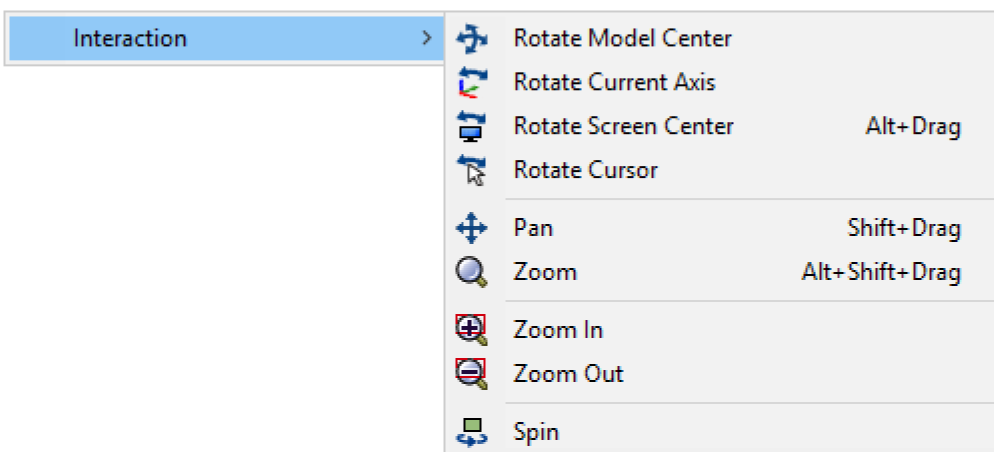


Thi View menu includes the options available on the Basic View Menu. The following additional options are available when a *Modeler* window is active. Note that a number of these options also appear in the View branch of the shortcut menu that appears when you right-click in the Modeler window.

- **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**.

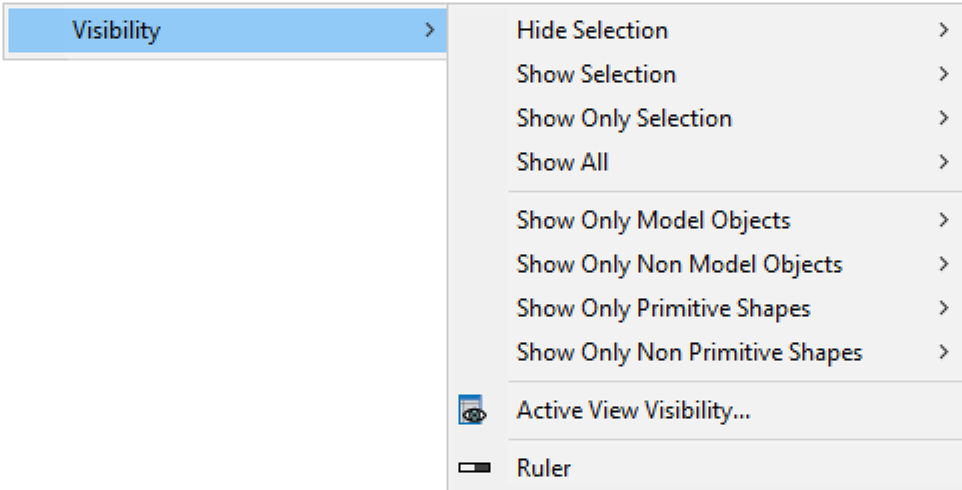


- **Interaction** – offers options to rotate, pan, zoom, and spin the drawing in the design area.

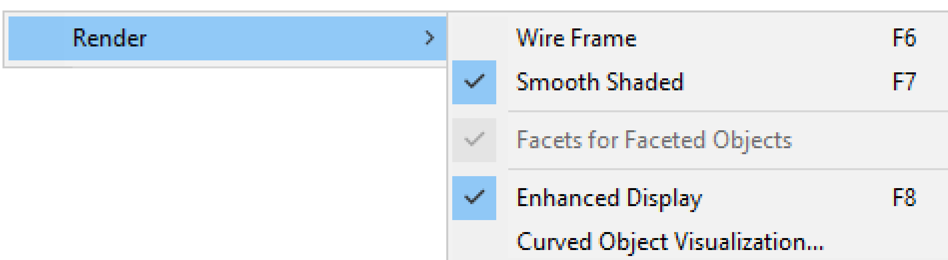


- **Fit All** – allows you to fit all objects to the view window.
- **Fit Selection** – allows you to fit selected objects to the view window.

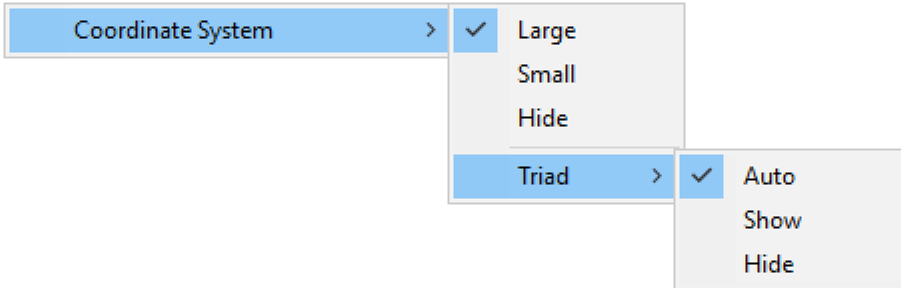
- **Visibility** – offers options to show and hide selections, objects, shapes, and the ruler.



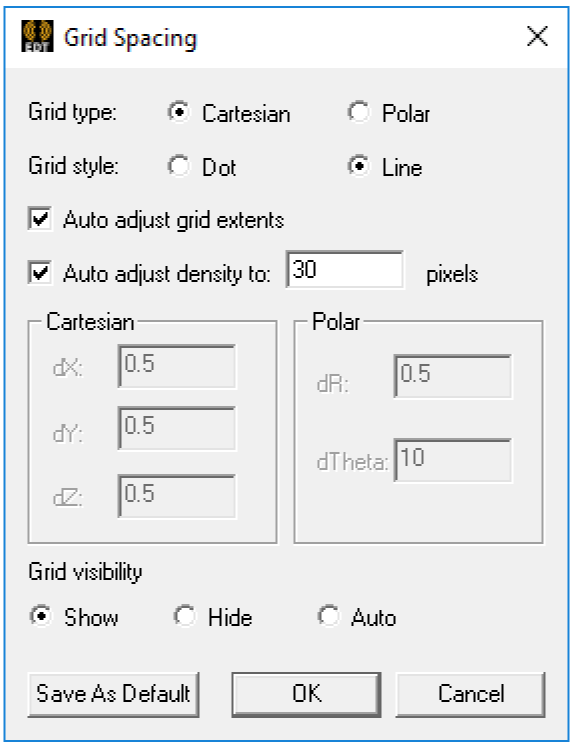
- **Animate** – opens the **Create Animation Setup** window, so that you can begin creating animations.
- **Clipping** – allows you to place a 3D coordinate system manipulator in the design area so that you can define clip planes.
- **Render** – offers options to switch between wire frame and smooth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



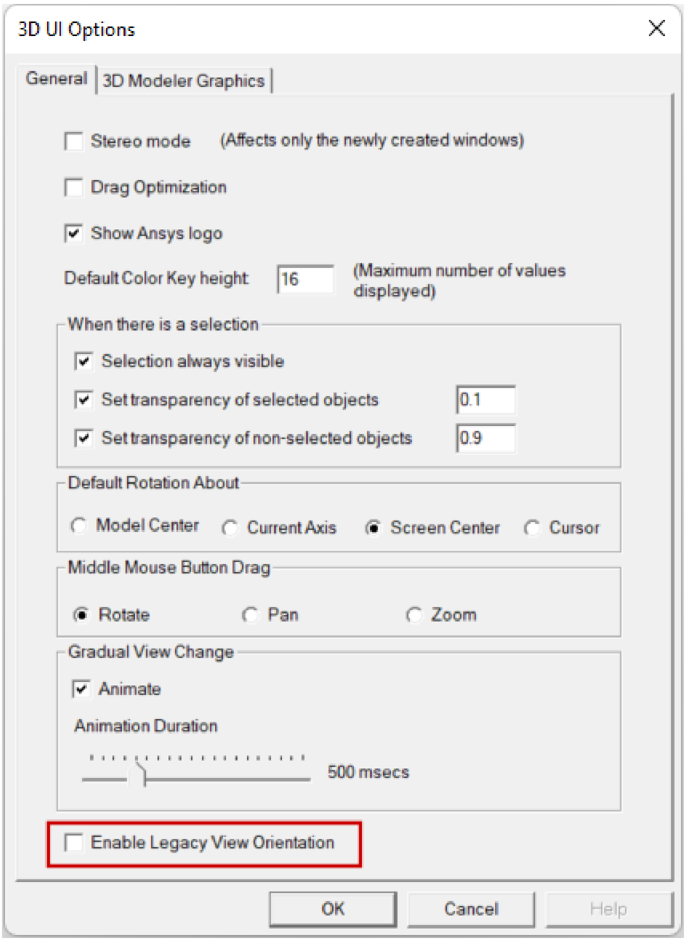
- **Coordinate System** – offers options to change how the coordinate system displays in the design area.



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

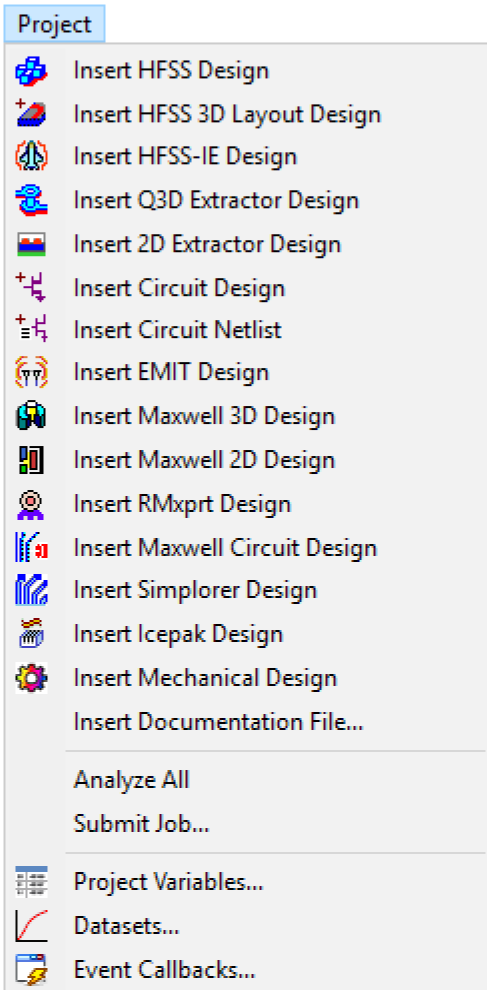


- **Options** – opens the **3D UI Options** window, where you can change additional settings.



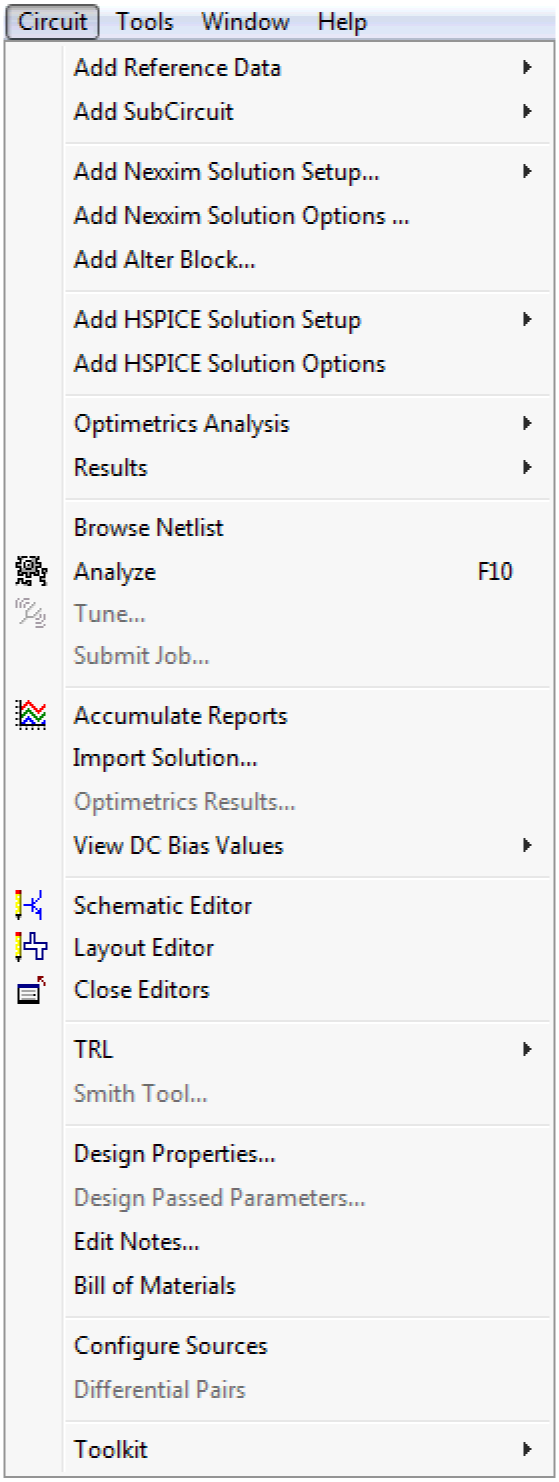
Project Menu

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.


Circuit
Tools
Window
Help
Add Reference Data
Add SubCircuit
Add Nexxim Solution Setup...
Add Nexxim Solution Options ...
Add Alter Block...
Add HSPICE Solution Setup
Add HSPICE Solution Options
Optimetrics Analysis
Results
Browse Netlist
Analyze
F10
Tune...
Submit Job...
Accumulate Reports
Import Solution...
Optimetrics Results...
View DC Bias Values
Schematic Editor
Layout Editor
Close Editors
TRL
Smith Tool...
Design Properties...
Design Passed Parameters...
Edit Notes...
Bill of Materials
Configure Sources
Differential Pairs
Toolkit

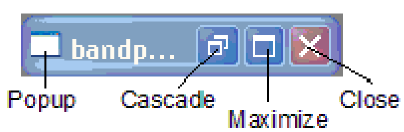
Working with Editor Windows

Each editor window in the Design Area has size controls in its top bar:

Clicking the white rectangle on the left opens a pop-up:

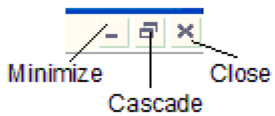
The **Move** and **Size** operations are not needed. To move an editor window, drag it by the top bar into the desired location. To resize an editor window, position the cursor at an edge or corner, and drag the border in or out.

- Click **Minimize** (or use the Minimize icon) to collapse the design window into an icon:



- Clicking the white rectangle on an iconized window opens the same pop-up as the one shown above. To restore an iconized window to its previous size (cascaded or maximized), click the white rectangle, then select **Restore** on the pop-up. Alternatively, use the Cascade or Maximize buttons on the icon to restore the window to the desired configuration. Use the Close button to close the iconized editor window. Reposition the icons anywhere in the Design area by dragging them with the left mouse button. To restore the icons to a neat row at the bottom of the Design Area, use the **Arrange Icons** command on the Window pulldown.
- To relocate an iconized window in the Design Area, drag it with the left mouse button.

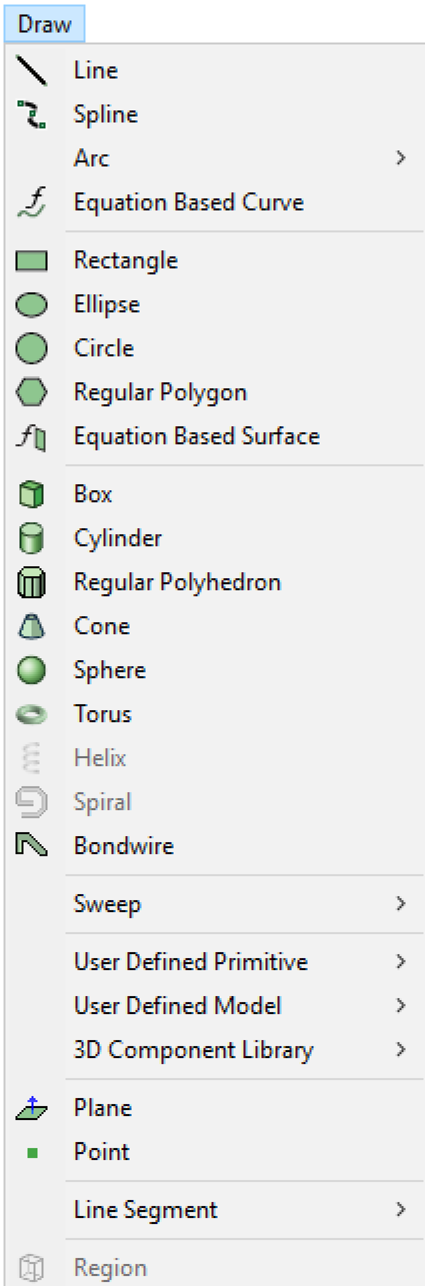
- Click **Maximize** (or use the Maximize icon) to enlarge the editor window to fill the entire Design Area. In this configuration, Minimize, Cascade, and Close icons appear at the upper right of the Ansys Electronics Desktop, on the same level as the Top Menu bar:



- To restore a maximized window to its initial size and top bar, use the **Cascade** operation.
- Click **Close** (or use the Close icon) to close an editor window.
- Click **Next** to move the active focus to another editor window.
- See Design Area for a summary of the available editor windows.

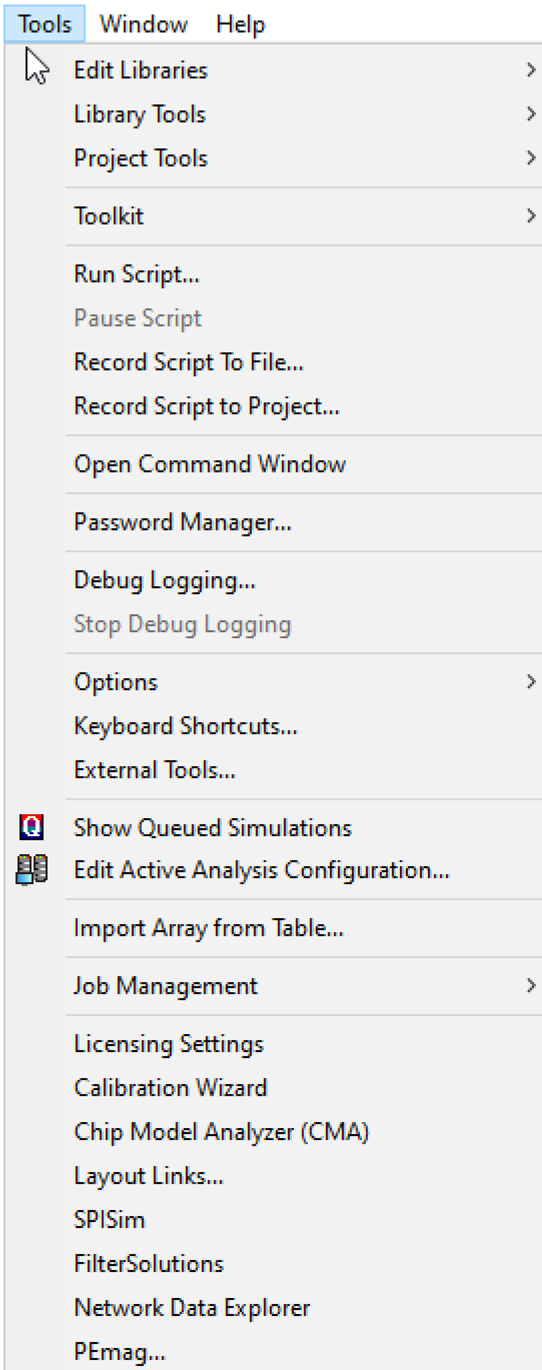
Layout Editor Draw Menu

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

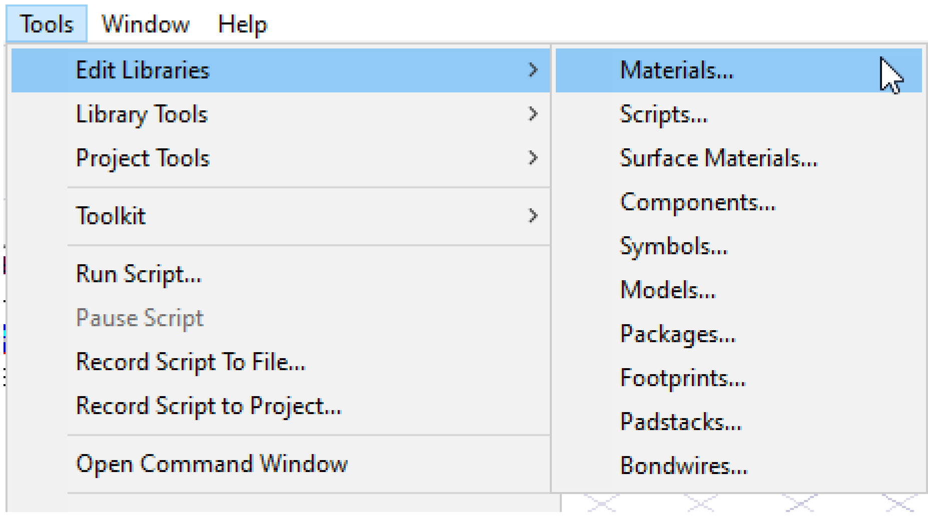


Tools Menu

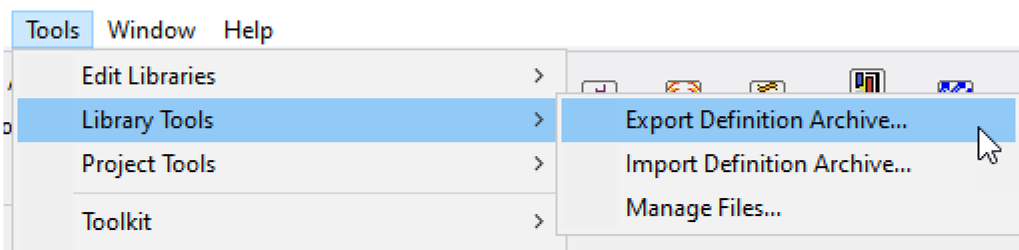
The **Tools** menu contains operations that are common to the analysis tools.



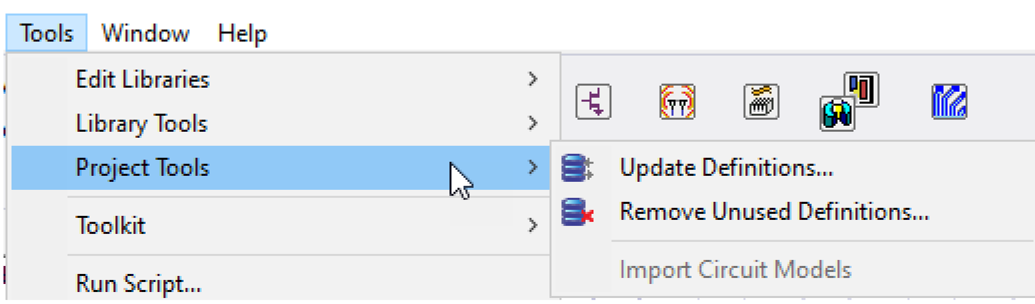
- Click **Tools>Edit Libraries** to view menus allowing access to the libraries available for the various Ansys Electronic Desktop solves, such as Materials, Scripts, and Components, Symbols and Models for Simplorer, Icepak, and Circuit.



- 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.

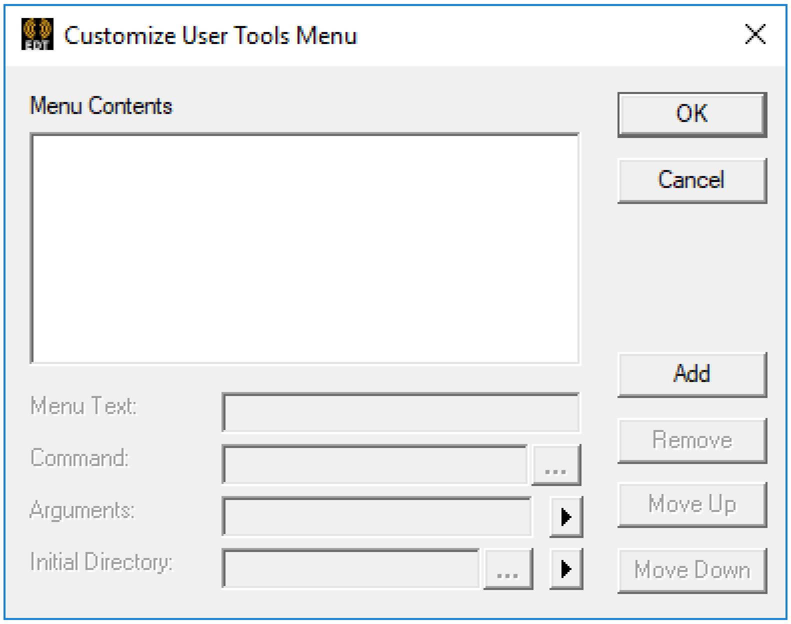
- Click **Tools> Command Window...** to open an IronPython command window. See the Scripting Help for more information.
- Click **Tools>Debug Logging** to enable debug logging via an environment variable. When enabled, the **Debug Logging** is disabled and **Stop Debug Logging** is enabled.
- Click **Tools> 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.

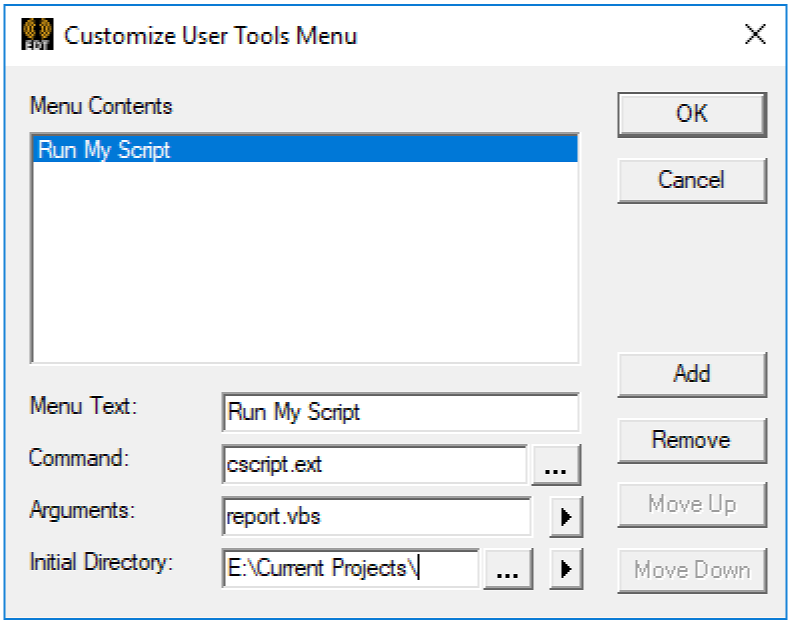
   Command field – displays the external executable. An ellipsis button [...] lets you navigate to the file location.

   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 Circuit or Cancel to close the dialog without changes.

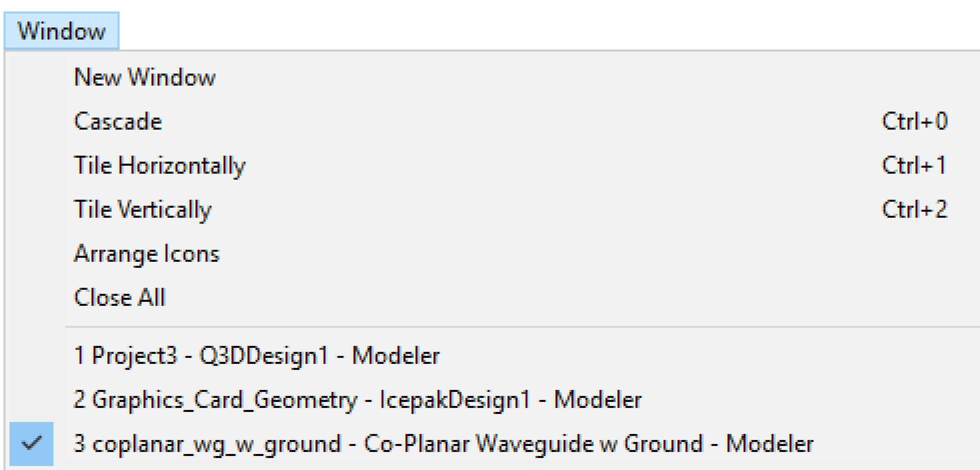
   You can also add scripts to the **Tools** menu. Assuming you have a script to generate custom reports called report.vbs, use the cscript.exe program to execute your script.



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:

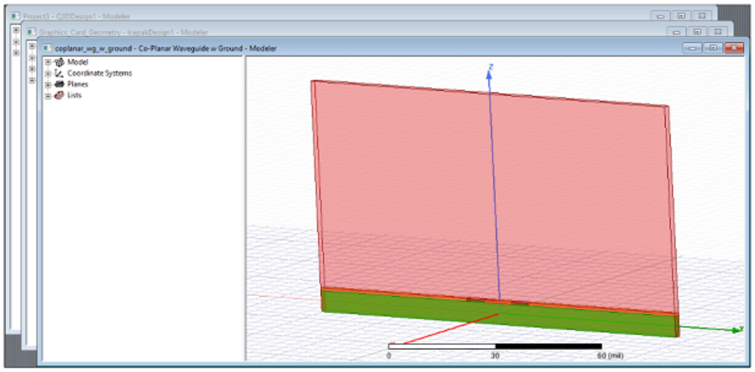
You can use the **Cascade** operation to restore all windows to their default sizes after one or more of them have been maximized (enlarged to fill the entire Design Area).

- 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:

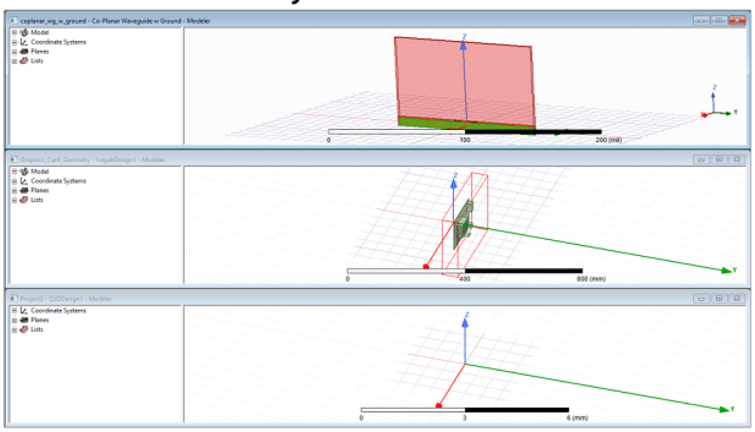
> **Tip:**
>
> You can use the shortcut key sequences Ctrl+0, Ctrl+1, and Ctrl+2 to execute the Cascade, Tile Horizontal, and Tile Vertical operations, respectively. The shortcuts can be used at any time, bypassing the Window menu.

- 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.

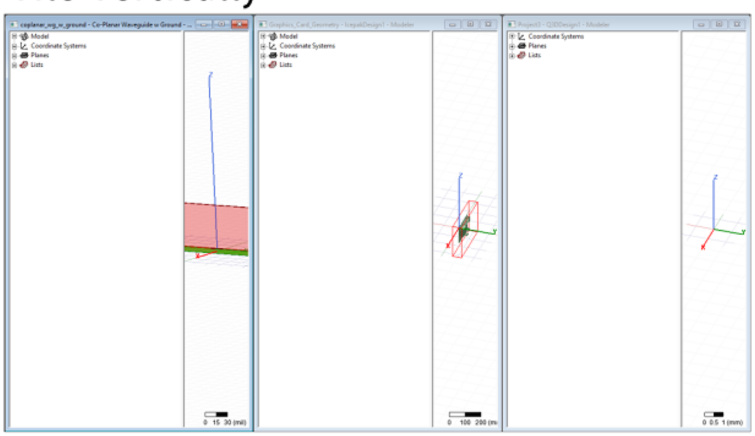
Cascade



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 Circuit are:

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

- **Circuit Help** – opens the Circuit help within the Electronics help system. You can also access PDF versions from within the help system.
- **Circuit Scripting Help** – opens the Circuit scripting help.
- **Circuit Getting Started Guides** – opens a list of links to the Circuit Getting Started Guides, which walk you through projects that demonstrate product features.
- **Circuit PDFs** – provides access to PDFs for Circuit, including the main help, scripting guide, and Getting Started Guides.
- **Components** – enabled when you insert a design to the active project, opens the Components help to a section that pertains to the component type.
- **Ansys Customer Support** – opens a browser page to the Ansys Customer Portal. At the website you can learn more about Ansys products and services and log on to contact Ansys technical support staff.
- **What's New in this Release** – opens a PDF that describes *What's New in Ansys Electronics Desktop* for 2024 R2.
- **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.

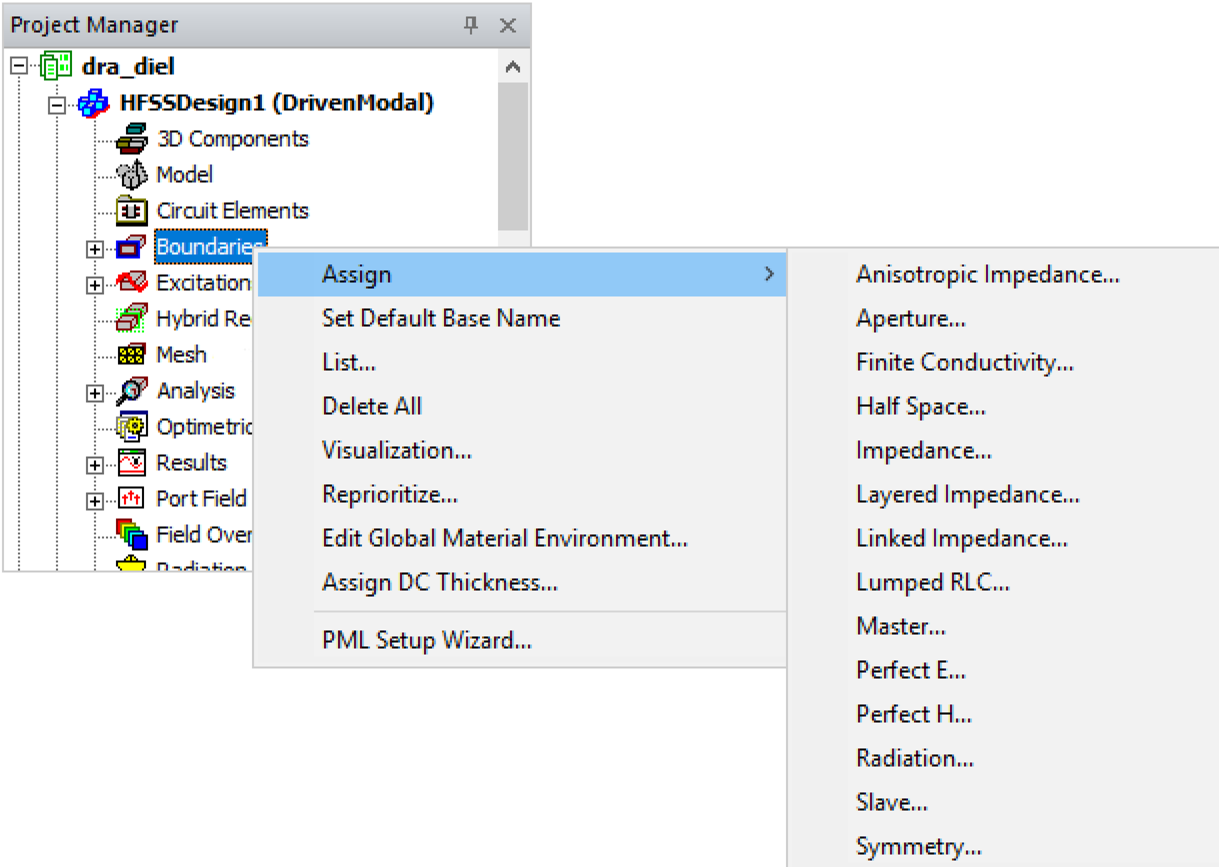
Shortcut menu in the 3D Modeler window	Use the shortcut menu in the **3D Modeler** window to select, magnify, and move objects (zoom, rotate, etc.); change the view; perform boolean operations; assign materials, boundaries, excitations, or mesh operations to objects; and work with field overlays.
Shortcut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

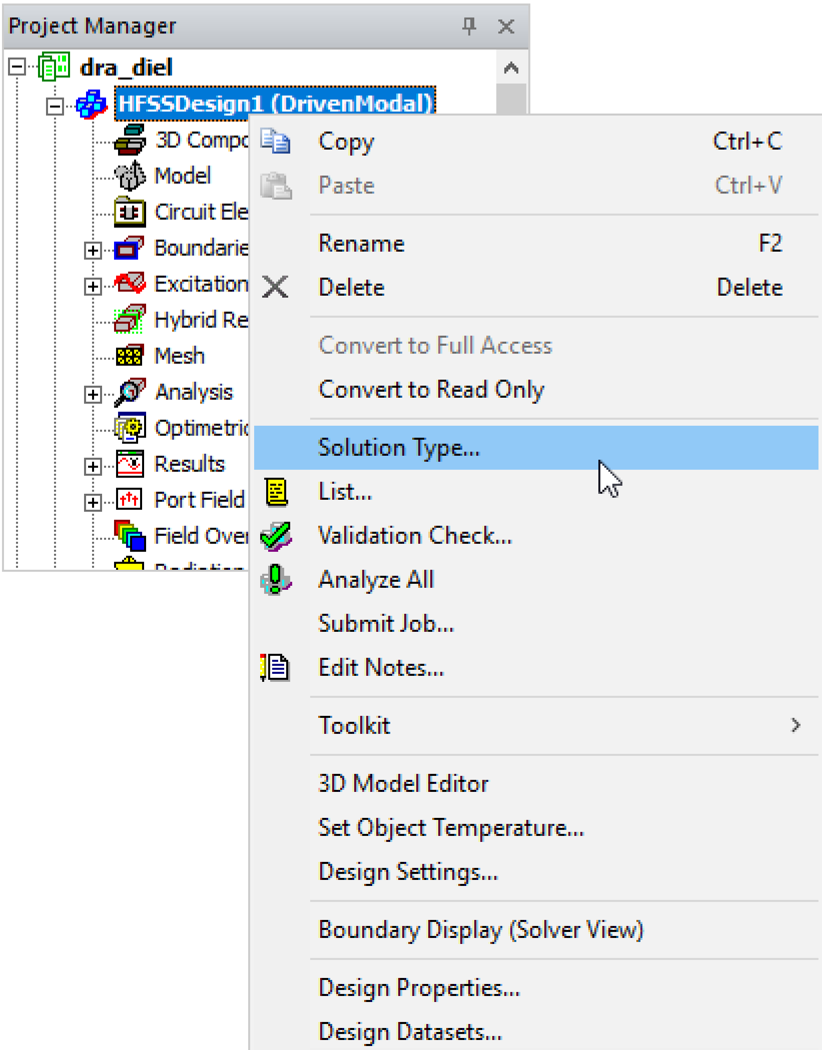
Shortcut Menus in the Project Manager Window

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**.

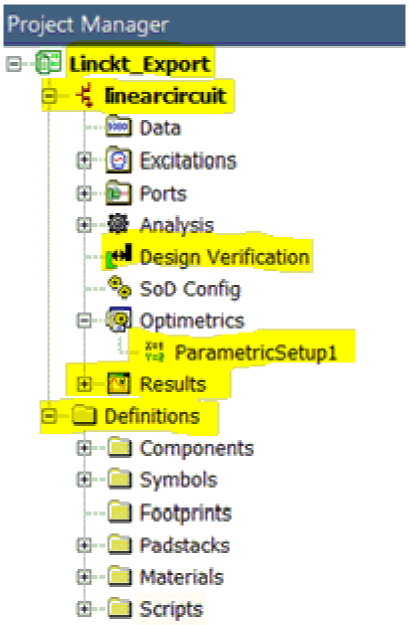
   Valid values are:

   - **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"

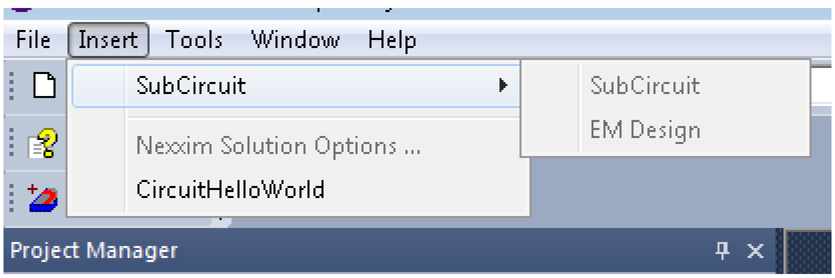
Sample XML Files

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

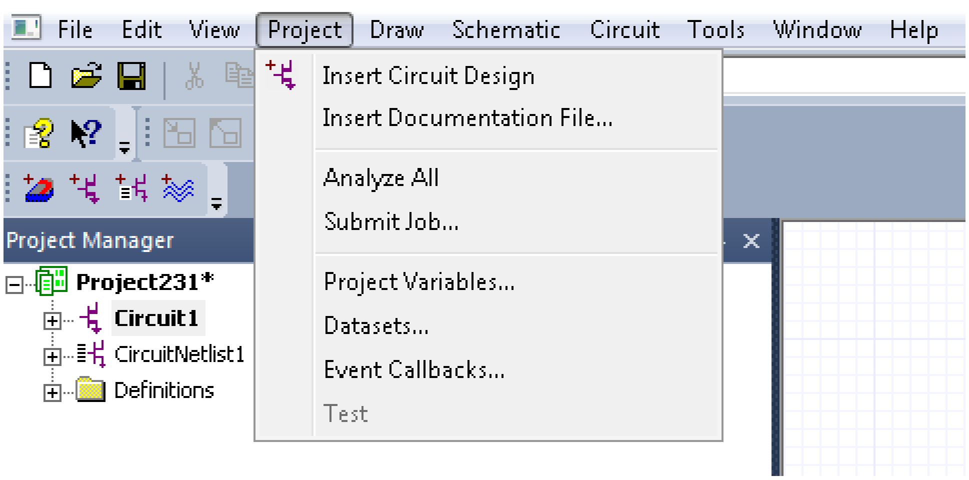


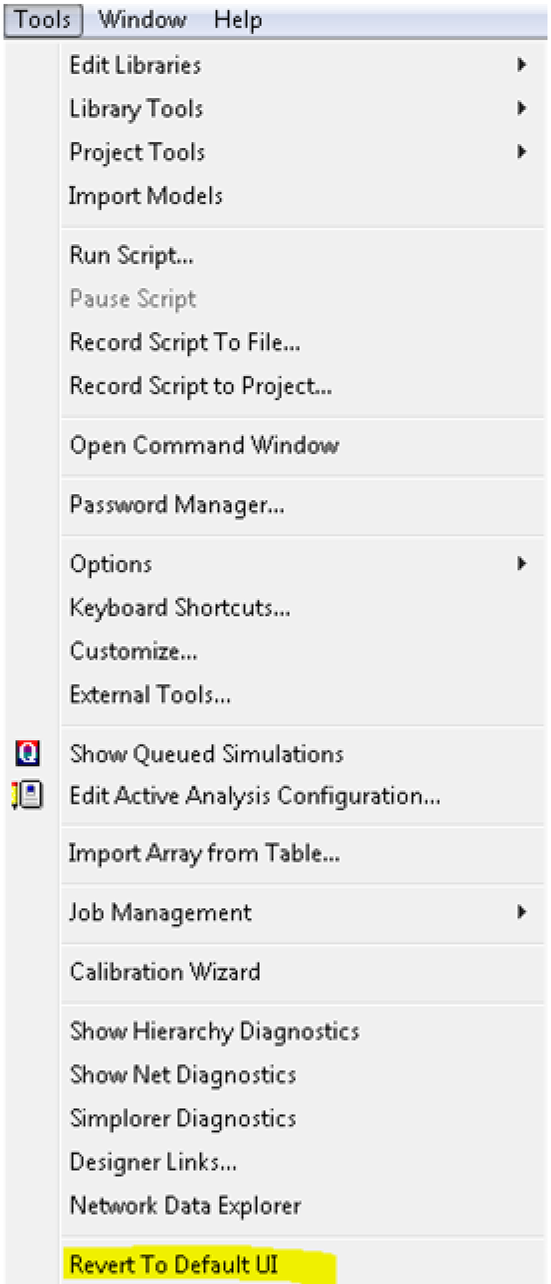
2. **Circuit Design.xml Sample**

```
<?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;Project</MenuName>
      <LeafMenu action="delete">
                 <MenuName>Insert HFSS Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
```

```
            <MenuName>Insert HFSS 3D &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



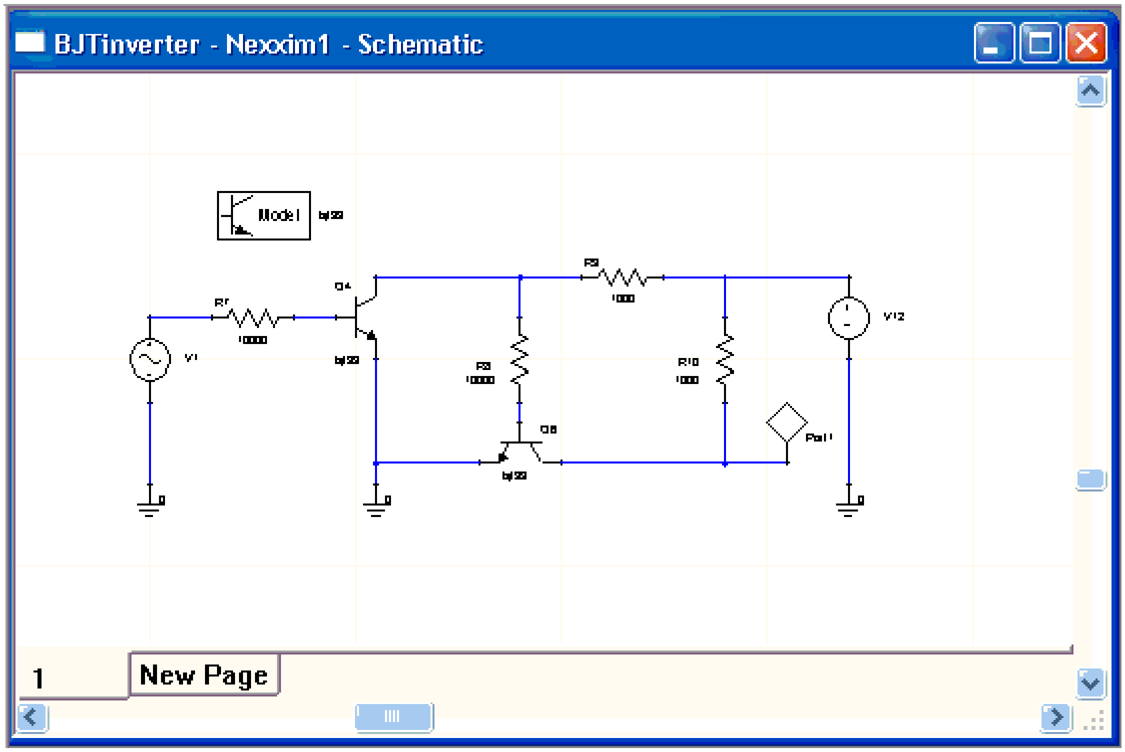


Ansys Electronics Desktop Design Area

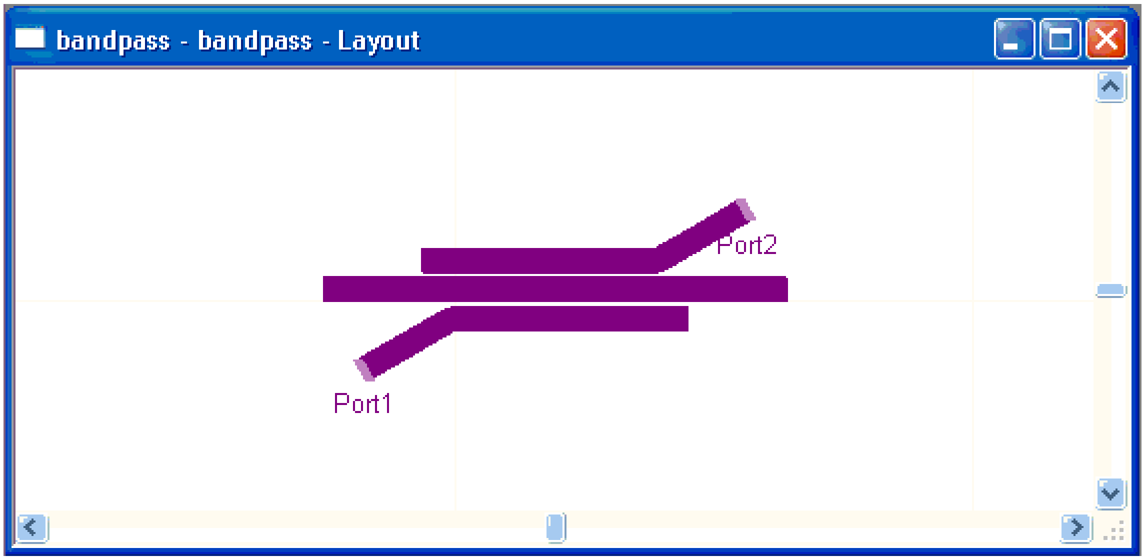
The Design Area of the desktop can display one or more editor windows and report windows, depending on the type of designs you create or load. See the Desktop Windows topic for ways to manipulate the windows in the Design Area.

Schematic Editor Window

The **Schematic Editor** window allows you to place components and wire them together.



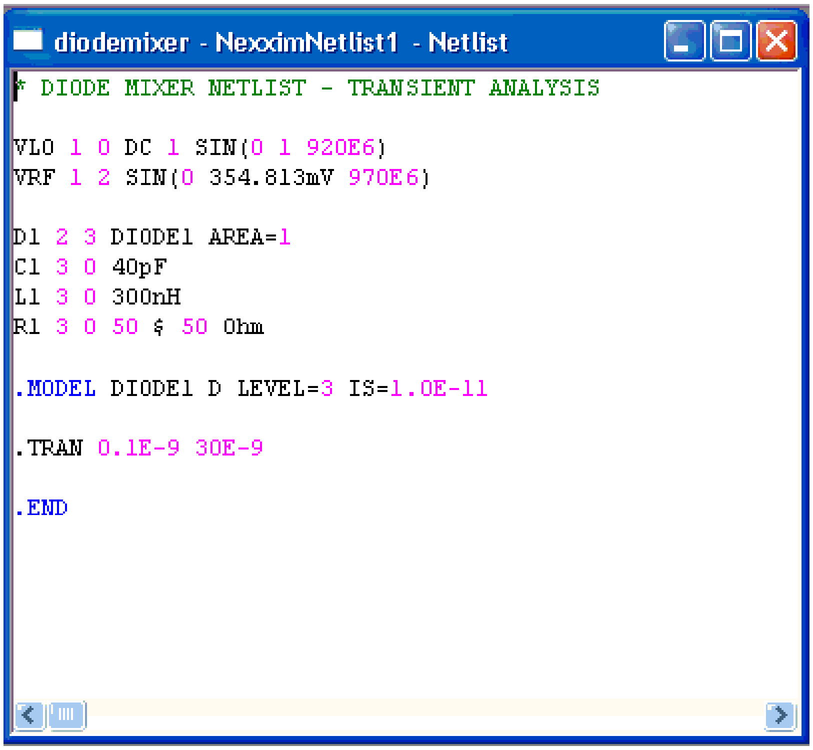
You can move components by simply selecting and dragging them. Copy and paste can be used on components and their wires within the schematic editor. You can also copy and paste to other schematics. For more information, see The Schematic Editor Window.



The Layout Editor shows the physical realization of the circuit. For more information, see Layout Editor.

Netlist Editor Window

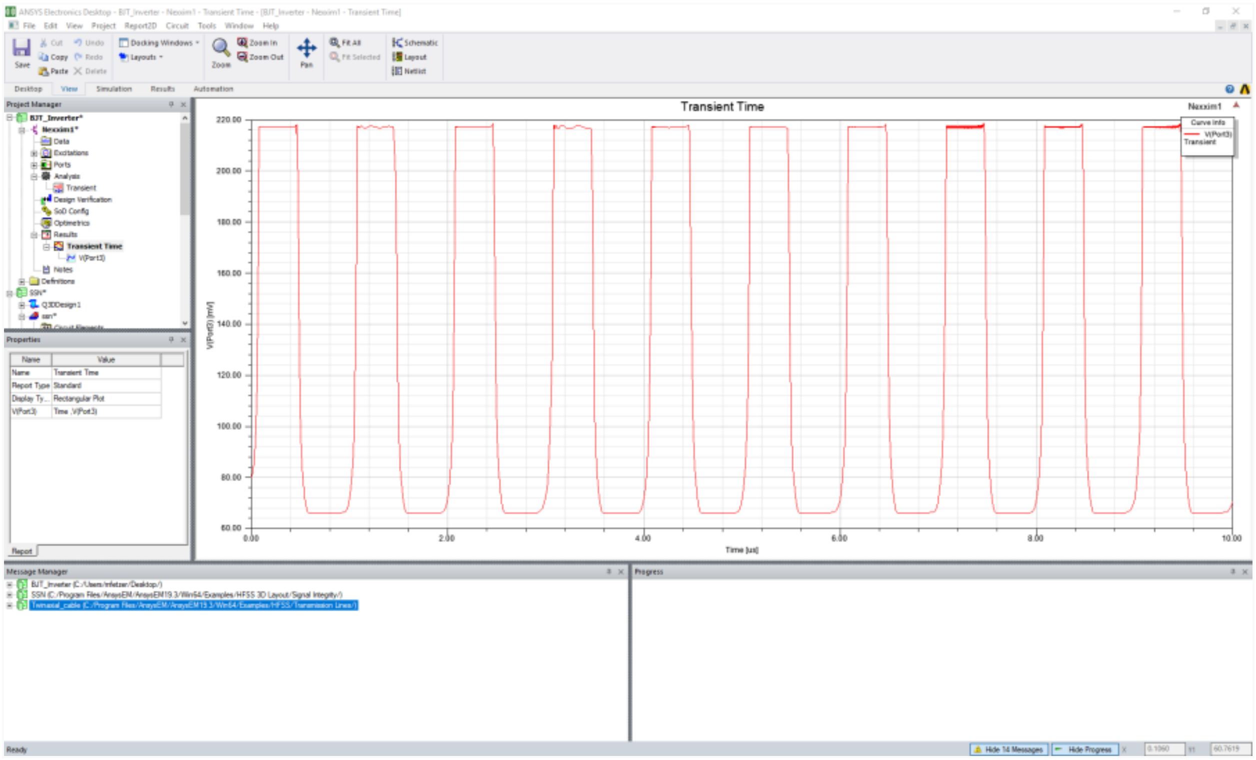
The Netlist Editor is available to edit Nexxim Netlist designs, and to view the netlist generated from the schematic for product designs.



For details, see Netlist Editor.

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:



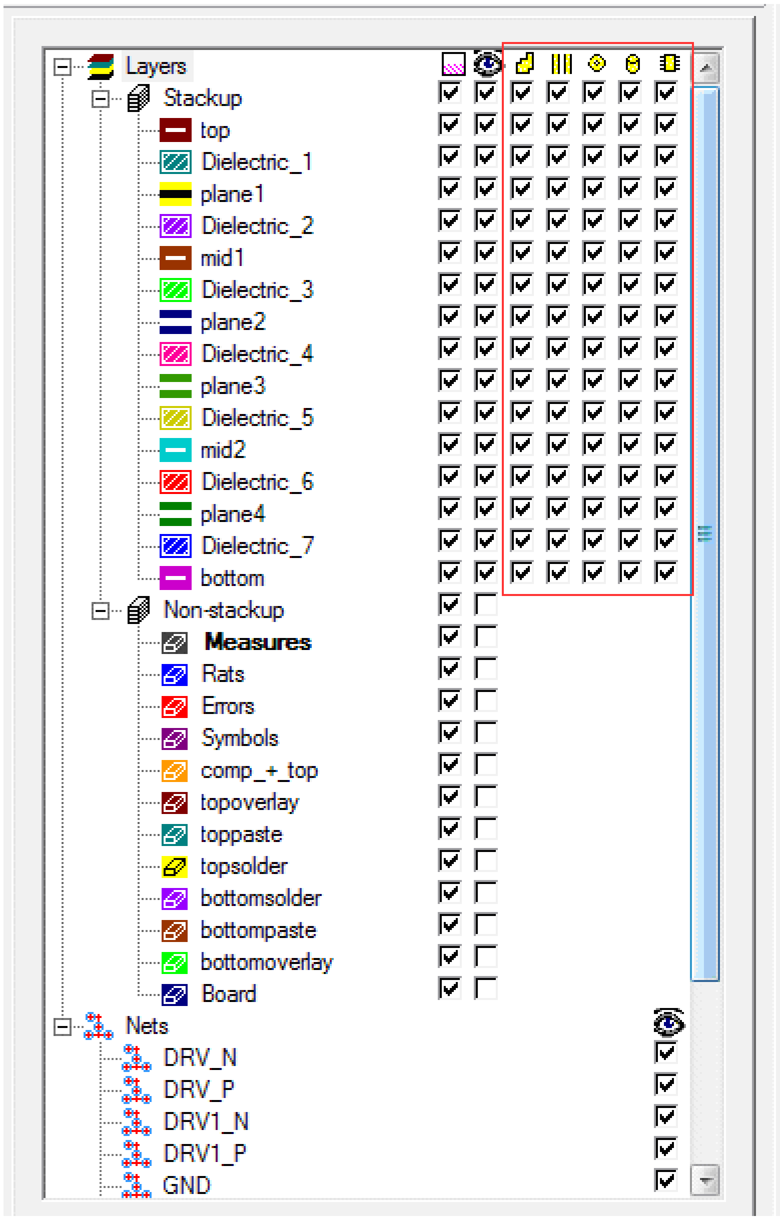
For more information, see Post Processing and Generating Reports.

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

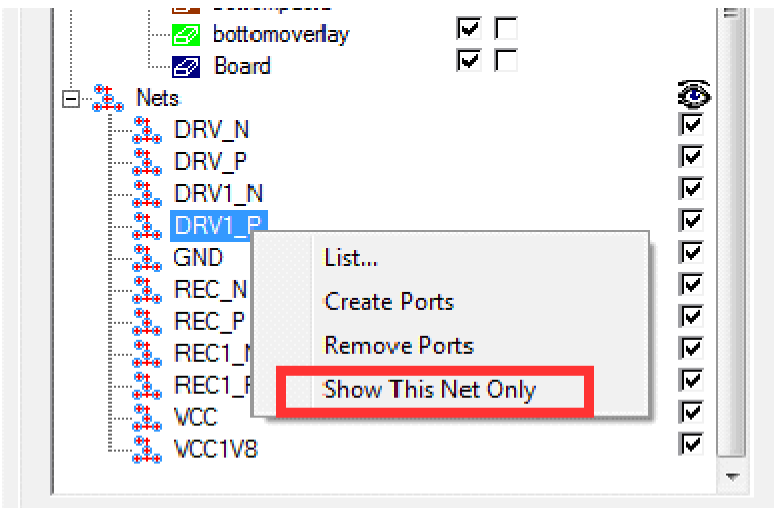
 Controls the visibility of lines (paths)

Controls the visibility of pads

Controls the visibility of holes

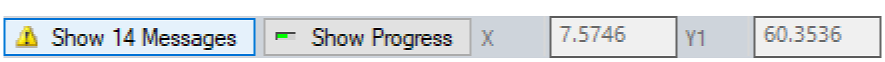
Controls the visibility of components

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

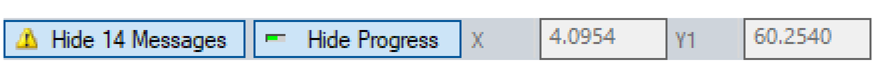


Ansys Electronics Desktop Status Bar

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:

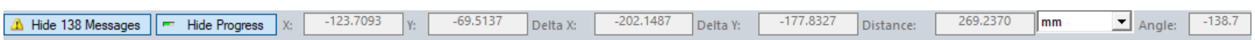
The status bar also displays helpful information about the current selection or command.

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.

Working with the Status Bar and the Layout Editor

The **Status Bar** is located at the bottom of the application window. It displays information about the command currently being performed at lower left, and also displays the following editable values that pertain to the display and positioning of objects selected or drawn in the Layout Editor.

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

- X and Y coordinates
- Delta X and Delta Y
- Distance

- Default Units drop-down menu
- Angle

You can turn off the status bar display by de-selecting the **Status Bar** check box on the View Pulldown Menu. For more information, see Ansys Electronics Desktop.

Keyboard Shortcuts

The following keyboard shortcuts apply to 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

Schematic Shortcuts

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

The following shortcut key combinations are available when the **Schematic** window is active:

Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + X	Cut
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + A	Select all
Ctrl + "+"	Zoom in
Ctrl + "-"	Zoom out
Ctrl + Q	Zoom area
Ctrl + D	Fit drawing
Ctrl + W	Add wire
Ctrl + K	Cross-probe for selected components in layout editor
Ctrl + MMB + drag	Pan (see note below)
Shift + MMB + drag	Zoom (see note below)

Note:

For view navigation shortcuts (last two rows of the table above):

- **MMB** = Middle Mouse Button.
- These are the default view navigation assignments for 2024 R2. Optionally, you can choose to use the legacy shortcuts that were appliable to version 2023 R2 and earlier. See Choosing the View Navigation Options for more information.

Layout Shortcuts

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

The following shortcut key combinations are available when the **Layout** window is active:

Ctrl + A	Select all
Ctrl + Shift + A	Unselect all
Ctrl + C	Copy
Ctrl + D	Fit drawing
Ctrl + "+"	Zoom in
Ctrl + "-"	Zoom out
Ctrl + K	Cross-probe for selected components in schematic editor
Ctrl + M	Align microwave ports
Ctrl + Q	Zoom area
Ctrl + R	Rotate
Ctrl + V	Paste
Ctrl + W	Add connection
Ctrl + X	Cut
Ctrl + Y	Redo
Ctrl + Z	Undo
Delete	Delete
MMB + drag	Rotate (see note below)
Ctrl + MMB + drag	Pan (see note below)
Shift + MMB + drag	Zoom (see note below)

Note:

For view navigation shortcuts (last three rows of the table above):

- **MMB** = Middle Mouse Button.
- These are the default view navigation assignments for 2024 R2. Optionally, you can choose to use the legacy shortcuts that were appliable to version 2023 R2 and earlier. See Choosing the View Navigation Options for more information.

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

Netlist Editor 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 Netlist Editor window is open:

Ctrl + A	Select all traces
Ctrl + B	Delete all bookmarks
Ctrl + C	Copy
Ctrl + F	Find
Ctrl + G	Go to line number
Ctrl + R	Replace
Ctrl + V	Paste
Ctrl + X	Cut
Ctrl + Y	Redo
Ctrl + Z	Undo
Delete	Delete
F2	Next bookmark
Ctrl + F2	Toggle bookmark
Shift + F2	Previous bookmark

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):

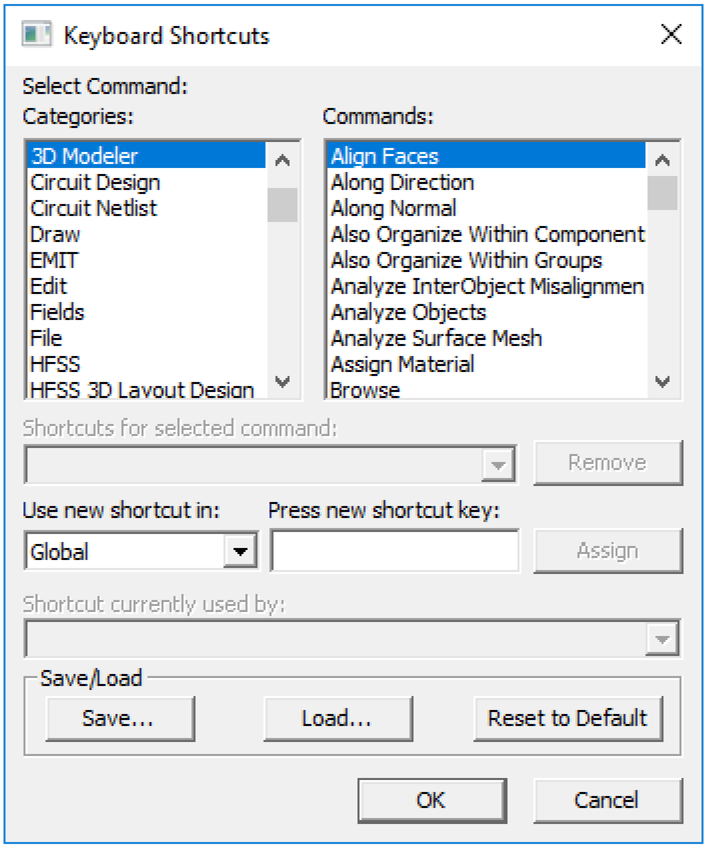
> **Note:**
>
> The *"Current"* assignments are available whether or not the *Enable Legacy View Navigation* option is selected:

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.

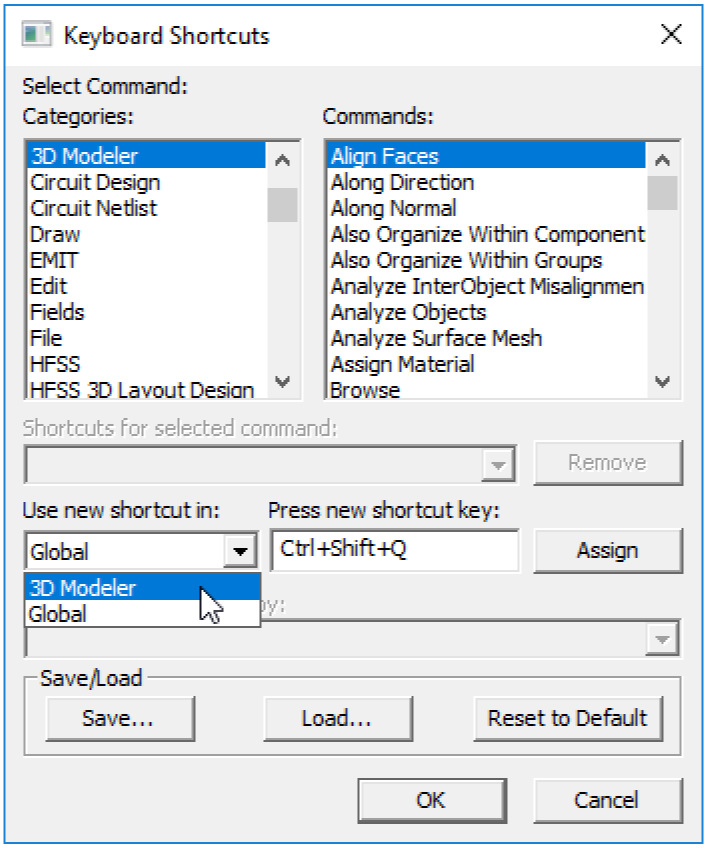
To create a new shortcut key:

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.

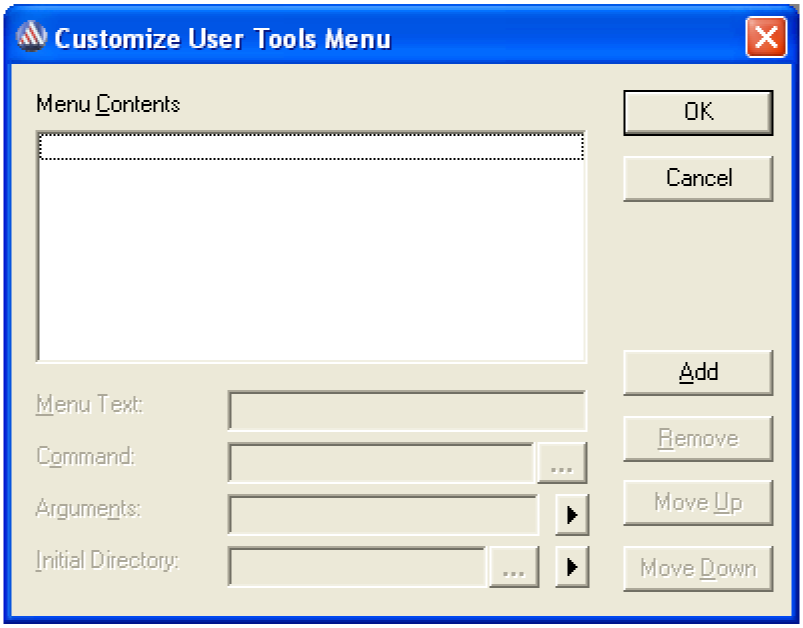
   If you have an existing *.aks file, use the **Load...** button to locate and select the file.

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

Adding External Tools

To add an **ExternalTools** menu to Ansys Electronics Desktop:

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



If a **User Tools** menu has been previously defined, its contents are displayed in the **Menu Contents** window. Navigation buttons allow you to **Add**, **Remove**, **MoveUp** and **MoveDown** items which are displayed in the **Menu Contents** window.

2. Click **Add**. This enables the following fields:

Menu Text — Displays the text you want to appear in the **UserTools** menu.

Command – Displays the external executable. Click the browse button to navigate to the file location.

Arguments – Specifies the argument to be associated with the command. Click the right-arrow button to choose from the following: File Path, File Directory, File Name, File Extension, Project Directory, or Temp Directory.

Initial Directory – Specifies an initial directory for the command to operate in. Click the browse button to navigate to the file location. Click the right-arrow button to navigate through your desktop or across the network.

3. Click **OK** to add the **ExternalTools** menu to Ansys Electronics Desktop, or click **Cancel** to close the dialog box without changes.

> **Note:**
>
> To execute a custom script using the **ExternalTools** customization, enter the name of the program that will execute the script in the **Command** field and enter the location of the script in the **Arguments** field. For example:
>
> For Windows, enter the name of the non-graphical Windows Script Host — 'cscript' — in the **Command** field and enter the location of the script in the **Arguments** field:
>
> <u>VBScript</u>
>
> Command: cscript
>
> Arguments: C:\scripts\myvbscript.vbs
>
> <u>JavaScript</u>
>
> Command: cscript
>
> Arguments: C:\scripts\myjavascript.js

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

5. In the Password Manager, select an existing resource to highlight it.
6. Click **Encrypt File**.

   A file browser window appears.

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

   The choices are Circuit files (*.lib) and Ansoft Library files. Any existing resources in the selected directory will appear.

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

  If migration from ACIS to Parasolid is needed, and –outputfolder is not specified, the original file will be saved in the AnsysEM_Backup subdirectory (located in the same directory as the project being opened). Then, the file being opened will be overwritten with the migrated file. (If –outputfolder is specified, the original files are not modified because the modified files would be written to the output folder.)

  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 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.

  preservehistory=<true or false>: preserve geometry history during migration.

- **-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***>:<***GPUsOnMachine***>**

      Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. 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***>:<***GPUsOnMachine***>**

  Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. 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\AnsysEM\v242\Win64\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\AnsysEM\v242\Win64\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 2024 R2 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
- For -batchoptions Use: HFSS 3D Layout Options with Paths

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

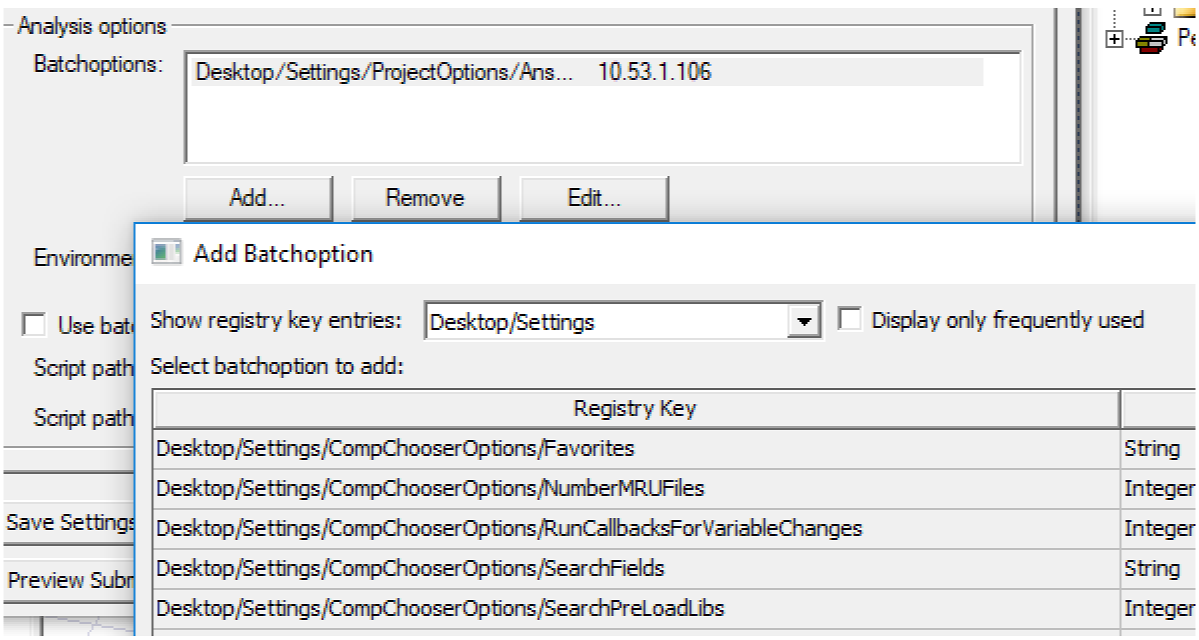
For -batchoptions Use: TempDirectory

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

Registry Key	Default Value	Units or Values	Description
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
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

Registry Key	Default Value	Units or Values	Description
Desktop/Settings/ProjectOptions/DrawStateIconsInProjectTree3	1 (true)	0 (false) or 1 (true)	Change icon when selection does not match active window
Desktop/Settings/ProjectOptions/ExpandMessageTreeOnInsert	1 (true)	0 (false) or 1 (true)	Ensure that new messages are visible in the message window tree
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.

Batchoptions: Override Registry Entry

Any registry entry can be overridden via a command-line option.

Example: batchoptions specified on a command line

```
ansysedt -batchsolve -batchoptions
"'Desktop/Preferences/ProjectOptions/NumberOfProcessors'=4 'HFSS 3D
Layout Design/NumCoresPerDistributedTask'=2" projectname.aedt
```

Example: batchoptions specified in a file

```
ansysedt -batchsolve -batchoptions numprocfile projectname.aedt
```

where numprocfile contains:

```
$begin 'Config'
  'Desktop/Preferences/ProjectOptions/NumberOfProcessors'=4
  'HFSS 3D Layout/NumCoresPerDistributedTask'=2
$end 'Config'
```

Running Ansys Electronics Desktop from a Windows Remote Terminal

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

Before running a job you must 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.

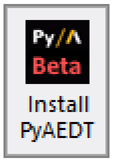
Documentation for PyAEDT can be found online at: https://aedt.docs.pyansys.com/version/stable/

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.

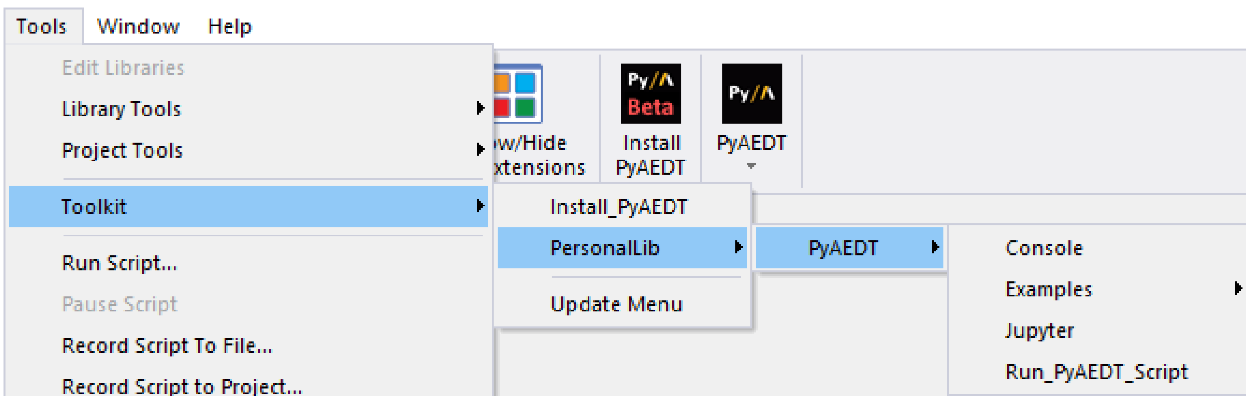
To install PyAEDT:

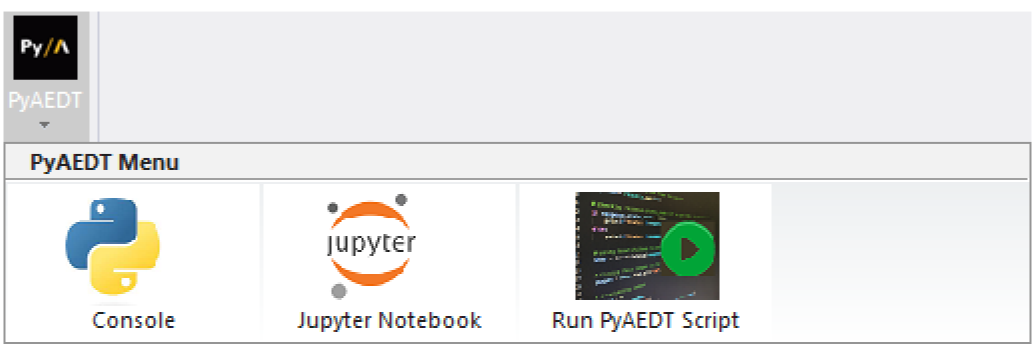
- From the **Automation** tab, click **Install PyAEDT**.



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:





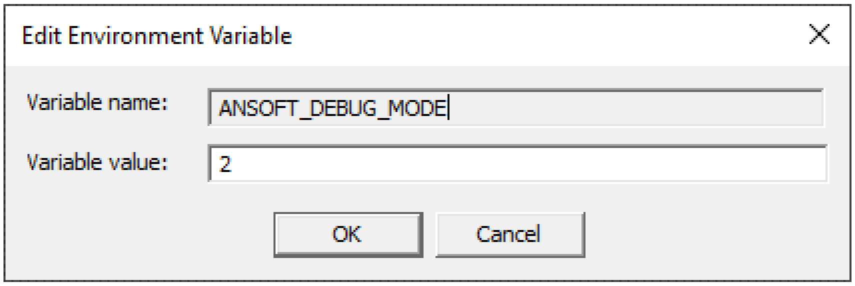
Debug Logging

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.

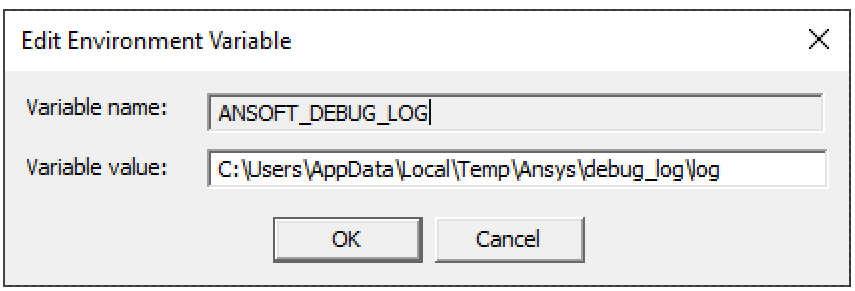


2. Enter a **Variable value**.

   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**.

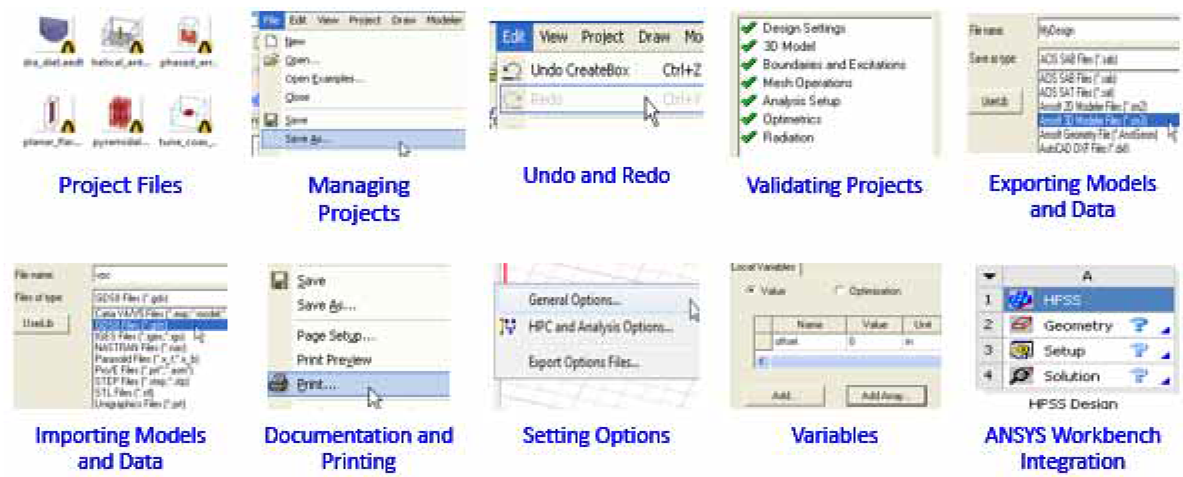
   Errors are logged to the specified folder until you disable logging.

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

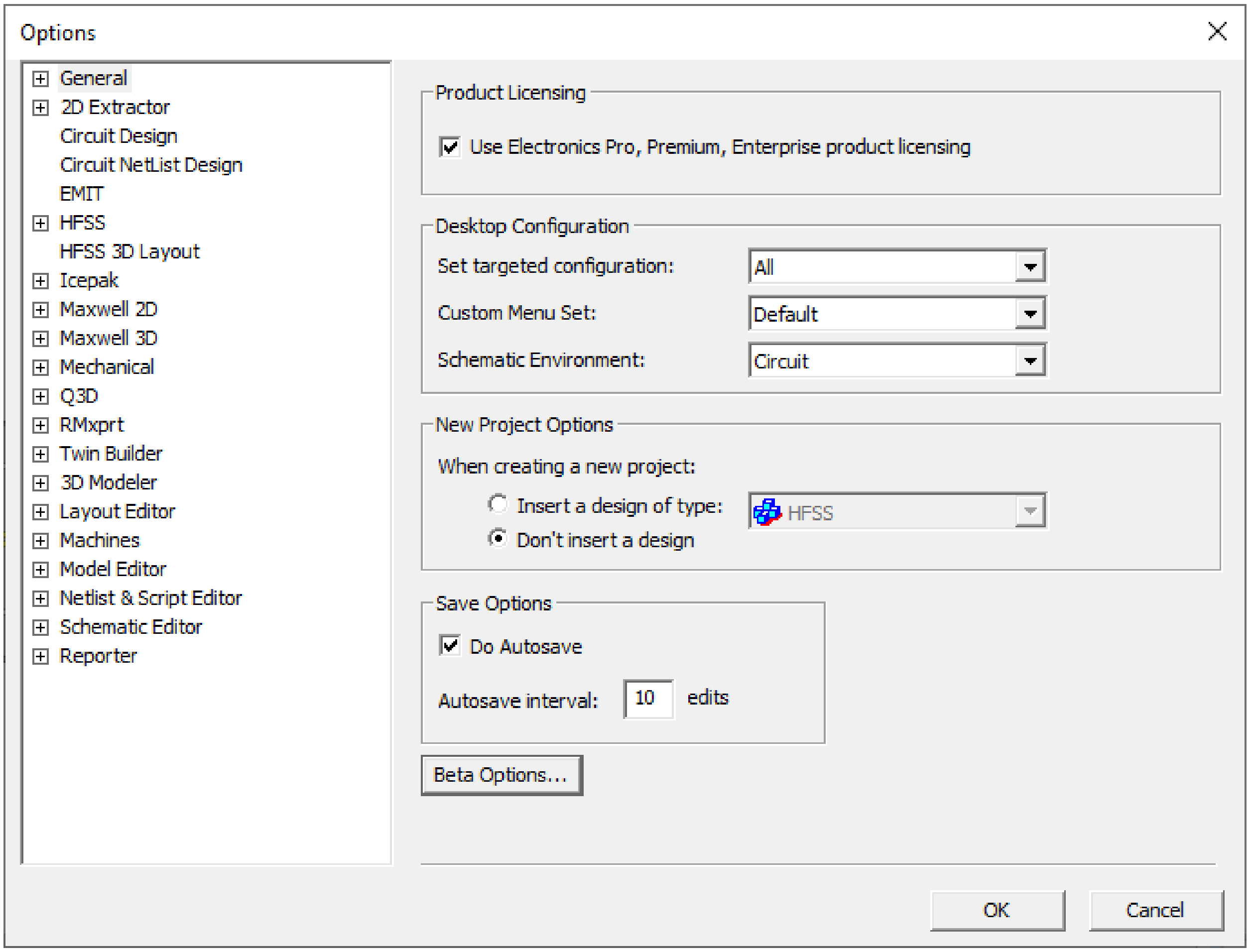
When you create any project in Ansys Electronics Desktop, it is given an .aedt file extension and stored in the directory you specify. For Legacy projects, the transition is handled as follows:

- 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.

- 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.

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 3D Layout

- HFSS 3D Layout
- Icepak
- Maxwell 2D
- Maxwell 3D
- Mechanical
- Q3D
- RMxprt
- Twin Builder
- 3D Modeler
- Layout Editor
- Machines
- Model Editor
- Netlist and Script Editor
- Schematic 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.

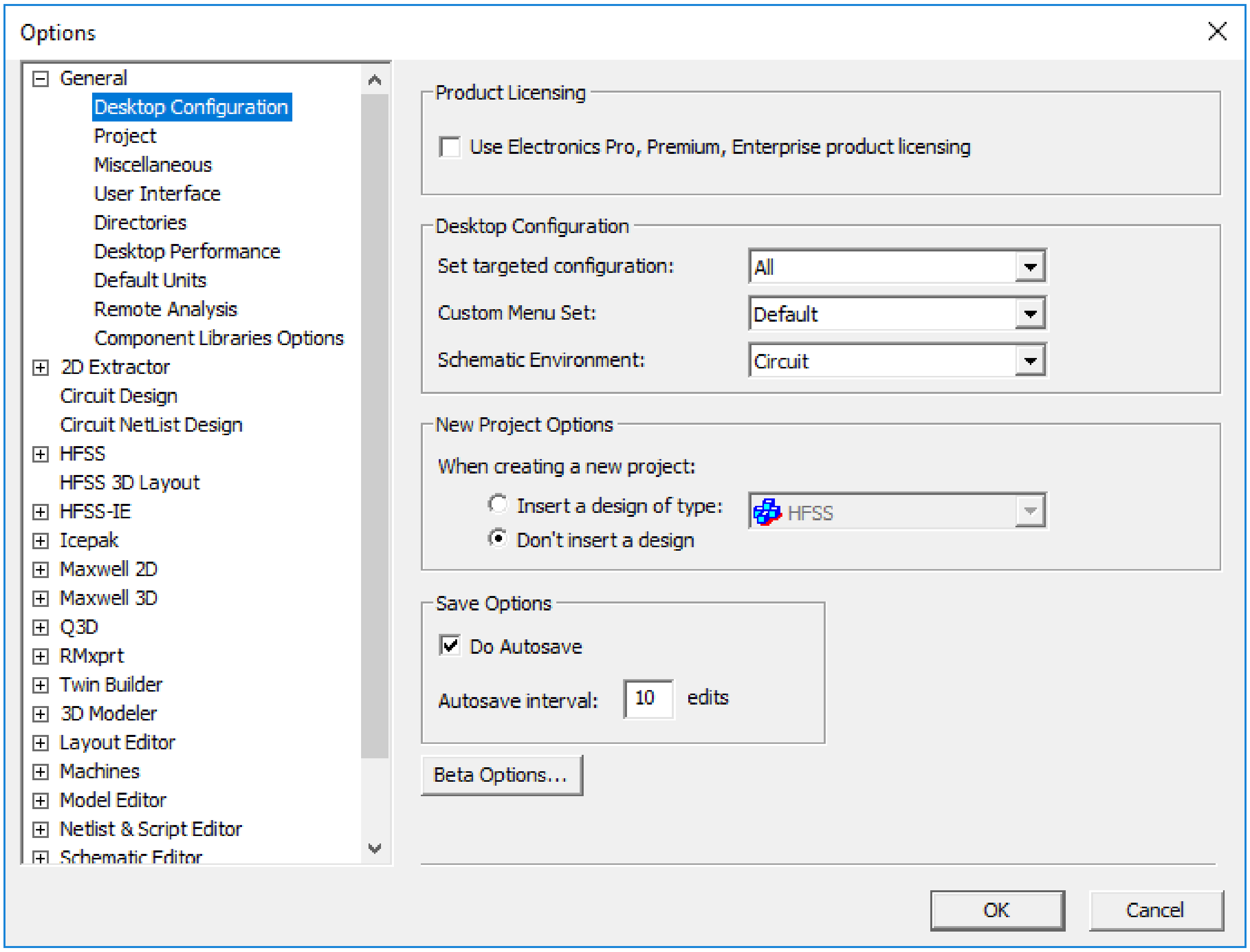
The **Tools** > **Options** > **Export Options Files** command writes xml files containing the Options settings at all levels to the specified directory. The **Tools** > **Options** > **Export Options** feature is intended to make it easier for different users to use Ansys Electromagnetics tools installed 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.



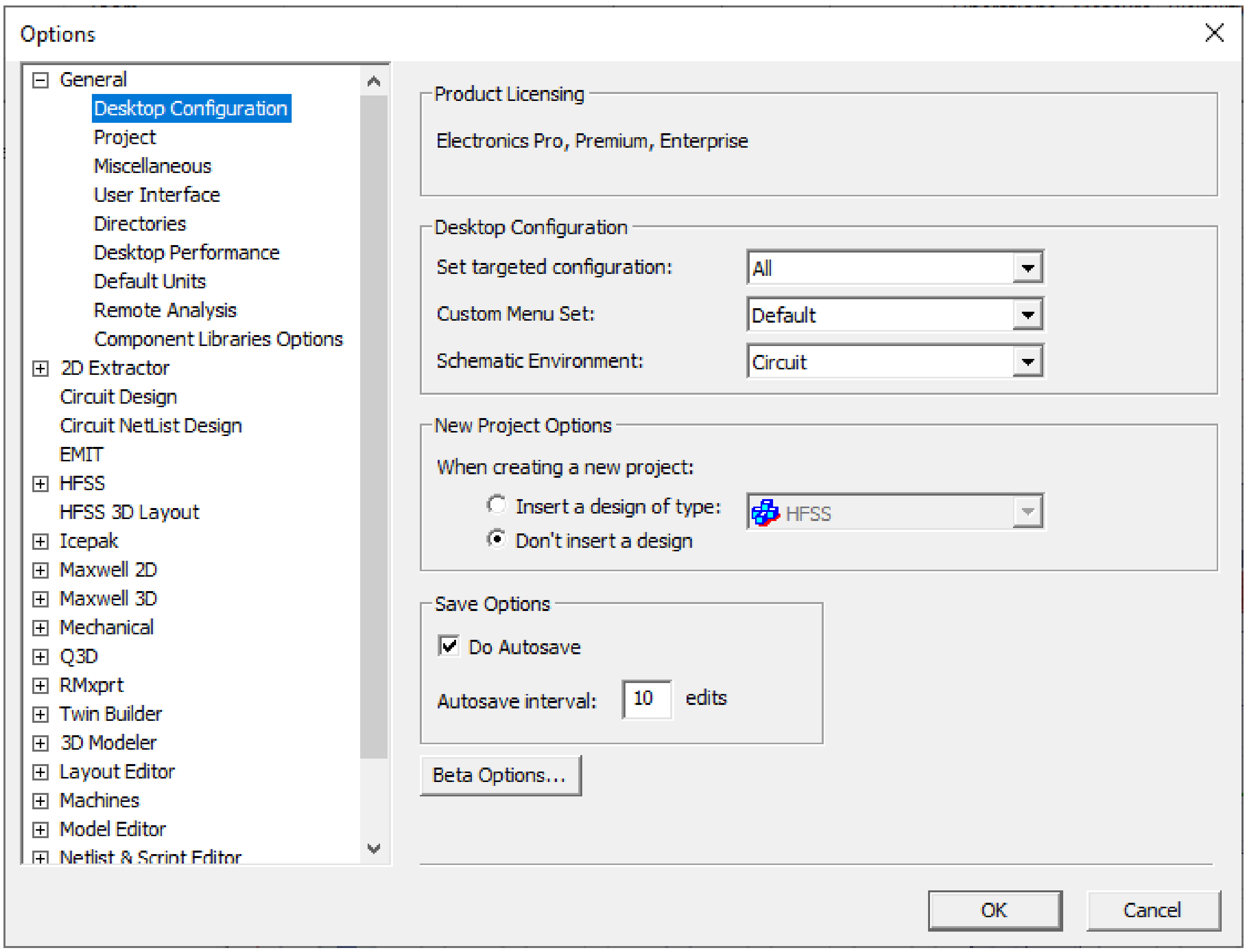
In the left pane, select the entries below **General** to display the associated options:

- 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** to apply your preferences.

General Options: Desktop Configuration

Under General Options, **Desktop Configuration** options allow you to customize Electronics Desktop in a way that suits your work priorities. These include options for 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**.

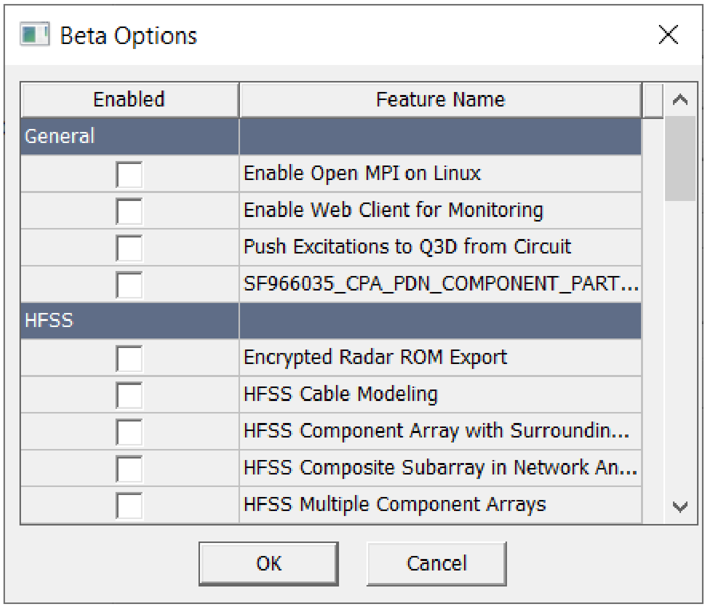
In the **New Project Options** area, options include:

- **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.

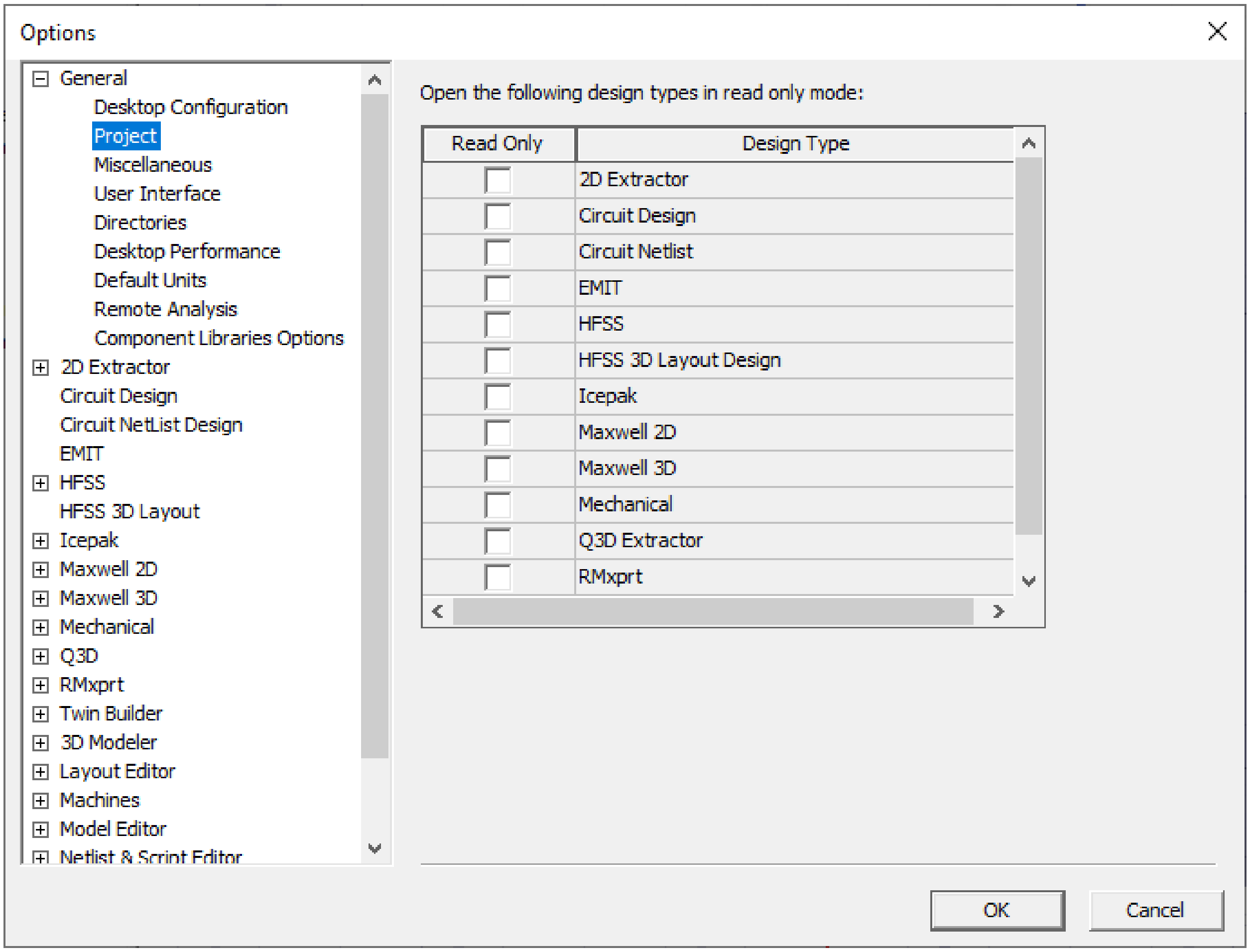
Click the **Beta Options** button to open a window listing beta options. You may need to scroll or size the window to view all options.



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

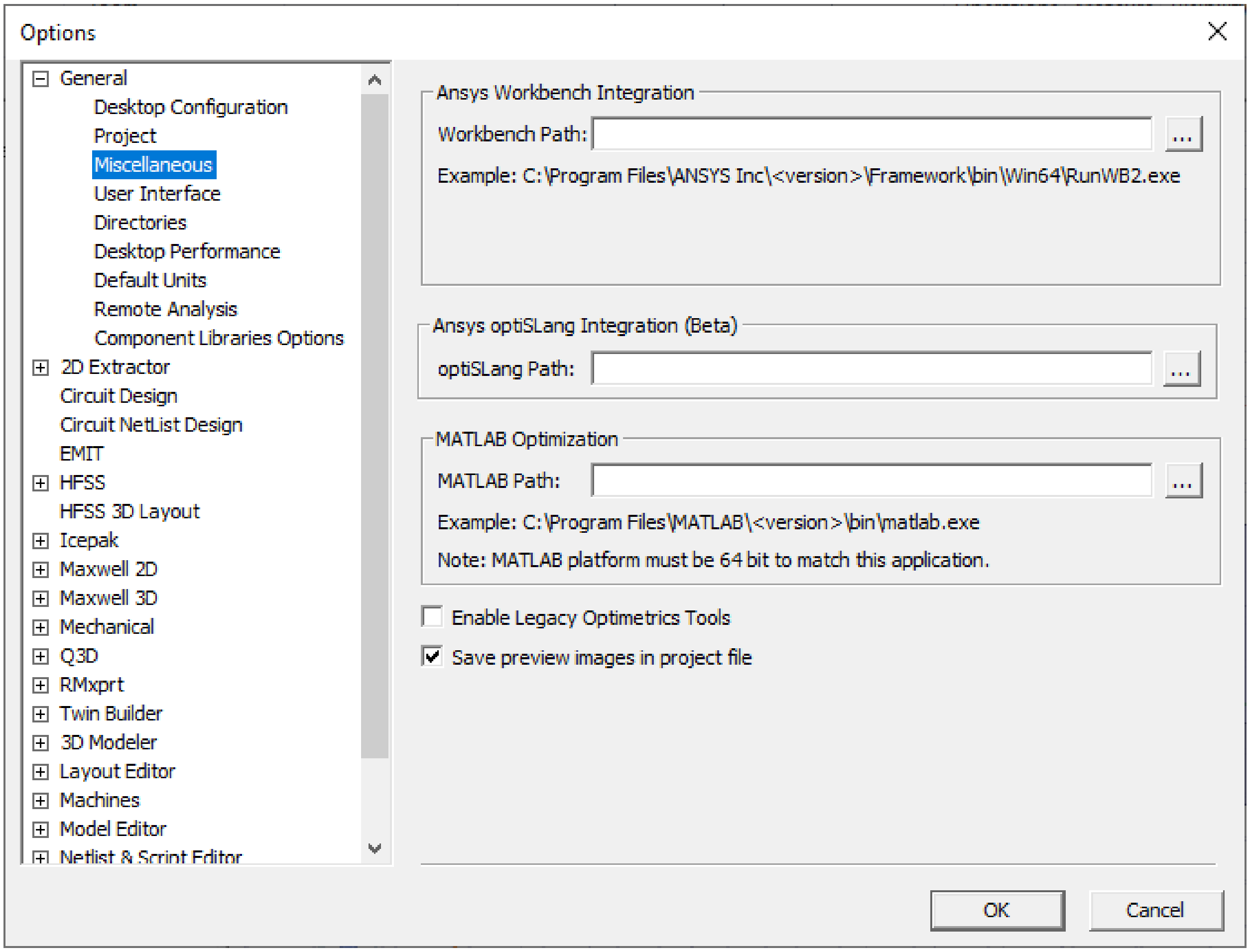
General Options: Project Options

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.



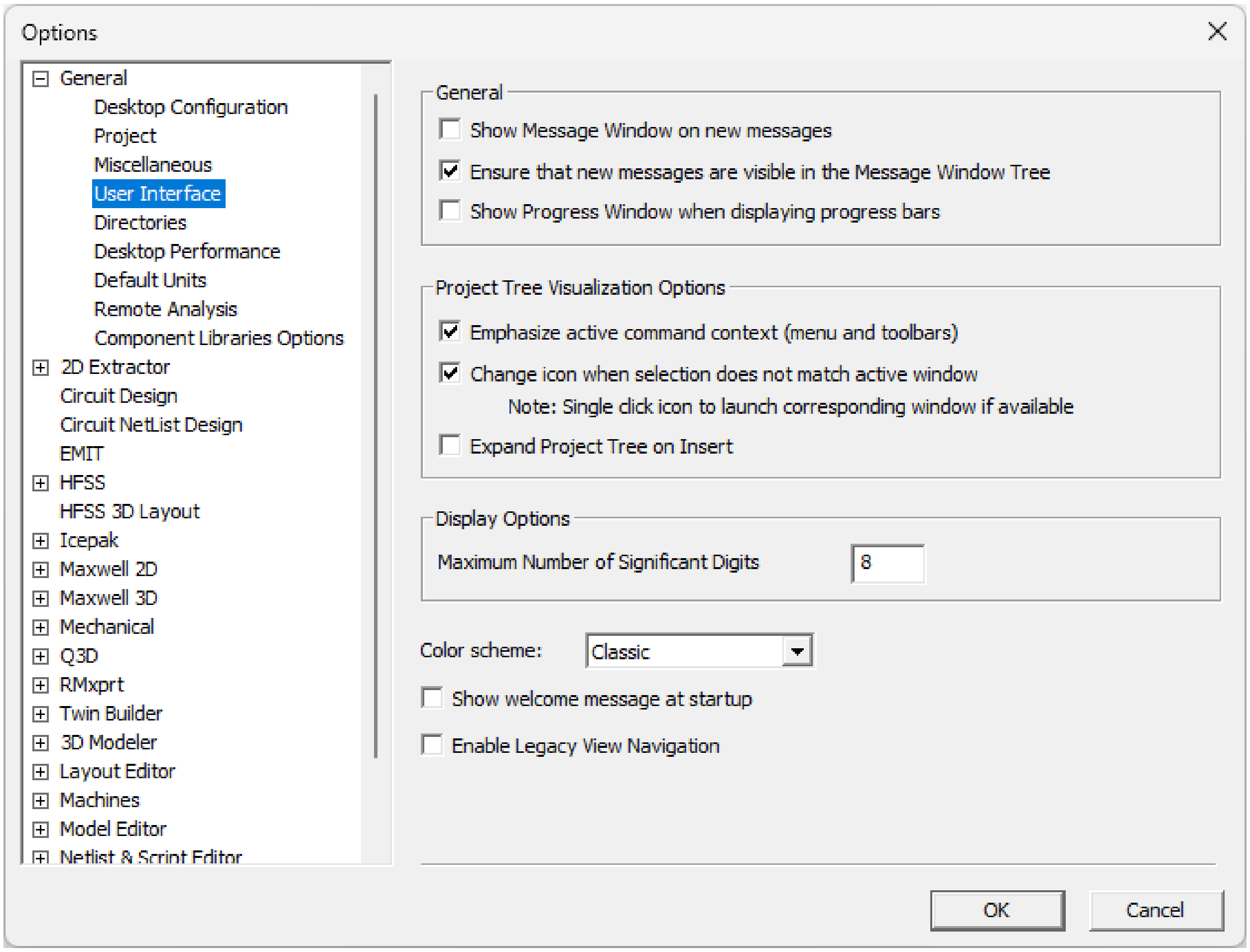
Ansys Workbench Integration allows you to specify the path to your Ansys Workbench installation. This path can be used in Optimetrics for connecting to the Design Xplorer.

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.

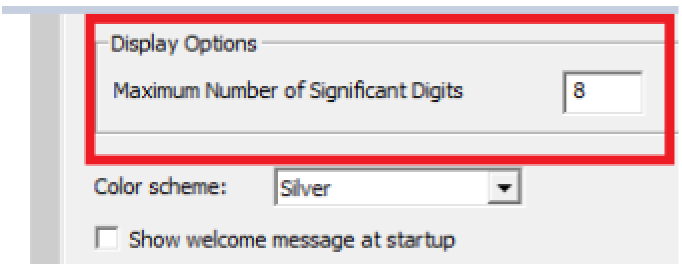
In the **Project Tree Visualization Options** area, options include:

- **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

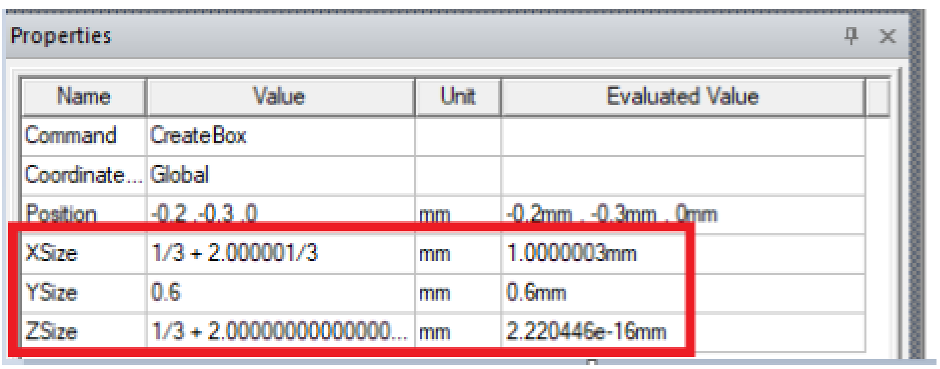
selected project item (data affecting the same model is considered to be related). Clicking the icon opens the window and brings it into focus.

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

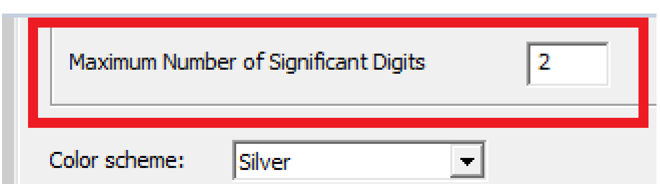
In the **Display Options** area, specify the **Maximum Number of Significant Digits** to display. The default is 8 and the maximum is 20. This affects the digits displayed in the **Solutions** dialog box, evaluated variable values, Animation dialog boxes, Optimetrics, Reports, and so forth. For example,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



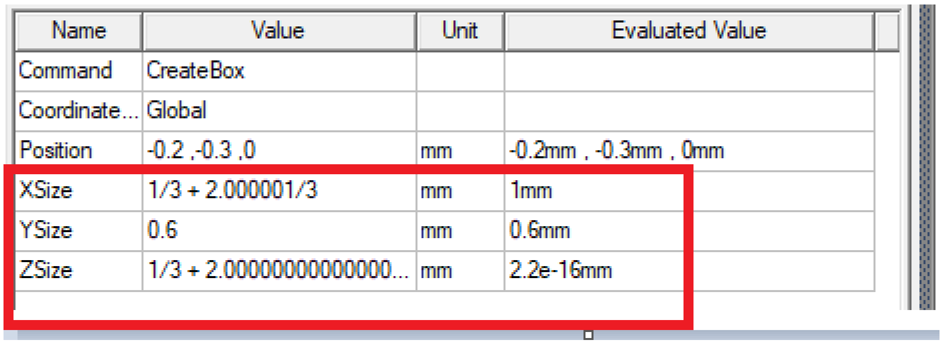
So some evaluated variable values with significant digits appear as follows.



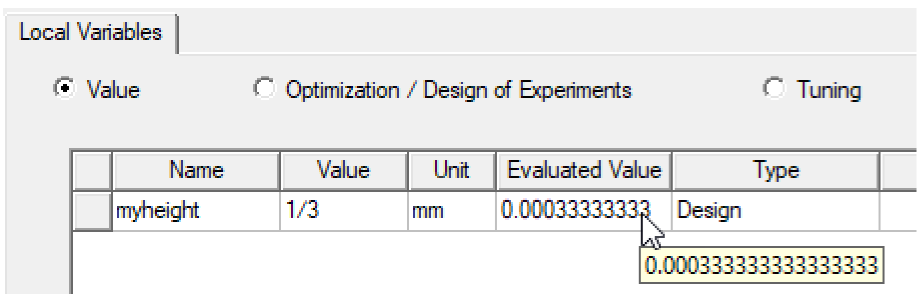
Change Precision = 2. This lets you display near-zero values in the Properties window as 0.



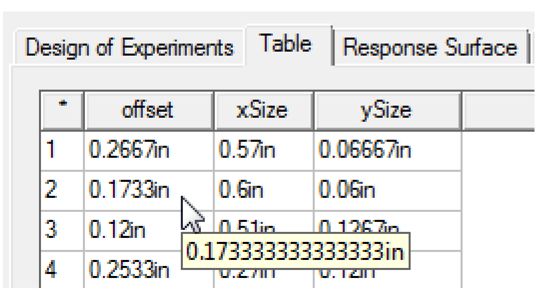
You can see that XSize is now rounded off to cleanly display 1mm. The decimal part of ZSize also shows cleaner value, but it is not rounded off to zero.



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

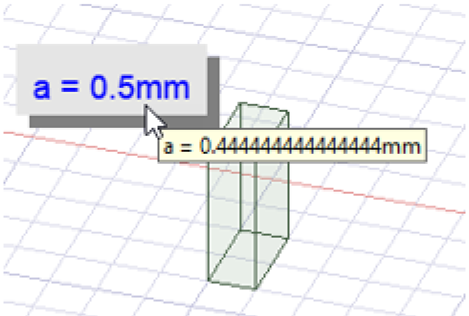


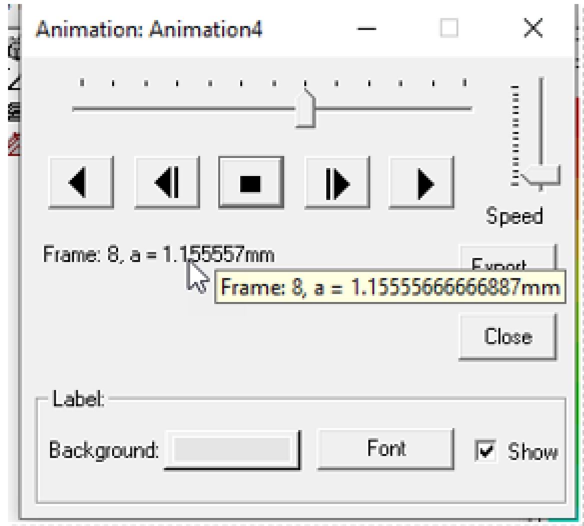
In the case of variable values, if you have assigned more significant digits, you will see these when editing the variable value. In the case of table displays of values, the tooltip display shows all available digits when the mouse pointer is over a result:



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.

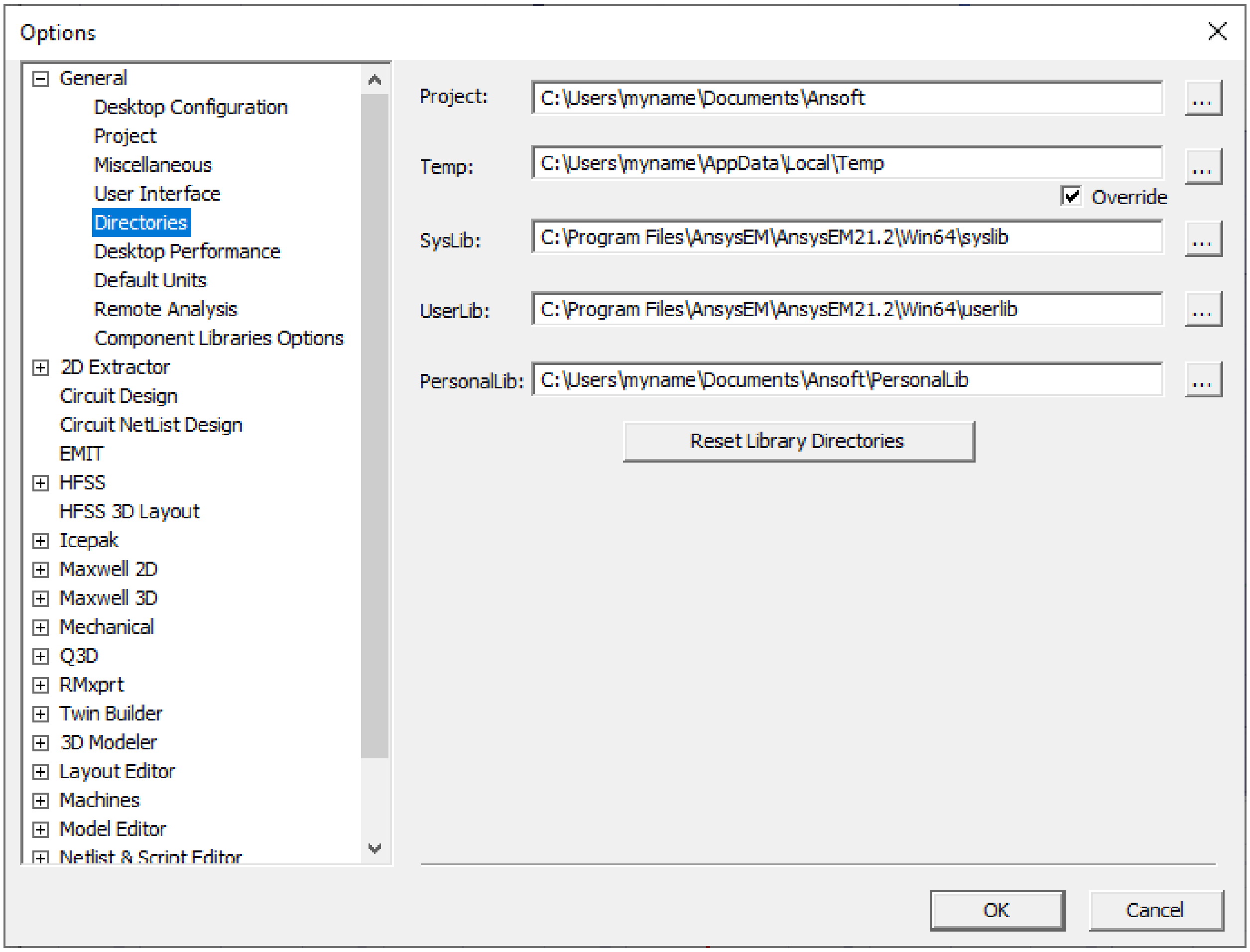
Enable legacy view navigation enables you to choose between two schemes for view navigation keyboard shortcuts and mouse button assignments, as follows:

- **Cleared:** Use only the view navigation mouse-button and hotkey assignments introduced in Ansys Electronics Desktop version 2024 R1 and applicable to subsequent versions. 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.



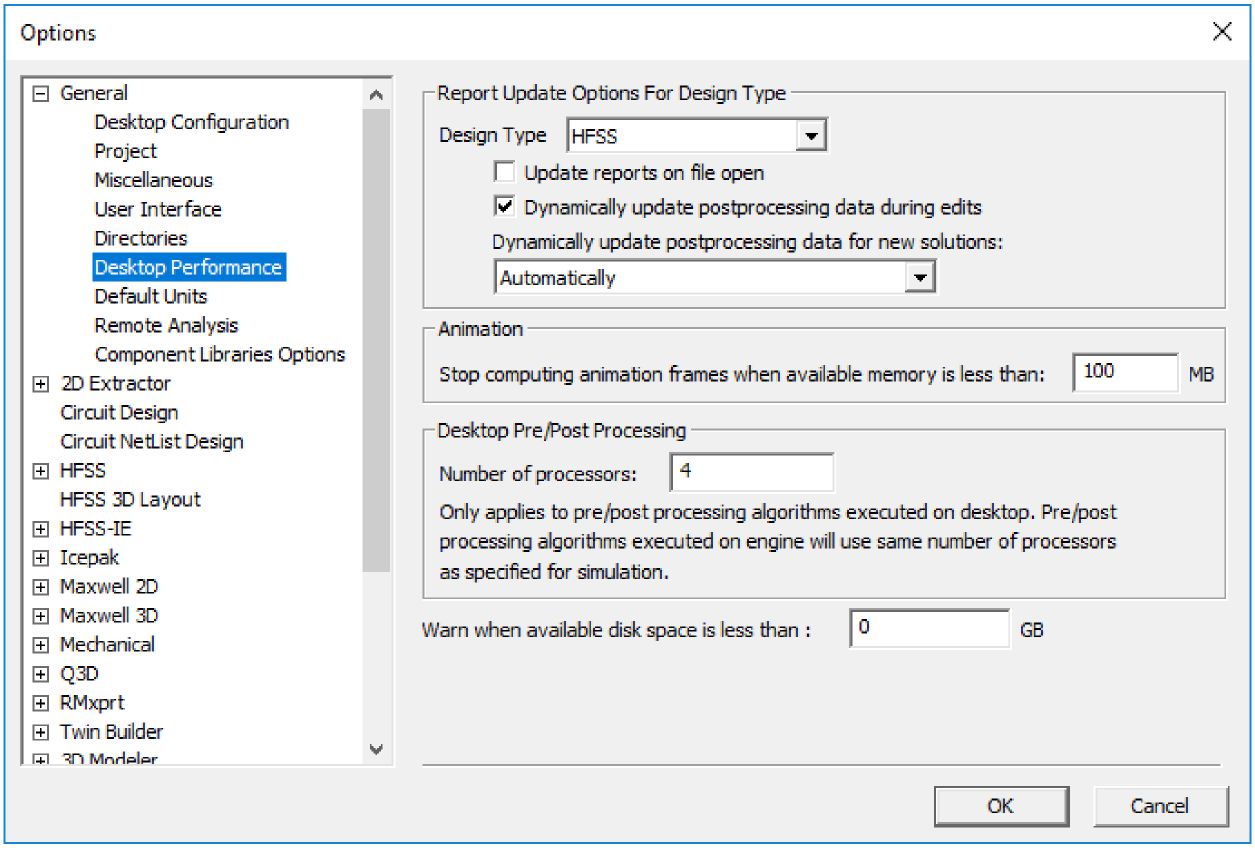
Use the **[...]** buttons to browse to paths and click **OK**. You will need to select **Override** to change the Temp directory.

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.

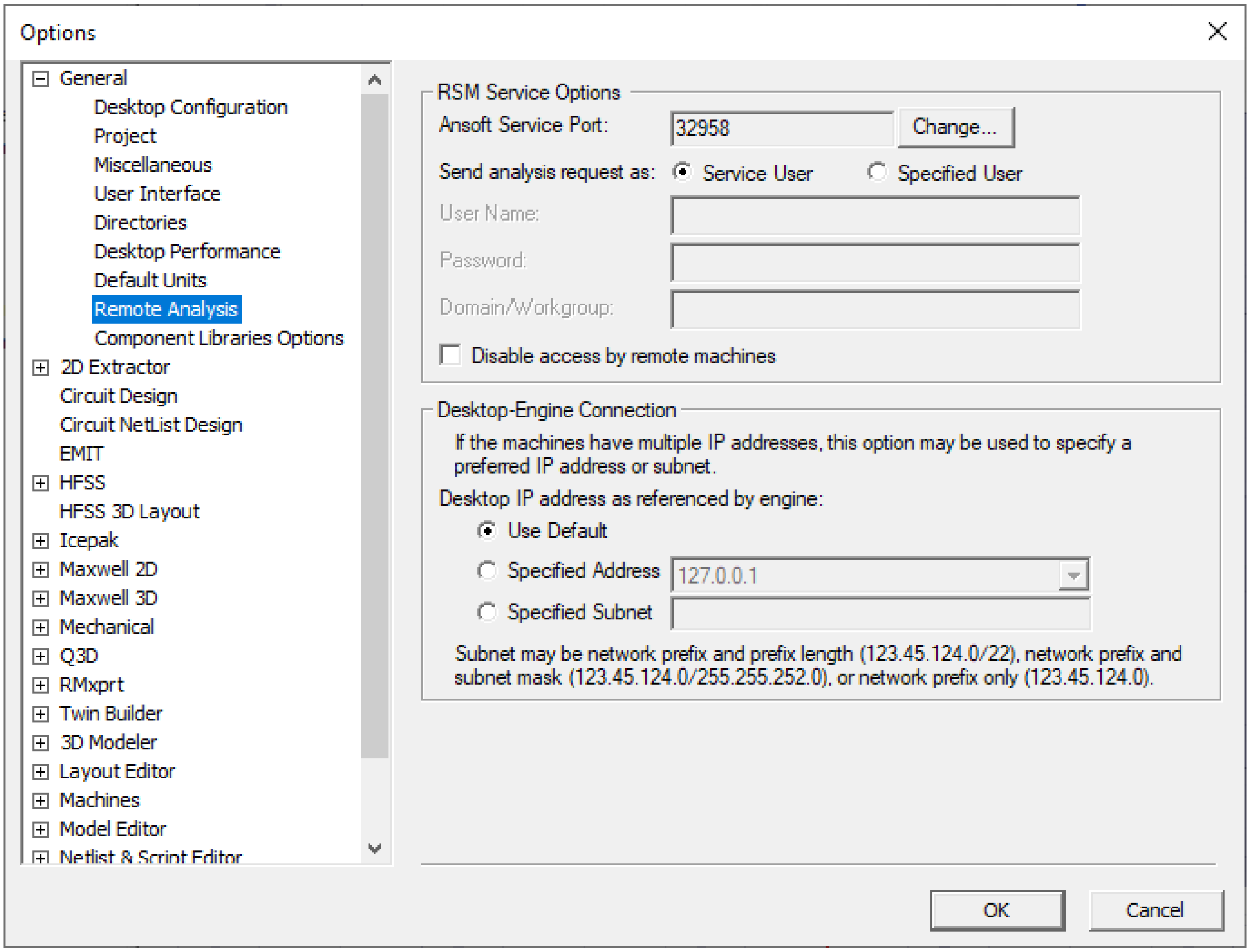
General Options: Default Units

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:

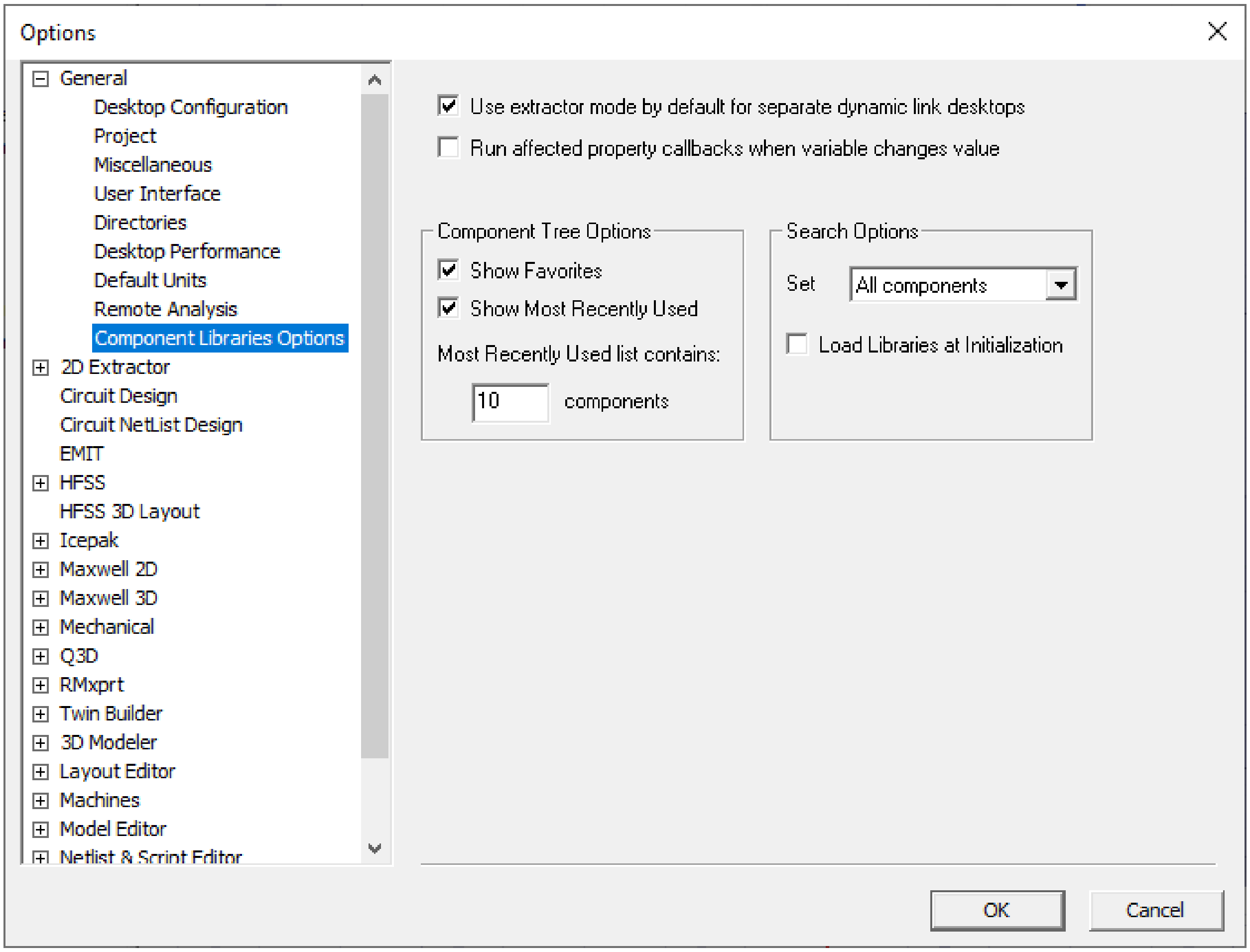
Specify the ListenPort within a 'CommDetails' block, which must be within a 'Default:CommDetails' block, which must be within the top level block of the file (the 'AnsoftCOMDaemon' block). The following example changes the listen port from 32958 to 32957, with these blocks at the beginning of the file:

```
$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

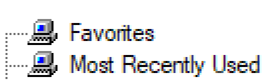
Under General Options, **Component Libraries** options allow you to change how Electronics Desktop handles components.



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.

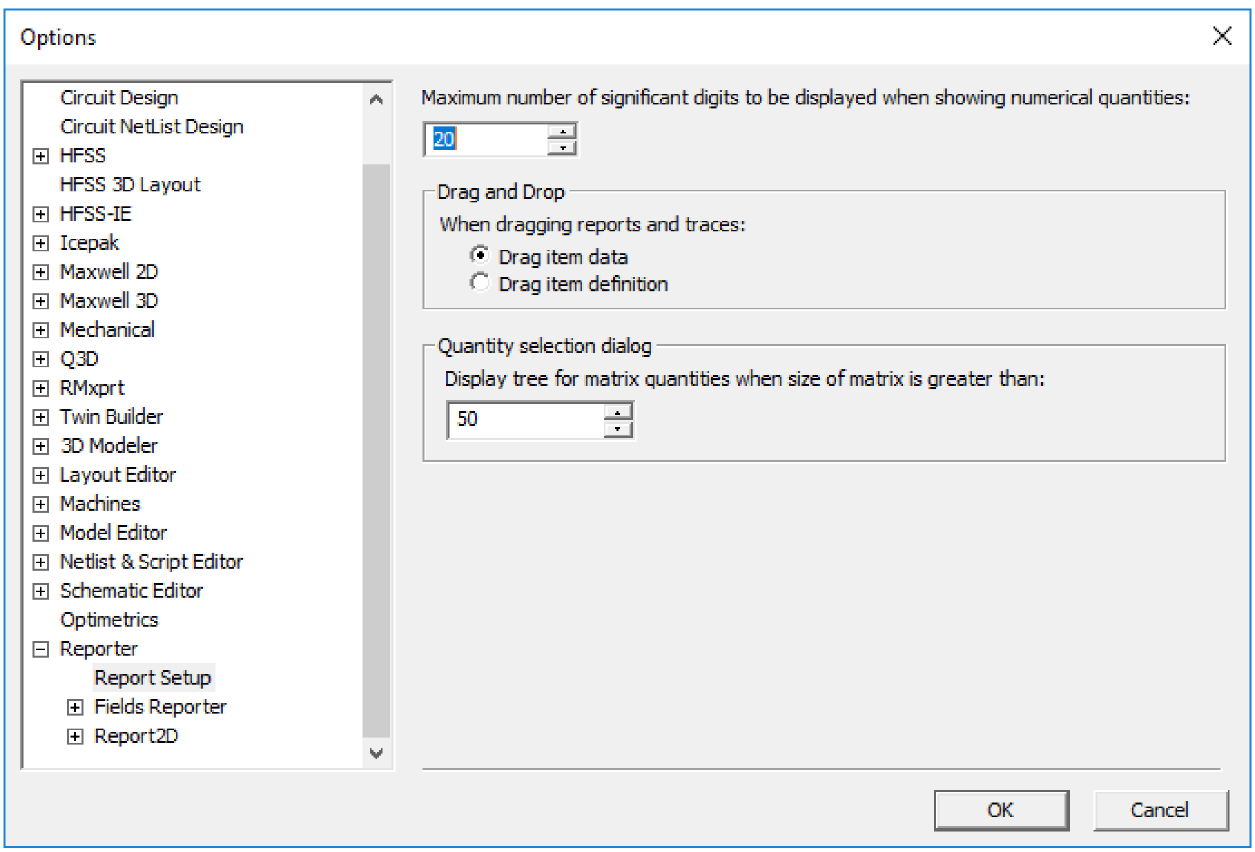
Setting Reporter Options

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.

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.

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: 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

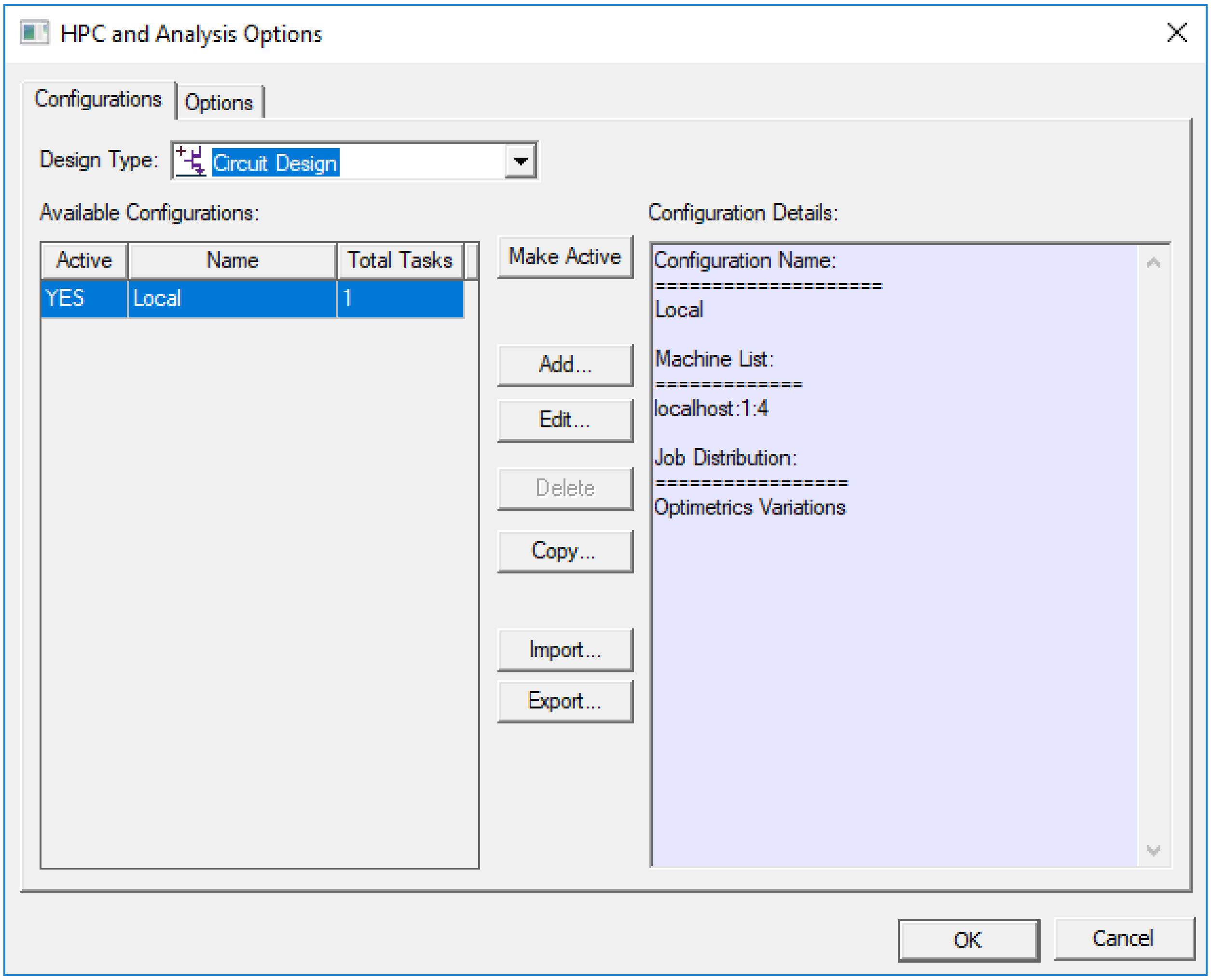
All analysis parameters are accessed via a single window. The machine list and options settings have been integrated into analysis configurations. The default configuration is for solving on a single, local machine. You can create many analysis configurations for remote and distributed solutions, and 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.

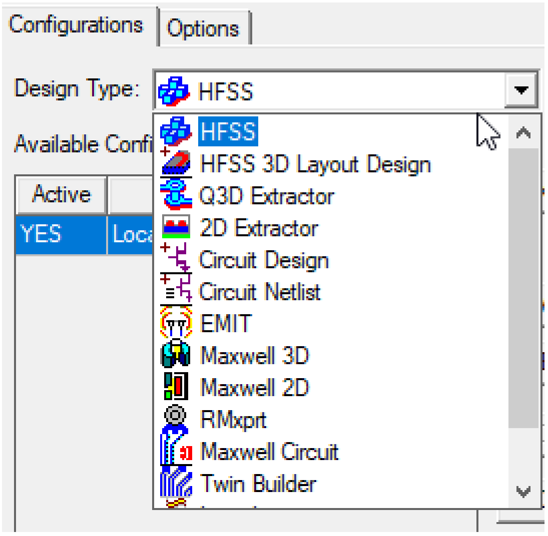


Configurations Tab

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.

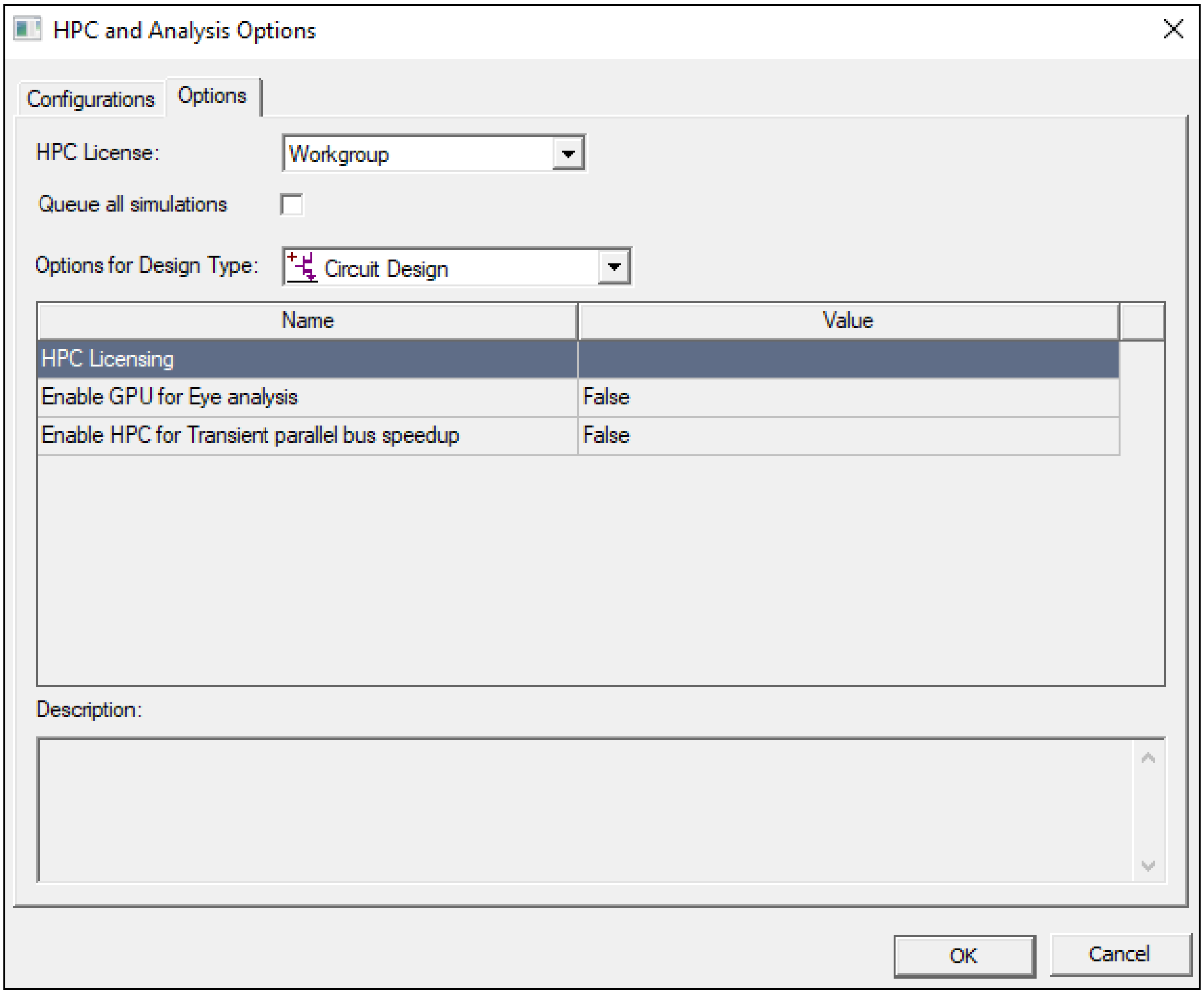
- **Copy** – creates a new analysis configuration, and launches a dialog box to edit it. If the dialog box is canceled, the new analysis configuration is not created.
- **Import** – allows you to import an *.acf file to create an analysis configuration.

  > **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

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.

> **Note:**
>
> In Nexxim, the HPC license type set works as follows:
>
> - When HPC type is set to **Workgroup** or **Pack**, the "Enable GPU" and "Transient speedup" settings are used as specified.
> - The number of processors is always passed to the solver as is, because Nexxim offers some multiprocessing capabilities without requiring an HPC license.
> - The number of processors setting in "Design Options" will override the setting in the HPC dialog.

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. For Linux authentication, specify the Remote Spawn command as RSH or SSH (the default).
2. Electronics Desktop supports GPU acceleration for transient, frequency domain, SBR+, and Maxwell 3D eddy current matrix solutions.

   For details on the requirements for GPU use, see Transient GPU Acceleration
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, Circuit 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 Circuit 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.

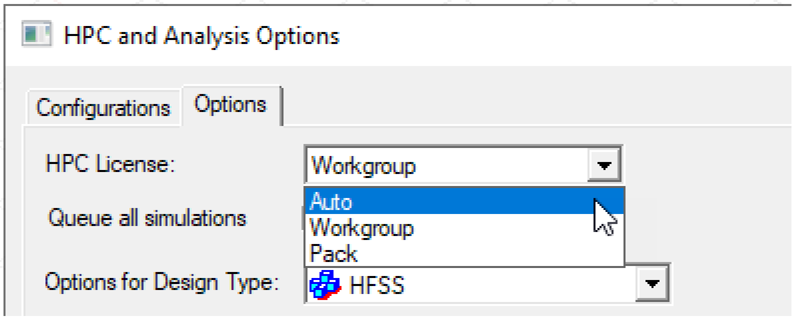
Setting MPI Version

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.

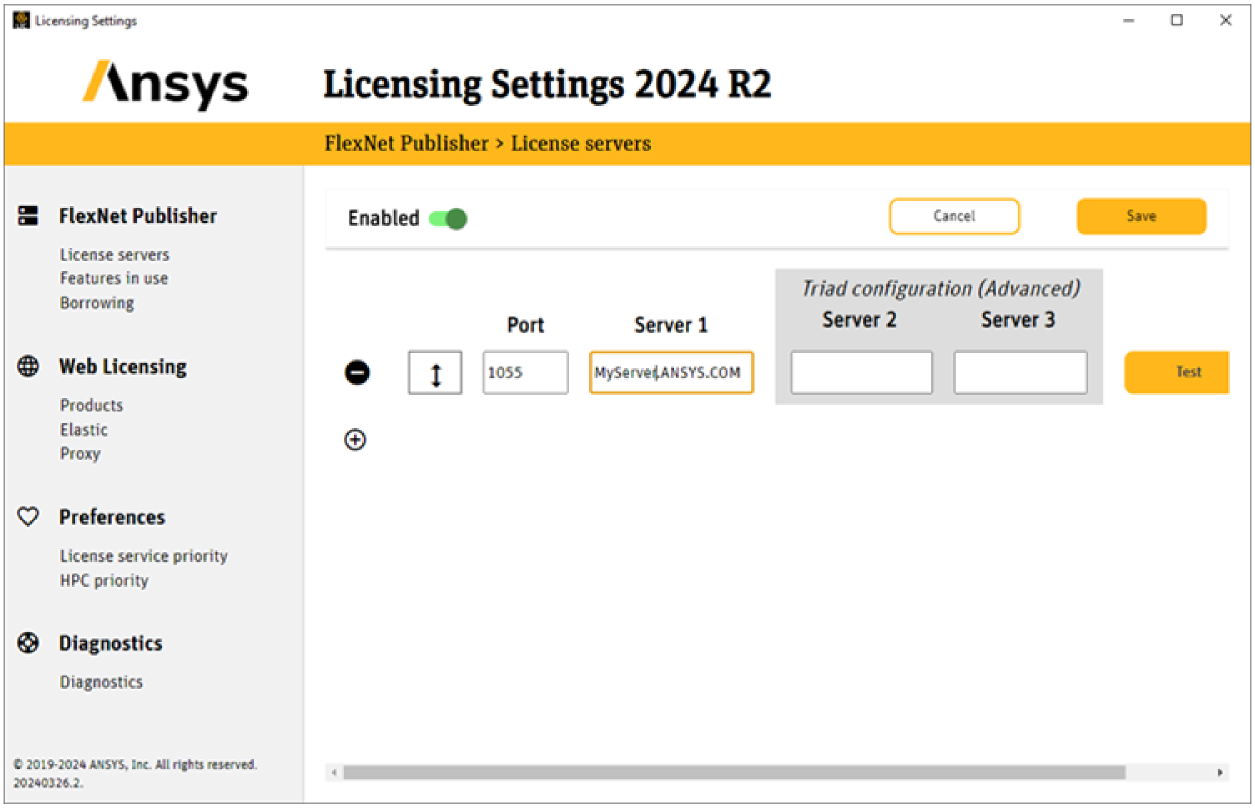


"Auto" means delegate the choice of Workgroup or Pack licensing to the Ansys Licensing Settings tool which is a separate application installed along with Electronics Desktop, accessible from the **Tools** menu. The HPC settings specified by the Licensing Settings tool are shared by other Ansys products. 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.

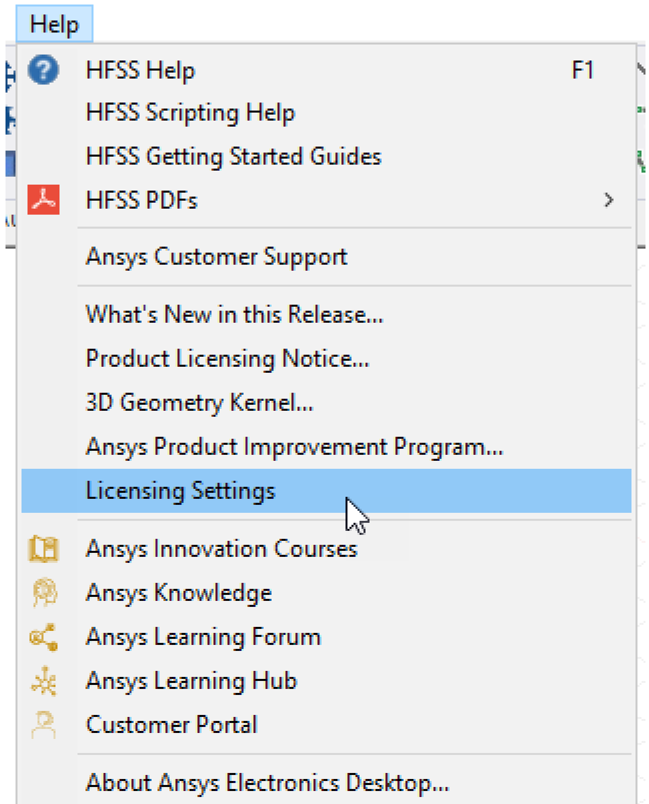
Launching the Licensing Settings Tool from the Main Menu

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*>\<*version*>\<*platform*>\config. The configuration files for host-dependent and host-independent **user** options reside in a subfolder of the user's Documents folder (for Windows) or the user's HOME folder (for Linux). See the tables below for specific Windows and Linux file names and paths.

Products with Multiple Desktop Versions

For products that have multiple Desktop versions, each 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	<*InstallationDirectory*>\v<*version*> \Win64\config	<*InstallationDirectory*>/v<*version*> /Linux64/config
installation default	install	default.XML		

Note:

- ***<hostname>*** is the name of the computer on which the Electronics Desktop software is installed
- **$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 ElectronicsDesktop2024.2)
- **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
- ***<InstallationDirectory>*** is the root folder where the Electronics Desktop software is installed (typically, C:\Program Files\AnsysEM, on Windows, or /opt/AnsysEM, on Linux)
- ***<Version>*** is the last two digits of the product version's year followed by the minor release number, without a decimal point (such as 221)

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\ElectronicsDesktop2024.2\config	/home/jsmith/Ansoft/ElectronicsDesktop2024.2/config
host independent user options	user	user.XML		
host dependent default options	install_machine	host123.XML	C:\Program Files\AnsysEM\v242\Win64\config	/opt/AnsysEM/v242/Linux64/config
installation default	install	default.XML		

> **Note:**
>
> As with the temporary file location configuration files, the settings in these options files have precedence in the following sequence: user_machine (highest precedence), user, install_machine, install (lowest precedence). The first time you start and then exit the application, the file at the "user_machine" level is created (<*hostname*>_user.XML). The other files are only created if you use the UpdateRegistry tool to specify an option at the "user," "install_machine," or "install" level. If the temporary directory is set to an empty string in a configuration file, then that setting is ignored.

Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

UpdateRegistry is a command line tool used to modify option settings in the options config files. You can use this command to add, change or remove settings from any of the option config files. This tool is included in the installation directory of each product. This feature makes it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives.

The UpdateRegistry command has multiple command line formats, as shown below.

The following command line options are *mutually exclusive*:

- `-Set`
- `-Get`
- `-GetKeys`
- `-Delete`
- `-FromFile`

UpdateRegistry -Set Command

This command is used to add or modify an option setting in an option config file. If the option config file does not exist, it will be created. If the setting does not exist in the specified config file, it will be added. If the setting already exists in the specified config file, then the value will be changed to the specified value.

Example:

```
UpdateRegistry -Set -ProductName <name> -
RegistryKey <keyPath> -RegistryValue <value> [
-RegistryLevel <level> ]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2024.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: ...\AnsysEM\<*Version*>\Win64\config\
`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory
`<value>`	The new value of the option, typically a string or a number. If the value contains spaces, it must be quoted.

Optional:

`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, the user_machine (host-dependent user options) file is modified.

UpdateRegistry -Get Command

This command is used to view an option value in an option config file. If the setting exists in the specified config file or files, then the value, the value type and the config file where the value was found will be reported. If no value is found, then that will also be reported.

Example:

```
UpdateRegistry -Get -ProductName <name> -
RegistryKey <keyPath> [ -RegistryLevel <level>
]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2024.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: ...\AnsysEM\<*Version*>\Win64\config\
`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory

Optional:

`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, then all config files are searched in order of precedence.

UpdateRegistry -GetKeys Command

This command is used to view the allowed key names for all of the option settings, or to view a subset of the key names that match a string. For each key displayed, the current value, if any, is also reported. If a key has a value in multiple config files, then only the highest precedence value is reported.

Example:

```
UpdateRegistry -GetKeys [ <pattern> ] -
ProductName <name> [ -Case ]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2024.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: ...\AnsysEM\<*Version*>\Win64\config\

Optional:

`<pattern>`	If no pattern is specified, then all allowed key names are reported. If a pattern is specified, then only keys that match the pattern are shown. For example, Settings/Project. If the name contains spaces, it must be quoted. By default, the pattern match is case insensitive.
`-Case`	If this command line option is specified, then the pattern match is case sensitive.

UpdateRegistry -Delete Command

This command is used to remove an option setting from an option config file. If the setting does not exist in the specified config file, the file will not be changed. If the setting exists in the specified config file, then it will be removed. A setting may need to be removed from an option config file, to allow the setting from a lower priority file to be used by the application.

Example:

```
UpdateRegistry -Delete -ProductName <name> -
RegistryKey <keyPath> [ -RegistryLevel <level>
]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2024.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: ...\AnsysEM\<*Version*>\Win64\config\
`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory

Optional:

`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, the user_machine (host-dependent user options) file is modified.

UpdateRegistry -FromFile Command

You can use this form of the UpdateRegistry command to set multiple key-value pairs from a file with a single UpdateRegistry command. You specify the -FromFile command line option. This option must be followed by a filename. The file may contain multiple entries, where each entry contains a registry key and a registry value. The key-value pairs are added to the registry level specified by the -RegistryLevel command line option; if no -RegistryLevel is specified, then the default registry level (user_machine) is used.

UpdateRegistry File Format

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

The file format is similar to the -batchoptions file format. An example UpdateRegistry file is shown below:

```
$begin 'AddEntries'
   'TempDirectory'='C:/temp/AnsysEM'
   'Desktop/Settings/ProjectOptions/HPCLicenseType'='Pool'
```

```
$end 'AddEntries'
```

Additional notes on the file format:

- The file may contain an arbitrary number of entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.

Registry key pathname:

- The registry key pathname appears before the equal sign "=" on each line.
- Each registry key pathname must be enclosed in single quotes.
- This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help.

Registry value:

- The registry value appears after the equal sign on each line.
- Integral registry values must not be enclosed in quotes.
- All other registry values are treated as strings, and must be enclosed in single quotes.
- The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.
- The back slash "\" is used as an escape character in the value string. That it, this character removes the special meaning of the following character.
- The single quote character normally ends the value string. The back slash may be used to remove this special meaning, and include a single quote in the string.
- To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

Alternative UpdateRegistry File Format:

- Analysis Configuration File format, which is exported from the HPC and Analysis Options dialog box.

> **Note:**
>
> If a current registry does not exist, Ansys Electronics Desktop can detect earlier minor versions of same application on the same machine. If such a registry exists (and does not involve -help, -batchoptionhelp, IsBatchMode(), -regserver, -unregserver, running a script, or non graphical mode), a prompt displays allowing you to port the registry from an earlier version.

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 ElectronicsDesktop2024.2
```

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 ANSYSElectronicsDesktop2024.2
 -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 ElectronicsDesktop2024.2
 -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 ElectronicsDesktop2024.2
 -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 ElectronicsDesktop2024.2
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryLevel user
```

User Options and the UpdateRegistry Tool

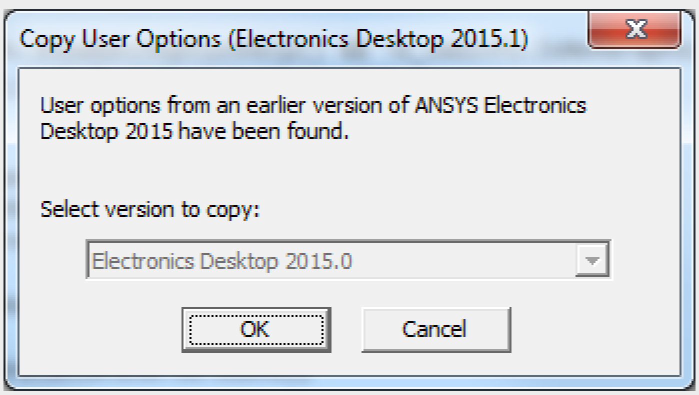
> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

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 ElectronicsDesktop2024.2
-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 ElectronicsDesktop2024.2 -
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 ElectronicsDesktop2024.2 -
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 ElectronicsDesktop2024.2
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryValue 250 -RegistryLevel user
```

Setting the Temporary Directory

The temporary directory may be viewed or set using the Electronics Desktop user interface (UI), or from the command line.

To set the directory via the UI:

> Navigate to **Tools** > **Options** > **General Options**. In the tree at the left side of the dialog box, expand the **General** branch and select **Directories**. Then, use the **Temp** field and

Override check box to enter a desired directory path. Values set in this manner are written to the user_machine level configuration file for the temporary directory. If the **Override** check box is cleared, clicking **OK** changes the user_machine level setting for the temporary directory to an empty string. This enables settings from the next highest precedence configuration file. The file that provides the currently active temporary directory setting is shown under the **Temp** edit box.

To set the temporary directory from the command line, using the -batchoptions command line option. See: Running Ansys Electronics Desktop from a Command Line and -Batchoptions Command Line Examples.

The temporary directory may be configured with an installation default value, as well as a host-dependent default value, a host-independent user-specified value and a host-dependent user-specified value. Temporary directory settings are stored in different files from the other option settings. These files are located in the same directories as the configuration files for the other option settings. The following table shows the directories and files used to store temporary directory settings.

Config File	Level Name	File Name	Windows Directory Path	Linux Directory Path
Host-dependent, user-specific temporary directory	user_machine	<*hostname*>.cfg	%UserProfile%\Documents\Ansoft \<*ApplicationName&Version*>\config	$HOME/Ansoft /<*ApplicationName&Version*>/config
Host-independent, user-specific temporary directory	user	default.cfg		
Host-dependent, default temporary directory	install_machine	<*hostname*>.cfg	<*InstallationDirectory*>\v<*version*> \Win64\config	<*InstallationDirectory*>/v<*version*> /Linux64/config
Installation default temporary directory	install	default.cfg		

Note:

- ***<hostname>*** is the name of the computer on which the Electronics Desktop software is installed
- **$HOME** is the user's home directory on Linux
- ***<ApplicationName&Version>*** is the product name (without spaces) followed by the four-digit year of the version, a decimal point, and the minor release number (such as ElectronicsDesktop2024.2)
- **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
- ***<InstallationDirectory>*** is the root folder where the Electronics Desktop software is installed (typically, C:\Program Files\AnsysEM, on Windows, or /opt/AnsysEM, on Linux)
- ***<Version>*** is the last two digits of the product version's year followed by the minor release number, without a decimal point (such as 221)

As with other options, the settings in these files have precedence in the following sequence: user_machine (highest precedence), user, install_machine, install (lowest precedence). The installer creates the file at the "install" level (default.cfg). The first time you start and then exit the application, the file at the "user_machine" level is created (<*hostname*>.cfg). The other files are only created if you use the UpdateRegistry tool to specify an option at the "user" or "install_machine" level. If the temporary directory is set to an empty string in a configuration file, then that setting is ignored.

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.

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/
'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.

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/
'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.

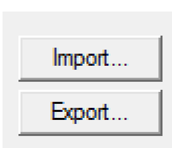
> **Note:**
>
> - Options are arranged as keys and values (in a structure similar to the *Windows Registry*). However, these options are not a part of the *Windows Registry* but are separately stored and maintained by the Ansys Electronics Desktop software. For more information concerning the configuration 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***>\v**<*version*>**\Win64**

2. Type and enter the following command:

   UpdateRegistry -GetKeys -ProductName ElectronicsDesktop20*xx.y* **>** <*text_file_path*>**\Batchoptions.txt**

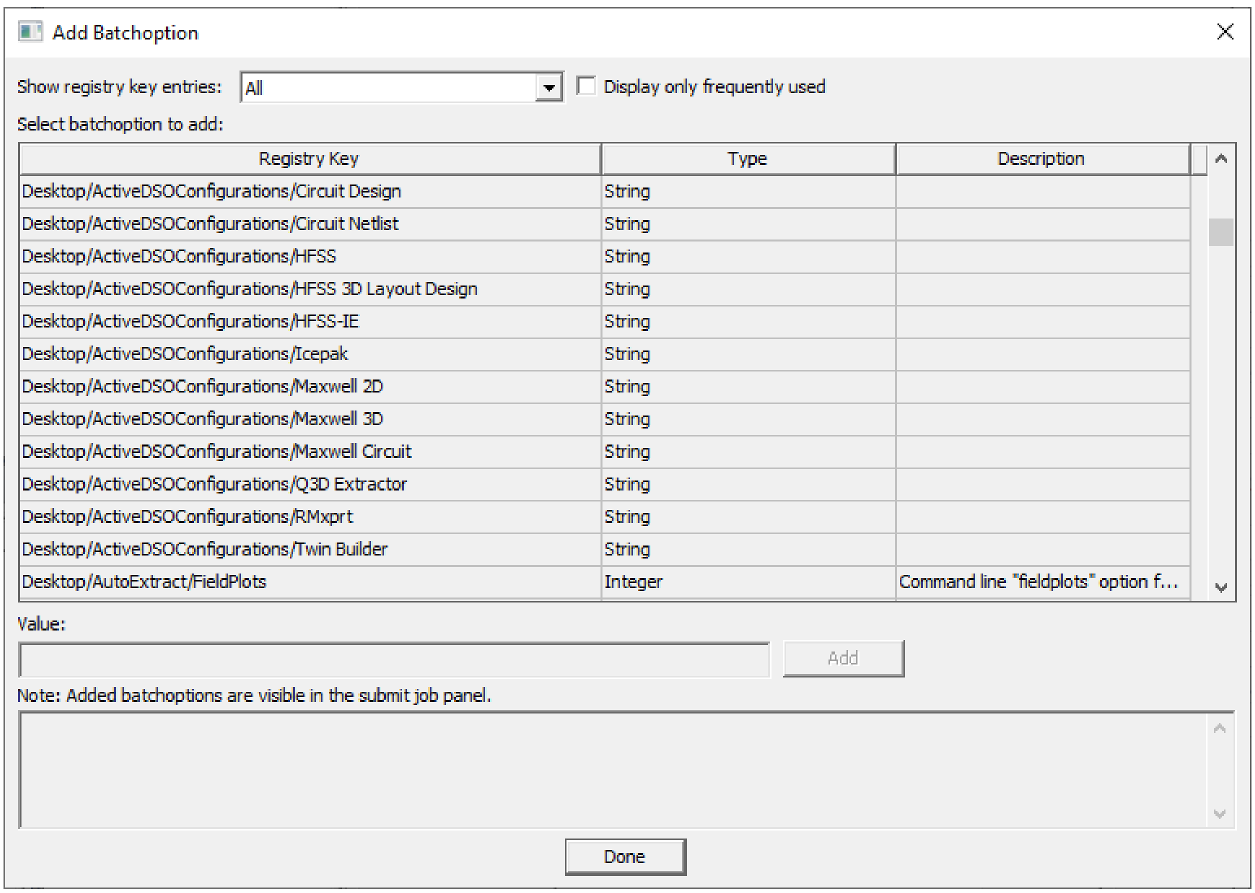
   Substitute the last two digits of the installed product version year and the minor release number for *xx.y* (such as **22.1**). Also, substitute the desired *text_file_path* (such as **%UserProfile%\Desktop**).

   The resulting **Batchoptions.txt** file will contain a nearly complete list of available options.

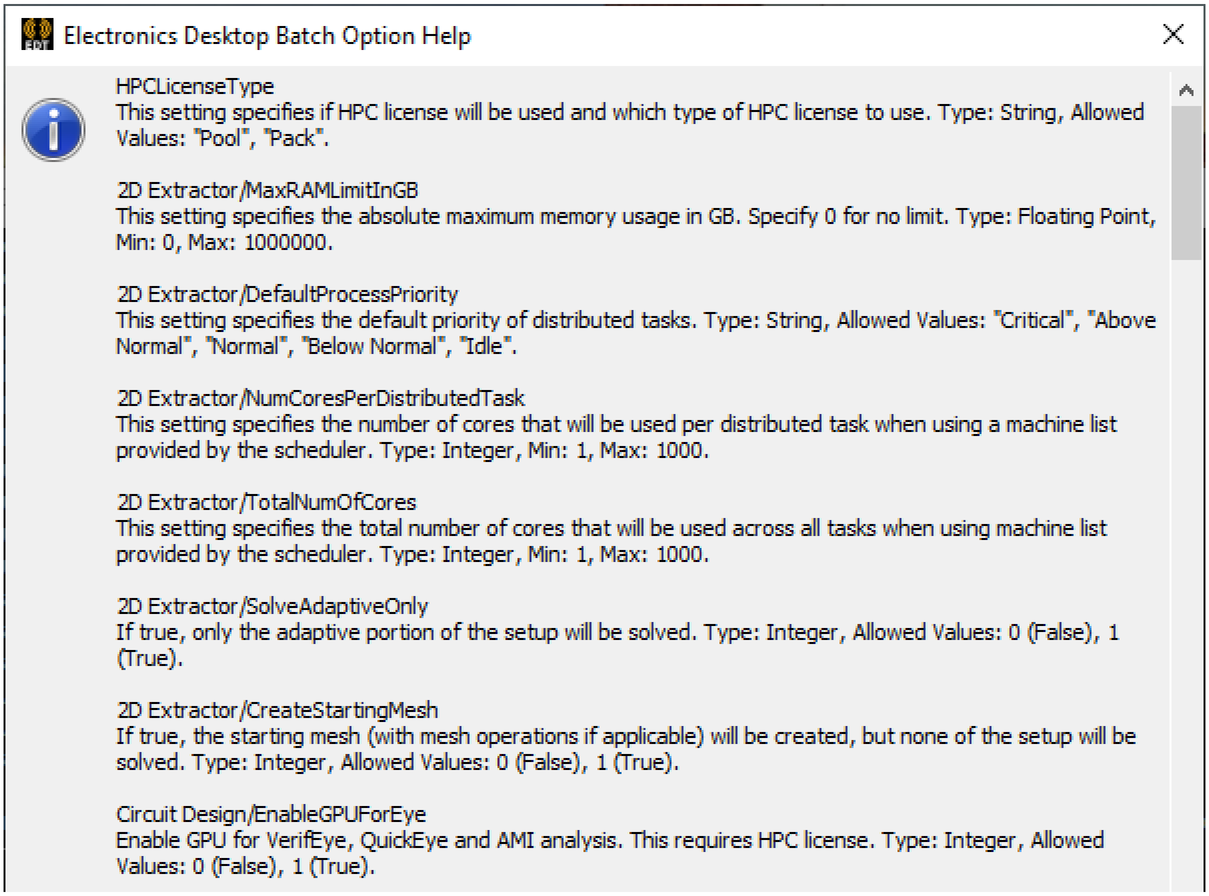
This procedure also works from the <*installation_directory*>/v<*version*>/Linux64 folder on the Linux platform.

> **Note:**
>
> For special product-specific options that are neither available from the GUI nor listed by the UpdateRegistry tool, see Special Batch Options. Even though the -GetKeys switch does not list these special options, you can still use the UpdateRegistry tool to set them.

When you submit jobs to a remote computer or cluster, you can specify batch options using the job submission GUI. When using the GUI, you select the batch options from a list and therefore avoid typographical errors. For the most commonly used batch options, there is detailed information about the allowed values. Click **Submit** on the **Simulation** ribbon tab to access the *Submit Job To* dialog box. Then, click **Add** to access the **Add Batchoption** dialog box pictured below:



To assist users who need to specify batch options and are unable to use the job submission GUI, a new help option has been added. If the Electronics Desktop application is launched with the **-batchoptionhelp** command line argument, a message box is displayed which lists and describes the most common design type-specific batch options:



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:

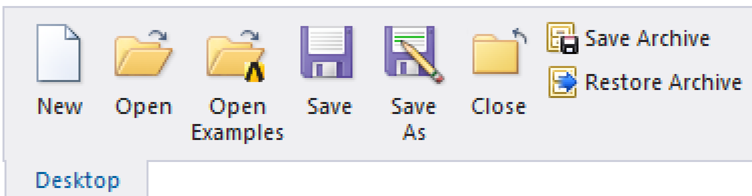
- Opening Projects
- Creating Projects
- Saving Projects
- Importing and Exporting Projects and Data
- Setting Project Options
- Validating Projects
- Copying and Pasting Projects and Designs
- Renaming Projects
- Deleting Projects or Designs
- Setting Read Only Designs

- Updating Design Components
- Undoing and Redoing Commands
- Inserting a Documentation File
- Saving Project Notes
- Printing

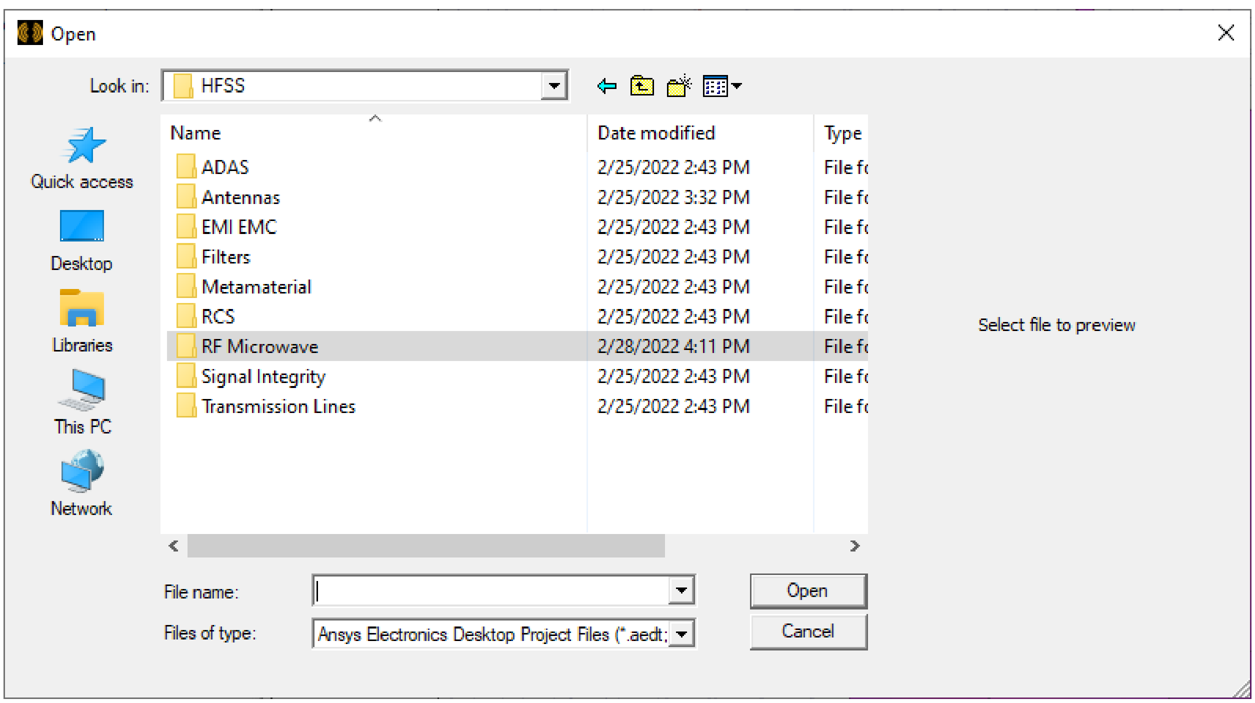
Opening Projects

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

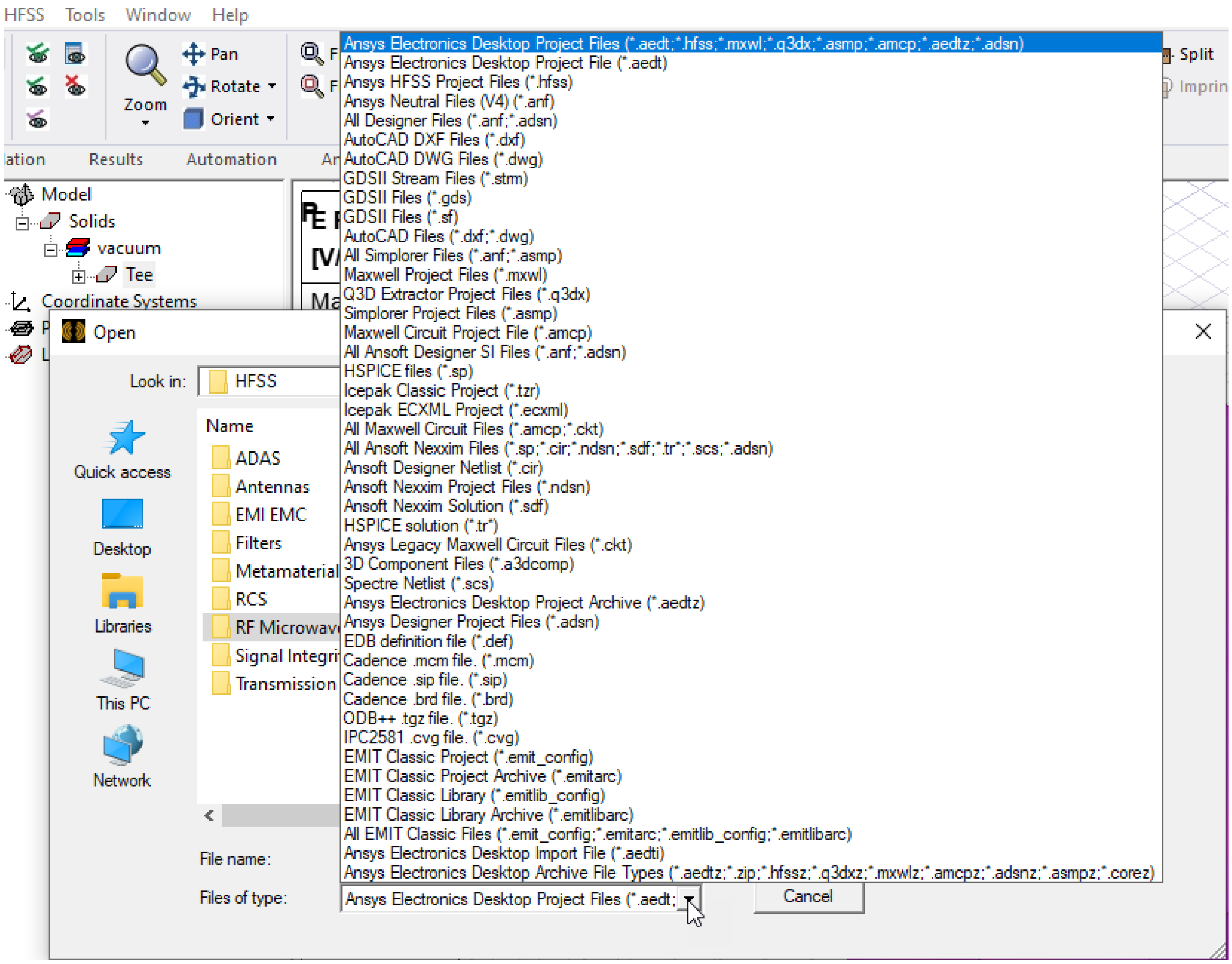
1. Click **File** > **Open** or click the **Open** icon on the **Desktop** tab of the ribbon.



2. Use the file browser to find the project file.

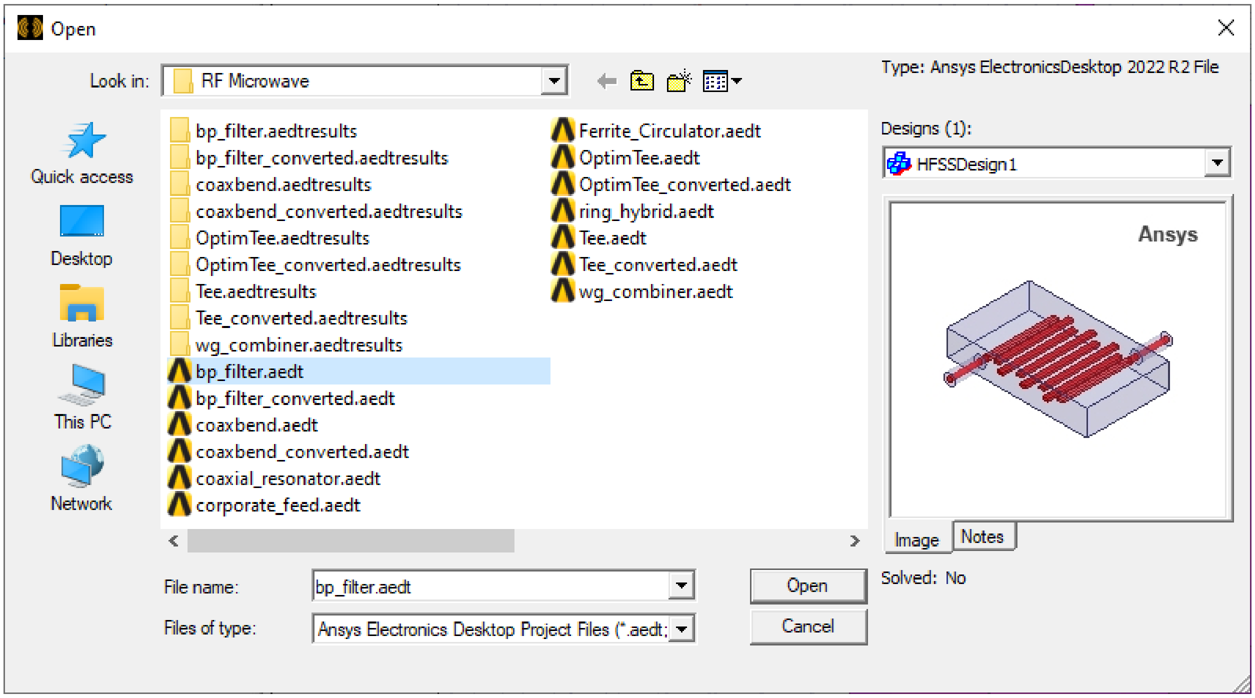


By default, only files that can be opened or translated display. *.aedt is the main default, although, you can also view several types of legacy files. The dropdown for Files of type lists the kinds file and archive formats for which you can select filters.



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.

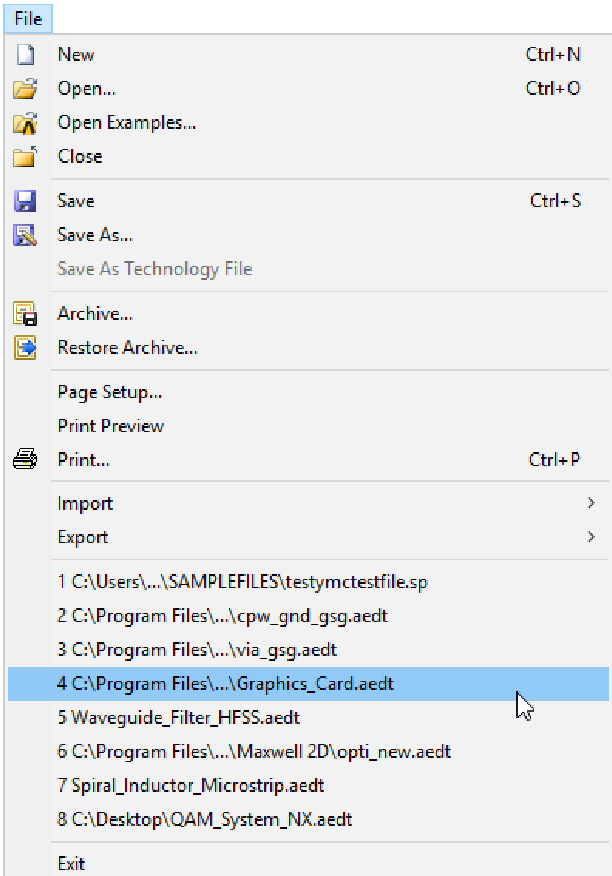
If you open another project without editing the automatically created project, the automatically created project is removed.

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.

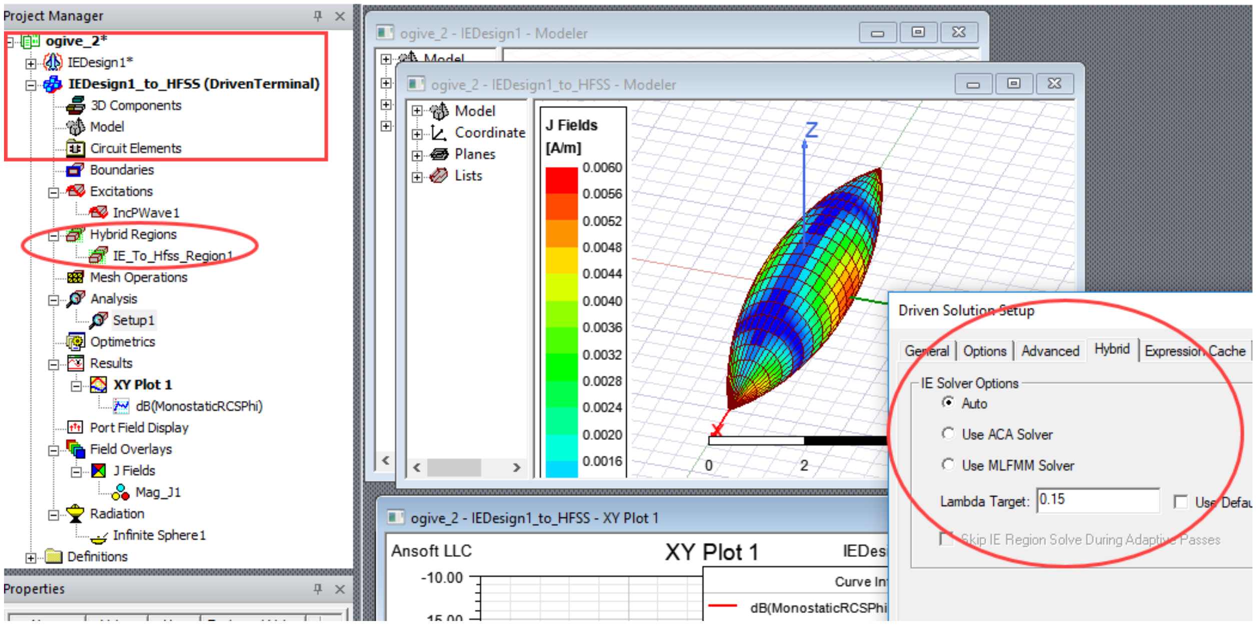


Opening Example Projects

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 HFSS-IE Projects

If you open an existing HFSS-IE project in Ansys Electronics Desktop, it is automatically converted to an equivalent HFSS project with IE Regions assigned.



The primary changes in the converted project are that an IE Region is assigned to the objects, and that the **Hybrid** tab of the Setup has the IE Solver Options. All Excitations, Plots, Field Overlays, and Radiation are all equivalent to those in the original project. If you select the original project, the HFSS-IE menus will still appear for that project. However, going forward, HFSS-IE will not be supported. IE Regions should be used.

HFSS-IE included a Physical Optics (PO) solver for use with metallic designs excited by an incident plane wave, a far field wave, or a near field wave. The PO functionality can be achieved using PO Regions. This solver provides first-order scattering information, such as an approximate RCS. If that result is not accurate enough (for example, for a back lobe calculation), you can use the regular IE solver. The types of reports are the same; it is the accuracy that changes. When selected, it solves for only one pass, performing no adaptive refinement.

Opening Legacy Projects

HFSS cannot open projects created in Ansoft HFSS version 8.5 or earlier. HFSS 10 files can be opened, but saving them in the current format renders them unable to be reopened in version 10.

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.

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. • Object coordinate systems are created for objects assigned anisotropic materials in legacy projects. The coordinate system is defined at the same origin as the global coordinate system with the same orientation defined when the anisotropic material was assigned to the object in the legacy project. • Nonlinear materials from legacy projects that have magnetic saturation values greater than zero are treated as ferrite materials. Their properties

	are not modified.
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.

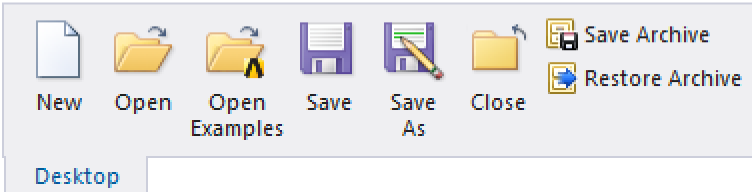
Note:

The **.ndsn** extension identifies projects created with the standalone Nexxim product. Nexxim circuit and Nexxim netlist projects are saved with the normal **.adsn** format. HFSS can open Nexxim-created **.ndsn** projects, but saves all projects in **.adsn** format.

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.

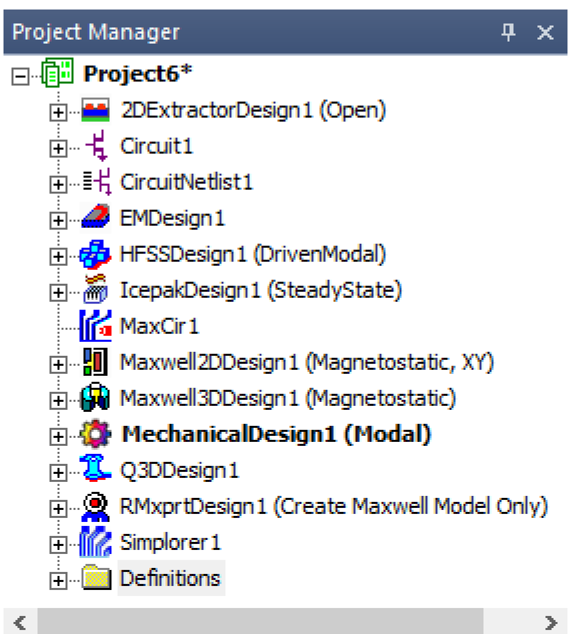
Creating Projects

To create a project, click **File** > **New**, or click the **New** icon on the **Desktop** tab of the ribbon.



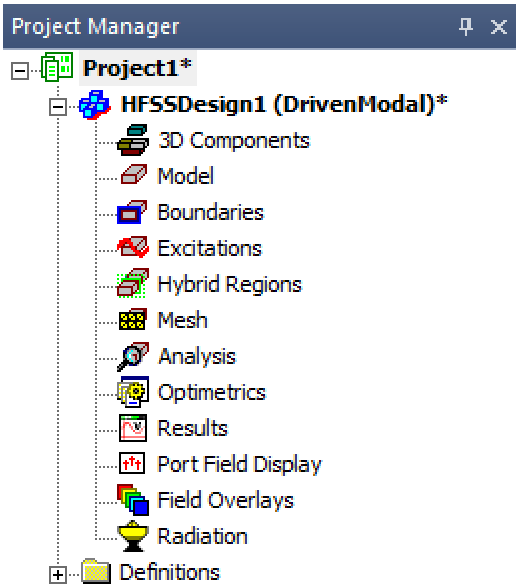
A new project is listed in the project tree. It is named Project*n* by default, where *n* is the order in which the project was added to the current project folder.

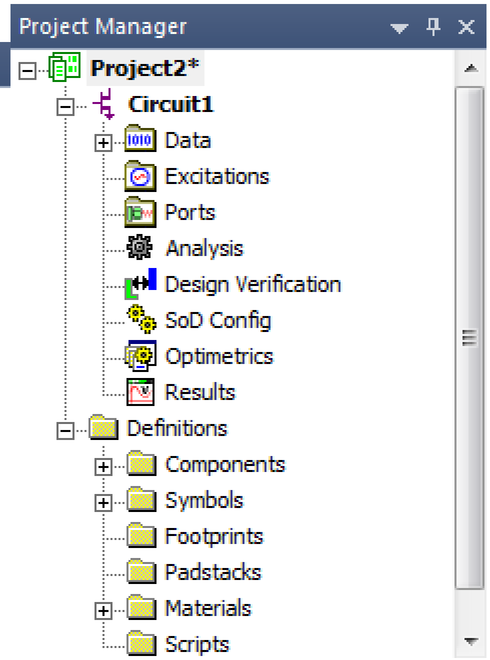
You can insert designs of any type into the project, where they are stored in the project tree.



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.



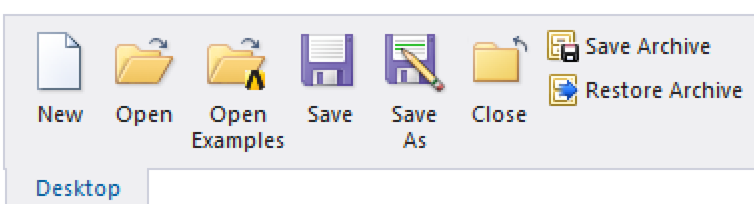


Saving Projects

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.

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

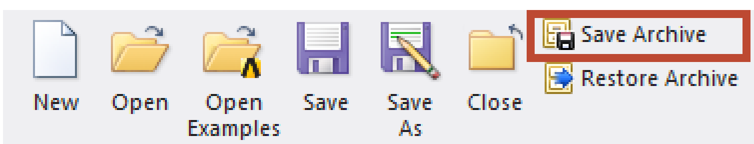
Internally, project archive files are .zip files, and are compatible with any program that can read .zip files. Project archive files have an extension that is unique for each product, and is generated by adding a 'z' to the project file extension (for example, *.aedtz, *.adsnz). This extension displays as the default when saving and restoring archive files.

Archive Preview

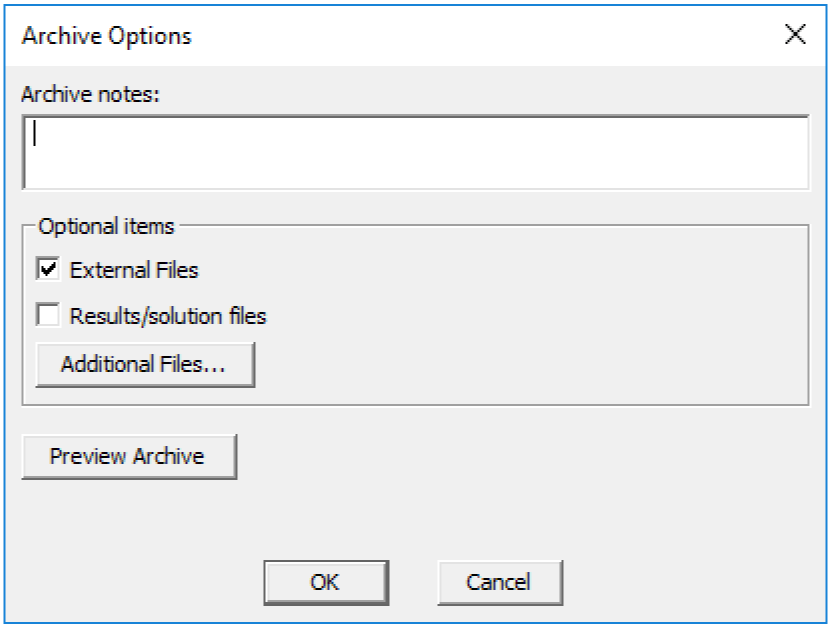
The **Archive** command includes a preview feature that lets you review the contents of a planned archive.

To archive the current project:

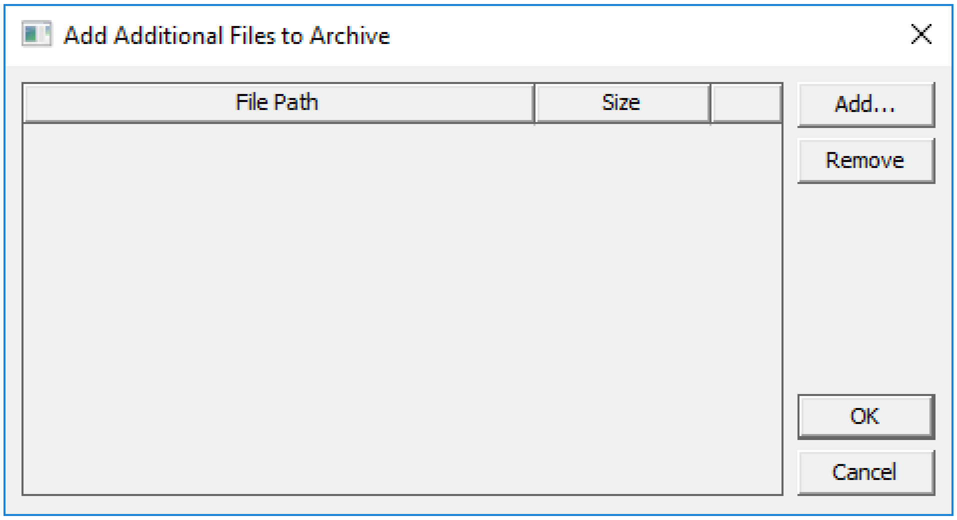
1. Click **File** > **Archive** or select the **Desktop** tab of the ribbon and click **Save Archive**.



The **Archive Options** dialog box opens.



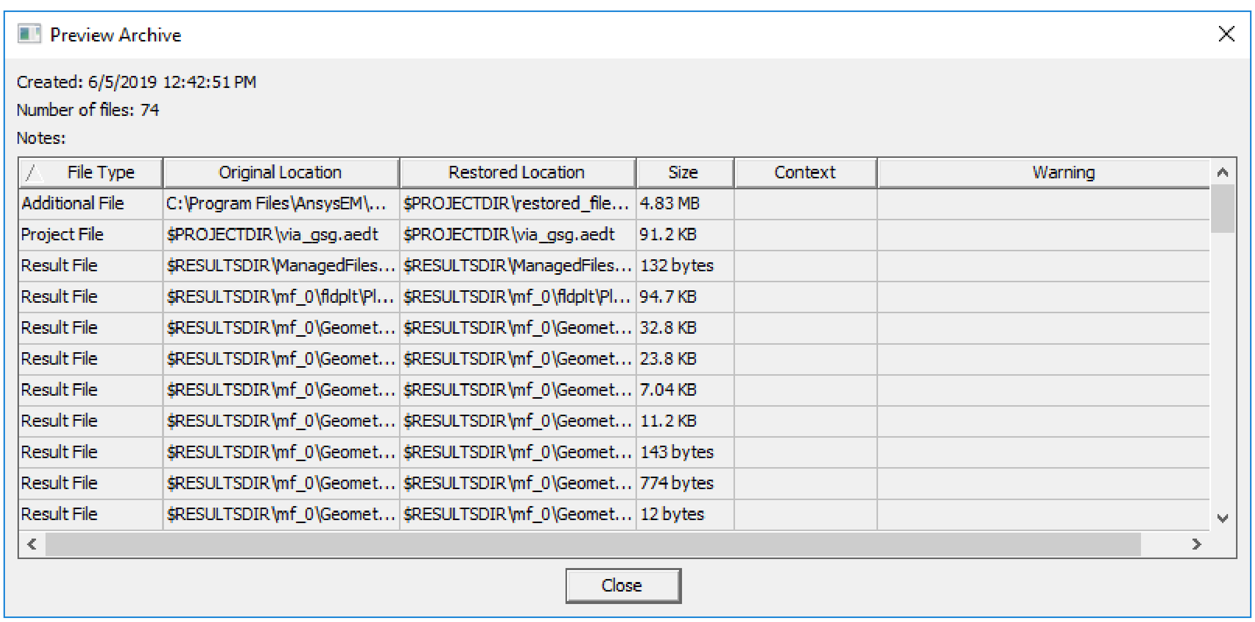
- **Archive Notes** – in this field, you can specify notes that will be visible when previewing the archive. These notes can be viewed from the preview dialog box without restoring the archive.
- **External Files** – selecting this check box causes all external files to be included in the archive. These 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

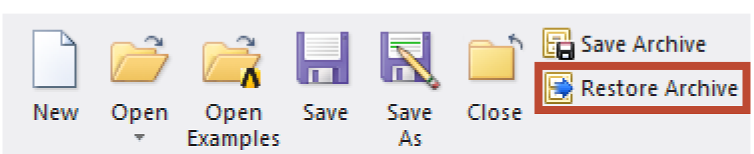
In a project to be archived, external files can be located anywhere on the user's system. One of the goals is for the restored project to be relatively self contained, and to NOT allow the restoring of an archived project to haphazardly write files anywhere on the restoring user's system.

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.

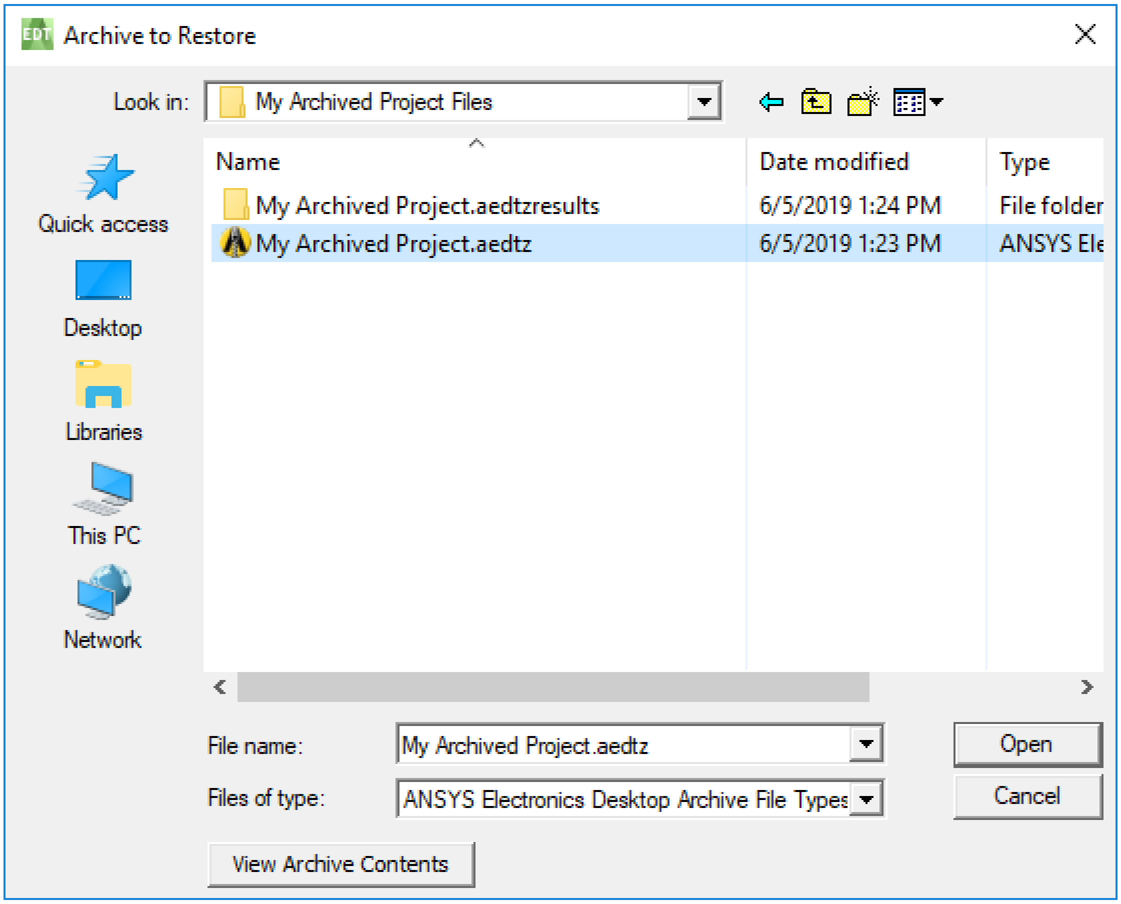
Restoring an Archived Project

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**.

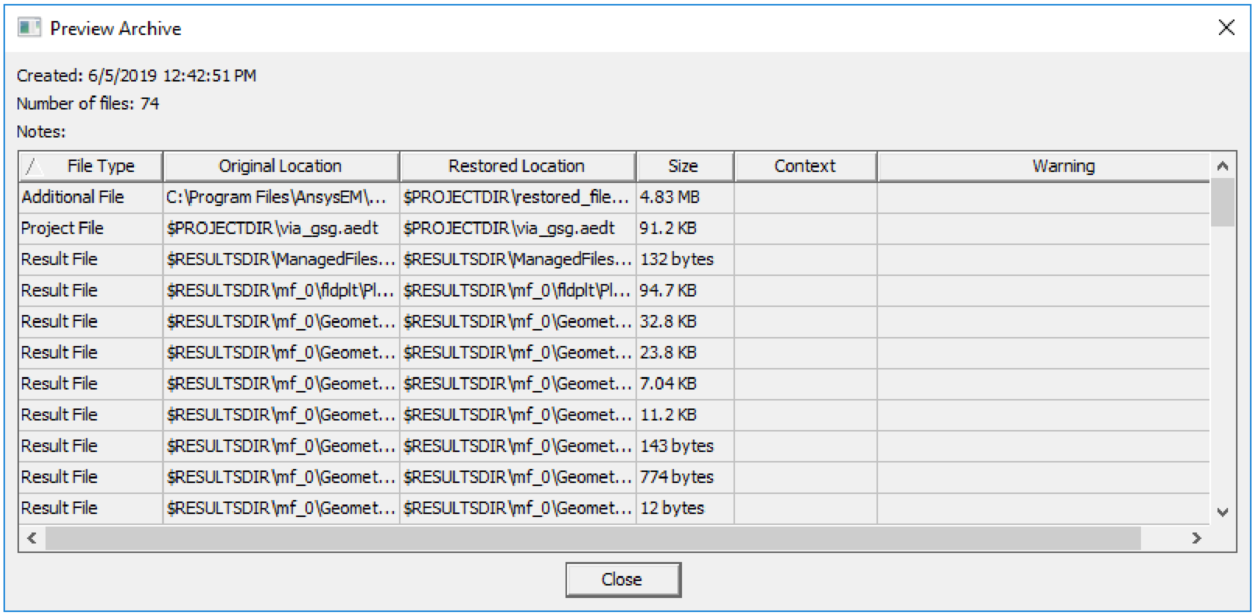


   This displays an **Archive to Restore** browser window that lets you navigate your file system for *.aedtz, legacy *.*<solversuffix>z* or *.zip archive files.

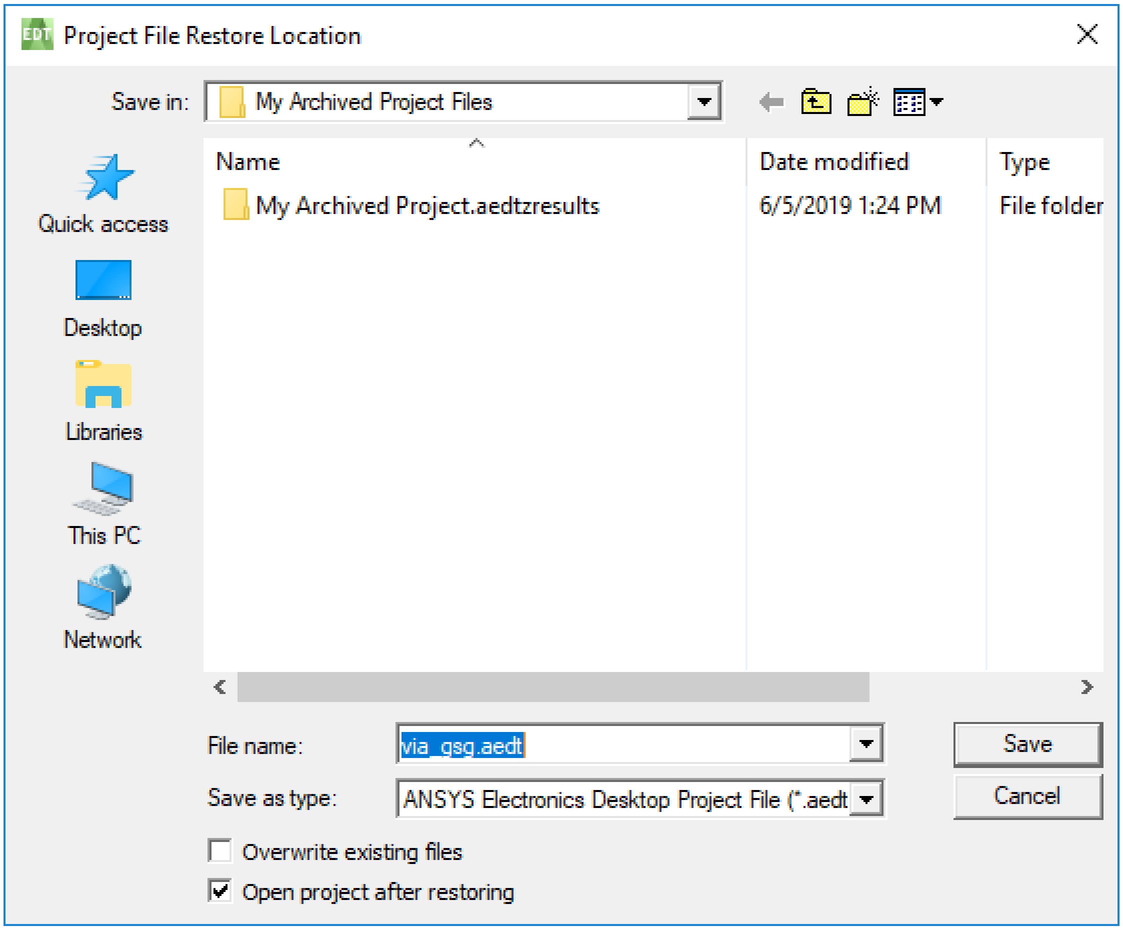


2. If you select a valid archive file, you can click **View Archive Contents** to preview the contents.

   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.

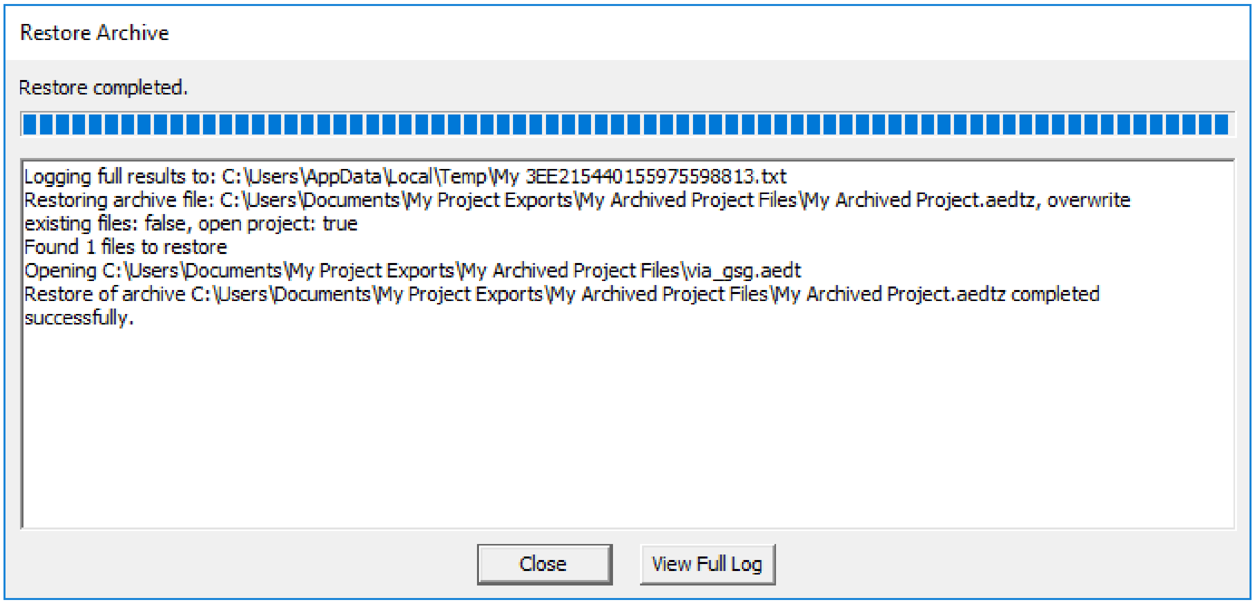


You can edit the file name, and select options:

- **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:

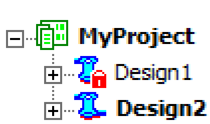
1. In the Project Tree, select a project or design.
2. Click **Edit** > **Delete**, or press the **Delete** key.
3. Confirm the warning box to complete the delete operation.

   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:

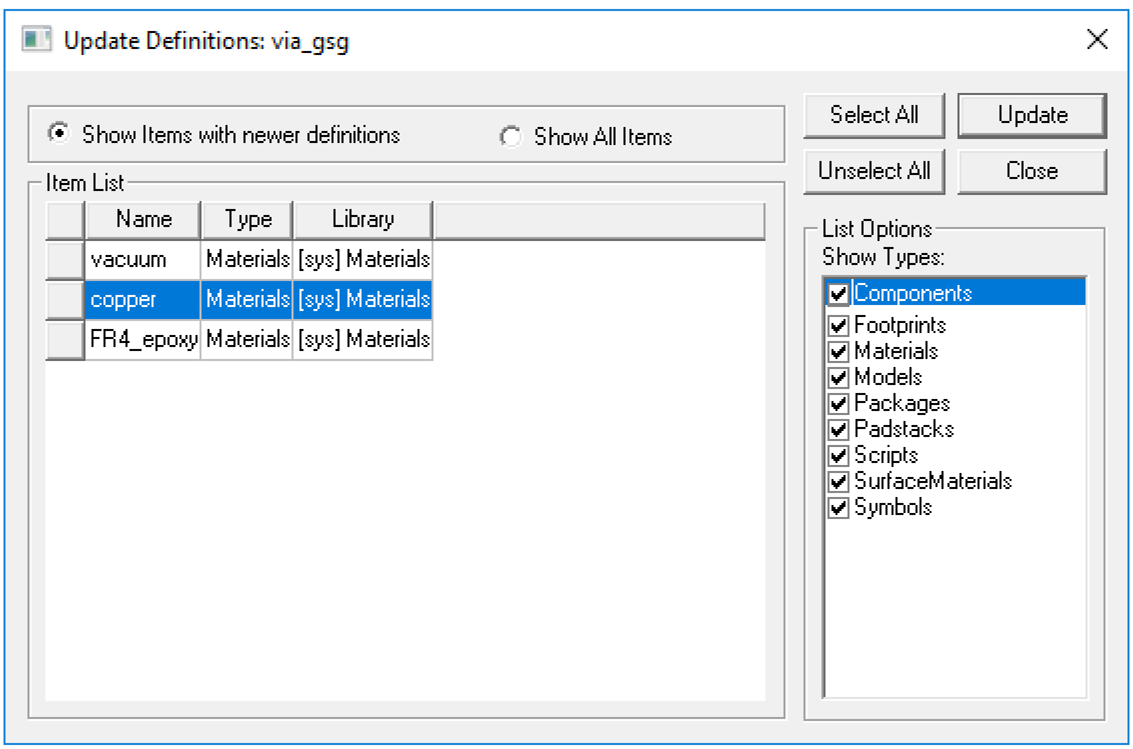
Read Only settings are not saved when you save the project file.

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 **Circuit** > **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.

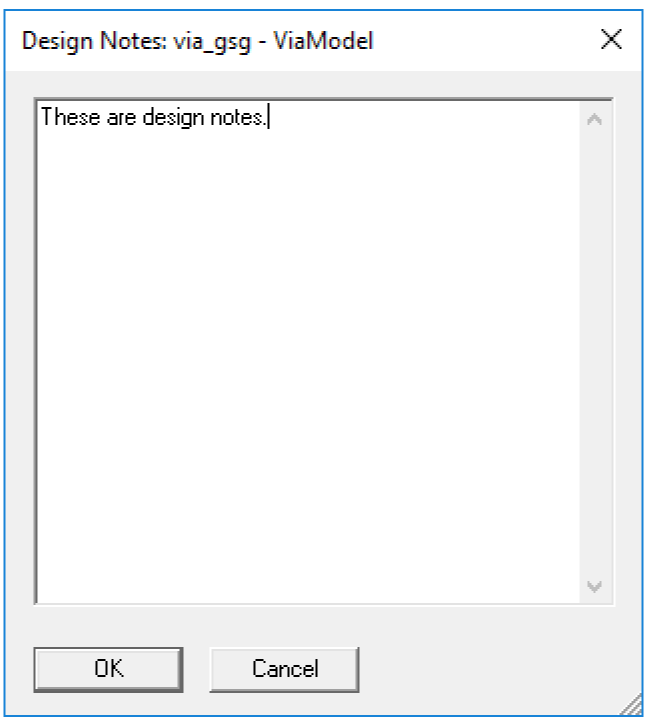
Working with Design Notes

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 > **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.

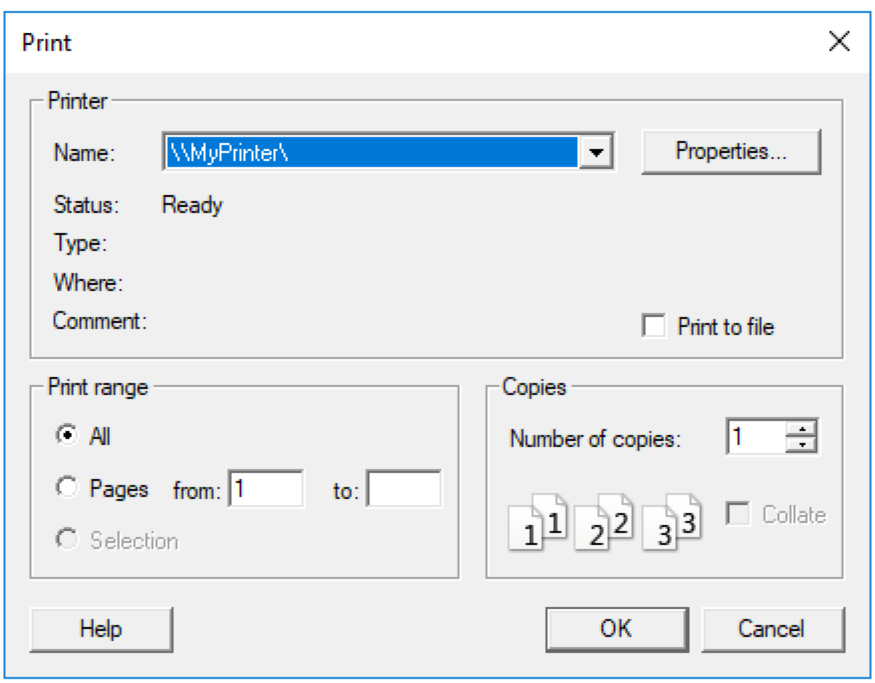
Printing

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.

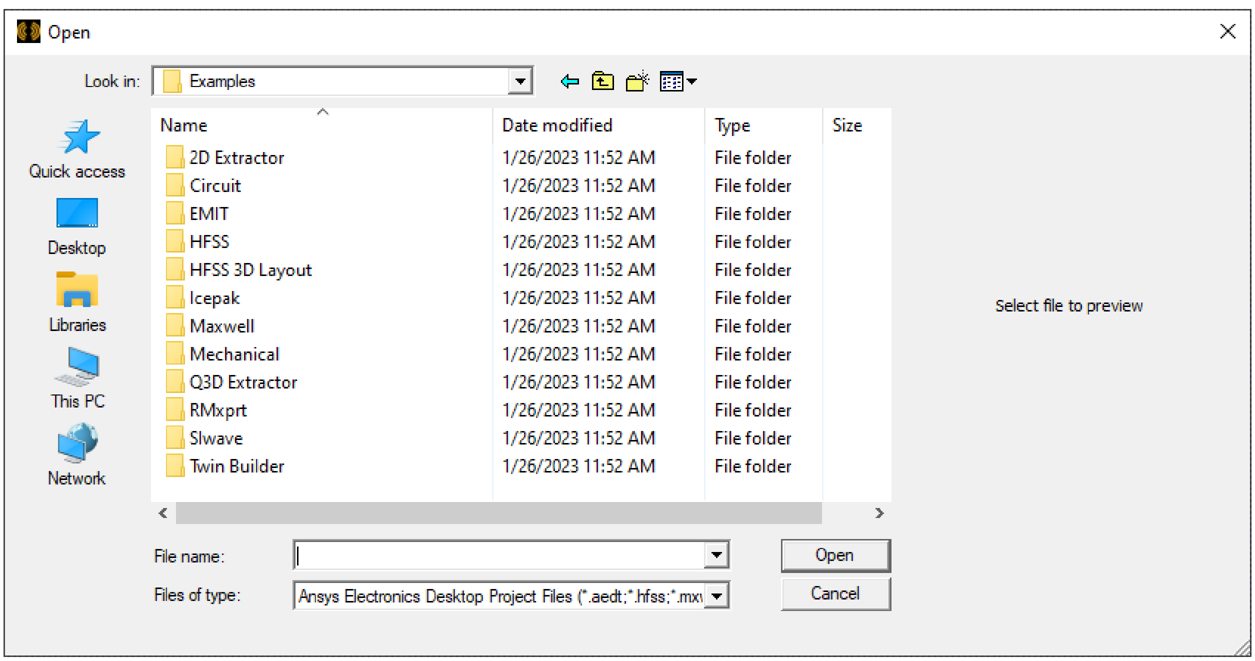
To change additional print settings, click **File** > **Page Setup**.

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. The examples under Circuit, HFSS, and HFSS 3D Layout include subfolders, organizing the projects according to different applications:

- Circuit has subfolders for Automation, High Speed Digital, Low Noise Amplifier, RF Microwave, and Signal Integrity examples. The High Speed Digital folder has further subfolders for PCIe and USB projects. And the RF Microwave folder has further subfolders for Amplifiers, DC-IV, Filters, Misc, Mixers, Oscillators, and System projects.

- HFSS has subfolders for ADAS, Antennas, EMI EMC, Filters, Metamaterial, RCS, RF Microwave, Signal Integrity, and Transmission Lines.
- HFSS 3D Layout has subfolders for Antennas, Component, Filters, FSS, and Signal Integrity.

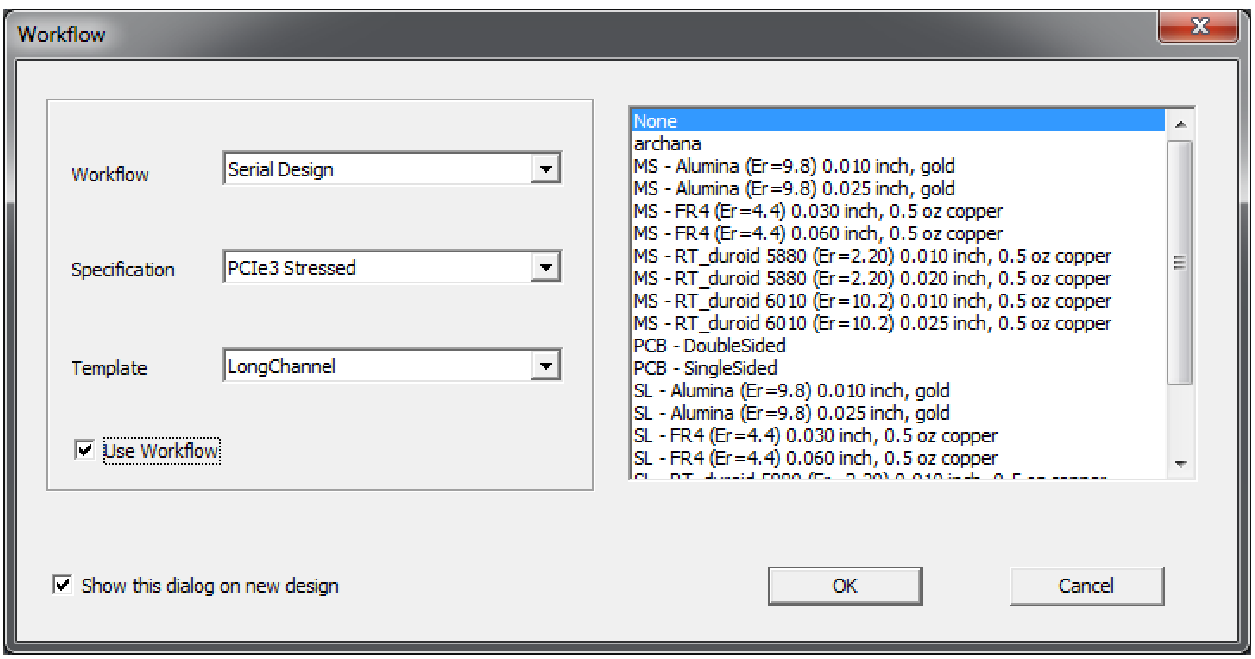
Several of these projects are associated with detailed getting started guides.

Many other projects include descriptions in the help.

- Circuit Example Projects

Workflow Projects in Circuit Design

When you insert a circuit design to create a new project in the Schematic Editor, you can choose to associate the design with a workflow template (specification). The workflow dialog allows you to associate the new design with a workflow. When a new circuit design is inserted, the following dialog opens.



> **Note:**
>
> For the workflow dialog to open automatically when a new circuit design is created, the workflow template option must be configured using the General Options for Circuit designs. For more information, see **Show Workflow dialog on new design** in *Nexxim Circuit Design — Global Options*.

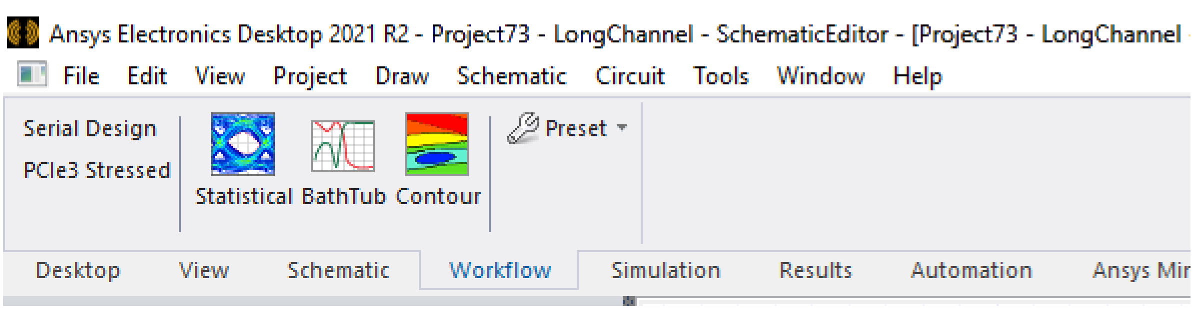
To save an active design as a workflow template in the Schematic Editor you can do either of the following:

- Click **Circuit > Save as Workflow Template**
- Right-click a design in the Project Tree and in the submenu that opens select **Save as Workflow Template**

The information for workflow, specification, and template items is based on data stored in the Workflow directory under Syslib, Userlib and PersonalLib. These data are used to populate the Workflow Ribbon.

Workflow Ribbon

The Workflow Ribbon displays a panel of buttons and menu items that are populated using the python scripts saved in the Workflow directory beneath Syslib, Userlib and Personallib.



- Statistical — Adds a VerifEye setup and displays a Statistical Eye plot.
- Bathtub — Adds a VerifEye setup and displays a Bathtub plot.
- Contour — Adds a VerifEye setup and displays a Contour plot.
- Preset — A drop-down menu that allows you to change between different preset options based on PCIExpress 3 specifications.

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')`

Working with Variables

A variable is a numerical value, mathematical expression, or mathematical function that can be assigned to a design parameter in **Circuit**. You can assign a variable to any dimension, material property, or output value. Variables are useful in the following situations:

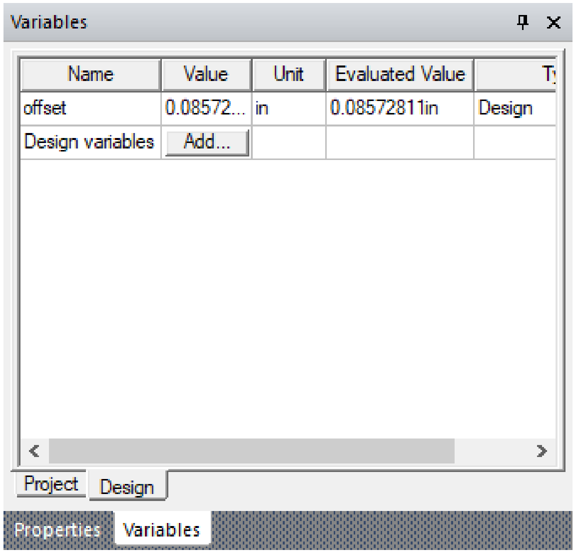
- You expect to change a parameter often.
- You expect to use the same parameter value often.
- You intend to run a parametric analysis, in which you specify a series of variable values within a range to solve.
- You intend to optimize a parameter value by running an optimization analysis or using Design of Experiments to generate a response surface.
- You intend to animate a plot against a variable.

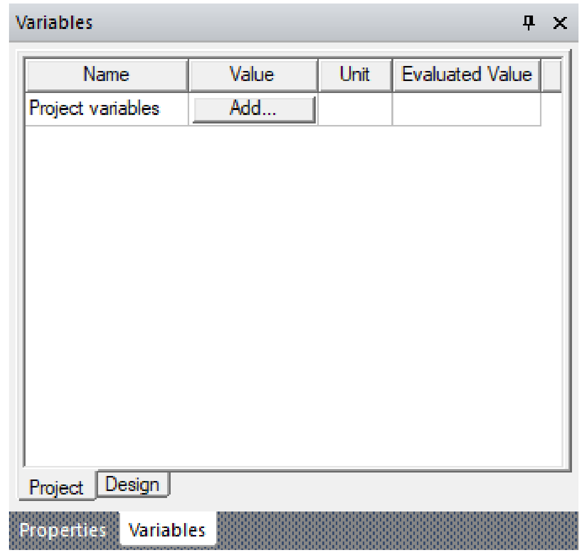
Variables can be regular variables, array index variables, or arrays. A project variable's value can reference other project variables or intrinsics. Values can be a number, variable, or mathematical expression.

There are two types of variables in Circuit:

Project Variables	A project variable can be assigned to any parameter value in the Circuit project in which it was created. Circuit 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 Circuit will automatically append the project variable's name after you define the variable. Project variables can be designated as Design, ArrayIndex or Separator variables .
Design (Local) Variables	A design variable can be assigned to any parameter value in the Circuit 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.

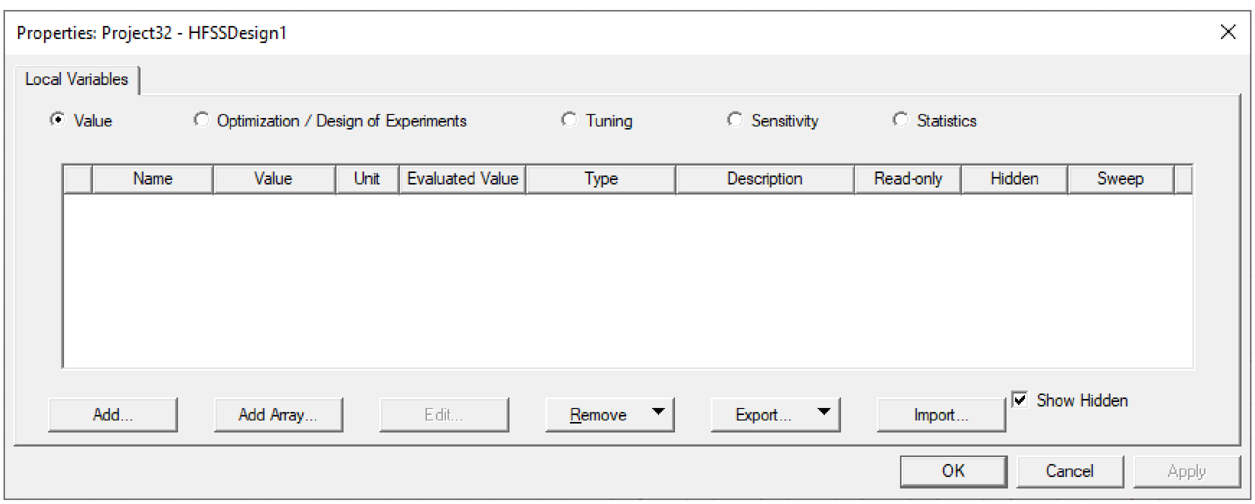
Clicking **View** > **Variables** brings up a dockable variable window that is associated with the active project and/or design. When there is an active project, there will be a corresponding project variable tab. When there is an active design, there will be a corresponding design variable tab. Each 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 Circuit project in which it was created. Circuit 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 Circuit will automatically append it after you define the variable.

The **Project Variables** menu item allows you to access three tabs:

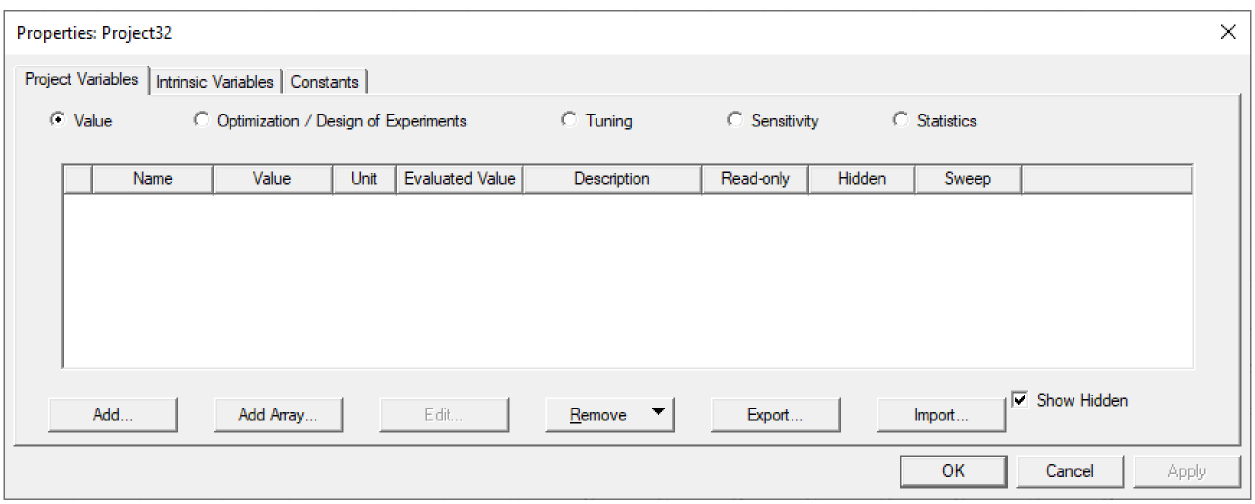
- **Project Variables** – variables, which you define, that apply to the entire project.
- **Intrinsic Variables** – pre-defined variables that cannot be changed. See: List of Intrinsic Variables.
- **Constants** – pre-defined constants that cannot be changed. See: List of Constants.

Adding a Project Variable

To define a Project Variable:

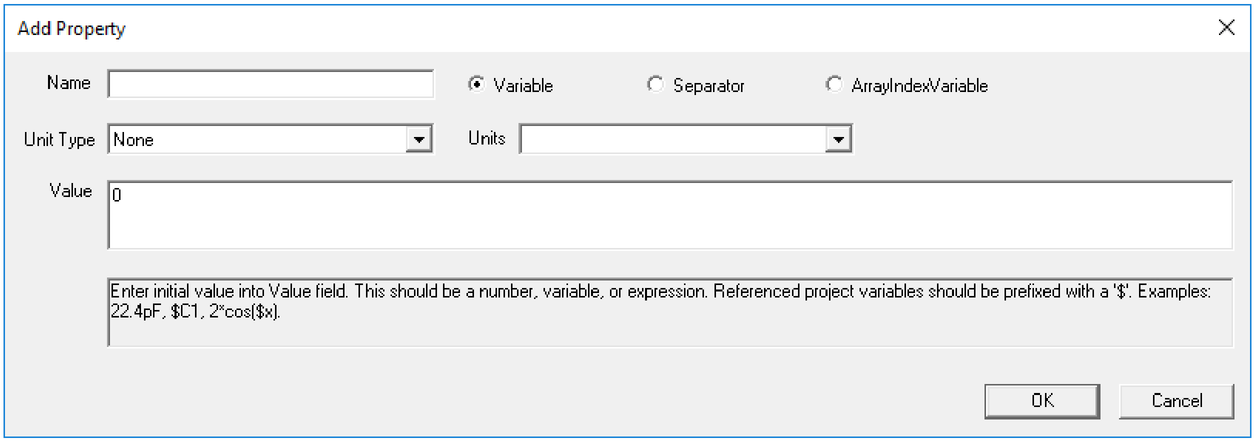
1. Access **Project Variables** one of three ways:
   - Click **Project** > **Project Variables**.
   - Right-click the project name in the Project Tree and select **Project Variables**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **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.
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**.

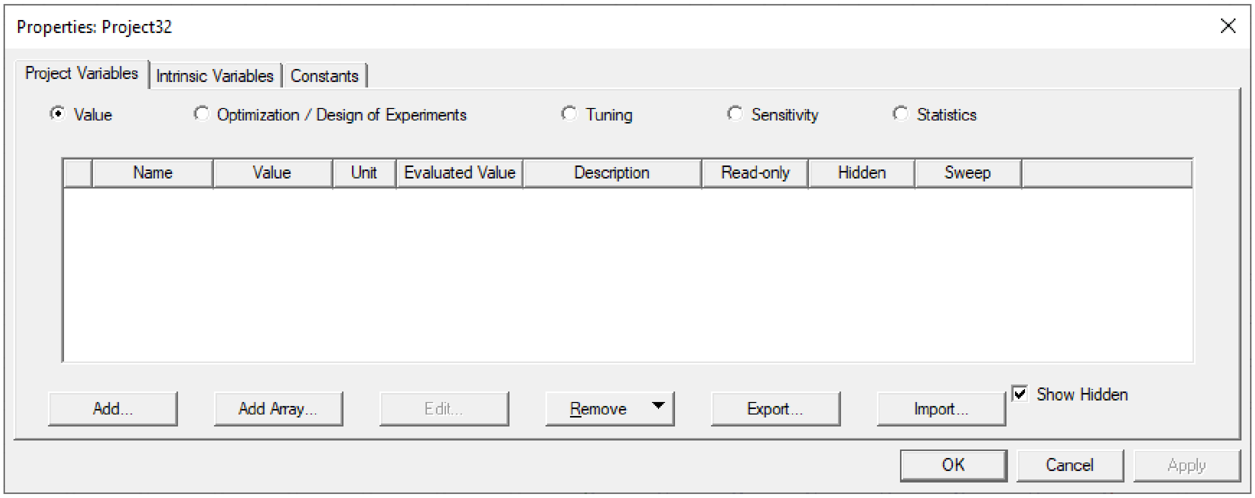
   The new variable appears in the list. You can sort project variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original 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

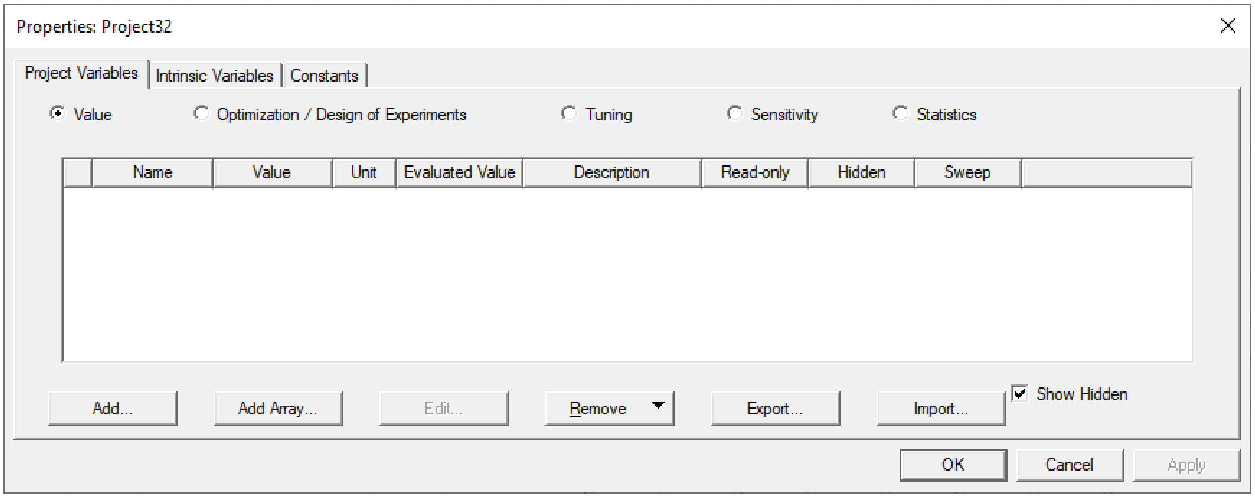
You can define array variables that contain numbers or strings. Number array variables can be used in component property expressions, while string array variables can be used in certain component property values.

> **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:

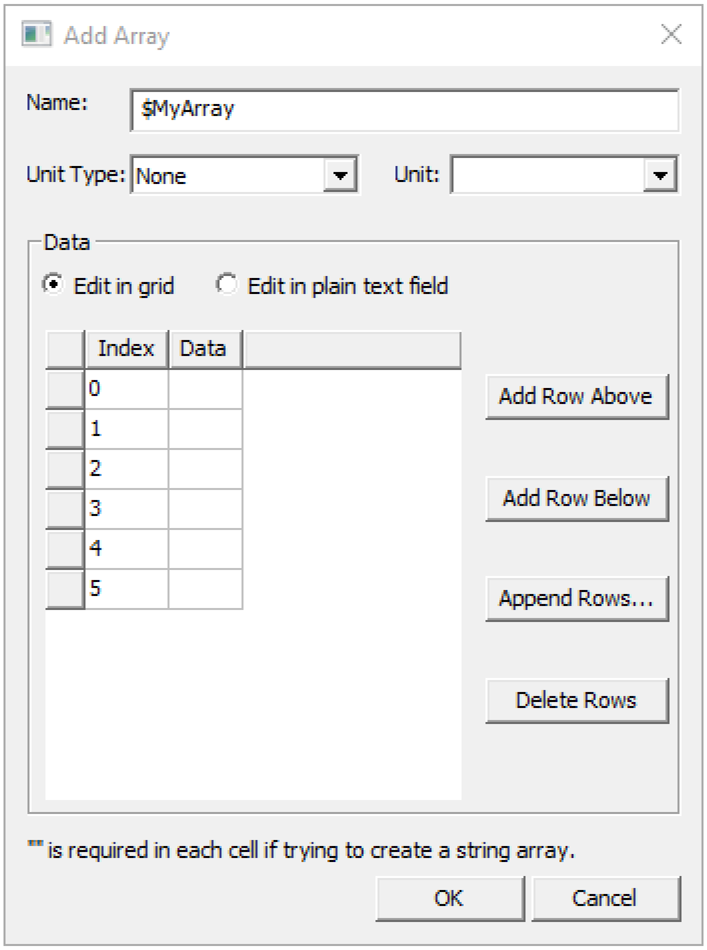
1. Access **Project Variables** one of three ways:
   - Click **Project** > **Project Variables**.
   - Right-click the project name in the Project Tree and select **Project Variables**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **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.

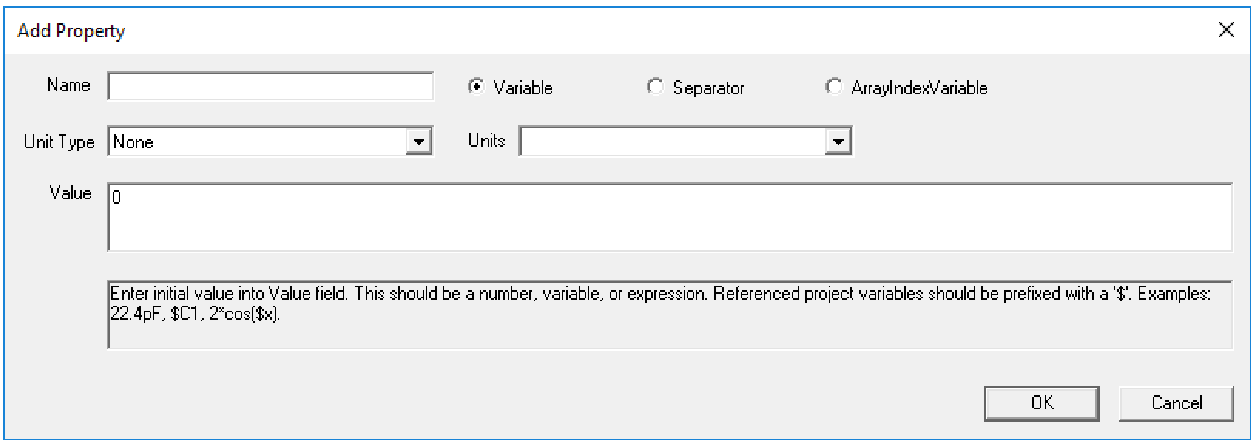
You can add a variable to the array by adding an ArrayIndexVariable on the **Project Variables** tab.

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

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 to _I9	mA	Terminal current in user-defined model (A).
_t		Variable to define parametric equation-based curve.
_u_,_v		Variable to define parametric equation-based surface.
_V1 to _V9	mV	Port Voltage in user defined model (V).
Ang	Ang	Angle. Post Processing variable
Budget Index		Post processing variables, not settable by the user.
Distance	mm	
Electrical Degree	deg	Electric degree of the rotating machine. Not settable by user.
F	GHz	Frequency
F1, F2, F3	GHz	Frequency tones 1, 2, 3 in harmonic balance analysis. (Hz).
FNoi	GHz	Offset noise frequency in harmonic balance noise analysis.
Freq	Hz	Frequency for analysis and post processing, not settable by user
Ia, Ib	mA	Post processing variables, not settable by the user.
Index		Identifier for a data point.
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		Post processing variables, not settable by the user.

Name	Unit	Description
OP	mW	Post processing variables, not settable 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	Post processing variable.
Temp	cell	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, not settable by the user.
X,Y,Z	mm	Point coordinate variables in the modeler.
ZAng	deg	A spherical coordinate system variable
ZRho		A spherical coordinate system variable.

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)
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 Circuit 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.

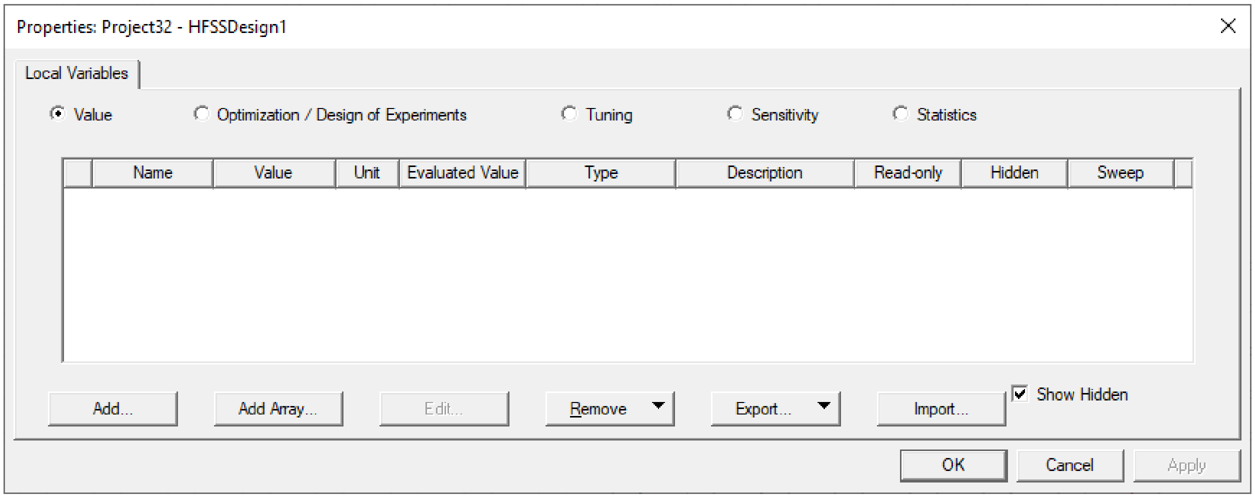
The **Design Properties** menu item allows you to access up to three tabs, depending on your design:

- **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 from a component or at the design level.
- **Local Variables** – variables, which you define, that apply to the current design only. Local variables can be defined from a component or 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.



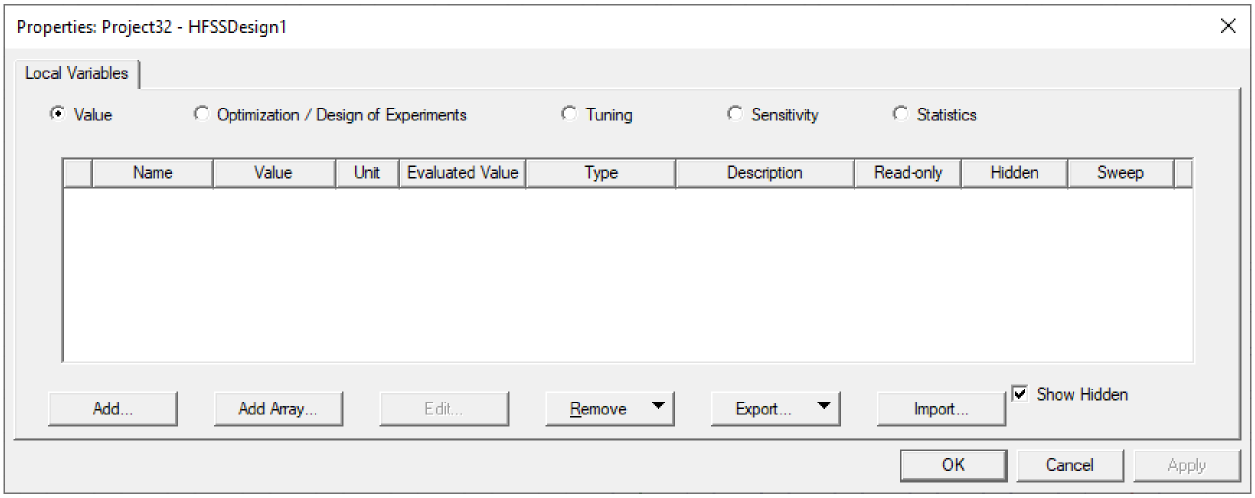
Defining Local Variables at the Design Level

Adding a Local Variable

To define a Local Variable:

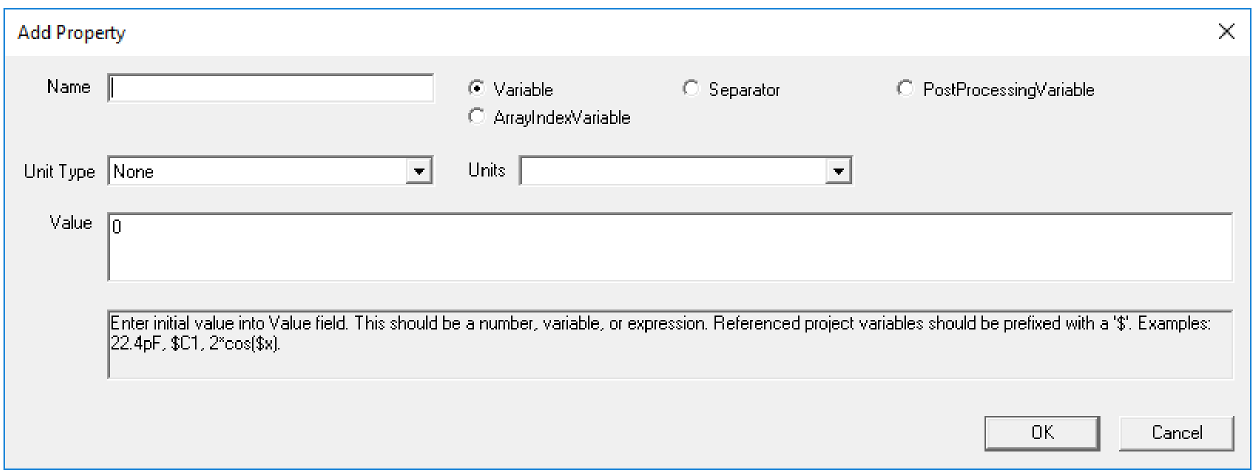
1. Access **Design Properties** one of three ways:
   - Click **Circuit** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **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**.

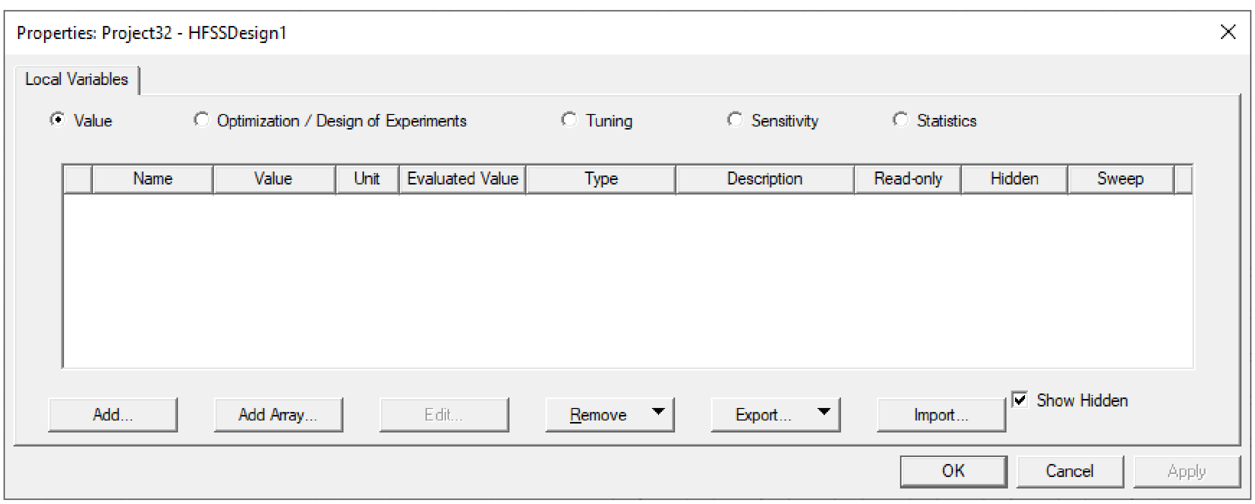
   The new variable appears in the list. You can sort local variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, 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

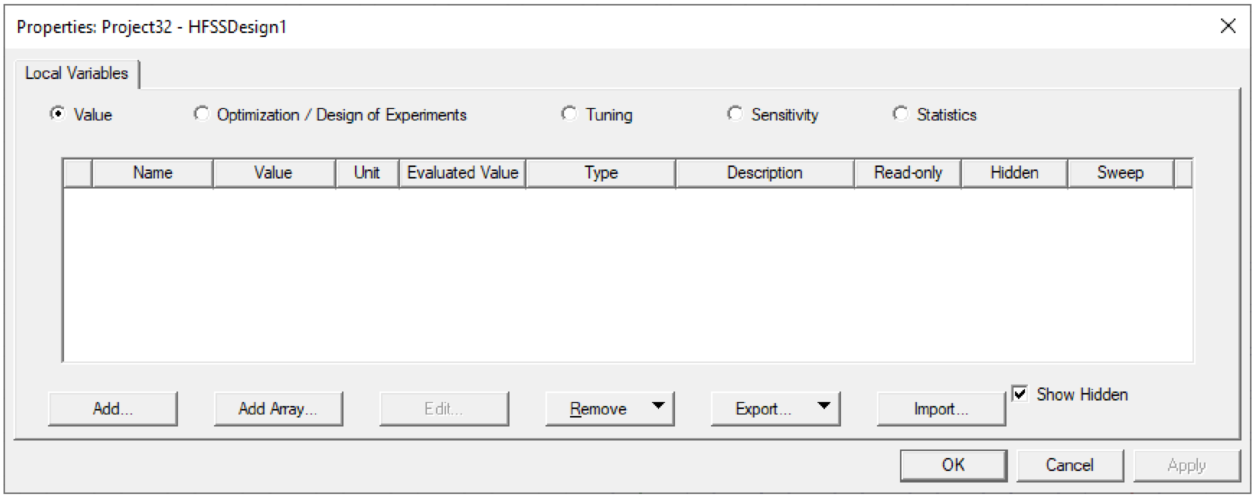
Defining Parameter Defaults at the Design Level

Adding a Parameter Default

To define a new parameter default at the design level:

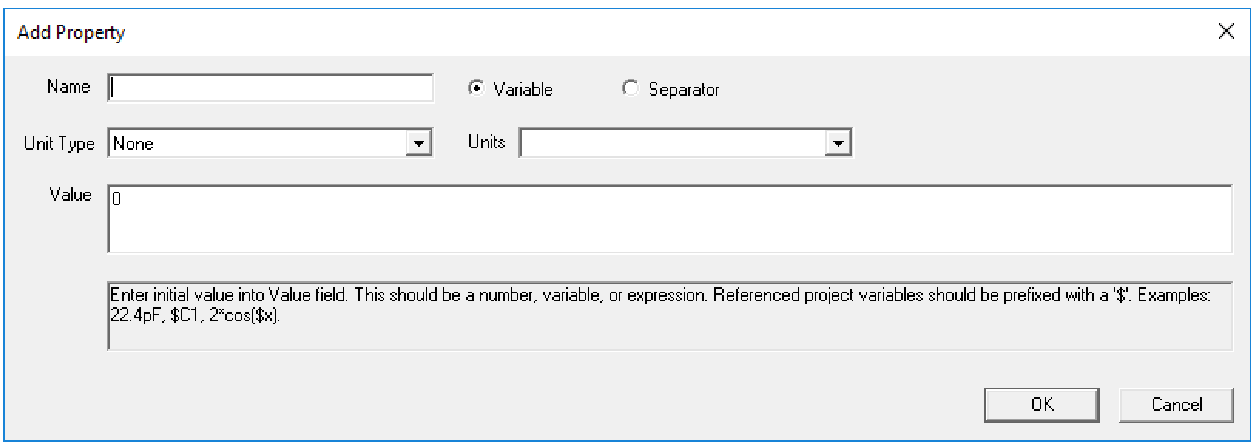
1. Access **Design Properties** one of three ways:
   - Click **Circuit** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **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 Circuit 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:

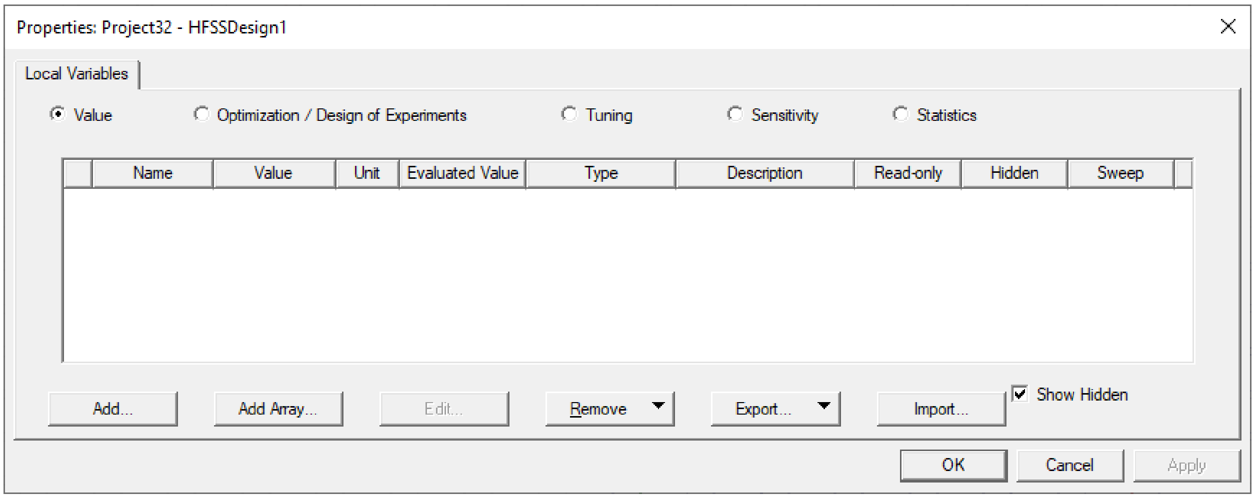
1. Access **Local Variables** one of three ways:
   - Click **Circuit** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.

- 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.

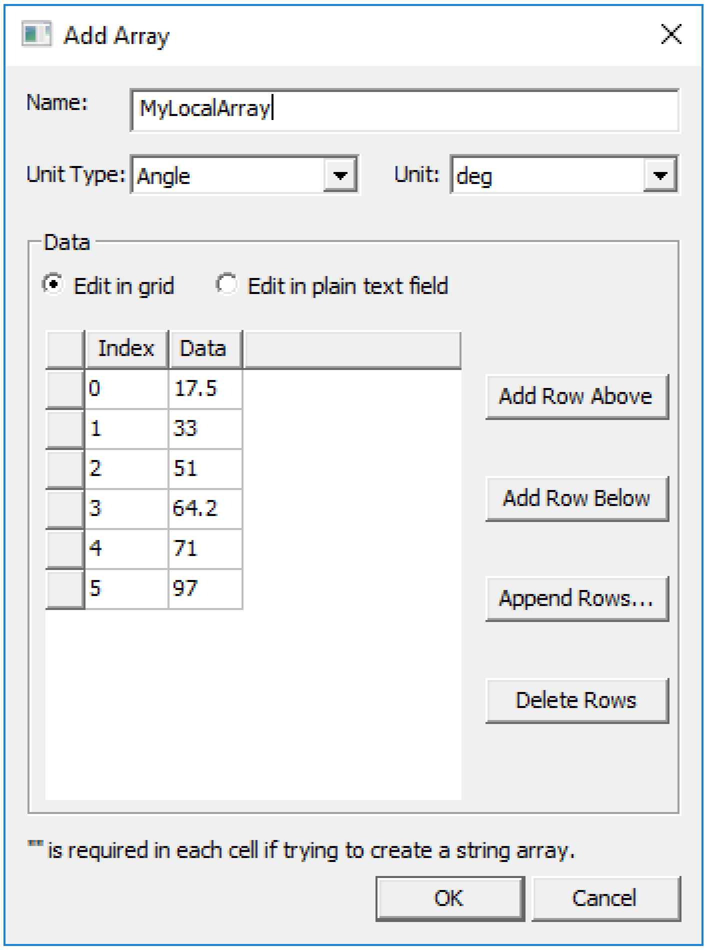
2. Select the **Local Variables** tab.

   The **Local Variables** tab displays.



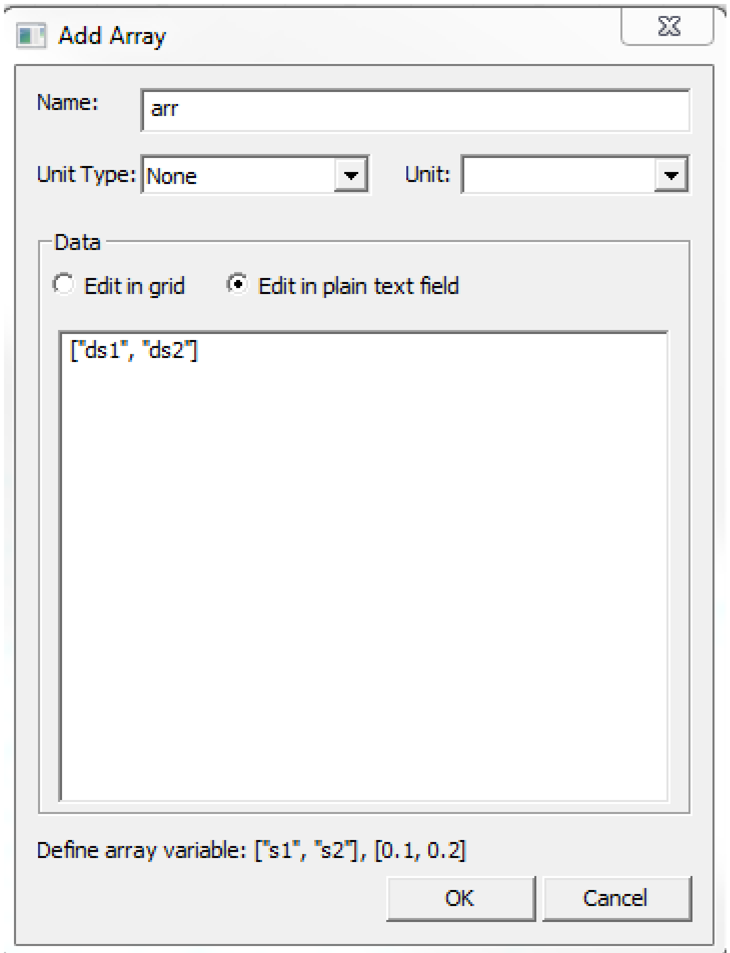
3. Click **Add Array**.

   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.

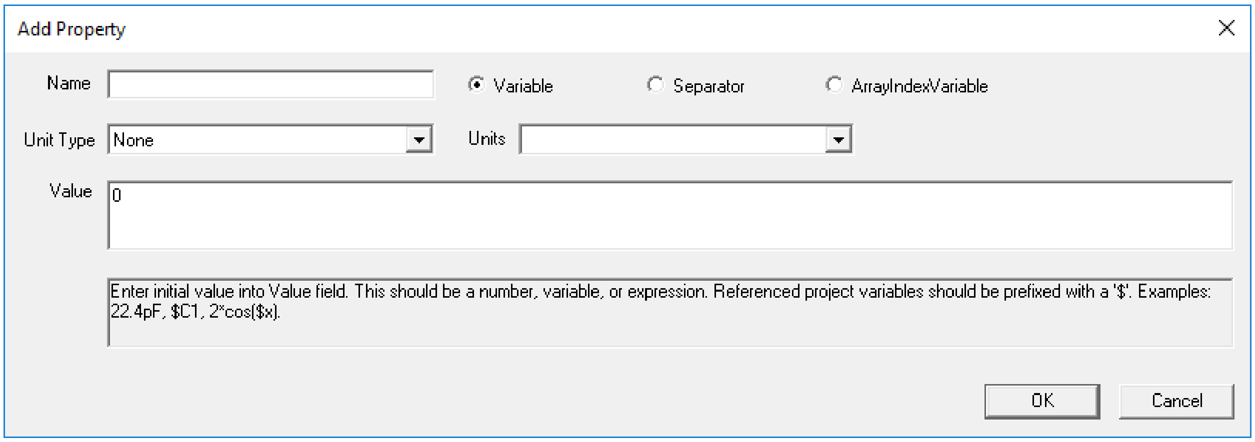
6. When you have finished entering values, click **OK**.

   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 **Circuit** > **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 **Circuit** > **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.

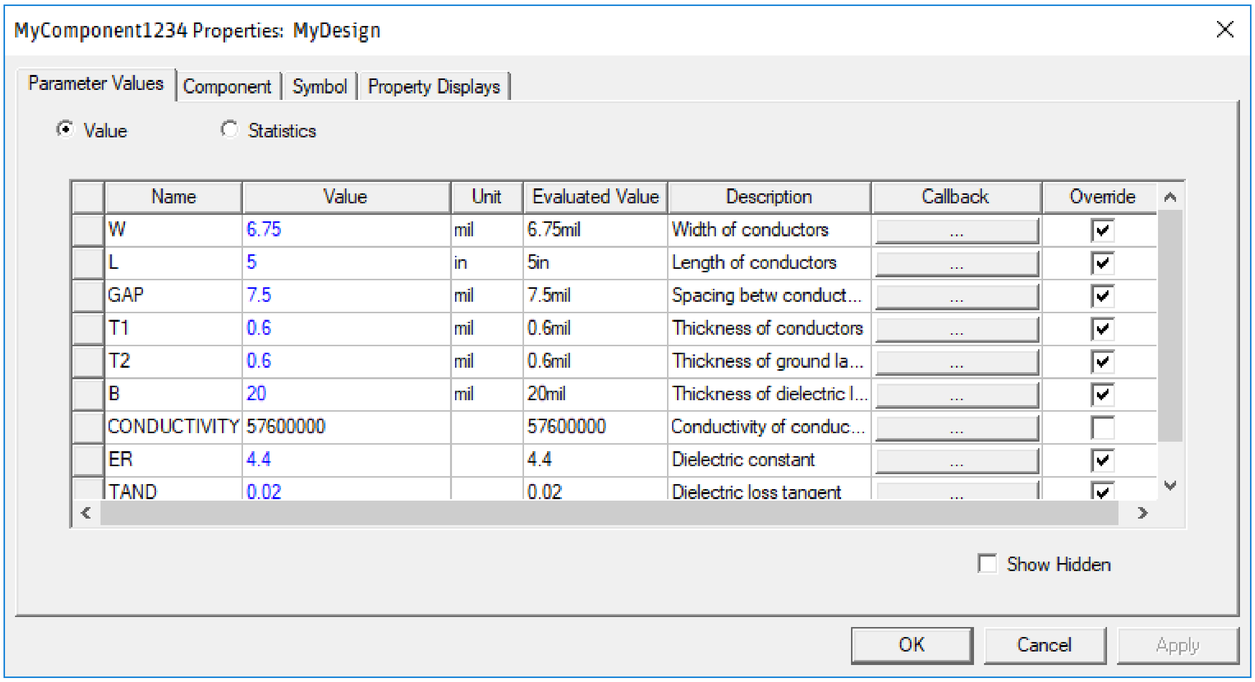
Adding Variables and Parameter Defaults from Components

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.



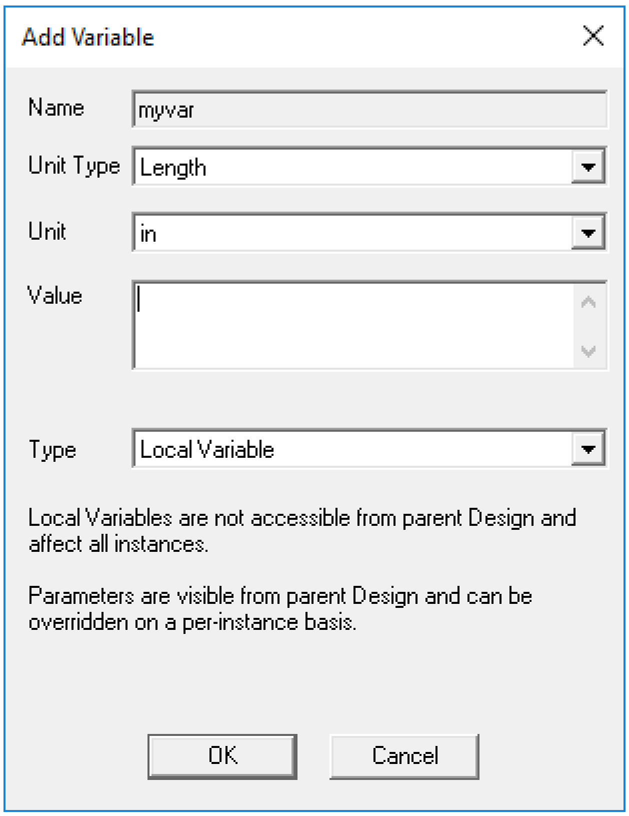
3. Select the **Value** radio button.
4. Select the **Value** field for the parameter you want to set equal to a local variable.
5. Type the variable name. The name may not be the same as an intrinsic variable or constant.

   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.

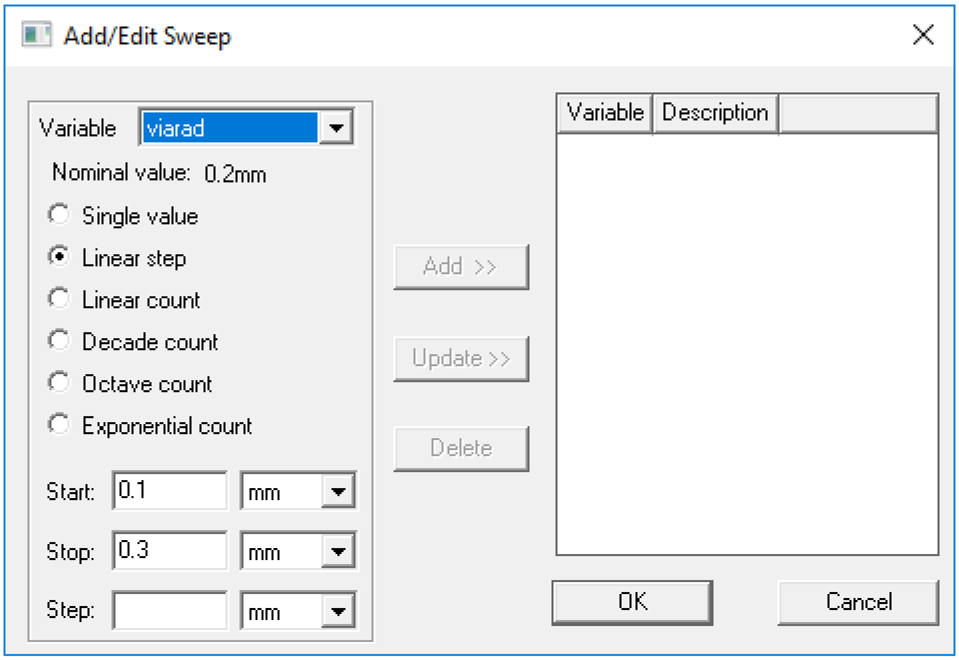
Non-Fixed Variable Sweeps

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.

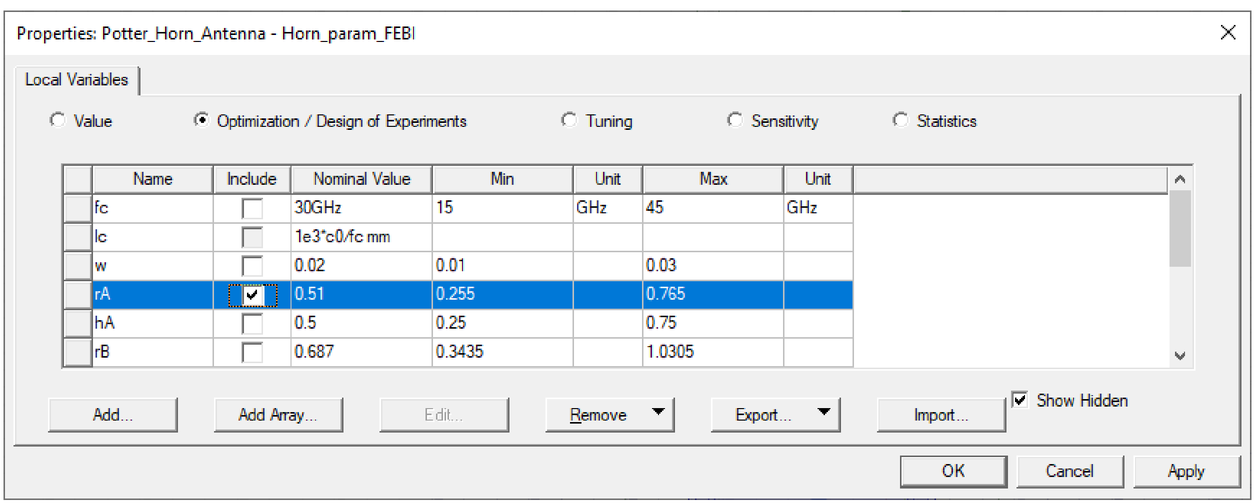
Choosing a Variable to Optimize

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.

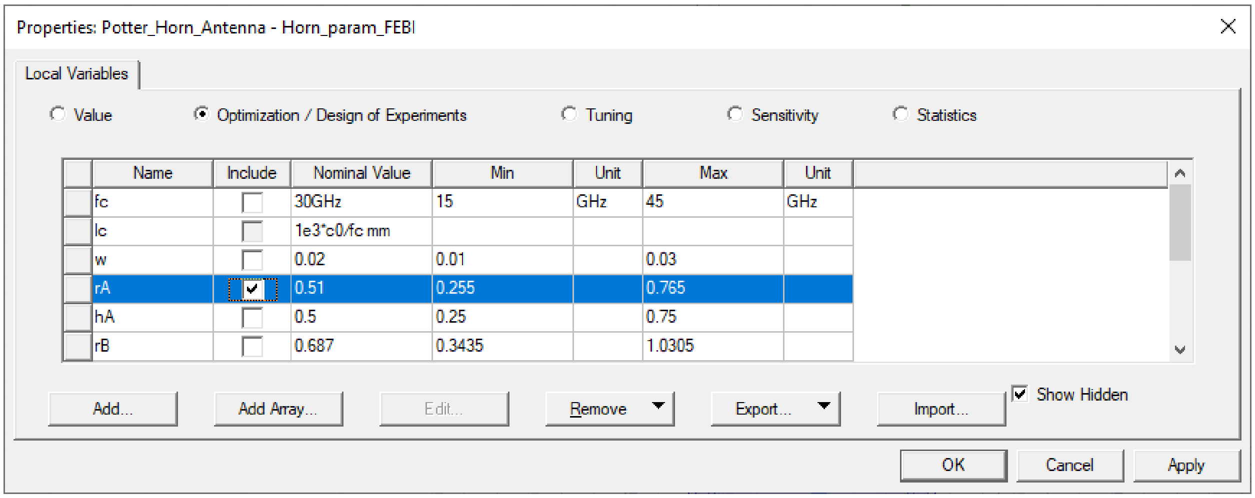
To specify that a variable is to be used during optimization:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Circuit** > **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:

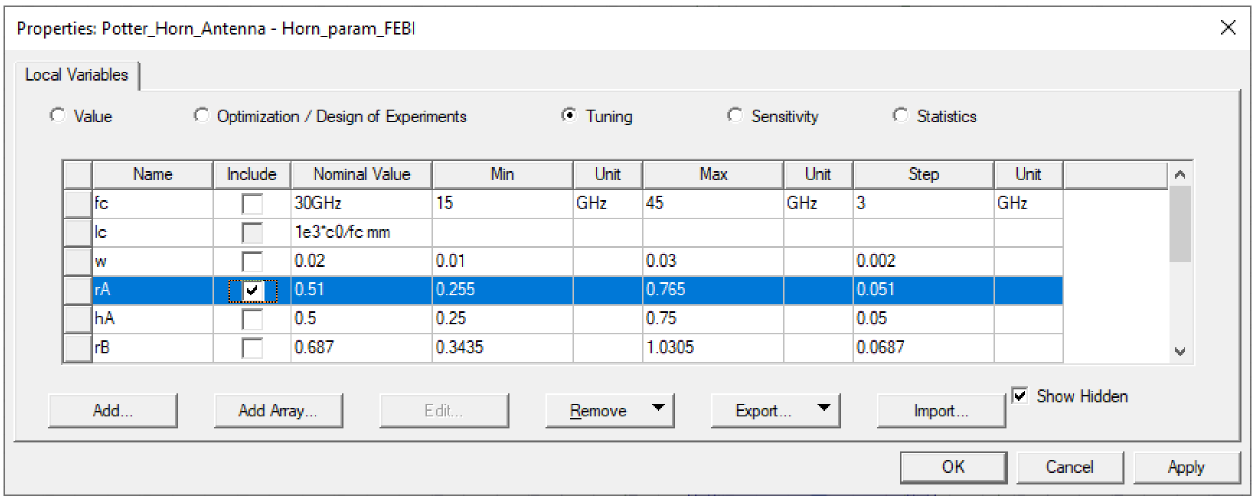
1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Circuit** > **Design Properties**).

   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

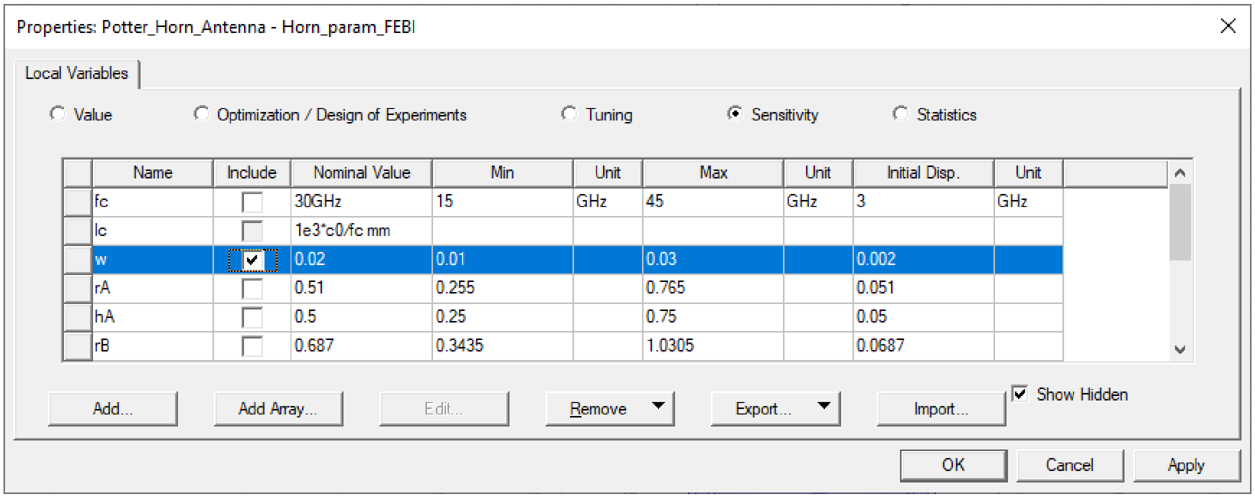
Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **Properties** window.

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 (**Circuit** > **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.

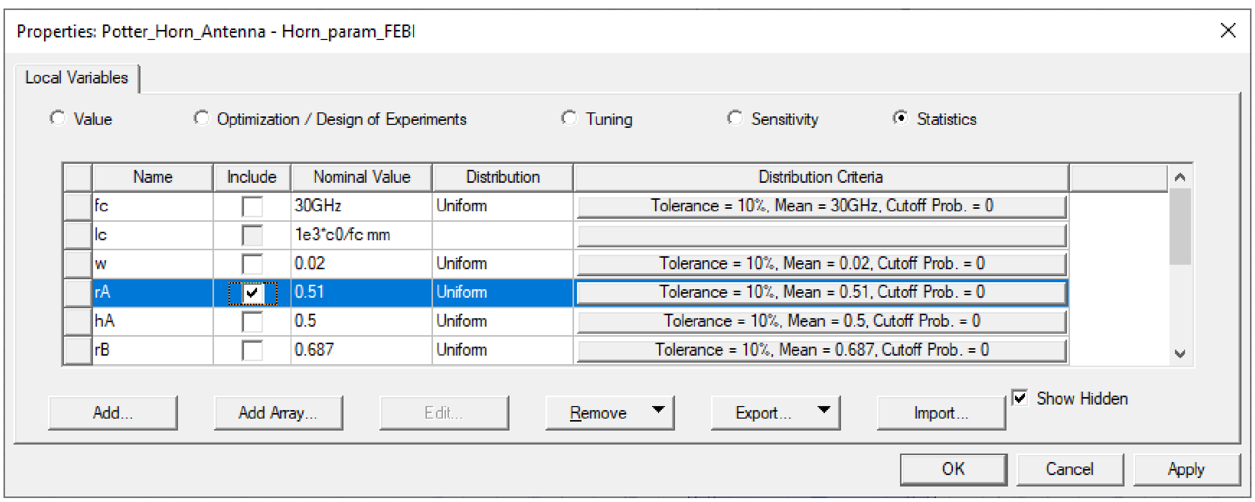
To specify that a variable is to be used during statistical analysis:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**Circuit** > **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**.

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.

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):

<table>
<tr><td>()</td><td>Parenthesis</td><td>1</td></tr>
<tr><td>!</td><td>Not</td><td>2</td></tr>
<tr><td>^ (or **)</td><td>Exponentiation
(If you use "**" for exponentiation, as in previous software versions, it is automatically changed to "^".)</td><td>3</td></tr>
<tr><td>-</td><td>Unary minus</td><td>4</td></tr>
<tr><td>*</td><td>Multiplication</td><td>5</td></tr>
<tr><td>/</td><td>Division</td><td>5</td></tr>
<tr><td>+</td><td>Addition</td><td>6</td></tr>
<tr><td>-</td><td>Subtraction</td><td>6</td></tr>
<tr><td>==</td><td>Equals</td><td>7</td></tr>
<tr><td>!=</td><td>Not equals</td><td>7</td></tr>
<tr><td>></td><td>Greater than</td><td>7</td></tr>
<tr><td><</td><td>Less than</td><td>7</td></tr>
<tr><td>>=</td><td>Greater than or equal to</td><td>7</td></tr>
<tr><td><=</td><td>Less than or equal to</td><td>7</td></tr>
<tr><td>&&</td><td>Logical and</td><td>8</td></tr>
<tr><td>||</td><td>Logical or</td><td>8</td></tr>
</table>

Using Intrinsic Functions in Expressions

Circuit 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)
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)

Variable Sweep

The DC, transient, harmonic balance, AC, and linear network analyses can be performed multiple times while sweeping the value of one or more variable quantities. Variable sweeps can be nested. The innermost nested variable runs through its specified sweep values at each of the specified values for the next outer sweep variable, and so on. For DC and transient analyses, a sweep can be combined with a Monte Carlo analysis. For details see Monte Carlo Analysis.

Related Topics:

Variable Types

Running Variable Sweeps from the Schematic Editor

Running Variable Sweeps from the Netlist Editor

Variable Sweep Netlist Examples

Sweeping Temperature with the TEMP Statement

Viewing a Sweep Analysis in a Report

Restrictions on Sweeping a Choice of Subcircuits

Variable Types

Variables of three types can be defined in project variables, local variables, and parameter defaults.

A *project variable* is available across all hierarchical levels of a project, and can be identified by its dollar sign ($) prefix, as in $C1. If, for instance, a project variable $C1 has been defined as equal to 4.32pF, a capacitance property for a component anywhere in that project can be set equal to 4.32pF by typing $C1 in the appropriate **Value** field.

A *local variable* is available only within the design for which it is defined. If, for instance, a local variable R2 has been defined as equal to 4316Ohms, a resistance property anywhere in that design can be set equal to 4316Ohms by entering R2 in the appropriate **Value** field.

A local variable takes the same value across multiple instances of a design. If three subcircuit instances contain a local variable R2 that is defined as 4316Ohms, redefining R2 as 1625Ohms in any of the instances redefines R2 as equal to 1625Ohms in all three instances.

A *parameter default* is a local variable with a default value that can be uniquely overridden in different 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*.

If you define a variable as an expression that evaluates to a constant, whether a project, local or parameter variable, the expression will be retained in the variable list, rather than being evaluated and replaced with a constant. This allows you to identify and modify the expression in the future.

> **Note:**
>
> Whether you are defining project, local, or parameter variables, intrinsic names (f, freq, lb, etc.) are reserved and cannot be used or entered into the source and port dialogs.

In this chapter on Variable Sweep we show how to use local variables as the values of parameters to be swept. The uses of project variables and parameter defaults are not described in this help topic.

Running Variable Sweeps from the Schematic Editor

In the Schematic Editor, variable sweep may be specified as part of the Solution Setup. The types of sweep available from the Schematic Editor depend on the type of simulation that is being set up.

- DC Analysis: Local variables, temperature
- Transient Analysis: Local variables, temperature
- Harmonic Balance: Local variables, temperature, frequency

Related Topics

Setting Up a Local Variable on a Component Parameter

All simulations allow a local variable to be swept. To create a local variable on a component for parameter sweep, perform the following steps:

1. Click on the component to bring up its **Parameter** window.
2. Click on the value of the parameter and enter the name of the local variable to be created. Press **Return**.
3. The **Add Variable** dialog box opens. The local variable name you entered in the **Parameter** window appears in the **Name** field. Enter a default value for the local parameter in the **Value** field. Click the **Local Variable** radio button. Click **OK**.
4. Repeat this procedure until all desired component parameters have been set up as variables.

Adding a Sweep of Temperature or of a Local Variable

Use the following steps with any analysis to add a sweep of temperature or of a local variable:

1. To add a sweep of a local variable or temperature, click **Add** on the **Sweep Variables** panel.
2. The **Add/Edit Sweep** dialog box opens. Select the temperature (**TEMP**) or a local variable from the **Variable** field.

   Sweep options are:

   - **Single Value** (enter the value and units)
   - **Linear Step** (enter **Start**, **Stop**, and **Step** values and units)
   - **Linear count** (enter **Start**, **Stop**, and **Count** values and units)
   - **Decade count** (enter **Start**, **Stop**, and **Count** values and units)
   - **Octave count** (enter **Start**, **Stop**, and **Count** values and units).
3. Click **Add**.
4. Repeat steps 2 and 3 until all parameter sweeps have been specified.
5. Click **OK**.
6. Complete the **Solution Setup** information (**Output Quantities**, etc.), then click **Finish**.

NOTE: if you specify two or more sweeps for the same variable, the dialog box will show the individual sweeps, for example:

```
Variable Sweep
A LIN 1 5 1 LIN 2 3 0.2
```

However, this specification turns into a single (POI) sweep when you browse the netlist:

```
.sweep A POI 9 1 2 2.2 2.4 2.6 2.8 3 4 5
```

Setting Up a Frequency Sweep with Harmonic Balance

Nexxim harmonic balance analysis supports a sweep of frequency using a local variable for the frequency value. (For DC analysis or transient analysis, no frequency parameters are predefined.) To perform a frequency sweep with harmonic balance analysis, perform the following steps:

1. Under **Circuit**, select **Add Nexxim Solution Setup**. Select **Harmonic Balance** for the **Analysis Type** and **1-Tone** or **N-Tone** for the **Category** as appropriate. Click **Next**.
2. The **Harmonic BalanceAnalysis** dialog box appears.
3. For N-Tone analysis, specify the No. of Tones. The **Max. Harmonic Number** table shows the frequency or frequencies. Specify the **MaxK** value for each frequency. To set up a sweep, click in the **Value** field for a frequency and enter the name of the local variable to be created. Press **Return**.
4. The **Add Variable** dialog box opens. The local variable name you entered in the **Parameter** window appears in the **Name** field. Enter a default value for the local parameter in the **Value** field. Click the **Local Variable** radio button. Click **OK**.
5. Repeat steps 3 and 4 until all desired frequencies have been set up as variables.
6. Locate the **Sweep Variables** panel at the right side of the **Solution Setup** box.
7. The **Add/Edit Sweep** dialog box opens. Select the local variable from the **Variable** field. Sweep options are:
   - **Single Value** (enter the value and units)
   - **Linear Step** (enter **Start**, **Stop**, and **Step** values and units)
   - **Linear count** (enter **Start**, **Stop**, and **Count** values and units)
   - **Decade count** (enter **Start**, **Stop**, and **Count** values and units)
   - **Octave count** (enter **Start**, **Stop**, and **Count** values and units).
8. Click **Add**.
9. Repeat steps 7 and 8 until all frequency sweeps have been specified. If desired, add sweeps of component-value local variables using the procedure given under Setting Up a Local Variable on a Component Parameter.
10. Click **OK**.
11. Complete the **Solution Setup** information (**Output Quantities**, etc.), then click **Finish**.

Setting Up a Frequency Sweep with Linear Network Analysis

Nexxim linear network analysis supports a built-in sweep of frequency. (For DC analysis or transient analysis, no frequency parameters are predefined.) To perform a frequency sweep with linear network analysis, perform the following steps:

1. Under **Circuit**, select **Add Nexxim Solution Setup**. Select Linear Network Analysis for the **Analysis Type** and accept Frequency Domain for the **Category**. Click **Next**.
2. The **Linear Network Solution Setup** dialog box appears.
3. For Linear Network Analysis, the **SweepVariables** panel has one frequency parameter (**Freq**). To set up a sweep of frequency, click **Add**.
4. The **Add/Edit Sweep** dialog box opens, with the selected frequency shown in the **Variable** field (which cannot be modified).

Sweep options are:

- **Single Value** (enter the value and units)
- **Linear Step** (enter **Start**, **Stop**, and **Step** values and units)
- **Linear count** (enter **Start**, **Stop**, and **Count** values and units)
- **Decade count** (enter **Start**, **Stop**, and **Count** values and units)
- **Octave count** (enter **Start**, **Stop**, and **Count** values and units).

5. Click **Add**.
6. If desired, add sweeps of local variables using the procedure given under Setting Up a Local Variable on a Component Parameter.
7. Click **OK**.
8. Complete the setup information (**Output Quantities**, etc.), then click **Finish**.

Running Variable Sweeps from the Netlist Editor

If you use the Netlist Editor to create a design, the netlist should contain the information for running the sweep analysis. No Solution Setup is required.

Related Topics

To Run a Sweep with Any Analysis

Variable Sweep Netlist Syntax

To Run a Sweep with Any Analysis

1. Specify the sweep as part of simulation statement (.DC, .TRAN, .HB, or .LNA), using the syntax described in the next section.
2. Run the simulation:

   a. On the **Circuit** menu, click **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a green progress bar appears.
   b. If the analysis is not successful, check the **Message Window** for an explanation, and then take corrective action.
3. Display results:
   a. On the menu bar, click **Circuit** and then click **Create Report**. The **Create Report** dialog box appears.
   b. When the **Report** dialog box appears, make the appropriate selections, click **Add Trace**, and then click **Done**.

Variable Sweep Netlist Syntax

Depending on the analysis, the netlist can specify a sweep of any of the following variables:

- The simulation temperature.
- A netlist parameter.
- An instance parameter for a circuit device.

To run a variable parameter sweep with any analysis, add the following netlist syntax at the end of the analysis statement:

`SWEEP variable1 sweep_spec1 [variable2 sweep_spec2 ...]`

Sweep Variable

Each *variable* entry specifies a variable to be swept:

- The keyword **TEMP** to sweep the temperature.
- The name of a netlist parameter as it is defined in a .PARAM statement elsewhere in the netlist.
- The name of a circuit device followed by the name of one of its instance parameters with a space separator, for example R5R for the resistance of resistor R5, or Q23area for the area of BJT Q23.
- For an independent voltage or current source, the name of the DC voltage or current parameter is optional. For voltage or current sources, if no parameter is given, the DC voltage or current is swept. For example, V21DC,V21V and V21 all sweep the DC voltage of source V1.

Sweep Range

sweep_spec specifies the range of values to be swept, using one of three syntax forms.

1. The first form of *sweep_spec* specifies the starting and ending values, and the incremental step:

`[START=]start_val [STOP=]stop_val [STEP=]step_val`

If you omit the **START=**, **STOP=**, and **STEP=** labels, the start value, stop value and step value must be entered in the order shown. If you use the labels, the values may be entered in any order, but then all three values must have labels. For example, the following sweep specifications both call for a sweep of variable parameter A from 1 to 10 in increments of 2:

```
SWEEP A START=1 STOP=10 STEP=2
SWEEP A 1 10 2
```

Both of these specifications generate simulations with A equal to 1, 3, 5, 7, 9, and 10.

However, the following sweep specification is not valid (if you use the labels, all three values must be labeled):

```
SWEEP A START=1 10 STEP=2
```

2. The second form of *sweep_spec* specifies a pattern of values to be swept with *numpoints*, *start_val* and *stop_val*, using one of the predefined sweep types **LIN**, **DEC**, or **OCT**:

LIN|**DEC**|**OCT** *numpoints start_val stop_val*

Numpoints must be a positive integer value. If you give zero or a negative value for *numpoints*, an error is generated. If you give a positive non-integer value for *numpoints*, it is truncated to an integer without warning.

The **LIN**, **DEC**, and **OCT** keywords specify sweeps as follows:

LIN

Linear sweep at *numpoints* different values, starting from the value given by *start_val* to the value given by *stop_val*. The spacing between test values is (*stop_val*–*start_val*) / (*numpoints – 1)*. For example, the following sweep specification calls for a linear sweep of variable A at 5 points starting at 1 and ending at 10:

```
SWEEP A LIN 5 1 10
```

This specification generates simulations with A equal to 1, 3.25, 5.5, 7.75, and 10.

DEC

Logarithmic sweep at *numpoints* different values in each decade interval (10× multiple of *start_val*) from *start_val* to the first point at or beyond *stop_val*. The spacing between test values is logarithmic within decades. For example, the following sweep specification calls for a logarithmic sweep of variable A at 2 points per decade starting at 1 and ending at 100:

```
SWEEP A DEC 2 1 100
```

This specification generates simulations with A equal to 1, 3.16, 2, 31.6, and 100. In this example, the *stop_val* is an exact decade of the *start_val*, and the sweep thus includes the *stop_val* exactly. See the next example for the case where the *stop_val* is not an exact decade or octave of the *start_val*.

OCT

Logarithmic sweep at *numpoints* different values in each octave interval (2× multiple of *start_val*) from *start_val* to the first point at or beyond *stop_val*. The spacing between test values is logarithmic within octaves. For example, the following sweep specification calls for a logarithmic sweep of variable A at 2 points per octave starting at 1 and ending at 2.5:

```
SWEEP A OCT 2 1 2.5
```

This specification generates simulations with A equal to 1, 1.414, 2, and 2.828. In this example, the *stop_val* is not an exact octave of the *start_val*, so the sweep includes the first point greater than the *stop_val* that also matches the OCT specification.

3. The third form of *sweep_spec* provides a space-separated list of *num_pts* values:

POI*num_pts pt1 ... ptN*

For example, the following sweep specification calls for a sweep of variable A at 5 points, 1, 2.7, 3.2, 5, and 11.5:

```
SWEEP A POI 5 1 2.7 3.2 5 11.5
```

Variable Sweep Netlist Examples

These examples show the various ways to specify a sweep using the different analysis types.

DC Analyses with Sweeps

This example performs DC analyses while sweeping the DC voltage of source VIN from zero to 5 Volts in increments of 1 Volt.

```
.DC SWEEP VIN DC START=0.0 STOP=5.0 STEP=1.0
```

This example performs DC analyses while sweeping two DC voltages: VIN from zero to 5 Volts in increments of 1 Volt, and VREF from zero to 1 volt in 5 increments.

```
.DC SWEEP VIN DC START=0.0 STOP=5.0 STEP=1.0 VREF DC LIN 5 0 1
```

Transient Analysis with Sweep

Transient analysis has a built-in time sweep that is part of the basic syntax. This example performs a series of transient analyses from time 0 to one microsecond in increments of 1 nanosecond, with the temperature sweeping over 10 values from 10 degrees to 30 degrees Celsius at each time point.

```
.TRAN 1e-12 1e-6 SWEEP TEMP LIN 10 10 30
```

Harmonic Balance with Sweep

These two examples run single-tone harmonic balance analyses with an input frequency of 1000 Hz over seven harmonics, changing the value of the area of BJT Q34 from 0.00005 to 0.00020 meter in increments of 0.00005 meter for each analysis. The first example uses a defined .PARAM to create the area sweep variable:

```
.PARAM bjtarea=1e-4
```

```
.Q34 10 10 0 0 bjtmodel1 AREA=bjtarea
.MODEL bjtmodel1 npn level=1
... $ Remainder of circuit not shown
.HB TONES=1000 MAXK=7
+ SWEEP bjtarea START=0.5e-4 STOP=2.0e-4 STEP=5.0e-5
```

The second harmonic balance example sweeps the AREA instance parameter directly. The effect is the same as in the previous example.

```
.Q34 10 10 0 0 bjtmodel1
.MODEL bjtmodel1 npn
... $ Remainder of circuit not shown
.HB TONES=1000 MAXK=7
+ SWEEP Q34 AREA START=0.5e-4 STOP=2.0e-4 STEP=5.0e-5
```

Linear Network Analysis with Sweep

Linear network analysis also has a built-in frequency sweep. The following example combines linear network analysis over a single decade with ten logarithmically distributed frequencies, with a linear sweep of an inductance value over five values at each frequency.

```
.LNA DEC 10 5e8 5e9 flag='LNA'
+ SWEEP L2 L START=0.0 STOP=0.2 STEP=0.05
.PRINT AC V(1) V(2)
```

Oscillator and Phase Noise Analysis with Sweep

Oscillator analysis has a phase noise option that can employ a sweep of temperature or other variable in addition to its built-in sweep of frequency. The following example combines oscillator phase noise analysis over ten linear frequency values with a temperature sweep of five points.

```
.OSC TONES=1E6 MAXK=3 NOISE_OUTPUT=[V(NET_3),I(VD)]
+ HARMONIC_NUMBER=1 LIN 10 0.5e6 2E6
+ SWEEP TEMP POI 5 15 20 25 30 35
```

Sweeping Temperature with the TEMP Statement

Instead of the SWEEP TEMP specification, the netlist can specify a sweep of temperature with the **.TEMP** statement:

```
.TEMP temp1 [temp2 ...]
```

For example, to sweep the ambient temperature over three values (10, 20, and 30 degrees Celsius):

```
.TEMP 10 20 30
.DC
```

is equivalent to

```
.DC SWEEP TEMP POI 3 10 20 30
```

The separate **.TEMP** sweep affects only those analyses that do not have a specific TEMP sweep in the analysis statement. The following example shows how the separate TEMP sweep statement affects other sweep specifications:

```
.PARAM A=1
.TEMP 10 20 30 $ Sweeps three temperature points
.DC SWEEP A LIN 4 1 4 $ Sweeps a total of 3x4=12 points
.DC SWEEP TEMP LIN 5 50 54 $ Sweeps a total of 5 points only
```

The first .DC statement is affected by the .TEMP sweep, so a total of twelve points are swept, three temperatures times four values of parameter **A**. The second .DC statement contains its own temperature sweep of 5 points, so the separate .TEMP statement is ignored and only the five temperature points are swept.

Viewing a Sweep Analysis in a Report

After running a sweep of a local variable, frequency, or temperature, you can plot any calculated or derived quantity against one or more values of the swept quantity.

By default, all values of any swept variable will be plotted. To control the display of sweep data, click the **Sweeps** tab in the **Report** dialog.

See Sweeping a Variable in a Report for details.

Restrictions on Sweeping a Choice of Subcircuits

When you sweep a selection of subcircuits and the instance names of those subcircuits are the same, any terminal name of one must either be unused in the others or used with the identical external connectivity.

The following example violates that rule in three different ways, any one of which is enough to produce unexpected results during simulation.

```
* subs_wrong.sp
* This is an example of three different incorrect
* uses of sweeping a subcircuit choice
.param S=1
V0 1 0 DC 5
X0 1 0 outer
.subckt outer a k
.if (S==1)
```

```
X1 a k inner1
.elseif (S==2)
X1 a k inner2
.elseif (S==3)
X1 a k inner3
.else
X1 k a inner1 / Problem #3
.endif
.ends
.subckt inner1 g h
R1 g h R=10
.ends
.subckt inner2 h g / Problem #1
C1 h g 1e-8
.ends
.subckt inner3 g f
R1 g h R=10 / Problem #2
C1 h f 1e-6
.ends
.dc SWEEP X 1 4 1
```

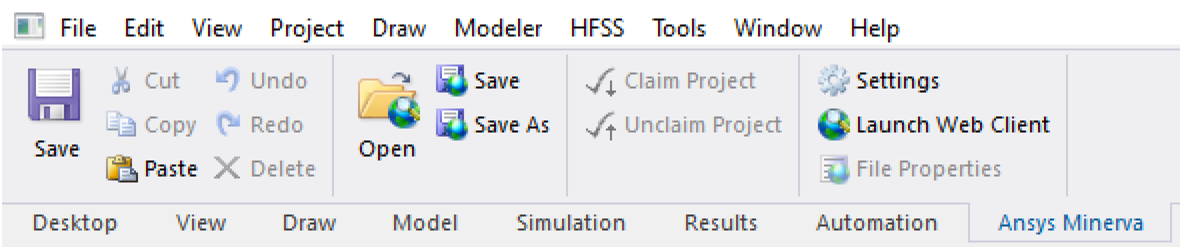
1. The definition **inner2** reverses terminals **g** and **h** compared to **inner1**, so that the resulting nodes X0.X1.g and X0.X1.h are cross connected between those two sweep steps.
2. The definition **inner3** has node **h** as an internal node, which conflicts with the use of **h** as a terminal of **inner1**.
3. The instance of **inner1** in the **.else** condition with **k** and **a** reversed in turn reverses the external connection of **g** and **h**, which causes the same cross connection as reversing them inside the subcircuit definition **inner2**.

All of these problems arise because the same instance name **X1** is used for the subcircuit for different steps. If you use a different instance name for each step, there is no restriction on terminal names.

When sweeping a choice of subcircuit, it may be convenient to use the same instance name every time, because it allows you to give the same name to terminals that are consistently connected or to any internal nodes. Then you have the ability to print and compare those nodes across the sweep. But when you choose to use the same instance name (**X1** in the example) you must be careful to use the terminal names (**g** and **h** in the example) consistently.

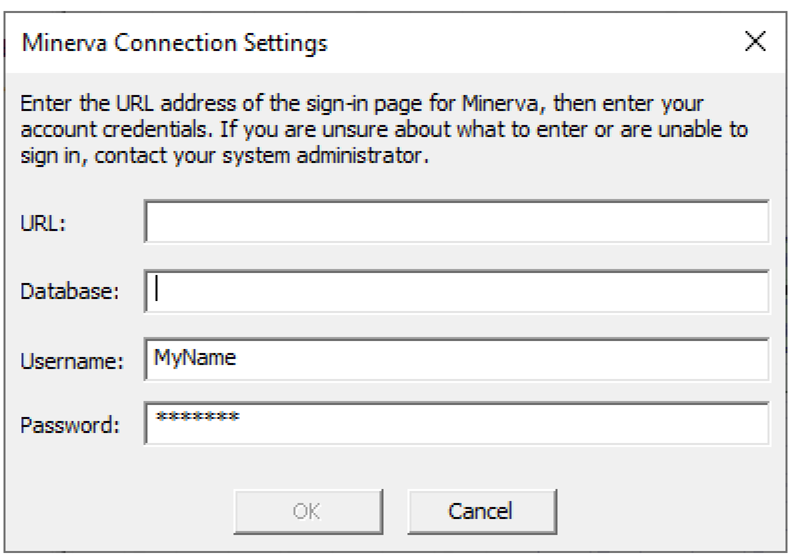
4 - Minerva Remote Storage Environment

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.



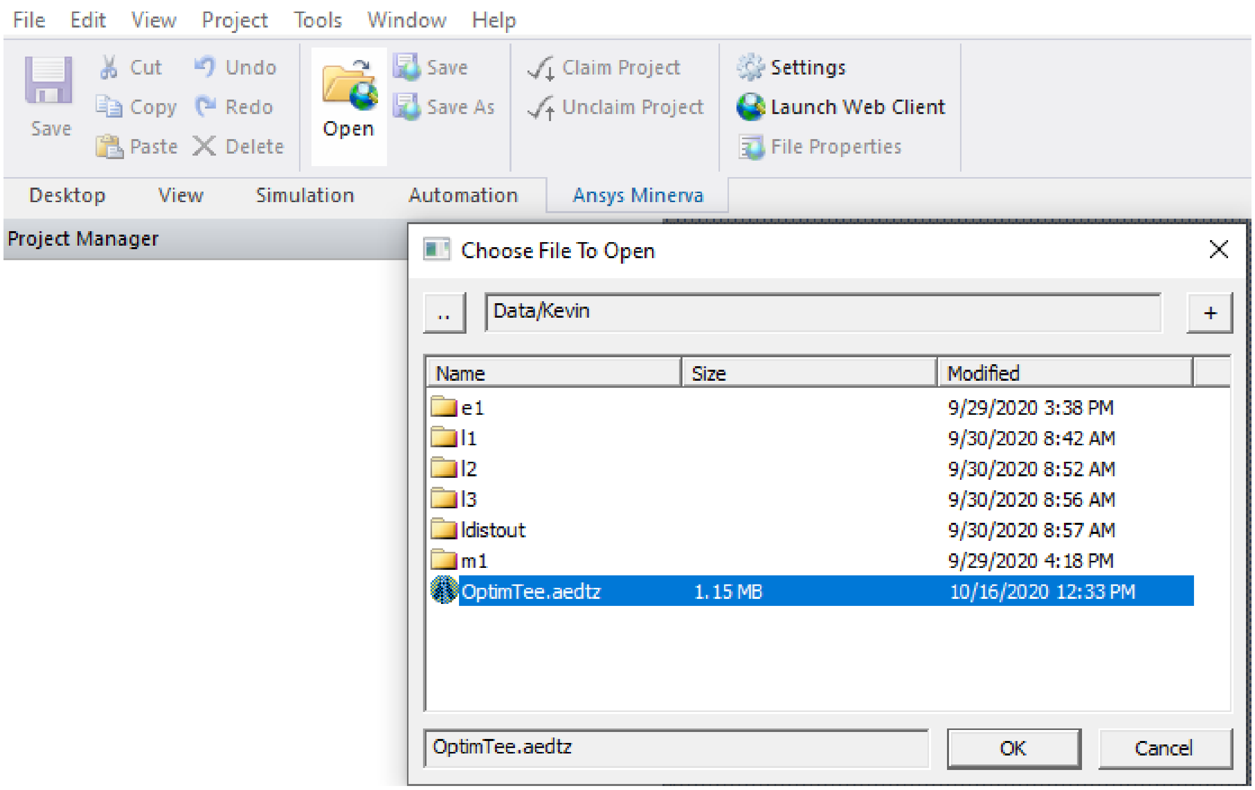
Minerva has its own HTML web interface with additional features as a knowledge management application that secures critical simulation data, and provides simulation process and decision support to simulation teams across geographies and functional silos.

Minerva Settings

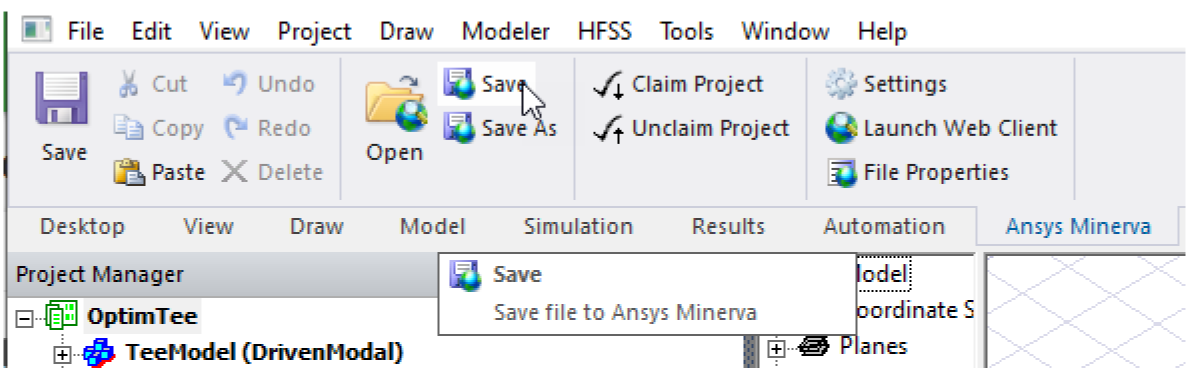


When you first login via the desktop, or to the Web Client, you must provide a Minerva URL, a database name, a user name, and a password. You will need to obtain these from your system administrator. Once you provide the connection settings, the other Minvera commands are enabled.

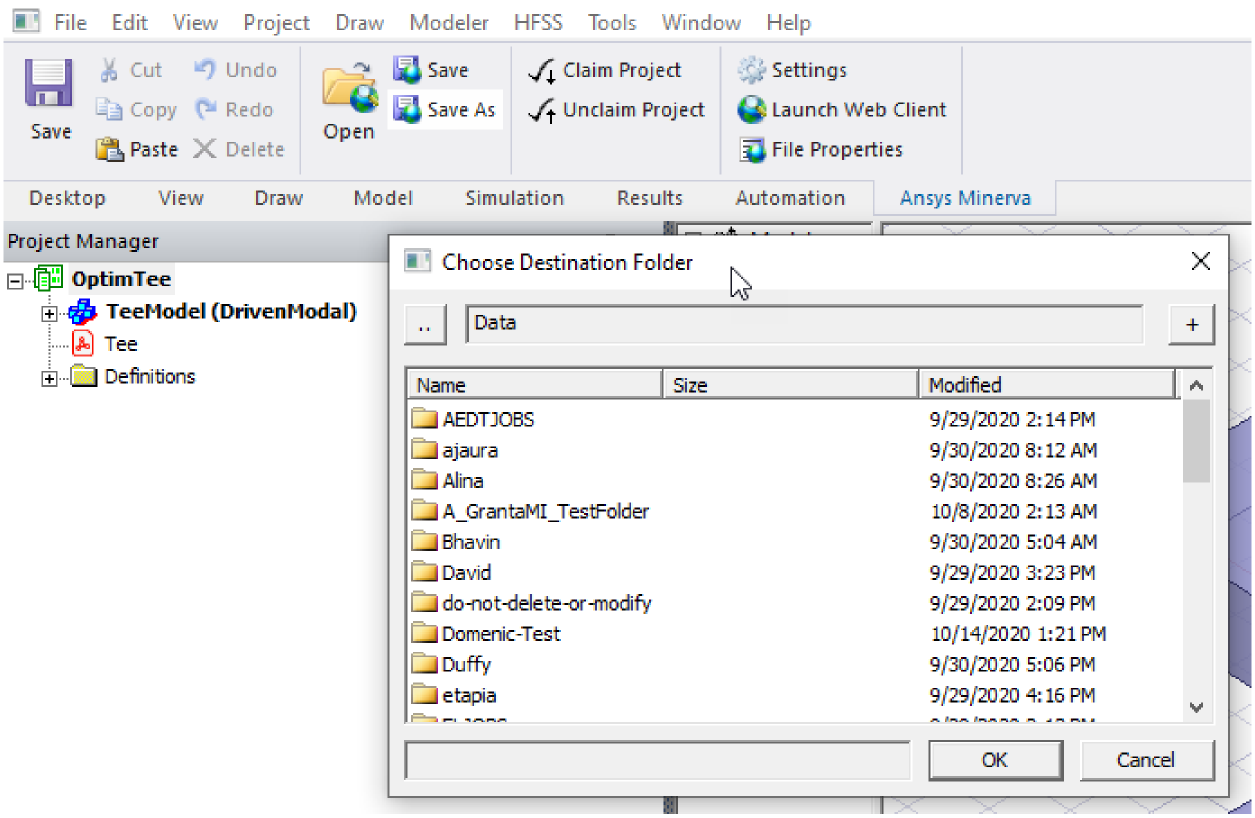
Open : opens the Choose File to Open browser that lets you select and download an archive file, extract it, and open it. The [..] button moves up one level in the file hierarchy. The [+] button creates a new folder at the current level.



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.

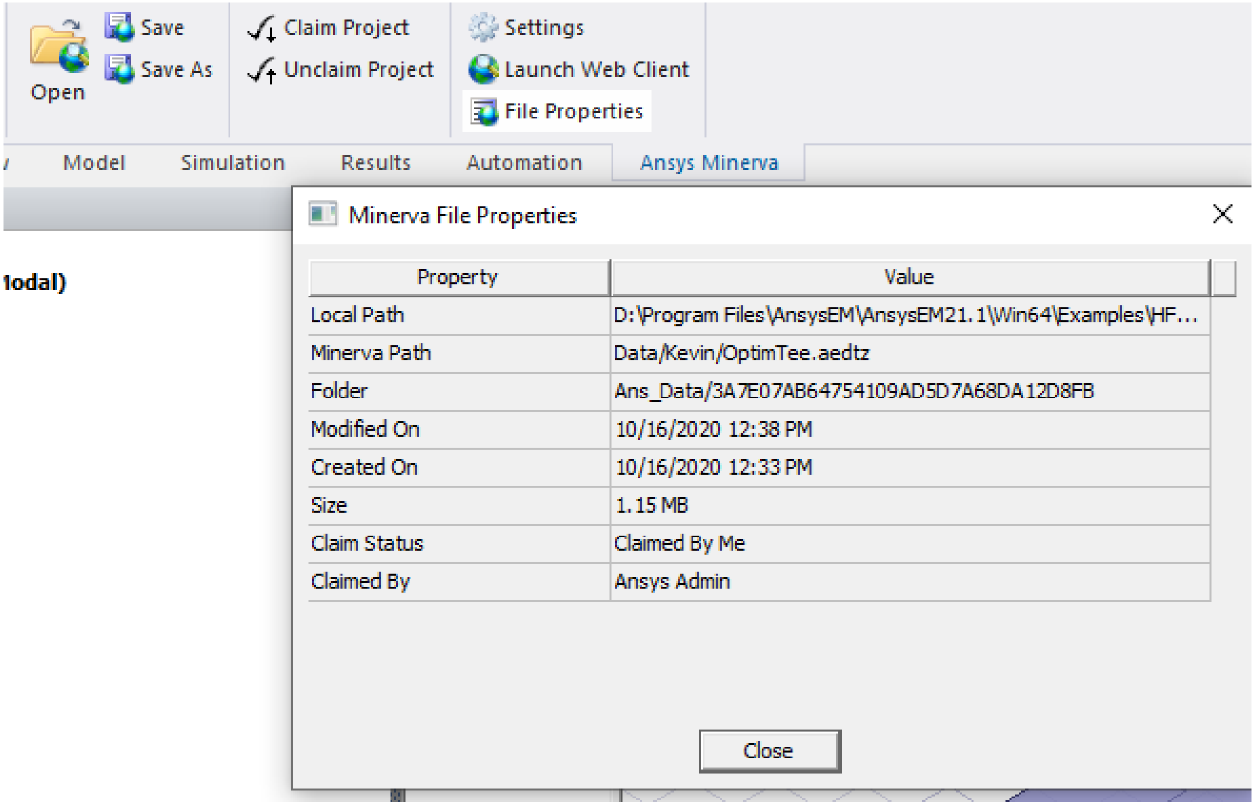
The recommended flow when working on a project is:

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.

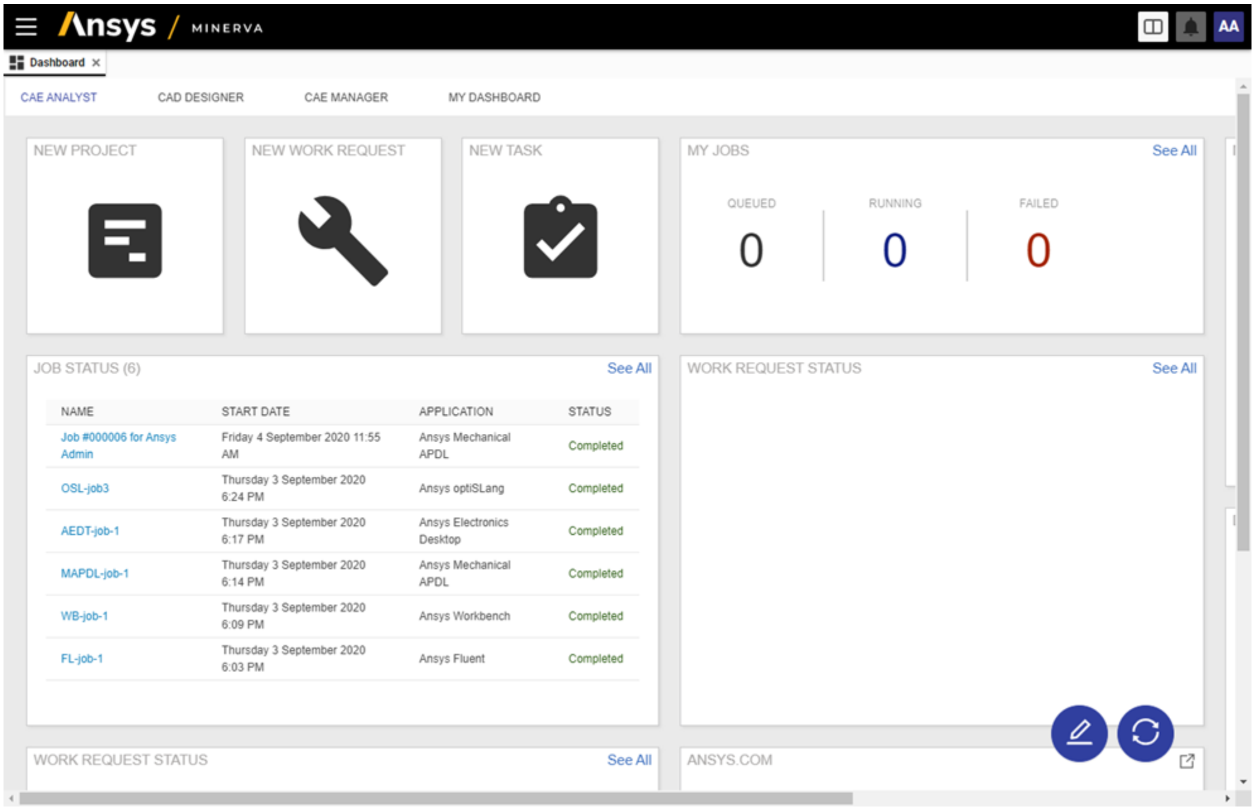
Unclaim Project: releases claimed file.

File properties: opens a dialog letting you view remote file properties. Notice the Claim Status and Claimed By properties.

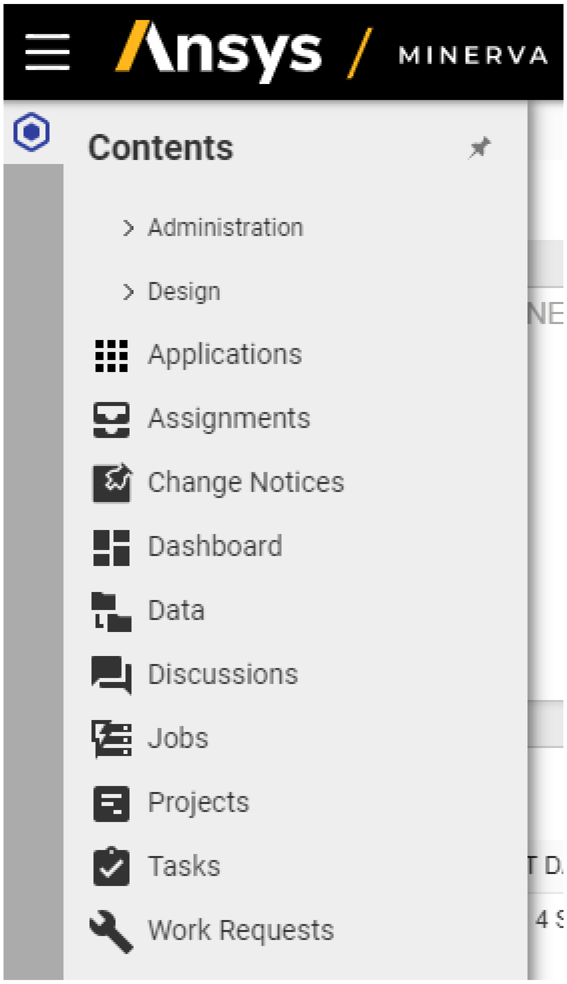


Launch Web Client:

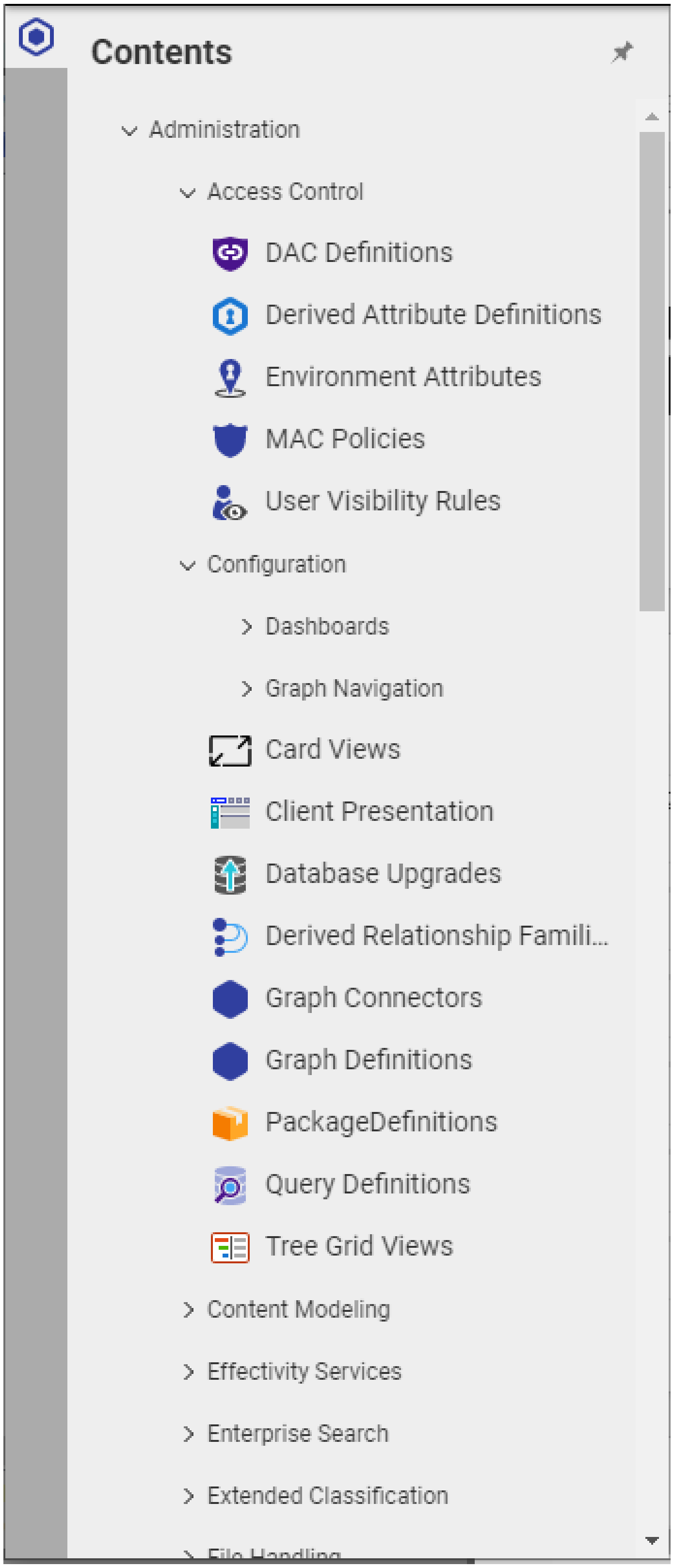
Ansys Minerva, powered by Aras, is a knowledge management application that secures critical simulation data, and provides simulation process and decision support to simulation teams across geographies and functional silos. Launching the Minerva Web Client gives access to these features. This opens a 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.


Contents
Administration
Access Control
DAC Definitions
Derived Attribute Definitions
Environment Attributes
MAC Policies
User Visibility Rules
Configuration
Dashboards
Graph Navigation
Card Views
Client Presentation
Database Upgrades
Derived Relationship Famili...
Graph Connectors
Graph Definitions
PackageDefinitions
Query Definitions
Tree Grid Views
Content Modeling
Effectivity Services
Enterprise Search
Extended Classification

5 - Power Module Characterization Tool

The Power Module Characterization tool (PMCT) aids in the building of various types of power module models so they can operate to a device manufacturer's specifications. The tool accepts inputs for various quantities available on the manufacturer and fits a numerical model to the data to provide accurate device simulation over a range of device excitations and thermal conditions without taking into account the knowledge of the topology, the control strategy, and the parameters of all the components.

The PMCT generates SPICE model to use in Nexxim and can approximate missing data. It provides accurate regulation, dynamic responses, and fast simulation time. This tool is presently only available for Microsoft Windows.

Power Module Characterization tool gives flexibility to model the following:

- Dynamic behavior
  - Inrush Current
  - Thermal model
- Non-linear static behavior
- Protection
  - Input Undervoltage
  - Input Over-Voltage
  - Output over-current
  - Output over-voltage
  - Thermal Protection
- Remote Control signals
- Start-up behavior (Soft-Start)
- Current Sharing
- Multiple Outputs

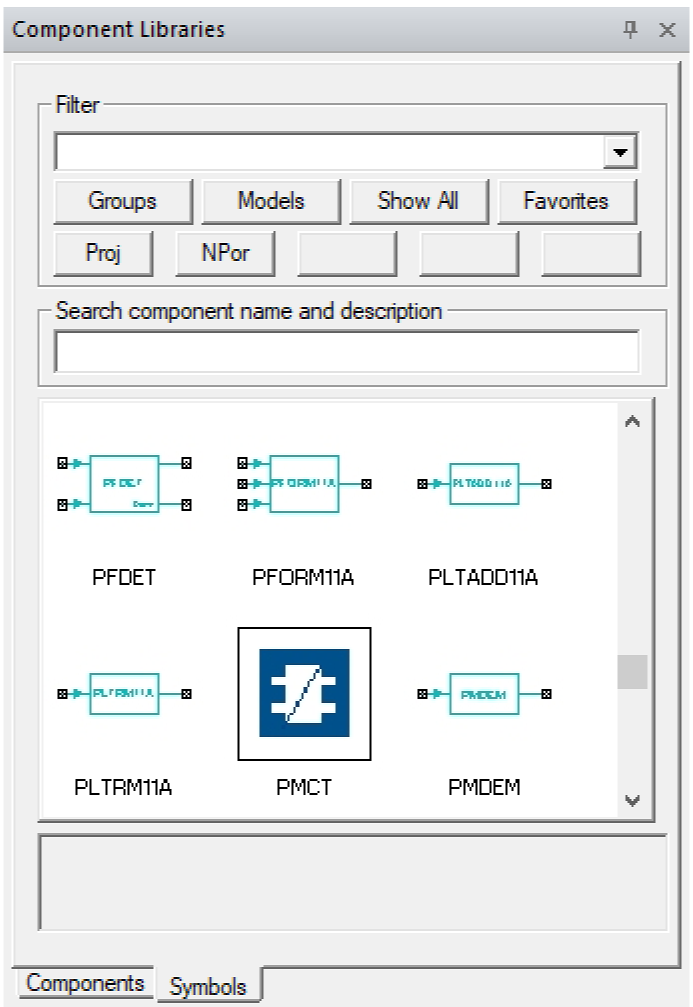
Getting Started with the Power Module Characterization Tool

The Power Modules Characterization tool (PMCT) includes an extensive library of DC-DC converter models from various manufacturers. You can view the characteristic data for these models in the tool. You can also to import these models into Circuits projects, to save models as discrete SPICE model files, and enable generation of sub-circuits for models.

Note:

If you have multiple versions of **Electronics Desktop** installed when you attempt to place a component on the PMCT, the Power Modules Characterization tool launches the newest version of **Electronics Desktop** and exit without placing any component. To place a component with an earlier version of AEDT, open a command line as an administrator and run the version of **Electronics Desktop** with which you want to launch PMCT with the option -regserver. For example:
%ProgramFiles%\AnsysEM\v242\Win64\ansysedt.exe -regserver

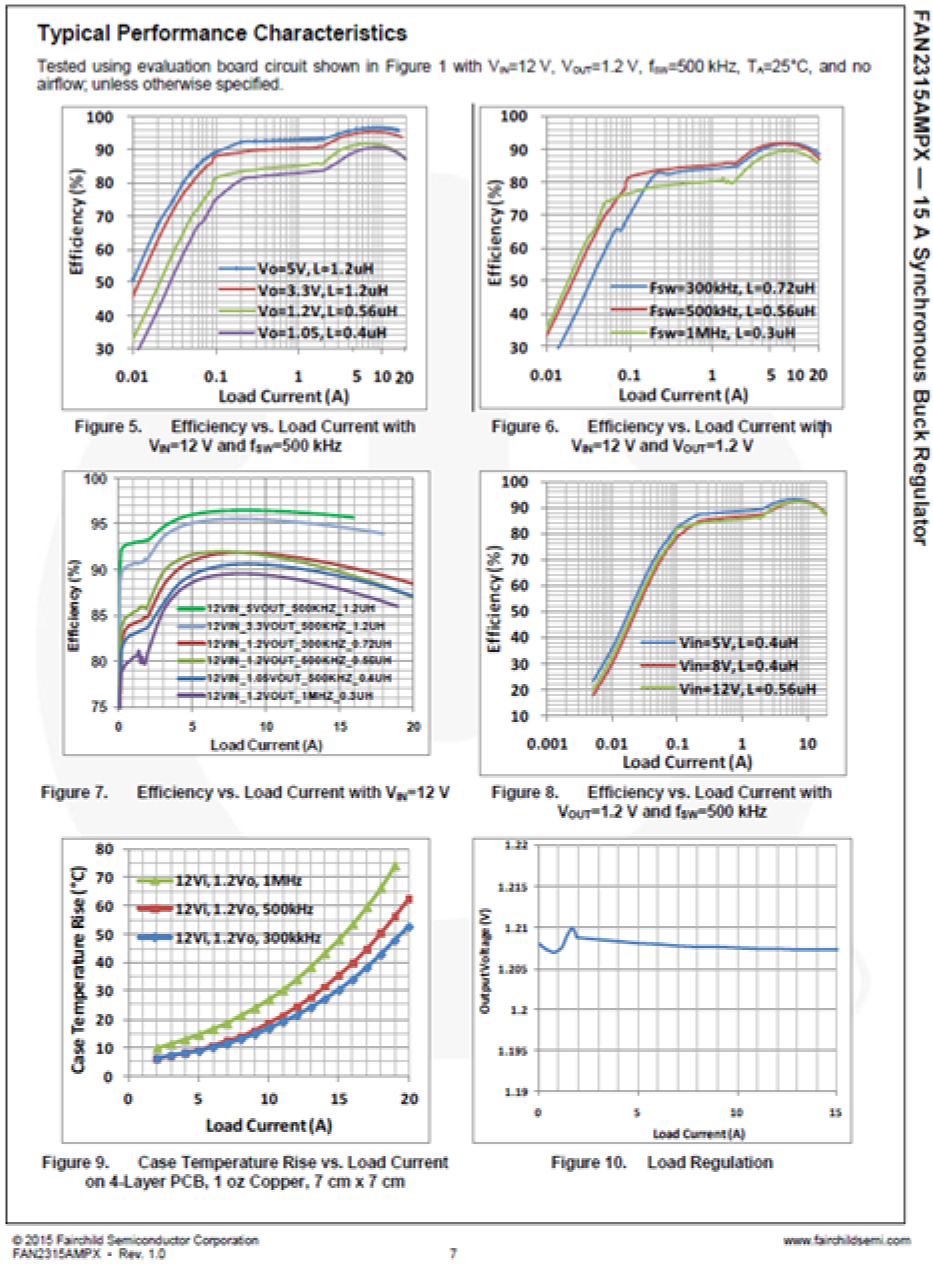
1. To launch the tool, click the **PMCT** tile in the **Components Libraries**.



2. In the Power Modules Characterization tool, either:
   - Select and modify a model on the system libraries or
   - Create a custom behavioral DC/DC converter model.
3. Adjust the model in any of the following ways as necessary:
   a. Define a Static Converter model.
   b. Define the Converter Model Dynamic Behavior.
   c. Define event driven behavior.
   d. Enable current sharing.
   e. Enable thermal modeling and thermal protection.
4. Generate SPICE code and import the model into Circuit. As the PMCT closes, the model is placed in the **Project Manager** window under **Definitions** > **Component**.

Power Module Characterization Tool Example

This example describes modeling a new DC-DC converter from OnSemi, the FAN2315AMPX-15 Asynchronous Buck Regulator. Every DC-DC converter has its own data sheet which typically covers **Efficiency vs Load Current** and **Load Regulation** characteristics, like this:



Extract **Data on Efficiency vs Load Current** and **Load Regulation** characteristics on the data sheet.

Figure 5-1 Efficiency vs Load Current Extracted Data in Excel

Load Current	Efficiency
0	89
0.1	90
0.2	92
0.25	92.5
0.5	92.8
1	93
1.5	93.2
2	93.25
3	94.6
4	95.6
5	96.1
6	96.4
8	96.6
10	96.5
12	96.4
15	96
18	95.3

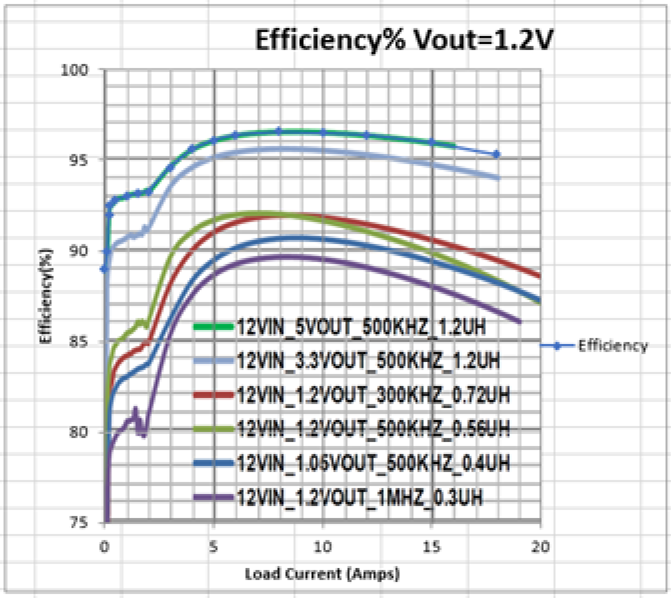
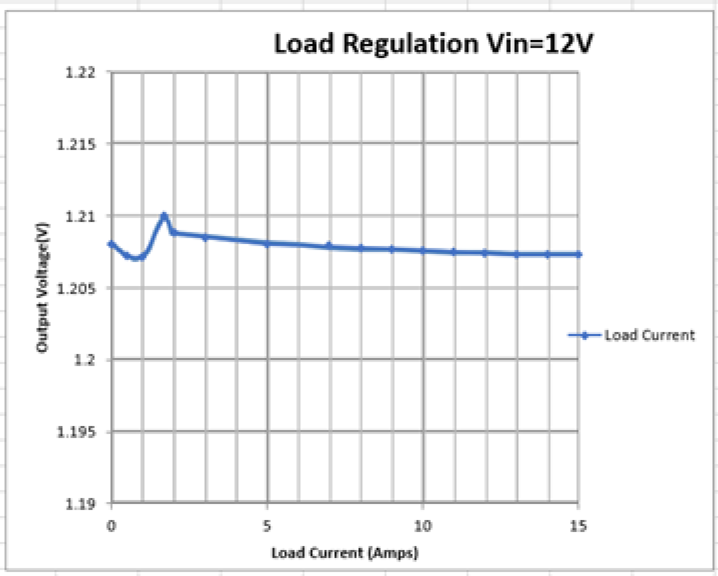


Figure 5-2 Load Regulation Extracted Data in Excel

Load Current	OutputVoltage
0	1.208
0.5	1.2072
1	1.2071
1.7	1.21
2	1.2088
3	1.2085
5	1.208
7	1.2079
8	1.2078
9	1.2077
10	1.2076
11	1.2075
12	1.2074
13	1.2073
14	1.2073
15	1.2073



Next, launch the Power Module Characteristics tool and create a new behavioral DC/DC converter model.

Use the extracted data and fill in the Efficiency and Load Regulation tables in the PMCT Tool. Set up the static model, then specify the dynamic behavior of the converter by entering in appropriate values:

Figure 5-3 Enable the Static Model

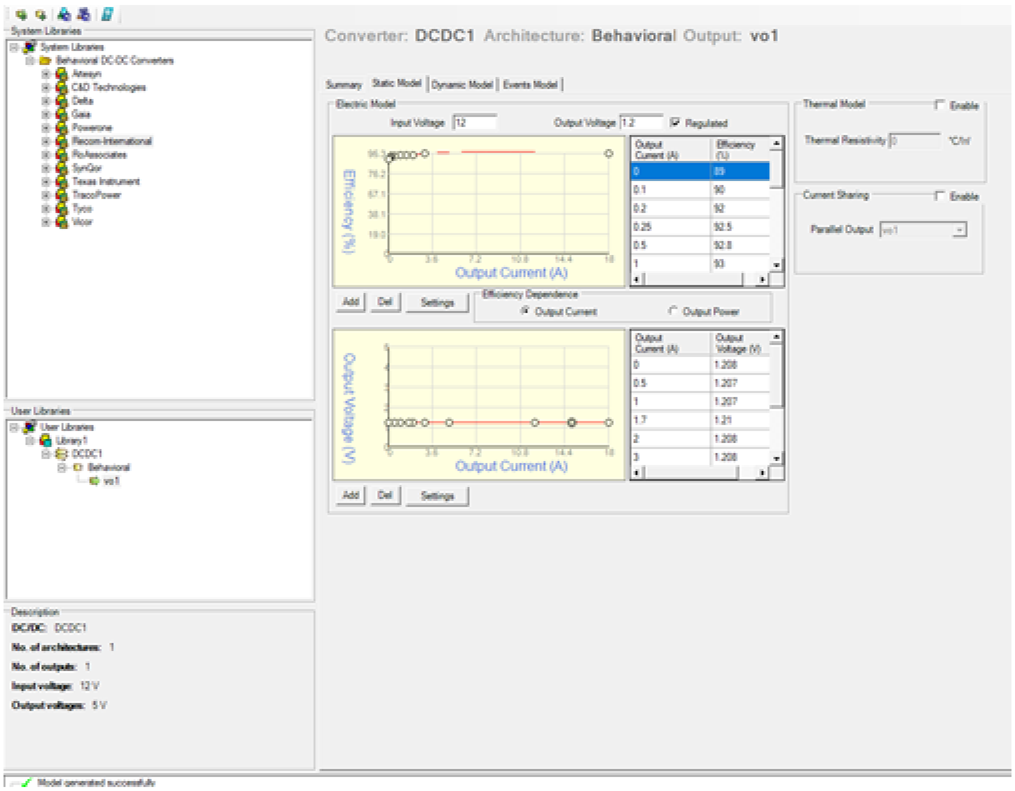
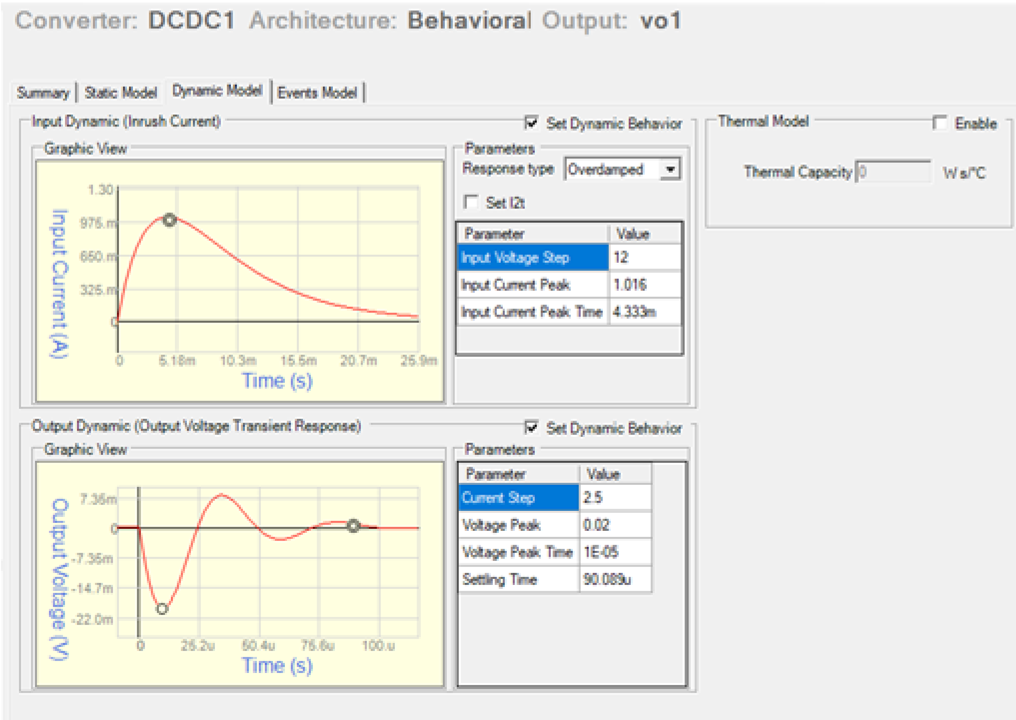


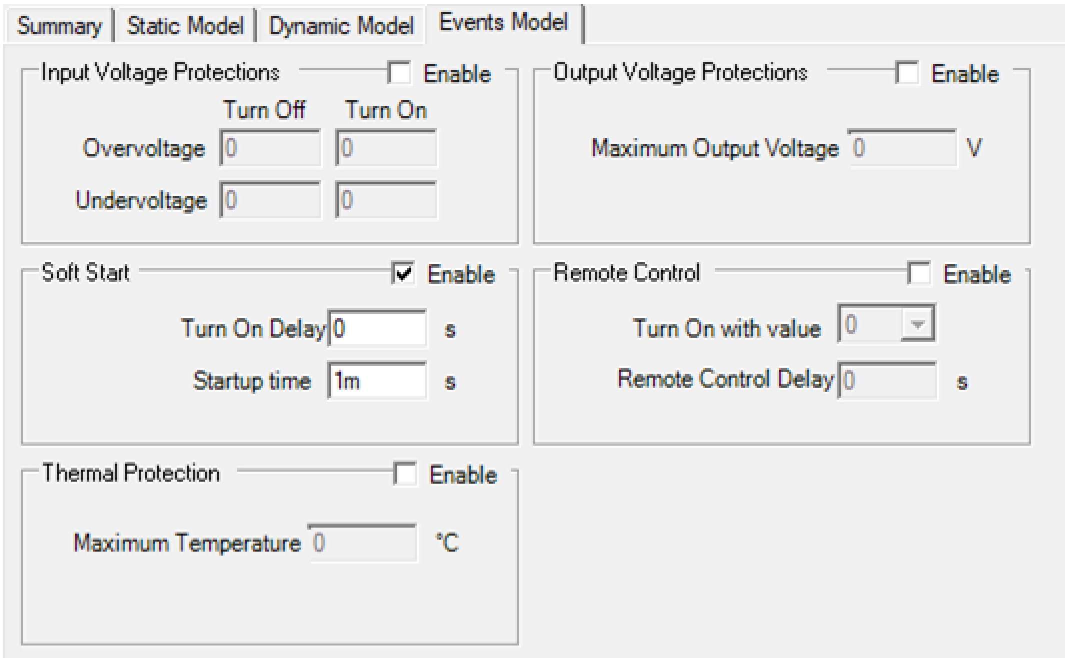
Figure 5-4 Enable the Dynamic Model



> **Note:** For this example thermal model is not enabled, but it could be enabled by checking the **Thermal model Enable** check box on the *Static Model* and **Dynamic Model** tabs.

From the Events Model tab, activate all the protections including **Thermal protection** according to the converter specification.

Figure 5-5 Enable the Event Model

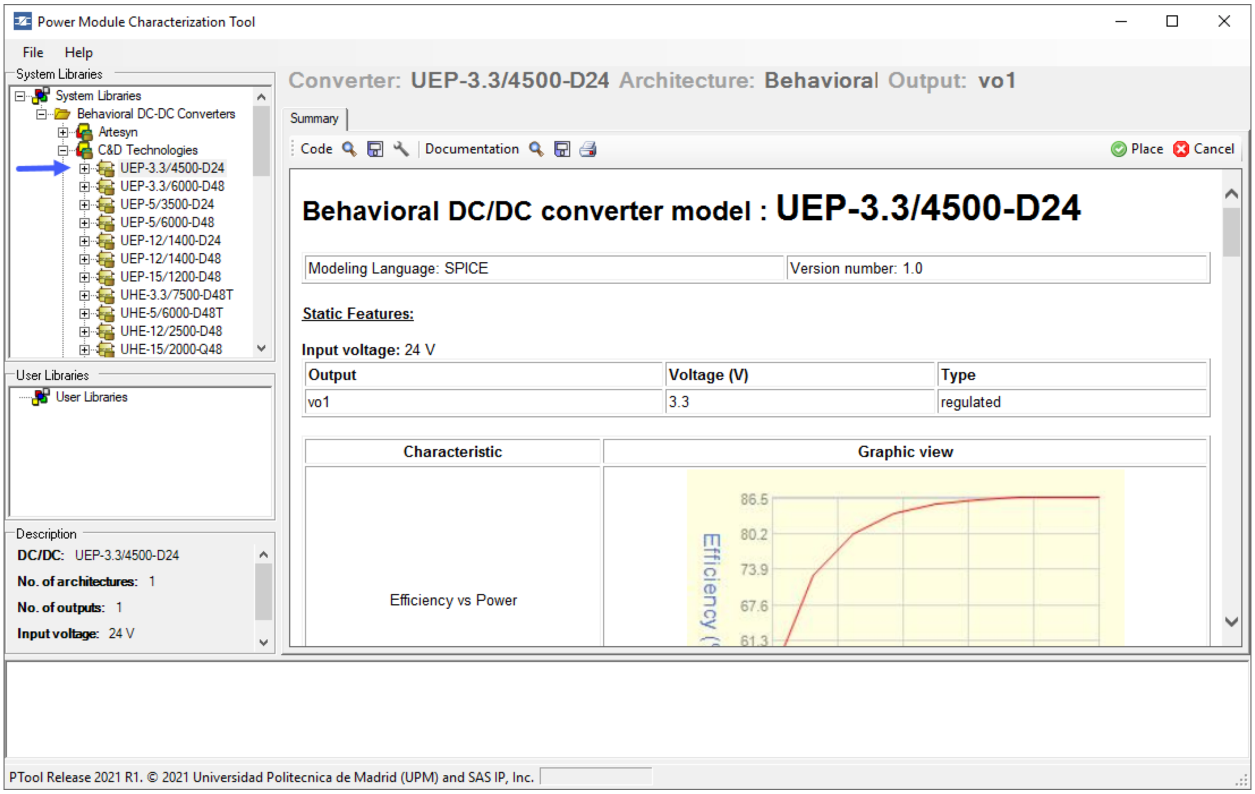


After choosing appropriate properties for the converter, on **Summary** tab, click the **Code**  icon to save the model in SPICE format. Click **Place** to place the model in Circuit.

Using System Library Power Module Models

The Power Modules Characterization tool includes an extensive library of DC-DC converter models from various manufacturers. You can view the characteristic data for these models in the tool. You can also to import these models into Circuits projects, to save models as discrete SPICE model files, and enable generation of sub-circuits for models.

1. To view the characteristics of power module models supplied in the System Library, select the appropriate models in the **System Libraries** model browser. The model editing group box displays the **Summary** tab for the model.



2. To view the model code (read only) in the **Summary** tab, click the **Code** icon.

3. To save the model in SPICE format, click the **Code** icon. In the resulting window, enter the model file name and choose a save location.

4. To enable generation of sub-architectures, click the **Code** icon, check **Enable**, and then select sub-architectures to include in the model.

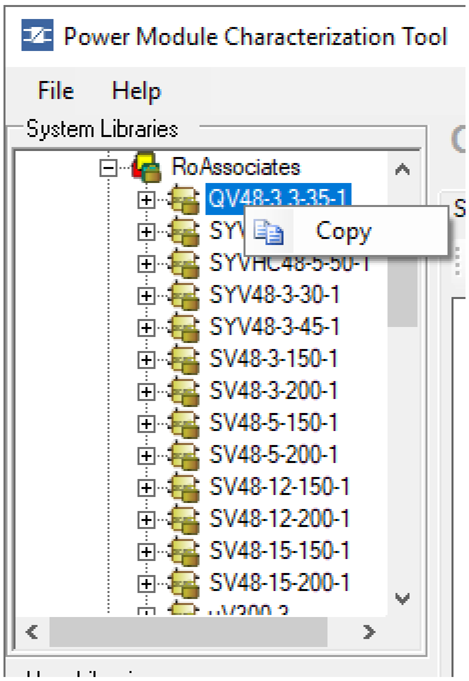
5. Click the **Documentation** icon to view the documentation in the tab. To print the documentation, click the **Documentation** icon, and export the documentation as an HTML file, click the **Documentation** icon.

6. To import the model into the active Circuit project, click the **Place** Place icon. The model is placed in the project as the PMCT closes.

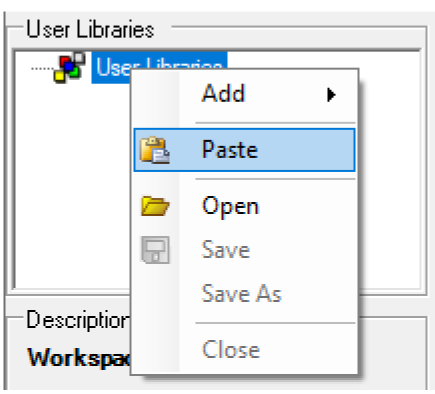
Modifying System Library Power Module Models

The models provided in the DC-DC Converters System Library cannot be modified directly. If you want to modify any of these models you must first place a copy of the appropriate model in the **User Libraries**. To modify a System Library DC-DC Converter:

1. Right-click the model you want to copy and select **Copy**.



2. Right-click the target library and select **Paste**.



3. Once the model is in the **User Libraries**, modify the model as shown in Creating a Power

Module Behavioral Model.

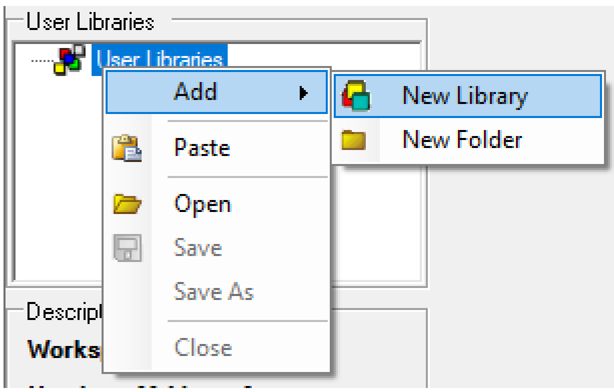
4. Import the model into Circuit when done.

Creating a Power Module Behavioral Model

Use the Power Module Characterization tool to create new behavioral models and model libraries in which to store them.

Creating a New Power Module Model Library

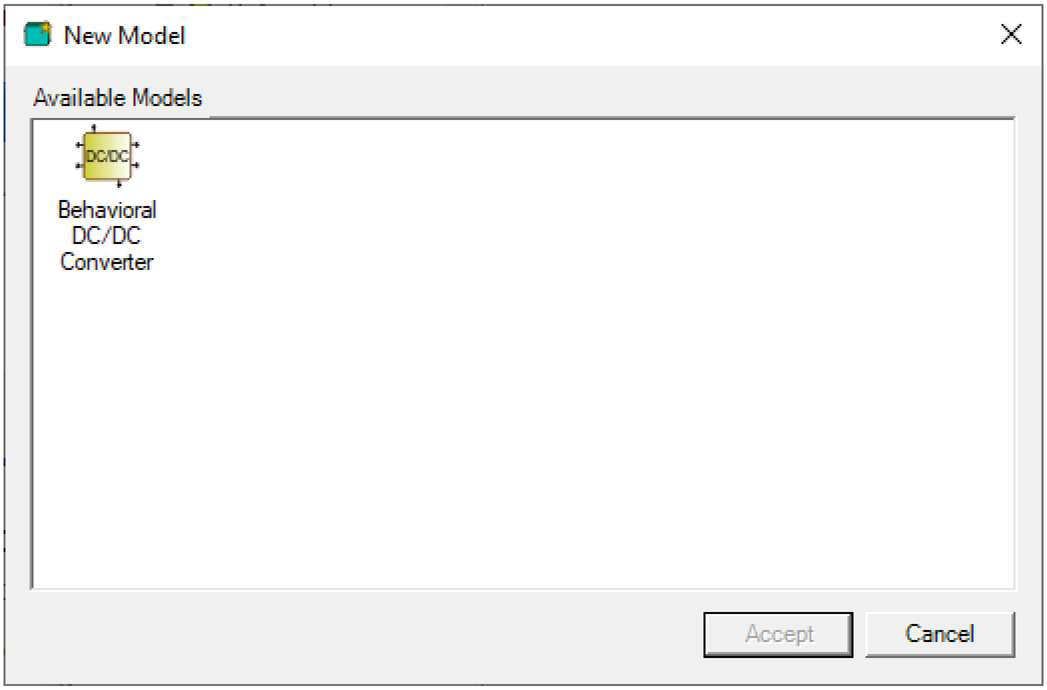
1. In the **User Libraries** group box, right-click **User Libraries** and select **Add** > **New Library** to create a new model library. A library with a default name appears under the **User Libraries** model browser.



2. Click a library name to select it, then click it again to rename it.

Adding a New Behavioral DC/DC Converter Model

1. From the model browser, right-click a library and select **Add** > **New Model** to open the **New Model** window.



2. Select the **Behavioral DC/DC Converter** model and click **Accept**. A model with a default name appears under the selected library.
3. Click the model name to select it, then click it again to edit the name.

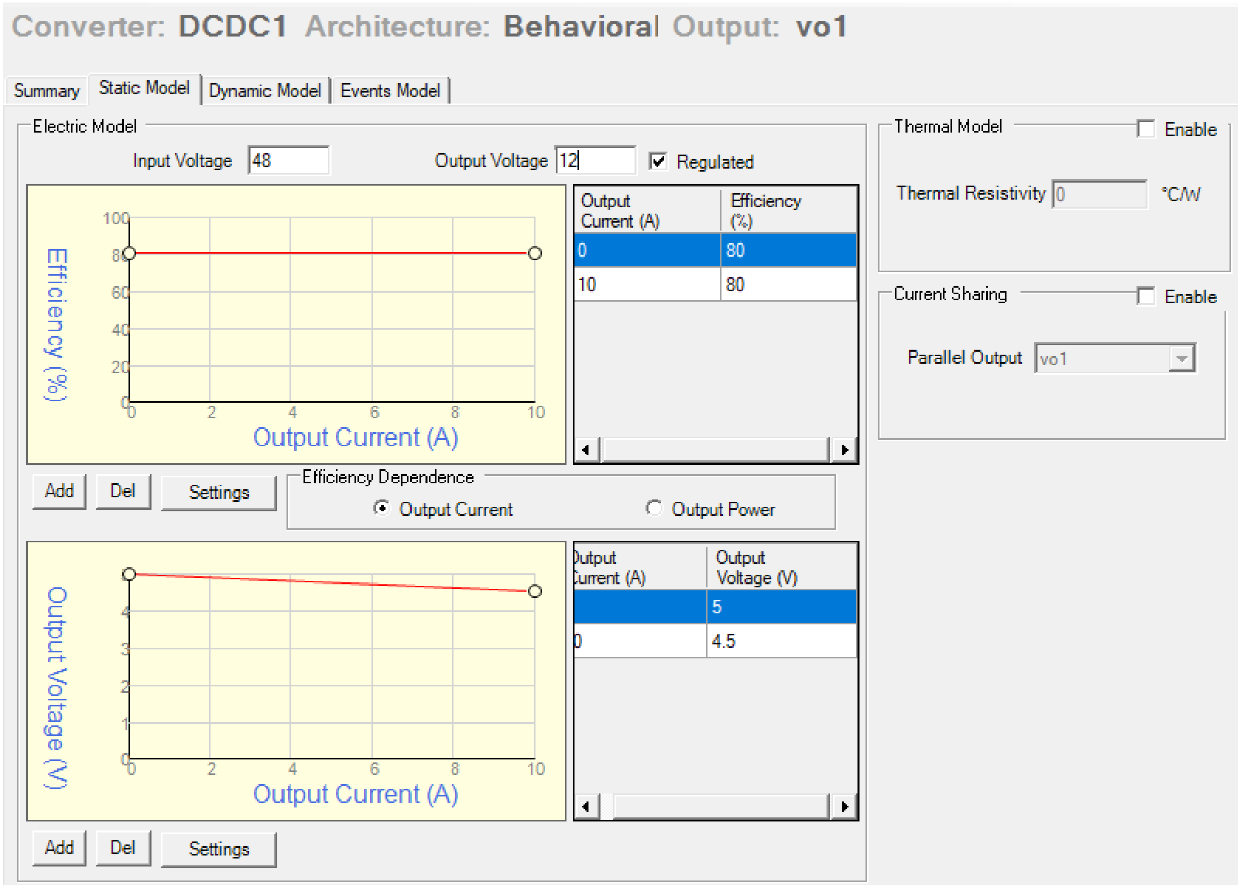
Every newly created DC/DC model has default parameters that correspond to a 12V input and 5V output voltage with a constant efficiency.

Defining a Static Converter Model

The following example describes how to define a static converter model having a 48V input voltage and a 12V output voltage. The efficiency is assumed to be a constant value.

1. In the **User Libraries** browser, select the DC-DC converter model that you want to edit to open an editing group box with four tabs—**Summary**, **Static Model**, **Dynamic Model**, and **Events Model**.
2. Select the **Static Model** tab.

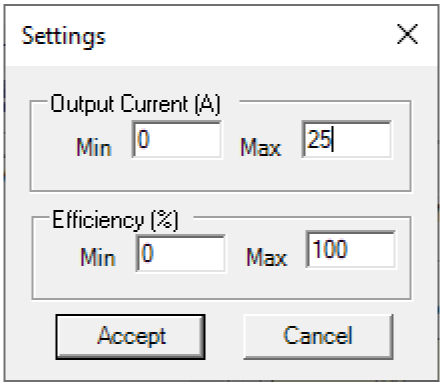
3. From the **Static Model** tab, define the basic characteristic of the converter.
   a. Change the value of the input voltage to **48V**.
   b. Change the output voltage to **12V**.



The **Static Model** editing group box shows two graphs: **Efficiency vs. Output Current** and **Output Voltage vs. Output Current**.

4. The first step to define a characteristic graph is to set its maximum and minimum values. Click **Settings** on the Efficiency vs. Output Current graph to open the **Settings** window.
   a. In the **Settings** window, set the **Output Current (A)Max** value to **25** (assuming that the converter supports a maximum output current of 25A).
   b. Enter **100** as the **Efficiency(%) Max** value. This corresponds to a maximum efficiency of 100%.

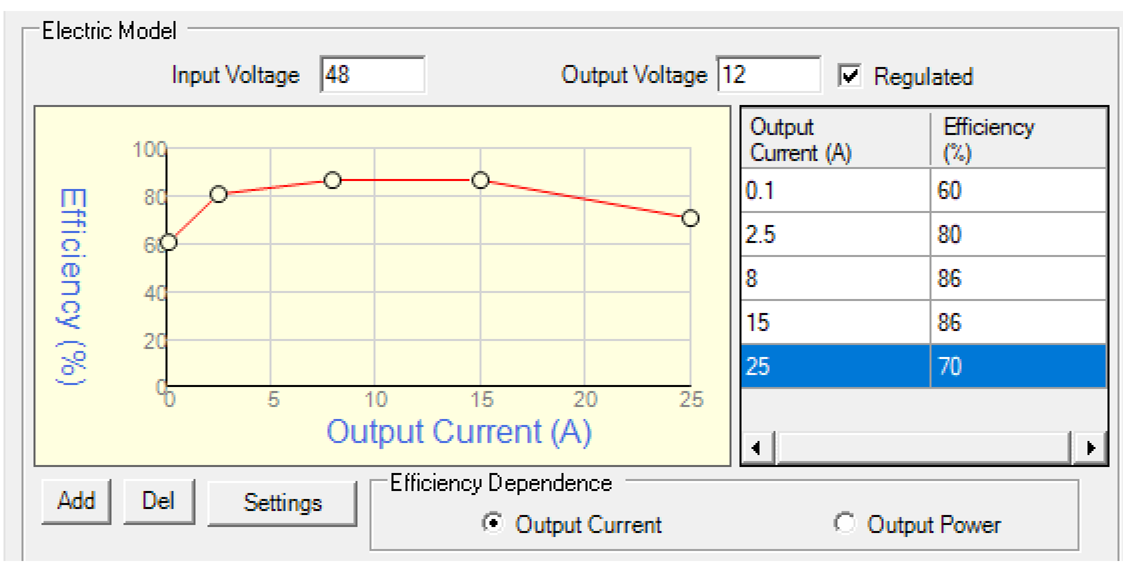
c. Click **Accept** to close the window.



5. Click **Add** on the efficiency graph to insert as many points as you need to define the behavior of your converter. You can also right-click in the graph and select **Add**.

   In this example, three more points is added. Set the values of the points by dragging the added points directly on the graphic, or specify the values directly in the table view. (Note that as you add points, the table entries sort in ascending order of output current magnitude.)

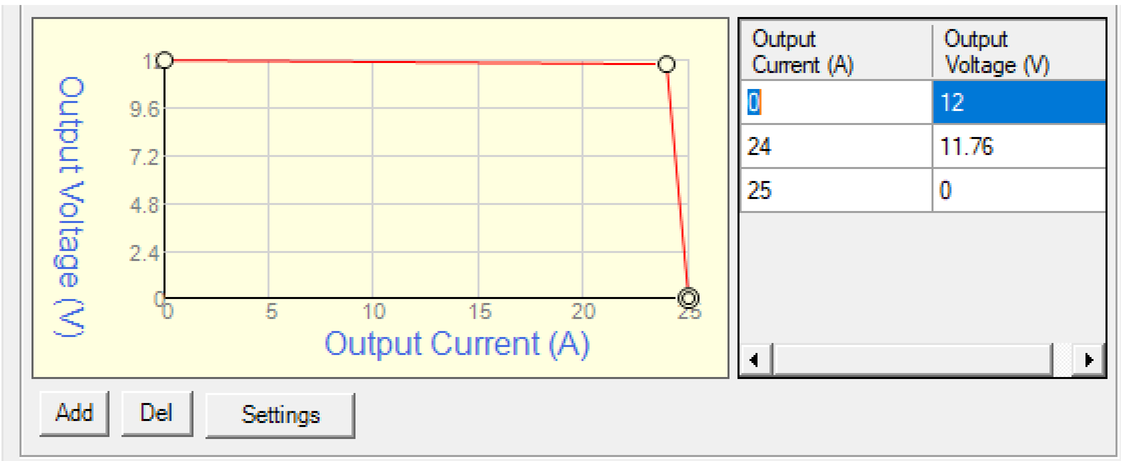
6. Enter the following values for the efficiency curve: **(0.1, 60)**, **(2.5,80)**, **(8,86)**, **(15,86)**, and **(25,70)**.



7. Similarly, define the behavior of the **Output Voltage vs. Output Current** graph.
   a. Click **Settings** on the **Output Voltage vs. Output Current** graph.
   b. In the **Settings** window, enter a maximum value of **25** for the **Output Current (A)**.

c. Enter a maximum value of **12** for the **Output Voltage (V)** (the output voltage value corresponds to the nominal output voltage).

8. In this example a load regulation of 2% at 24A is being defined. This means that the output voltage of the converter drops 0.24V when the converter is supplying an output current of 24A. The values to enter are **(0,12)**, **(24,11.76)**, and **(25,0)**. Note that the converter also has output protection that makes the voltage drop when output current exceeds 24A.



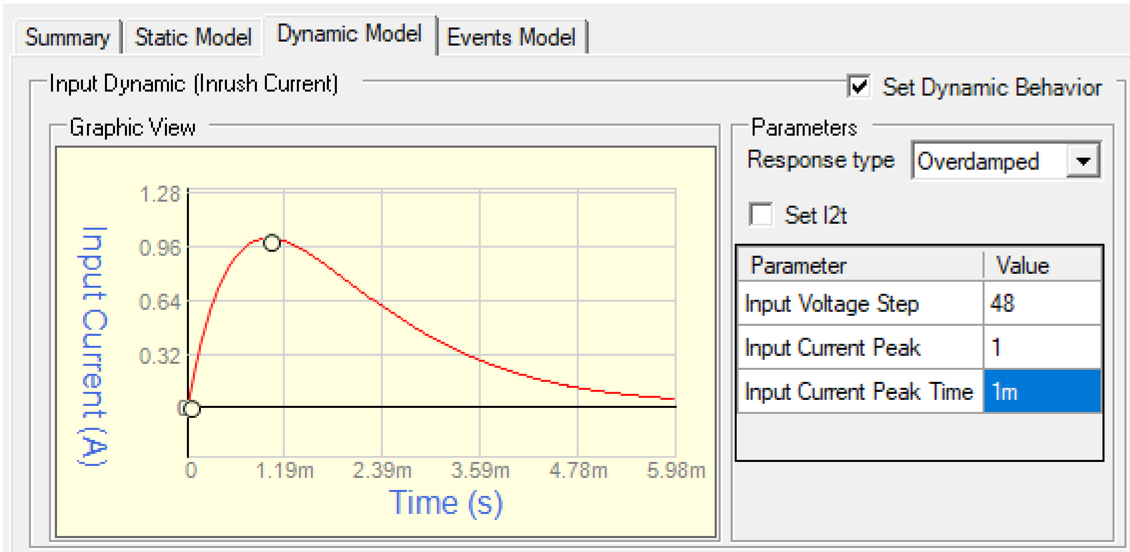
At this point, the minimum characteristic values needed to create a simulation model are defined, SPICE code for the model can be generated, and the model exported to Circuits.

Defining the Converter Model Dynamic Behavior

The behavior of the converter model is dependent on the inrush current.

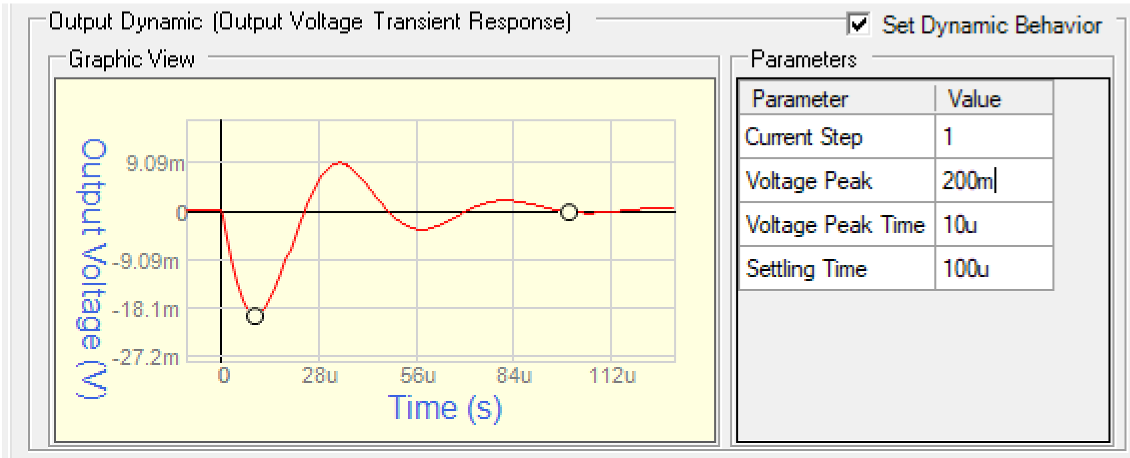
1. Select the **Dynamic Model** tab in the editing group box.
2. In the **Input Dynamic (Inrush Current)** group box, select the **Set Dynamic Behavior** check box.
   a. Set the **Input Voltage Step** parameter to **48V**. This value corresponds to a step in the input voltage when the converter is powered.
   b. Set the **Input Current Peak** parameter to **1A**.
   c. Set the **Input Current Peak Time** to **1ms**. You can also drag the points to graphically define the behavior of the input current.
   d. Right-click the input dynamic graph and select **Automatic Scale** to display the input

current graph in its best representation.



3. In the **Output Dynamic (Output Voltage Transient Response)** group box, select the **Set Dynamic Behavior** check box to include the behavior of the converter under load steps.
   a. Set the **Current Step** value to **1A**, this value corresponds to the load step under which the test was performed.
   b. Set the **Voltage Peak** to **200mV**. This value is the voltage drop peak when the load step occurs.
   c. Set the **Voltage Peak Time** to **10us**. This is the time when the voltage peak occurs.
   d. Set the **Settling Time** value to **100us**. This value represents the time that the converter takes to recover on the load step.
   e. Right-click the graph and select **Automatic Scale** to open the graph in its best

representation.



4. From the **Summary** tab, generate the model and import it to Circuit as needed.

Inrush Current in a Dynamic Model

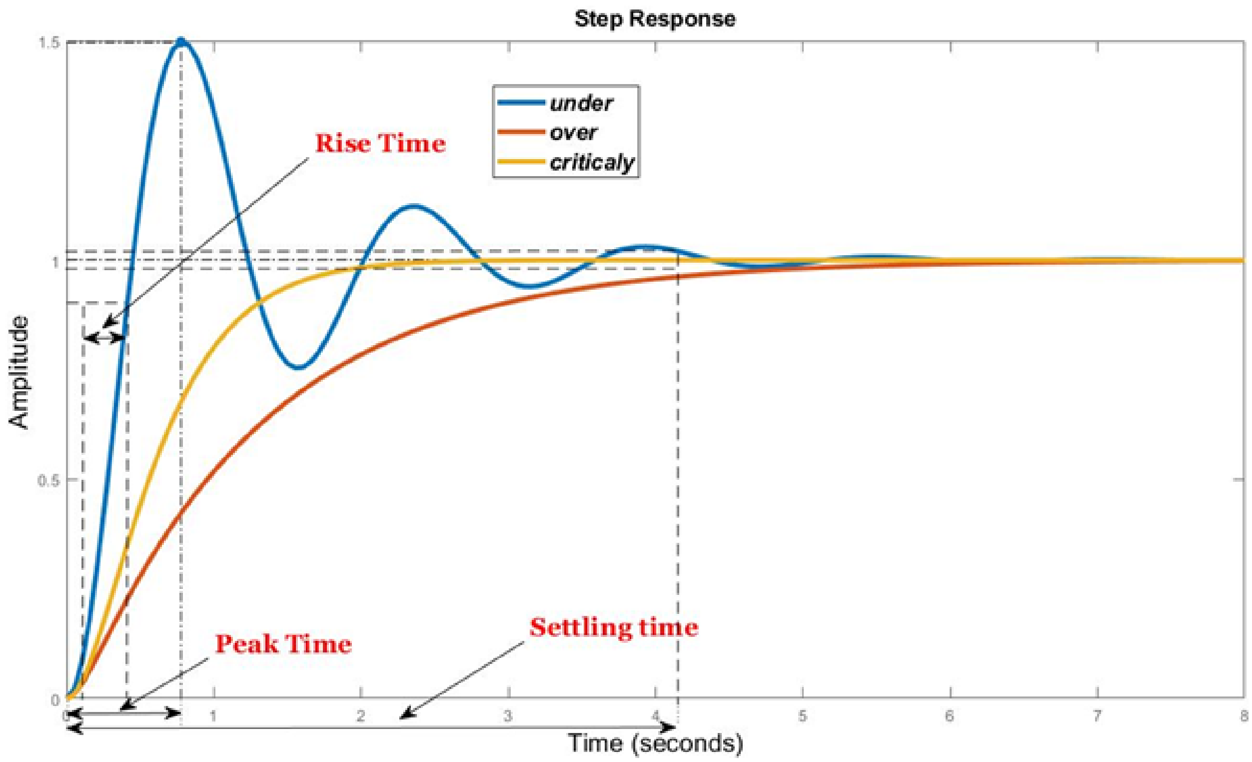
Inrush current is the spike of current drawn by a power supply when it is activated. A typical power system includes some capacitance connected across the input line. The DC-DC converter also includes capacitance connected across its input and output. The load may include additional capacitance. Each of these capacitors requires current to charge them to their steady state voltage. This current is the inrush current.

Methods of Reducing Inrush Current

Inrush current can be reduced by increasing the voltage rise time on the load capacitance and slowing down the rate at which the capacitors charge. Three different solutions to reduce inrush current are shown following: voltage regulators, discrete components, and integrated load switches. All three of these solutions center around increasing the voltage rise time which, as shown in Equation 1, leads to reduced inrush current.

Voltage regulators, DC/DC converters, and LDOs may have an integrated soft-start functionality. With this feature, the rise time can be increased, thereby reducing the inrush current. With a properly selected DC/DC converter or LDO, the inrush can be effectively managed to ensure system stability.

In a Dynamic Model



In this graph:

- Rise Time is the time it takes for the response to rise from 10% to 90% of the steady-state response.
- Settling Time is the time it takes for the error $|y(t) - y_{final}|$ between the response y(t) and the steady-state response yfinal to fall to within 2% of final.
- Peak is the absolute value peak of y(t)
- Peak Time is the time at which the peak value occurs.

This diagram demonstrates the typical Data Sheet values:

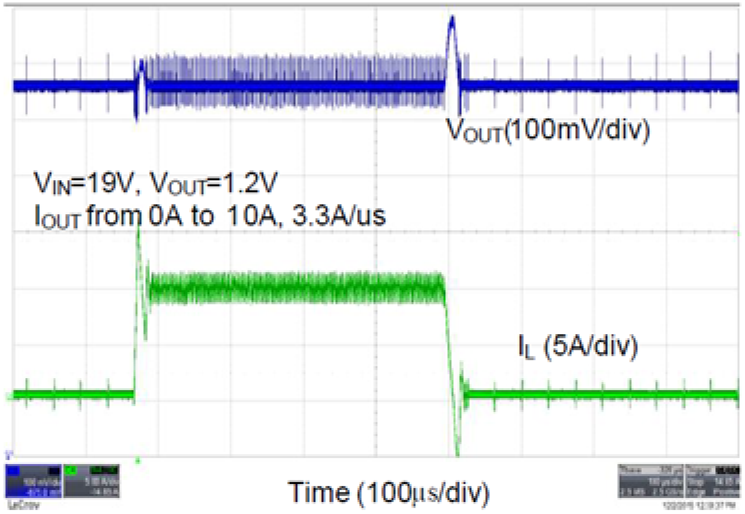


Figure 24. Load Transient from 0% to 50% Load Current

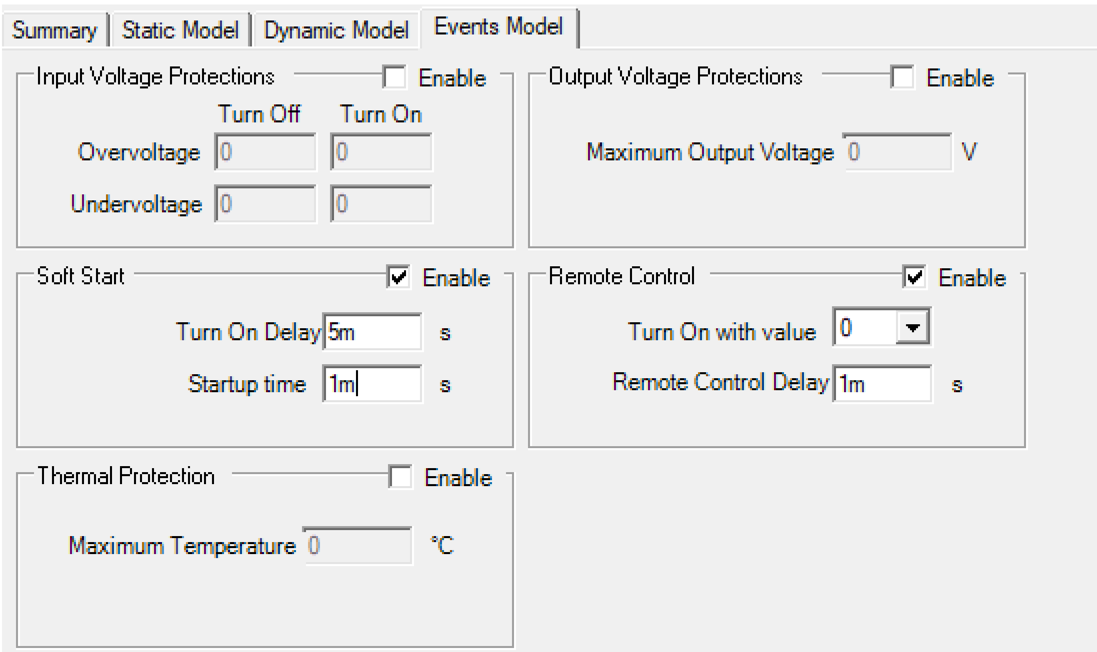
Defining Event Driven Behavior

The events included in this model are:

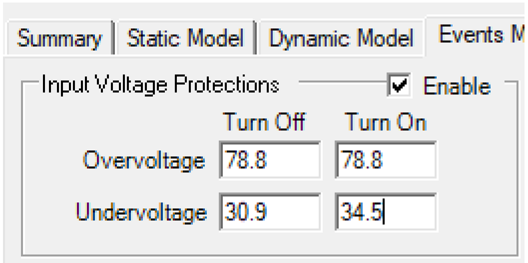
- **Input voltage protection** - This protection disables the operation of the converter when the input voltage is out of the operational limits of the converter to avoid malfunction.
  - **Over-voltage protection** (with and without hysteresis) - This voltage setting assures that all the components of the converter are working with designed voltage ranges.
  - **Under-voltage protection** (with and without hysteresis) - Underneath this voltage, the converter is deactivated to avoid high currents that can destroy it.
- **Over-current protection** - This protection tries to avoid high internal current due to high external loads. This protection is included inside the event-driven behavior in case of turning-off the converter. The protection based on lowering the output voltage as a function of the output voltage must be included in the static non linear model since it does not create an event
- **Output voltage protection** - This is usually an upper output voltage value that disables the converter to protect it against overvoltage caused by an external source.
- **Remote control** - This signal is usually provided to DC-DC converters to enable or disable them. Most DC/DC converters throughout the industry offer some type of remote control function to enable the converter to be activated or off by means of an external control signal.
- **Thermal protection** - In the case of high-temperature operation of the converter, this protection turns it off if the high-temperature limit is exceeded.

To define event driven behavior:

1. Click the **Events Model** tab in the editing group box.
2. In the **Soft Start** and **Remote Control** group boxes, click **Enable** to enable these features.
3. In the **Soft Start** group box:
   a. Set the value of the **Turn On Delay** to **5ms**.**Turn On Delay** represents the time after power-up before the converter begins supplying voltage.
   b. Set the **Startup time** to **1ms**. The **Startup time** is the time that takes the converter to raise the output voltage from zero to its nominal value.



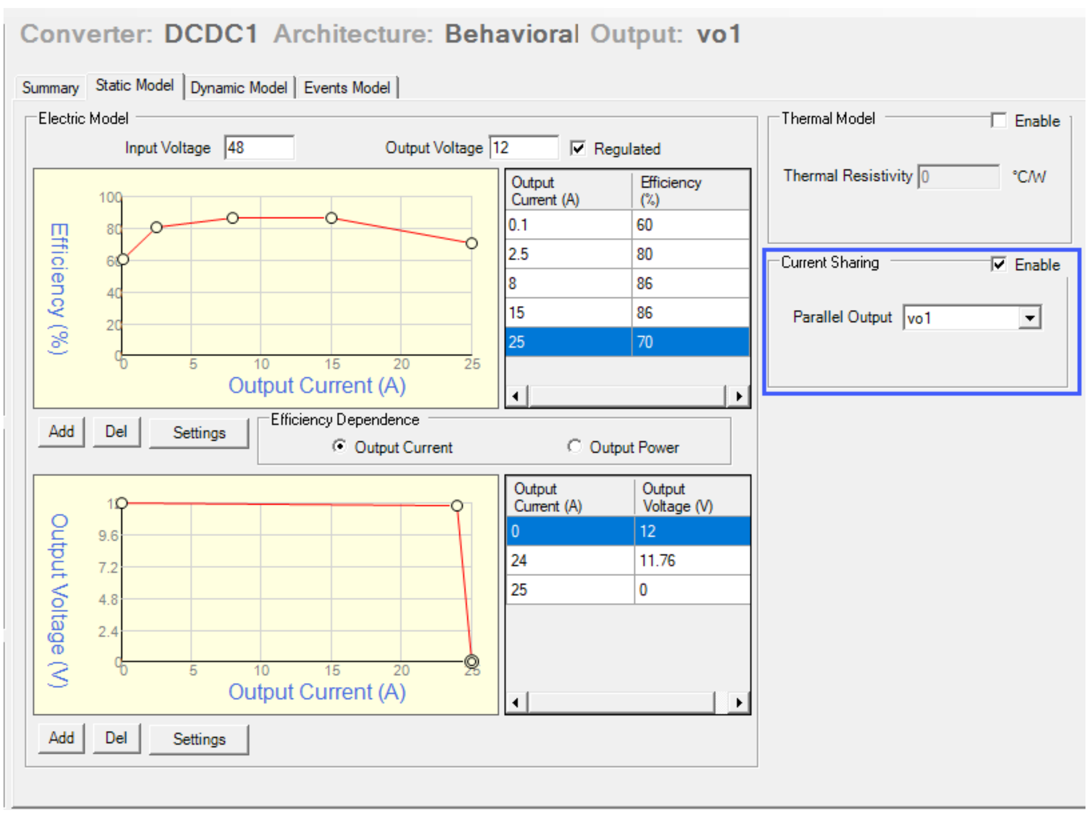
4. When **Remote Control** is enabled, a terminal is added to the Circuit model to control the converter. The **Turn On with value** combo box defines the value used to activate the converter. Set the **Remote Control Delay** to **1ms**. This value specifies the time that it takes for the converter to activate or off after the remote control signal is toggled.
5. Select the **Enable** check box in the **Input Voltage Protections** group box. This function defines the input voltage operating range of the converter.
   a. Set **Undervoltage Turn On** to **30.9V**.
   b. Set **Undervoltage Turn Off** to **34.5V**. These values mean that when the converter is deactivated it needs at least 34.9 V to activate; and once it is activated, if the voltage falls under 30.5V the converter turns off.
6. Set **Overvoltage Turn Off** and **Overvoltage Turn On** to **78.8V**. In this case, when the input voltage is above 78.8V the converter turns off and if it is under it turns on.



You can also enable other types of protection on the **Events Model** tab, such as **Output Voltage Protection** and **Thermal Protection**. In this example these protections are not used.

Enabling Current Sharing

1. Click the **Static Model** tab in the editing group box and select the **Current Sharing** check box.



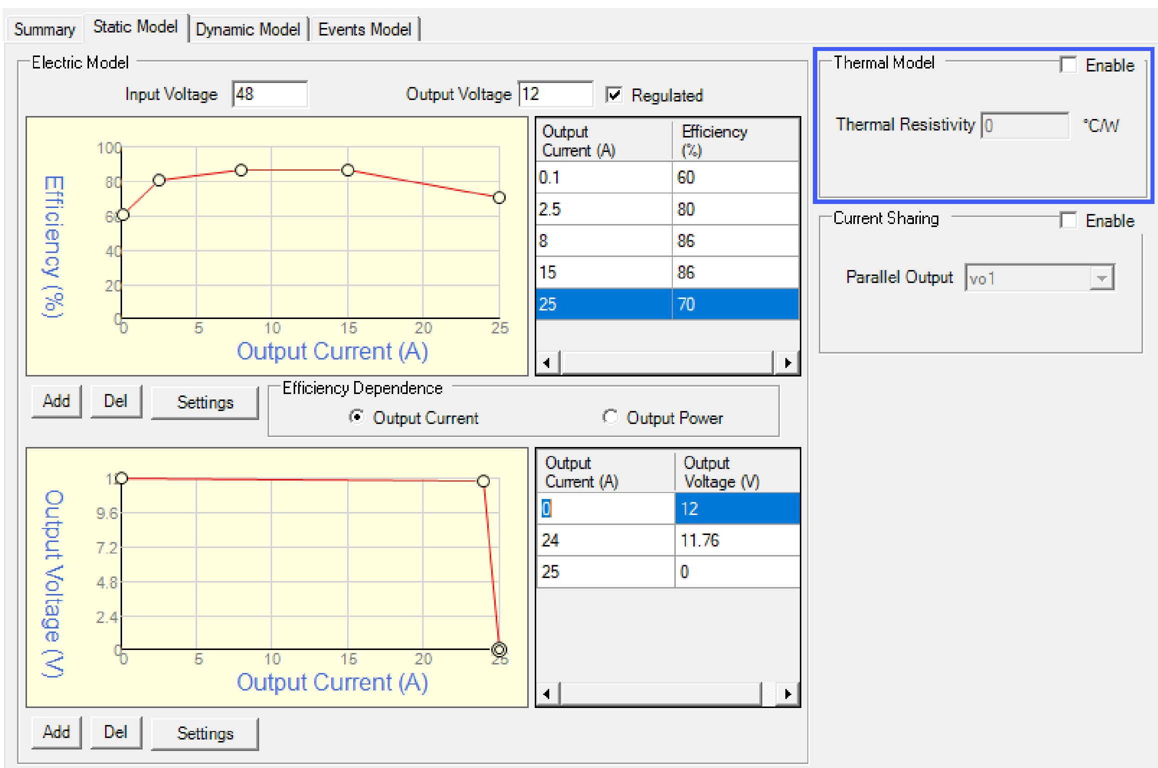
2. Select a **Parallel Output** on the .
3. Generate the code and import it to Circuit as specified in Generating SPICE Code and Importing the Model into Circuit.

4. Insert the model in a schematic in Circuit to create a simulation model.
5. In reality, two similar converters cannot be perfectly equal; they have physical differences. When you add the current sharing feature to a model, a corresponding **current_share_ deviation** parameter is also included in the model. This value defines a factor of deviation that simulates a real and imperfect current sharing among converters. Set this parameter to **0.001** on the **Param Values** tab of the **Properties** window in Circuit.
6. Similarly, to add the effects of thermal resistivity to the model, see Enabling Thermal Modeling and Thermal Protection.

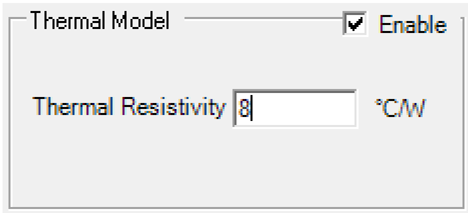
Enabling Thermal Modeling and Thermal Protection

To include thermal model and thermal protection, all thermal properties must be specified.

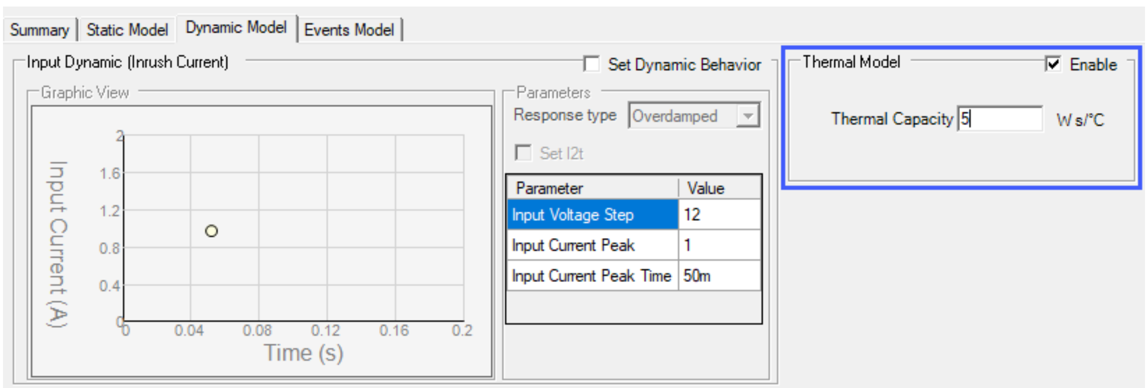
1. Click the **Static Model** tab in the editing group box and select the **Thermal Model** check box.



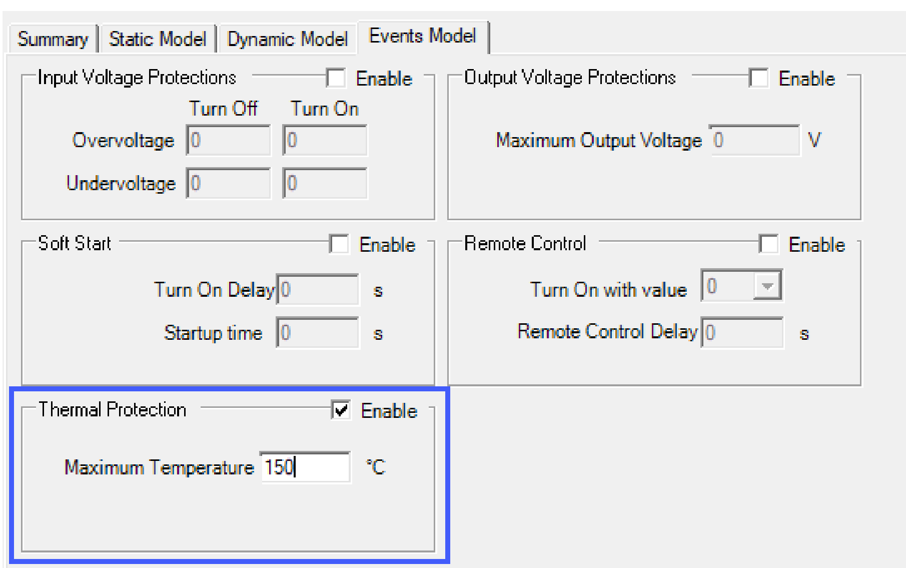
2. To add the effects of thermal resistivity to the model, check the **Thermal Model** check box and enter the appropriate **Thermal Resistivity** value.



3. To add the effects of thermal capacity to the model:
   a. Click the **Dynamic Model** tab in the editing group box.
   b. Check the **Thermal Model** check box.
   c. Enter the appropriate **Thermal Capacity** value.



4. To enable thermal protection, on the **Events Model** tab, enable **Thermal protection** and specify the **Maximum Temperature**.



Thermal Modeling and Thermal Protection

Solid state DC-DC power converters always have efficiency less than 100%, and therefore always waste a percentage of their input power. This wasted power is dissipated as heat and causes the temperature of the DC-DC converter to rise above the ambient system temperature. The temperature rise of the DC-DC converter must be considered during the system mechanical and thermal design to ensure the converter does not exceed its maximum rated operating temperature.

Thermal calculation requires information of power loss, package thermal resistance, and ambient temperature. Power loss is calculated in the same manner as the efficiency calculation: multiply the input/output voltage difference by the input current.

Thermal resistance is provided in most data sheets. If it cannot be found, consult the manufacturer of the linear regulator. The thermal calculation basically employs the thermal resistance θ_{ja}, between the chip (junction) and ambient. Some ICs provide thermal resistance θ_{jc} between junction and case. At any rate, a thermal resistance up to θ_{ja} must be determined. Finally, the ambient temperature can be an assumed temperature, such as 50°C, derived from ratings for the device in which the linear regulator is to be deployed. Under severe conditions, the ambient temperature must be obtained by direct measurement.

Thermal Characteristics

The thermal characteristics were evaluated on a 6-layer PCB structure (2 oz/2 oz/2 oz/2 oz) measuring 7 cm x 7 cm).

Symbol	Parameter	Typ.	Unit
Θ_{JA}	Thermal Resistance, Junction-to-Ambient	22.7	°C/W
Ψ_{JC}	Thermal Characterization Parameter, Junction-to-Top of Case	12.6	°C/W
Ψ_{JPCB}	Thermal Characterization Parameter, Junction-to-PCB	1.7	°C/W

$$T_j = \text{Power loss} \times \text{Termal resistance}\,\Theta_{\text{ja}} + T_a$$

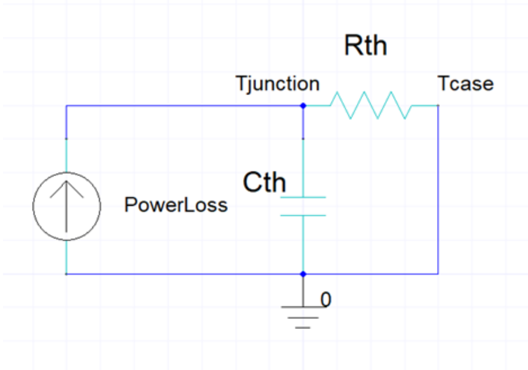
$$\text{Power loss} = (V_{\text{IN}} - V_O) \times I_{\text{IN}}$$

e. g. above condition: θ_{ja}= 50°C/W, T_a= 40°C

$$\{(5V - 3.3V) \times (0.2A + 5mA)\} \times 50^{\circ}C/W + 40^{\circ}C = 57^{\circ}C$$

At T_{jMAX} = 125°C, a 68°C margin is provided. With a termal calculation, ensure T_{jMAX} is not exceeded. THe amount of heat generation increases as input/output voltage difference and I_o rise.

The heat flow can be modeled by analogy to an electrical circuit where heat flow is represented by current, temperatures are represented by voltages, heat sources are represented by constant current sources, absolute thermal resistances are represented by resistors, and thermal capacitances by capacitors:



Defining Static Converter model:

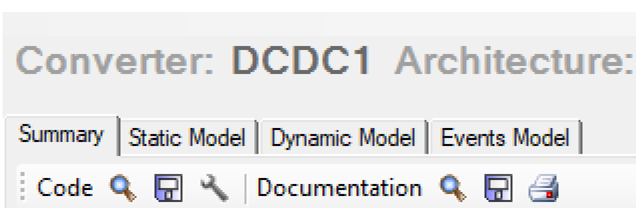
Using KCL:

$$i_c = i_{\text{Losses}} - i_{\text{Rtj}}$$

$$i_c = i_{Losses} - \frac{V_c}{Rtj} = \frac{Losses \times Rtj - V_c}{Rtj} = Cth \times \frac{dV_c}{dt}$$

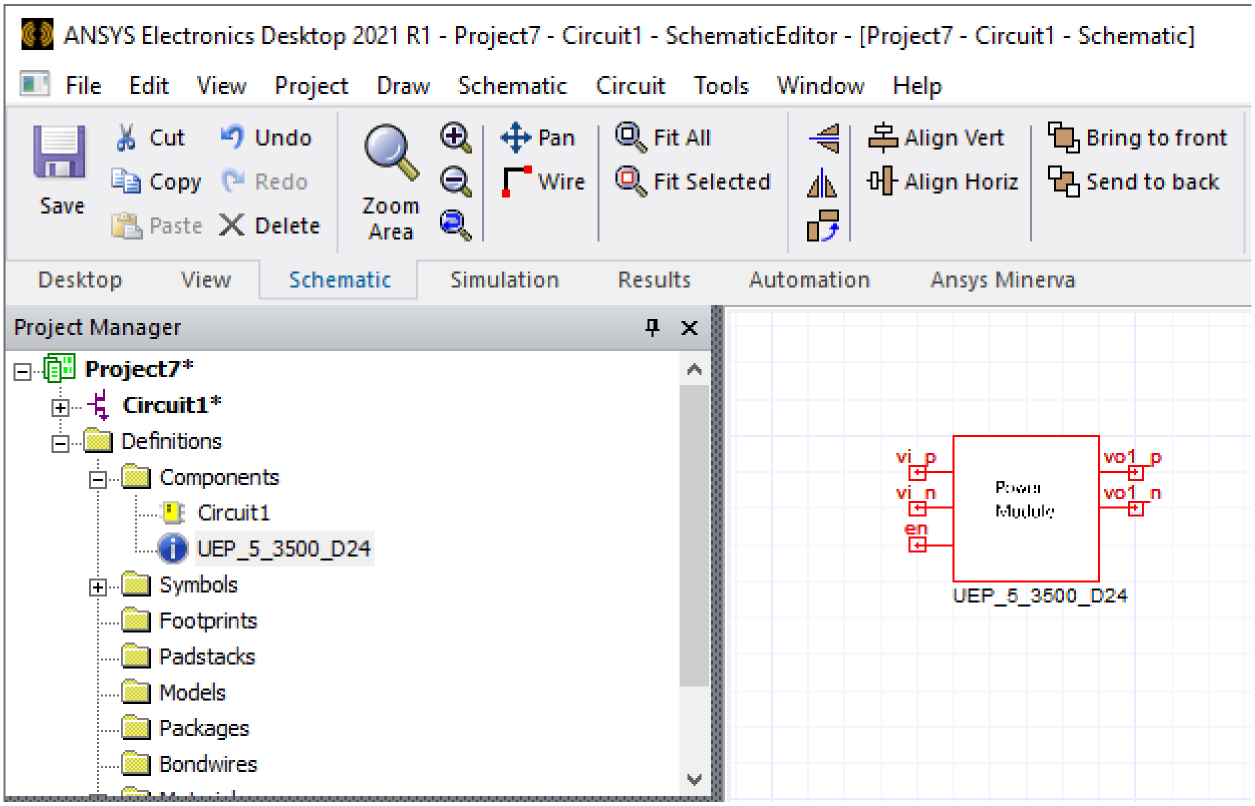
Generating SPICE Code and Importing the Model into Circuit

1. Click the **Summary** tab in the editing group box. The **Summary** tab contains options for the most common functions for code and documentation generation.



2. From the toolbar, click **Behavioral DC/DC** > **Code** > **View** or click the **Code**  icon. The SPICE code for the converter model appears (read-only) in the viewer.
3. To import this model into Circuit, click **Place**. This command adds the model to the currently active project in Circuit in the **Schematic Editor** and lists the component in the

Project Manager window under **Definitions** > **Components**.



Modeling of DC-DC Converters Based on Hybrid Wiener-Hammerstein Structure

Most modeling approaches of DC-DC converters are based on some kind of averaging technique that requires a knowledge of the topology, the control strategy, and the parameters of all the components. These kinds of techniques are very useful for design of the control loops but, because commercial DC-DC converters do not supply this information (it is part of their intellectual property), these modeling approaches cannot be applied to extract a model of a commercial DC-DC converter.

Other modeling approaches try to model the behavior of the converter based on the small-signal frequency identification of the converter. This approach is very accurate on the point of view of small-signal stability but it does not account for efficiency as a function of the input voltage and load, output voltage as a function of load or protections such as overvoltage, over-current, under-voltage and temperature. The following model proposed overcomes all of those limitations. The trade-off between simulation time and accuracy of DC-DC converters is successfully addressed by the use of a hybrid model as shown in the following figure:

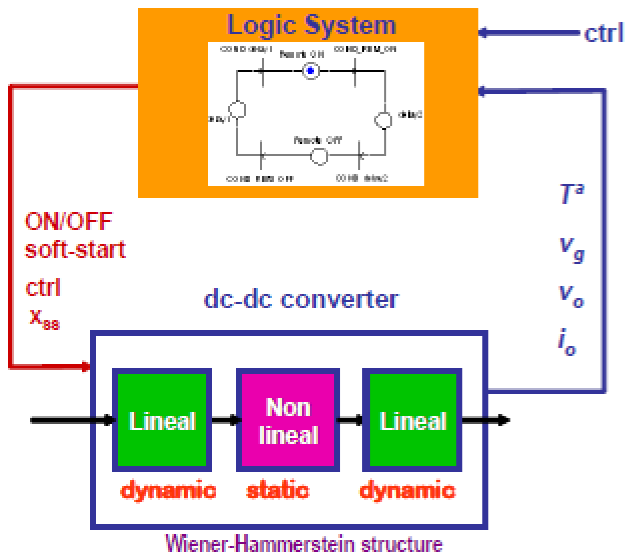


Figure 5-6 ***Proposed DC-DC Converter Model Structure***

Proposed Hybrid Wiener-Hammerstein Structure

Control signals, protection, and start-up behavior are extremely important in the assessment of the stability of distributed power systems. These parameters modify the behavior of the converter and must be considered in the model.

Taking into account the hybrid nature of DC-DC converters, the model is partitioned in two blocks:

The **logic system** (event driven behavior) manages the protection and remote control features. It is modeled by means of a state diagram. Its implementation on a circuit simulator depends on the modeling capabilities of the simulator. It measures the analog variables that can modify the behavior of the DC-DC converter (output voltage, output current, input voltage, temperature, et cetera) and, based on its current state and the analog inputs, it generates the control signal for the power stage and control.

The **continuous/discrete system** includes the power stage and its control - independently if it is implemented digitally or analogy.

The **power stage and control** is modeled by means of a Wiener-Hammerstein structure. This structure has the advantage of gathering all the non-linear static behavior of the converter, usually provided in datasheets, by means of one block. This non-linear behavior includes:

- Variation of the output voltage with respect to the output current and input voltage.
- Variation of the efficiency with the load and the input current.

Detailed Wiener-Hammerstein Structure

One of the most critical steps in the modeling process is to select the model structure. If the selected structure is wrong, it does not matter what optimization algorithm is used to fit the model – it never finds a correct solution.

In this case, the problem is how to find a model structure valid for all kinds of DC-DC converters (with and without zeros in the Right Hand Plane (RHP), hard switched PWM converters or frequency controlled resonant converters). The proposed model for the power stage is based on a Wiener-Hammerstein structure shown in the following diagram.

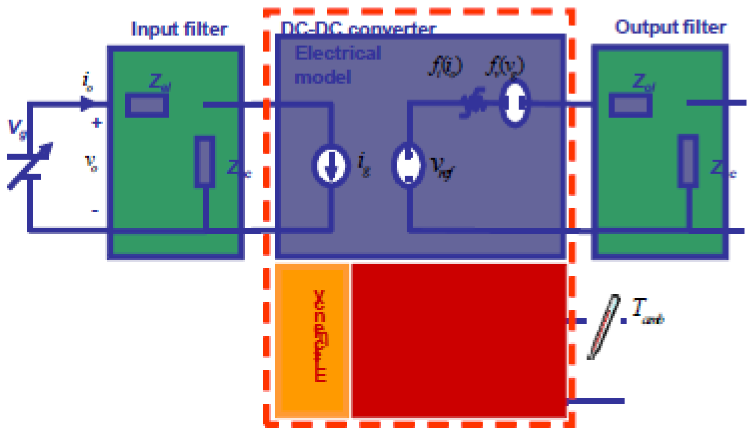


Figure 5-7 ***Proposed Wiener-Hammerstein structure for the dc-dc converter model***

It consists of:

- A non-linear static model that represents the steady-state behavior of the converter.
  a. Efficiency as a function of input voltage and output current.
  b. Output voltage as a function of the reference, the input voltage and the output current.
- A dynamic input block that models the high-frequency input impedance behavior of the converter and its initial inrush current.
- A dynamic output block that models the transient behavior of the converter under load changes.

Additionally, a simple thermal model accounts for the effect of thermal protection.

The equations for the static, nonlinear model are given by:

$$v_o = v_{ref} + f_v(v_g) - f_z(i_o) \quad (1)$$

$$i_{in} = \frac{1}{\eta(v_g, i_o)} \frac{v_o \cdot i_o}{v_g} \qquad (2)$$

The implementation of these equations in a circuit-oriented simulator based on information given by the manufacturers is straightforward. The dependence of the output voltage on the input voltage and output current can be implemented either by a 2-D lookup table, or by means of two 1-D lookup tables.[1] This solution, though an approximation, provides very good results in terms of accuracy and convergence.

Additionally, dependence of efficiency on the input voltage and output current also can be implemented by means of a 2-D lookup table or two 1-D lookup tables.

Linear dynamic output network

To propose a suitable network to consider the output dynamics of a generic converter the following assumptions are made:

1. **There is a capacitive element place at the output**. Actually, this is always the case to reduce output voltage ripple and reduce output impedance. This capacitive element need not be ideal, and the model can include the equivalent series resistance (ESR) and inductance (ESL) of the capacitor bank.
2. **Output voltage of the converter is controlled through the injected current**. This is almost true for current mode controlled converters up to the crossover frequency of the current loop. In voltage mode controlled converters seems less obvious but, as is shown the approach is also applicable - but with higher error.

Using these assumptions, the small-signal model of the output stage of a general current controlled converter can be represented as shown in the following diagram, where:

- Z_C is the equivalent impedance of the output capacitors (including parasitics).
- K_V is the output voltage controller transfer function.
- $\tilde{i}_c$ is the injected current.
- $\tilde{v}_{ref}$ is the reference voltage.
- $\tilde{i}_0$ is the current demanded by the load.

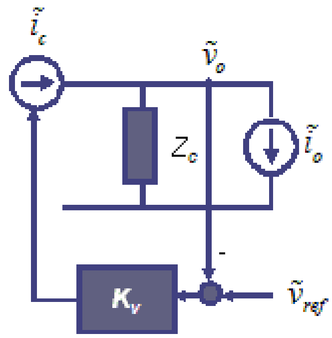


Figure 5-8 ***Equivalent Small-signal Circuit of the Output Stage of a Current-Controlled Converter***

According to the equivalent circuit shown above, and after some calculation, it is possible to arrive at the following expression for the output voltage (3):

$$i_{in} = \frac{1}{\eta(v_g, i_o)} \frac{v_o i_o}{v_g} \qquad (3)$$

This transfer function is also the same as the transfer function of the equivalent circuit of the following figure, just by making the series impedance Z_L equal to the inverse of the voltage loop regulator (4):

$$Z_L(s) = \frac{1}{K_v(s)} \qquad (4)$$

As a consequence, the proposed dynamic network to fit the output dynamics of the converter is the one shown in the following diagram.

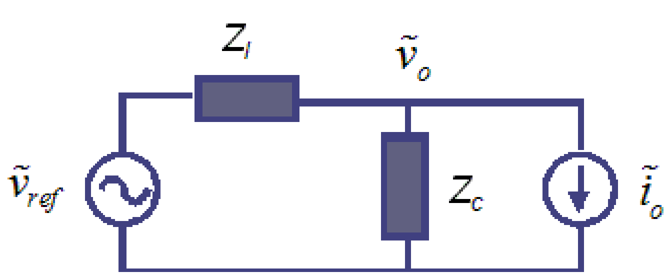


Figure 5-9 ***Equivalent small signal circuit of the output stage***

Because the output network does not change the static behavior of the converter, which is accounted for by the non-linear network, the condition that must be satisfied by this network is:

$$\frac{Z_L(0)}{Z_C(0)} = 0 \qquad (5)$$

The impedance of Z_L/Z_C must be 0. It can be achieved with $Z_C(0)$=infinite (capacitor) and/or $Z_L(0) = 0$ (inductance).

From the point of view of the parametric identification it is necessary to determine the structure of the series impedance, Z_L, to identify its components. To do that, the form of the transfer function K_v is analyzed.

Using the above assumptions and the small signal model shown above, the output voltage can be controlled by means of a PI controller:

$$K_v(s) = \frac{\omega_i}{s} \cdot \left(1 + \frac{s}{\omega_z}\right) \qquad (6)$$

where:

- ω_i adjusts the crossover frequency and
- ω_z sets the phase margin.

This controller, according to equation (4) yields the following equivalent series impedance:

$$Z_l(s) = \frac{1}{K_v(s)} = \frac{\frac{1}{\omega_i} s}{1 + \frac{s}{\omega_z}} \qquad (7)$$

The network that fits into this transfer function consists of an inductance in parallel with a resistor of the following values:

$$L = \frac{1}{\omega_i}$$

$$L = \frac{1}{\omega_i} \tag{8}$$

$$R_d = \frac{\omega_z}{\omega_i} \tag{9}$$

Applying this procedure to the current-controlled converter with equivalent output capacitors impedance (C, R_{ESR}), the linear output network has the structure shown in the following diagram. This output network can be easily identified based on the current step-response data provided by the manufacturer (or an equivalent measurement).

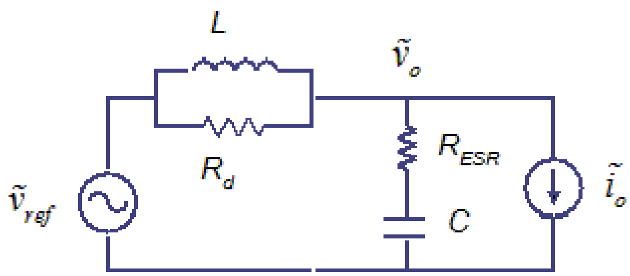
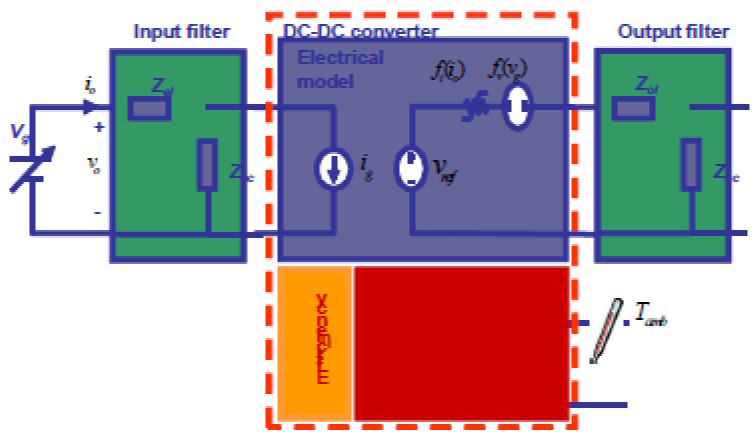


Figure 5-10 ***Equivalent Closed Loop Circuit***

Linear dynamic input network

The proposed structure of the input network is labeled **Input filter**. The selection of this network is based on the typical configuration of the EMI input filters for DC-DC converters.



This network must be selected under the constraint of not modifying the static behavior of the converter, given by the nonlinear static block. As a consequence, it must satisfy the same condition in steady state as the output network equation (5).

One possible network that can be fitted based on inrush current data is shown in the following [1-2]. This network is valid for both over-damped and under-damped inrush current behavior.

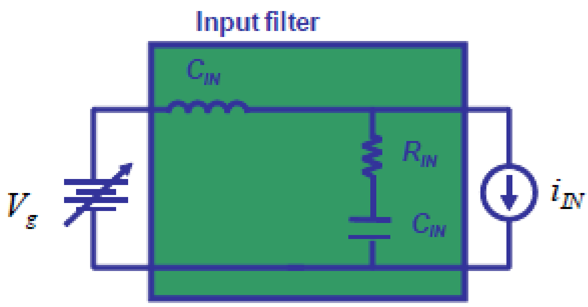


Figure 5-11 ***Candidate input filter network***

Event-driven Behavior

The behavior of the converter depends on its state (ON, OFF, PROTECTION, et cetera). It is possible to include this dependency on the behavior thanks to the mixed-signal capabilities of current circuit simulators. In this example, the state of the converter is modeled by means of the following state diagram that controls the values of the dependent voltage and current sources.

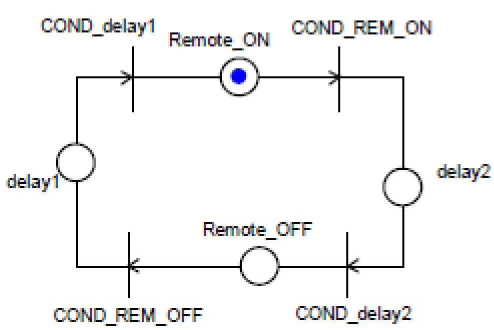


Figure 5-12 ***Remote ON-OFF state diagram***

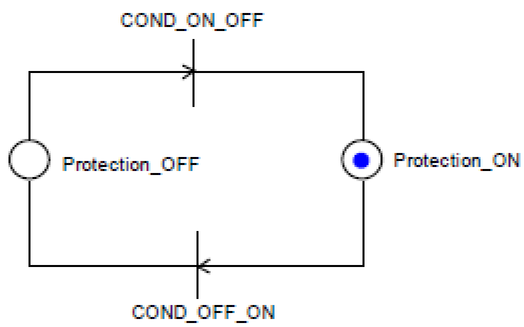


Figure 5-13 ***Protection state diagram - State Diagram for Event Driven Behavior Description***

In this example, most common events are included based on a careful search of many DC-DC converter manufacturers. The events included in this model are:

- **Remote control** - This signal is usually provided to DC-DC converters to enable or disable them.
- **Input voltage protection** - This protection disables the operation of the converter when the input voltage is out of the operational limits of the converter, to avoid malfunction.
  - **Under-voltage protection** (with and without hysteresis) - Under this voltage, the converter is deactivated to avoid high currents that can destroy it.
  - **Over-voltage protection** (with and without hysteresis) - This voltage setting assures that all the components of the converter are working with designed voltage ranges.
- **Output voltage protection** - This is usually an upper output voltage value that disables the converter to protect it against overvoltage caused by an external source.
- **Thermal protection** - In the case of high-temperature operation of the converter, this protection turns it off if the high-temperature limit is exceeded.
- **Over-current protection** - This protection tries to avoid high internal current due to high external loads. This protection is included inside the event-driven behavior in case of turning-off the converter. The protection based on lowering the output voltage as a function of the output voltage must be included in the static non linear model since it does not create an event.

References

[1] J. A. Oliver, R. Prieto, V. Romero, and J. A. Cobos, “*Behavioral modeling of dc-dc converters for large-signal simulation of distributed power systems,”* IEEE Applied Power Electronics Conference and Exposition, APEC 2006.

[2] J. A. Oliver, R. Prieto, V. Romero, and J. A. Cobos, “*Behavioral Modeling of Multi-Output DC-DC Converters for Large-Signal Simulation of Distributed Power Systems,”* Power Electronics Specialists Conference, 2006.

6 - Assigning Materials

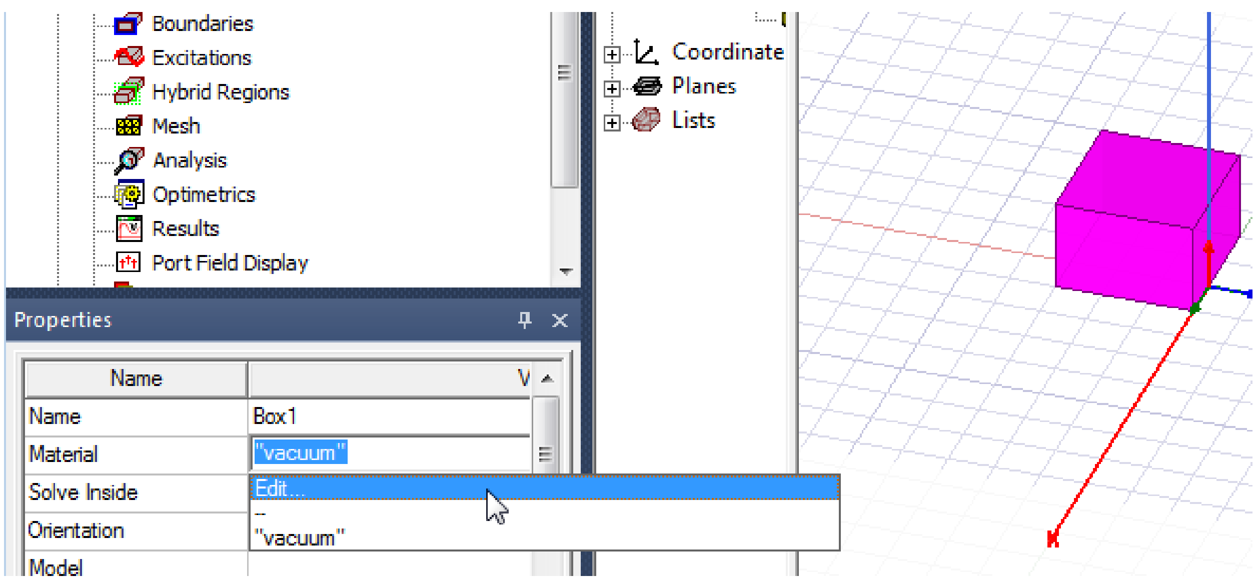
You can add, remove, and edit materials in two main ways:

- 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:

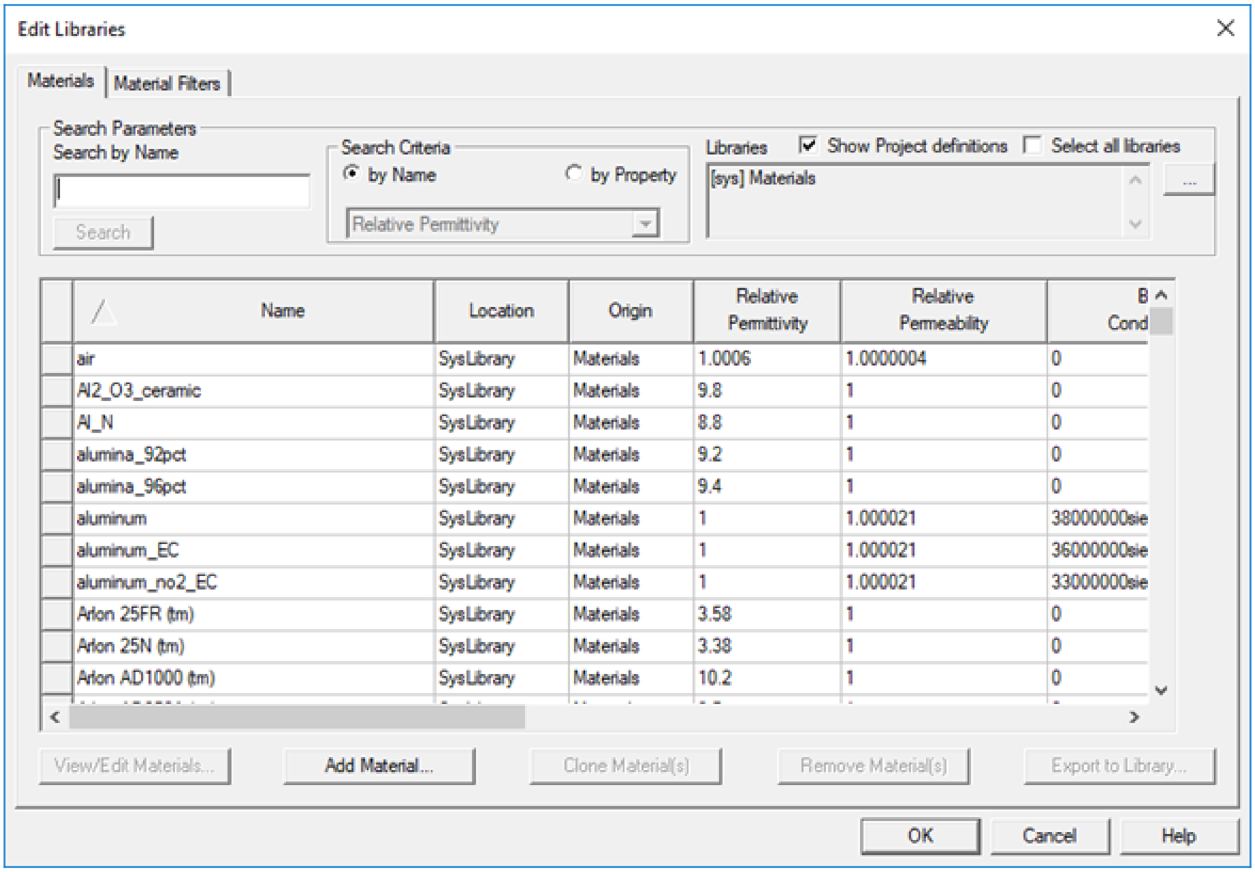
1. Select the object to which you want to assign a material.
2. Click **Modeler** > **Assign Material** or select the Material field in the **Properties** window for the selected object, and select **Edit...** from the drop-down menu:



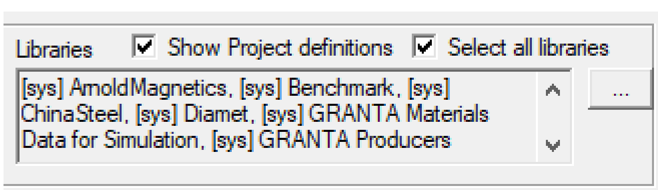
The **Select Definition** window appears. The current material is highlighted, with the Name, Location, Origin library, and parameter values shown.

> **Note:** You can resize the dialog box to show more of the materials and/or material properties without having to use the scroll bars.

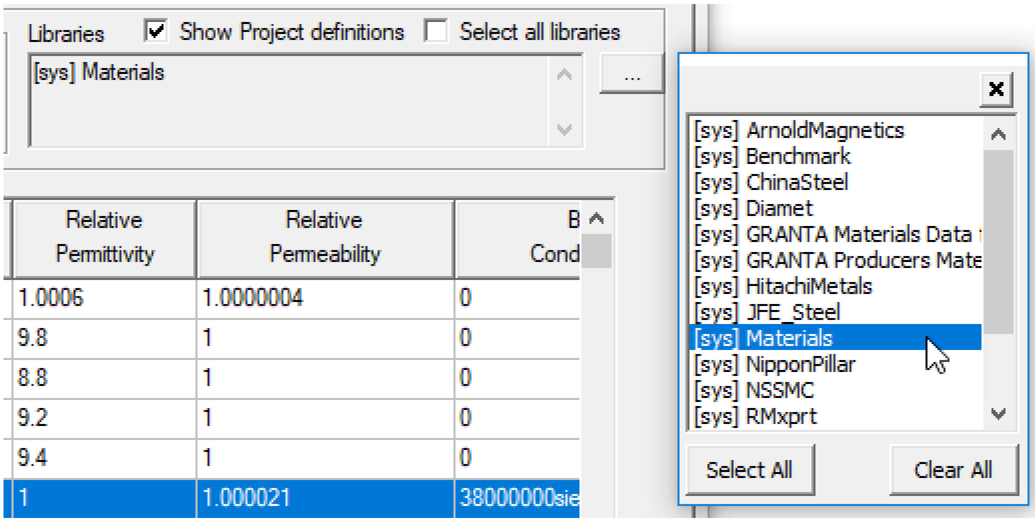
Select Definition Dialog Box - Materials



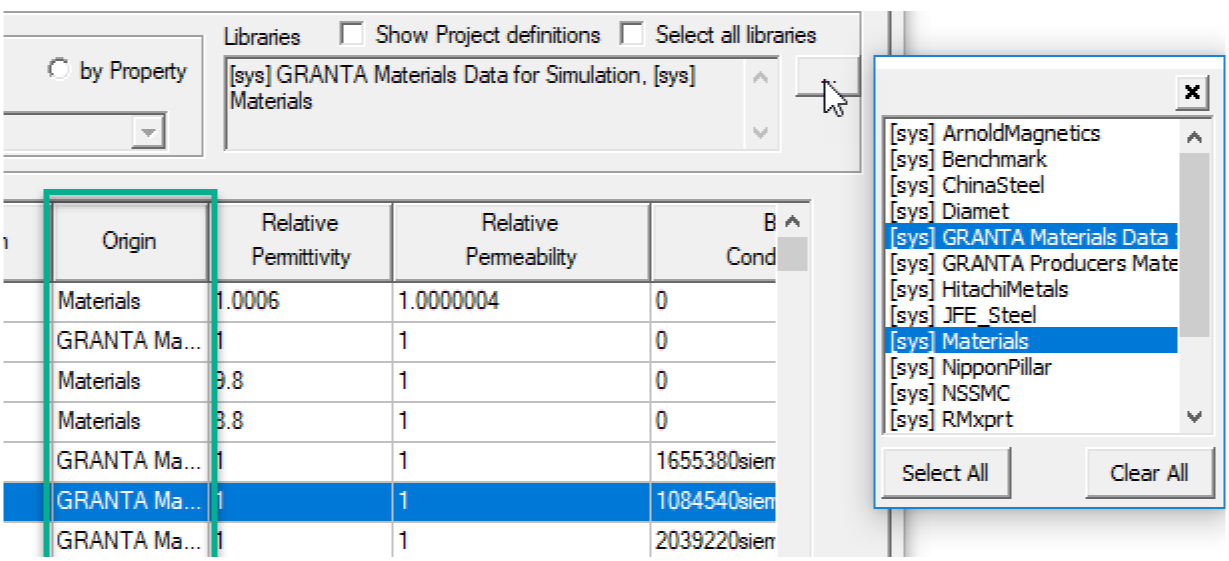
When the **Select all libraries** check box is selected, the window lists all of the materials in Ansys Electronics Desktop's global material library as well as the project's local material library



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.

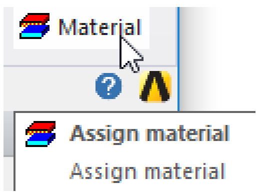


For further information on the materials and their intended uses, you can refer to the published information on materials from those libraries. For example, the Schott materials are described in detail in the Wily Series in Materials for Electronic & Optoelectronic Applications, *Microwave Dielectric 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:

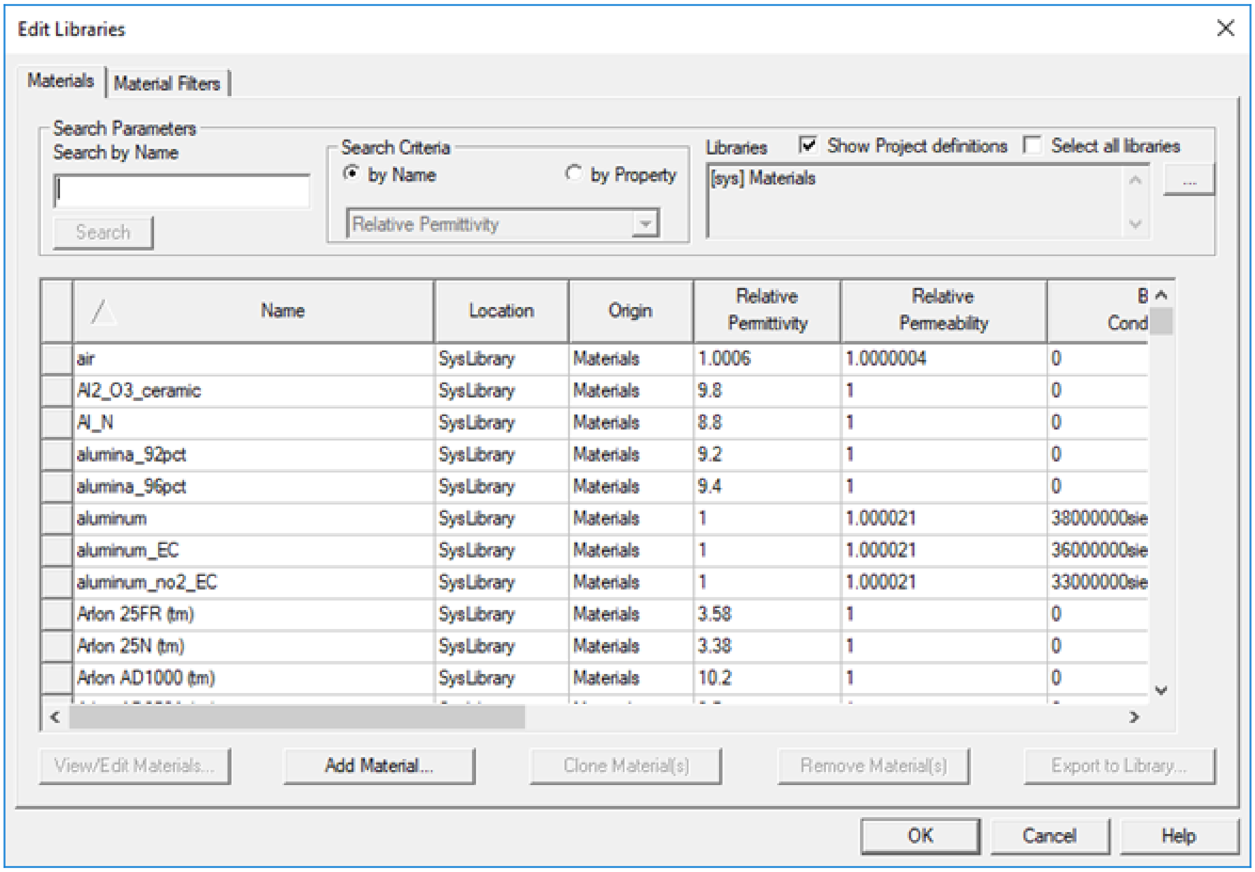
- In the **Properties** dialog box for the object, click the material name under the **Attributes** tab. A drop-down menu shows an **Edit...** button that opens the **Select Definition** window.
- With an object selected in the **Modeler** window, on the **Draw** ribbon, select the **Material** icon.



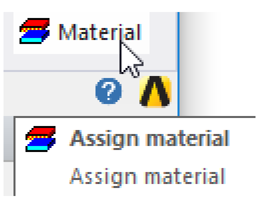
The menu also lists materials included in the current project. Selecting one of these materials provides another way to assign materials to an object.

- 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.

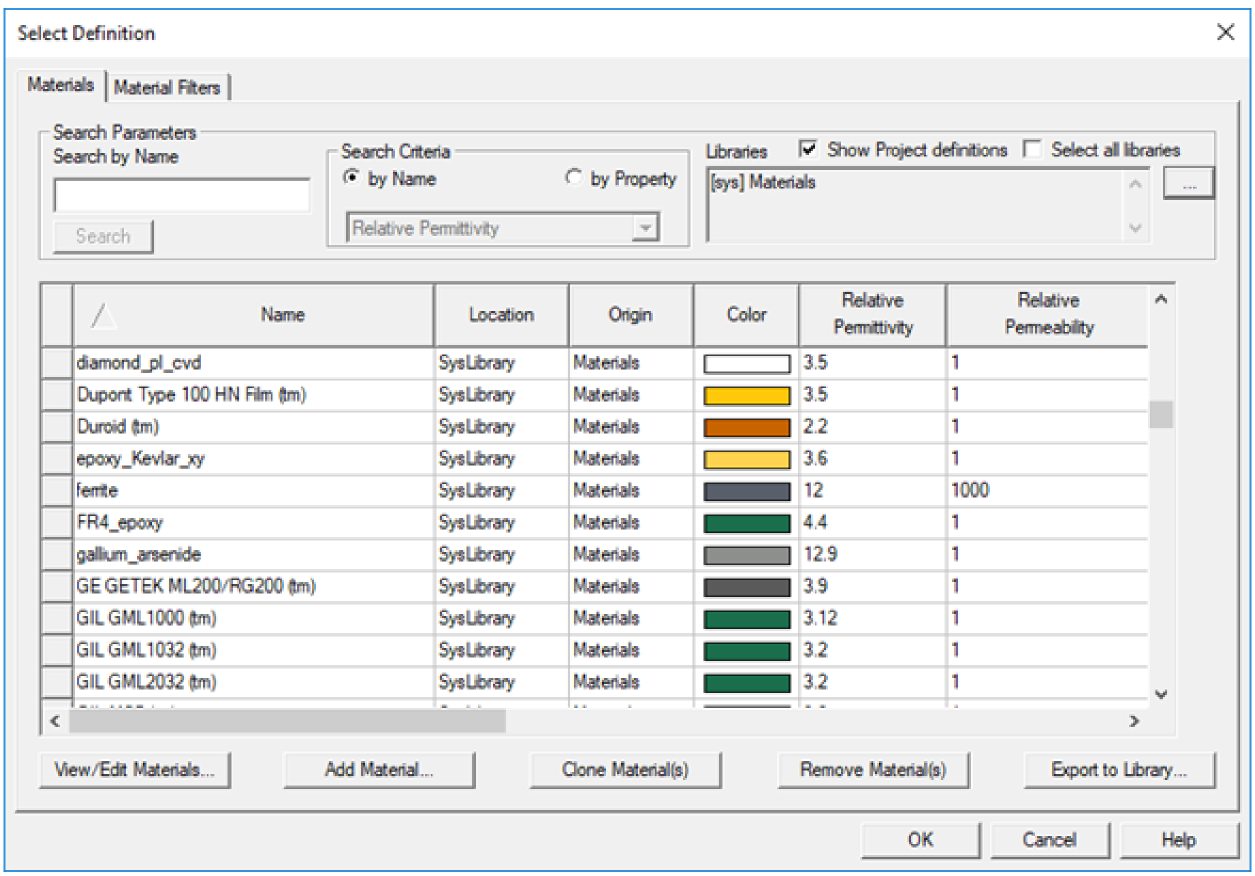
If the material you want to assign is not listed, add a new material to the global or local material library, and then select it.

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.

Solving Inside or on the Surface

When you assign a material to an object, you can specify whether to generate a field solution inside the object or on the surface of the object. If you elect to generate a solution inside the object, a mesh is created inside the object and generate a solution from the mesh. If you elect to generate a solution on the surface of the object, only a surface mesh is created for the object.

If you want a solution to be generated inside an object, select **Solve Inside** in the **Properties** window. Conversely, if you want a solution to only be generated on the surface of an object, clear the **Solve Inside** option in the **Properties** window.

By default, **Solve Inside** is selected for all objects with a bulk conductivity less than 10^5 siemens/meter and for perfect insulators. By default, the **Solve Inside** option in the **Properties** window is clear for perfect conductors.

To change the threshold for solving inside objects, do the following:

1. Click **Tools**>**Options**> **General Options**.
2. In the left pane, select **Circuit** > **Materials**.
3. Enter a new value in the **Solve Inside threshold** text box.

A finite conductivity boundary condition is placed on the surfaces of an object that should not be solved inside based on the material properties of that object. You can also incorporate a DC Thickness for the implicit boundary condition by setting an appropriate thickness value as described in Assigning DC Thickness.

Assigning DC Thickness

The **Assign DC Thickness** option allows you to more accurately compute the DC resistance of a thin conducting object for which Solve Inside is not selected. Skin impedance of the object will be calculated using the defined finite thickness, which can be Automatic (the default), Manually assigned per object, or Infinite per object. This option also exists for finite conductivity boundaries.

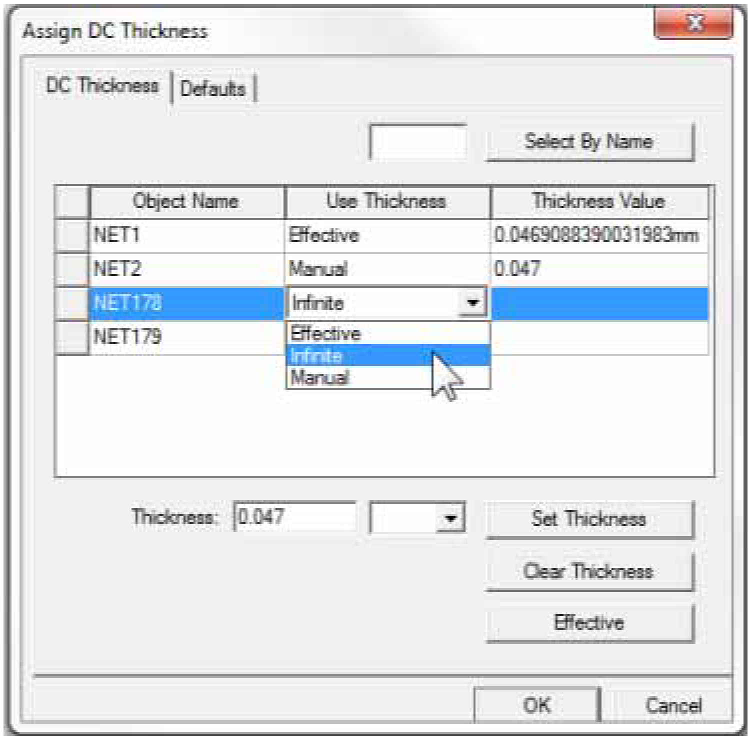
The **Assign DC Thickness** option on the **Boundaries** menu is enabled if at least one object contains a good conducting isotropic material (such as copper), and the **Solve Inside property** is not selected. If the object meets these conditions, you can assign a DC thickness, either by enabling the automatic default, or by specifying a value for a selected object.

To see the **Assign DC Thickness** dialog box:

1. Select **Circuit** > **Boundaries** > **Assign DC Thickness**.

   This displays the **Thickness of Objects for DC Resistance** dialog box with the DC Thickness tab selected. Objects to which the thickness can be applied are listed in the **Object**

Name column.



2. Select the objects to assign a value. You can select objects either by:
   - Clicking on the **Object Name** to highlight it.
   - Use the **Select By Name** field to type the object name, and click **Search**.

   The first object to match the name is highlighted.

Selecting an object highlights the **Thickness** field and the **Set Thickness** button.

3. In the Use Thickness column, you can specify that the value the object uses is Automatic, Infinite, or Manual.

You can disable automatic assignment on the **Defaults** tab of the **Assign DC Thickness** dialog box. The Automatic value is calculated as Thickness ~ 2* Volume / Surface Area

It should be noted that this is a calculation for an "effective" DC thickness to be used by the correction calculation. For a "thin" object this will work well. For example, a rectangular microstrip trace described by a box with dimensions 100 by 10 by 1 the volume is 1000 and the surface area is 2220 resulting in an apparent thickness of .9009, close to the geometric thickness of "1". For arbitrary shapes of "thicker" objects, this calculation will not work as well. For example, a cube with sides of 1 will have volume of 1 and surface area of 6, and a resulting apparent thickness of .3333. Another example is the case of a cylindrical wire (e.g., bond wire).

In this case, the automatic effective DC thickness will be about R_o, which gives us the best approximation of the DC resistance of a cylindrical wire.

The intention is that the auto-thickness will provide an accurate representation the majority of the time and is superior to not using any DC thickness setting. When the automatic value is not appropriate, you can override it using the manual technique.

DC thickness impedance is an approximation. It is accurate just for TE/TM waves when the widths are infinite which clearly never occurs in a "real" design. The fact that the object is finite causes an increase of the effective impedance due to current crowding/edge effects. Thus entering the exact geometrical thickness would actually underestimates the impedance. By returning a DC thickness smaller than the geometric thickness the automatic DC thickness compensates for this underestimation resulting in a slightly higher impedance as desired.

4. To manually apply a value, enter a **Thickness** value, select the units and click the **Set Thickness** button.

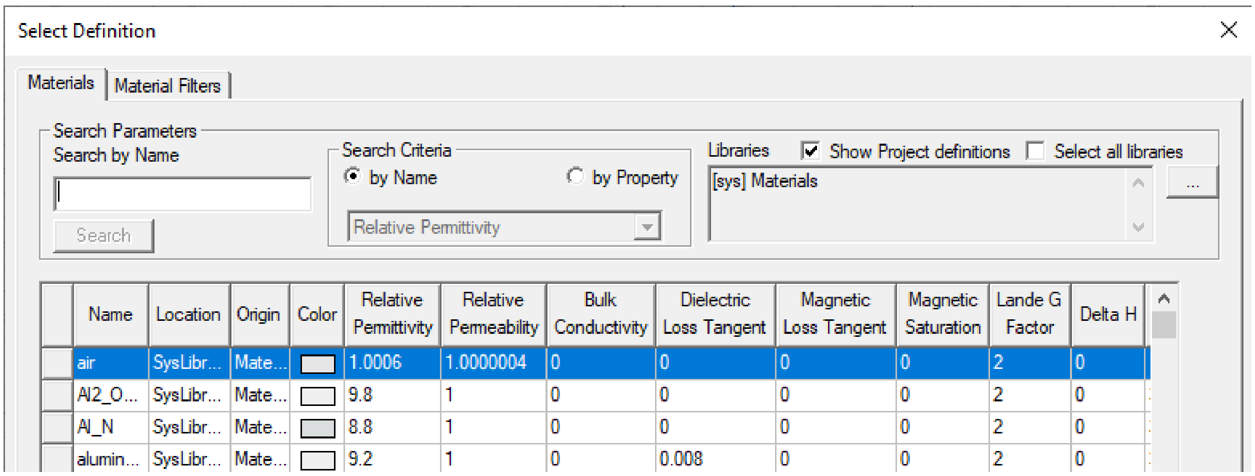
This applies the value to the selected object and changes the **Use Thickness** selection for that object to **Manual**.

> **Note:**
>
> If you enter a “0” for the thickness, HFSS gives a warning that this will cause infinite impedance that causes isolation.

5. To change the value, select the **Clear Thickness** button and then enter a different value. You can also manually select or deselect the box and manually enter or delete a thickness value in the table.
6. When you have assigned the values you need, click **OK** to close the 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.

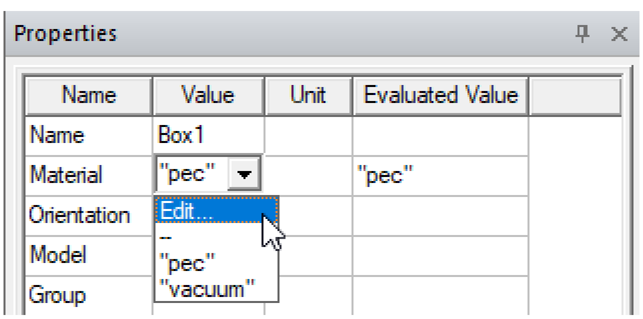


To search for a material **by Name**:

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):

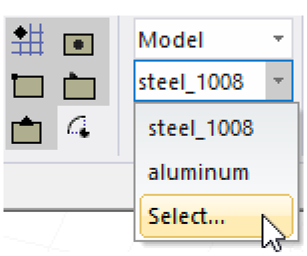
   - Click **Modeler** > **Assign Material**
   - Click the **Material** *value* in the docked **Properties** window, and select **Edit...** from the drop-down menu.

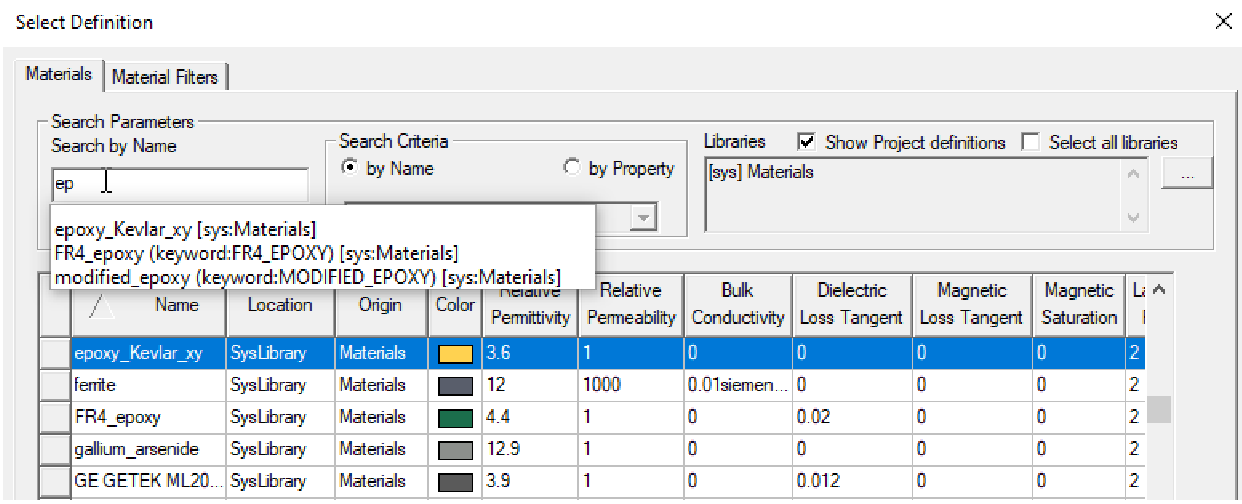
   - On the **Draw** ribbon tab, click **Assign material**.

   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.



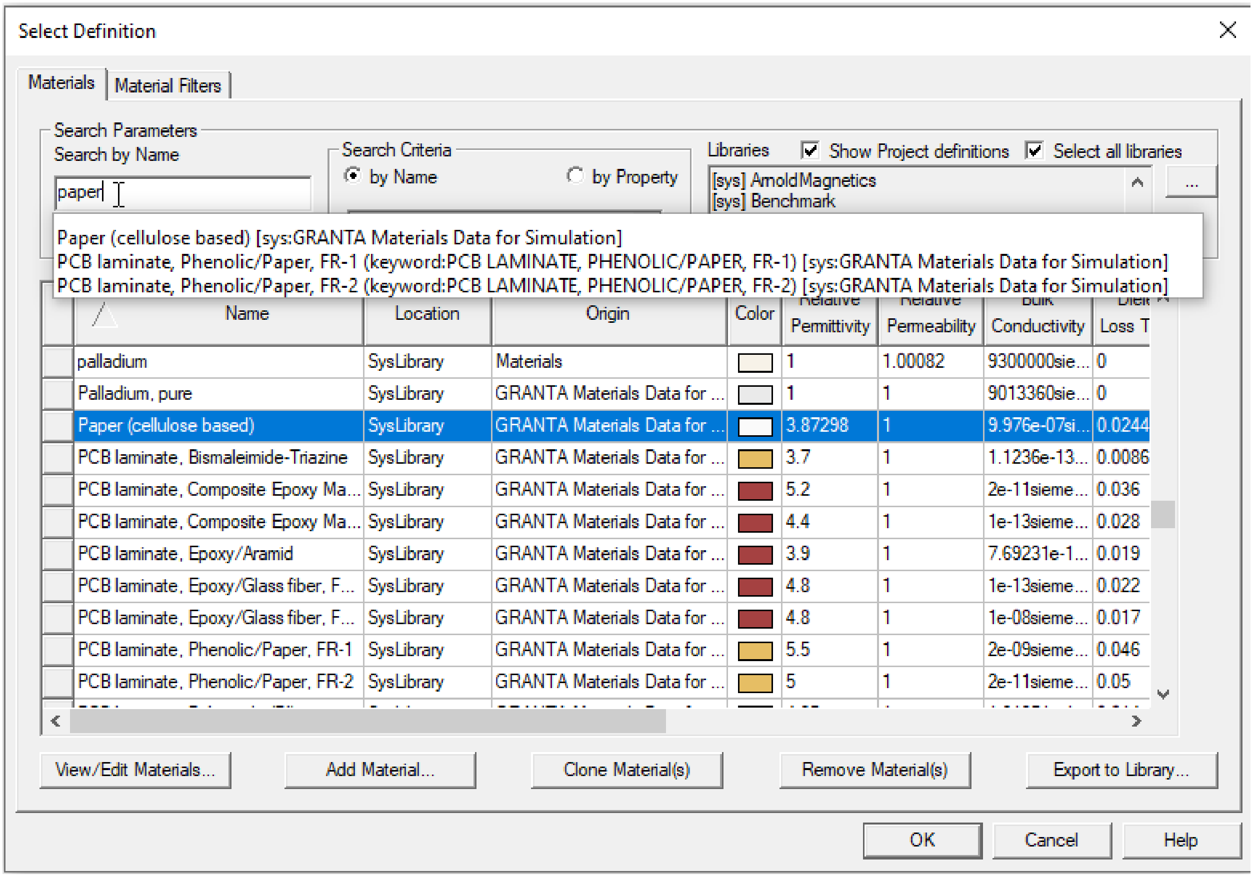
A drop down under the Search Parameters field will show the a list of the materials from selected libraries matching any part of the current text, and, if the Granta Materials library is selected, matching keywords in the material definitions. The first row containing the material name most similar to 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

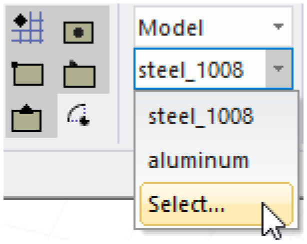
1. Access the **Select Definition** dialog box using one of the following methods:

   With one or more objects selected (that is, to assign a material to selected objects):

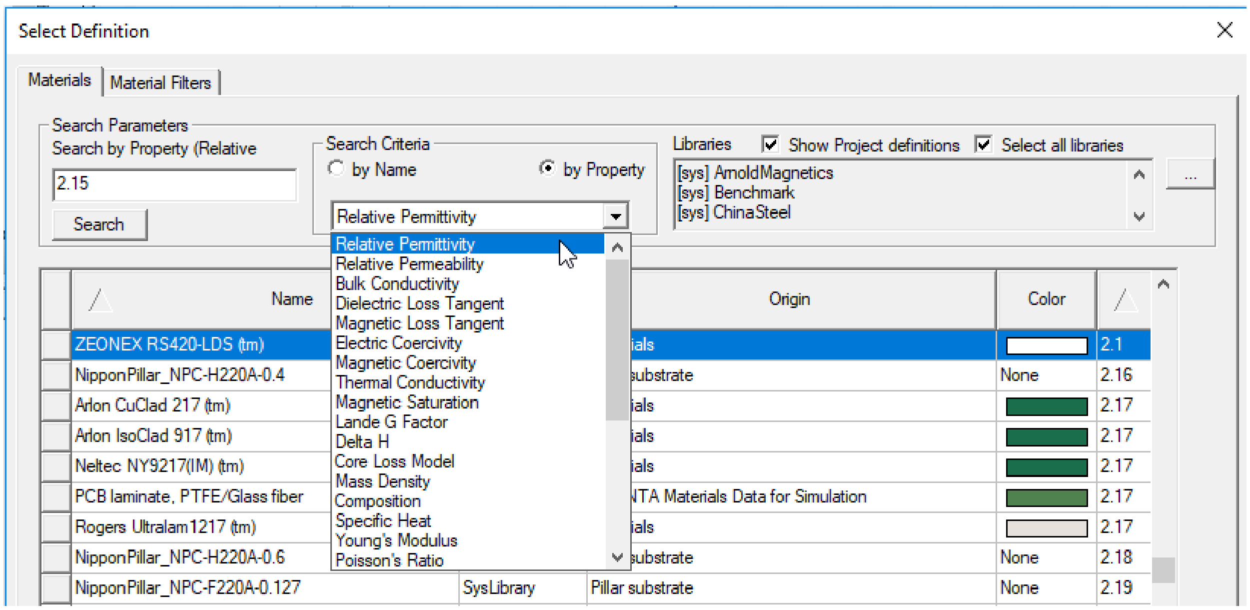
   - Click **Modeler** > **Assign Material**
   - Click the **Material** *Value* in the docked *Properties* window, and select **Edit...** from the drop-down menu.
   - On the **Draw** ribbon tab, click **Assign material**.

   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.

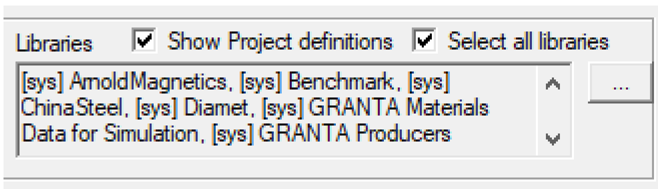
4. In the **Search Parameters** area, type a numerical value in the **Search by Property** text box and then click **Search**.

   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:

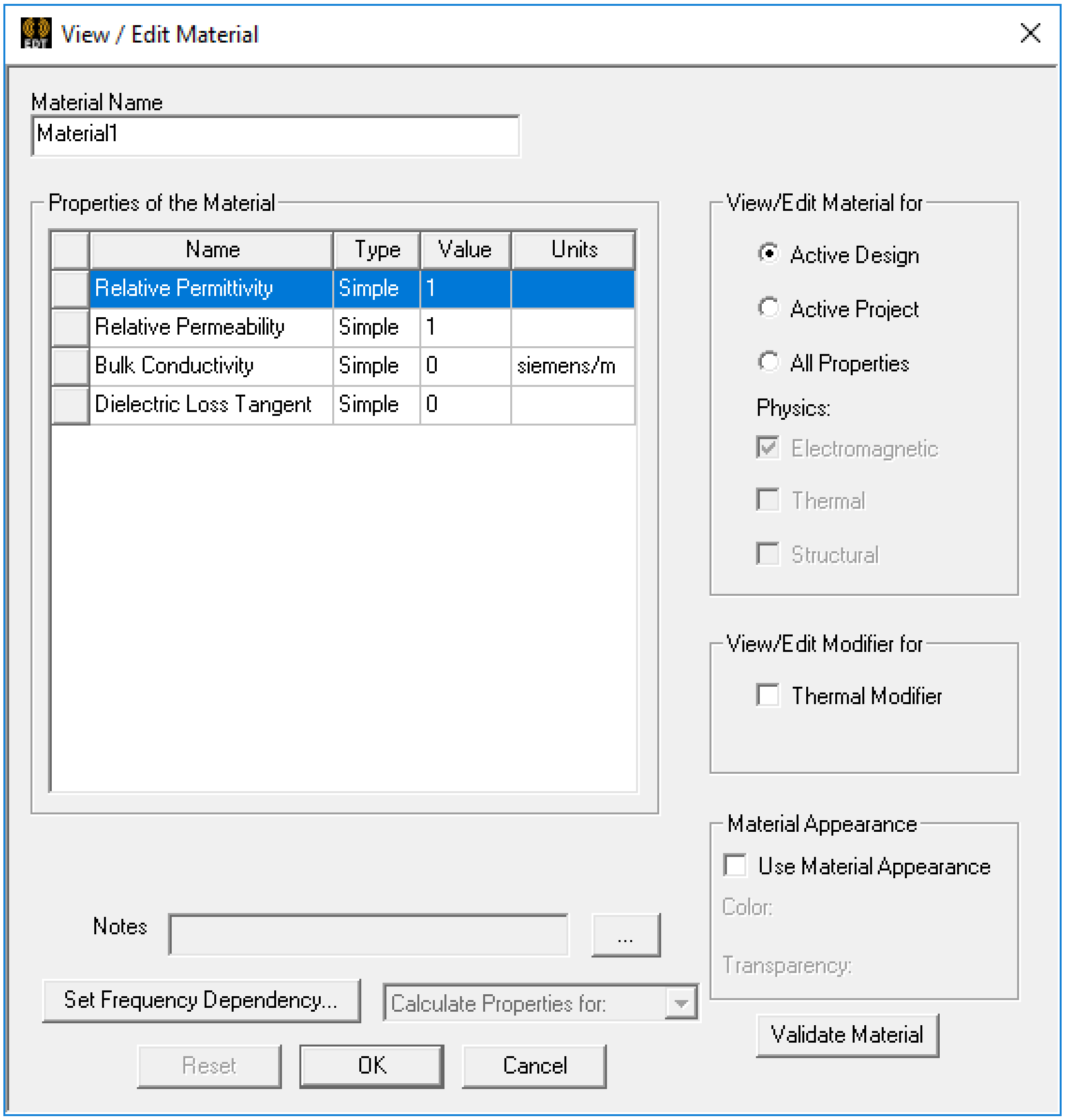
- Click **Tools** > **Edit Libraries** > **Materials**. In the project tree, you can also right-click **Materials**, and select **Edit Library**.

  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.

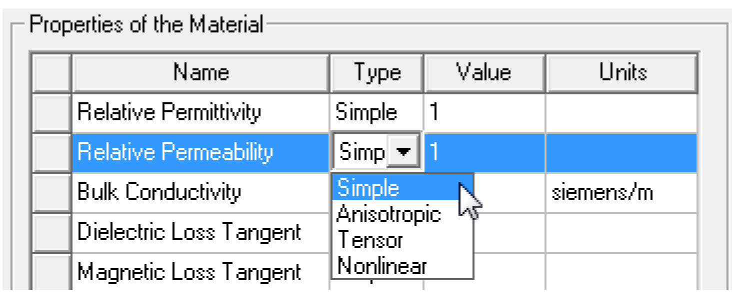
2. Type a name for the material in the **Material Name** text box, or accept the default.
3. Use the radio buttons in the **View/Edit Material for** section to specify whether the new materials apply to Active Design, Active Project, or All Properties. When **All Properties** is selected, **Physics** 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.



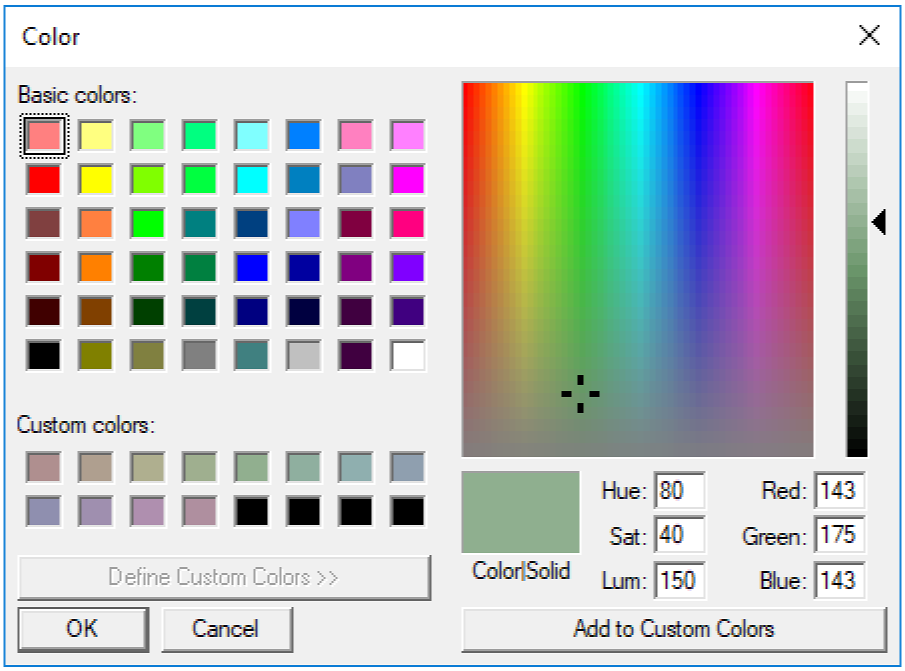
6. If the material is *linear*, enter a **Value** for the following material properties:
   - Relative Permeability
   - Relative Permittivity
   - Bulk Conductivity
   - Dielectric Loss Tangent
   - Magnetic Loss Tangent

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).

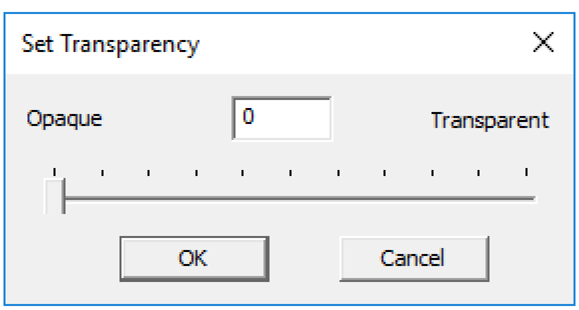
Note:

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:

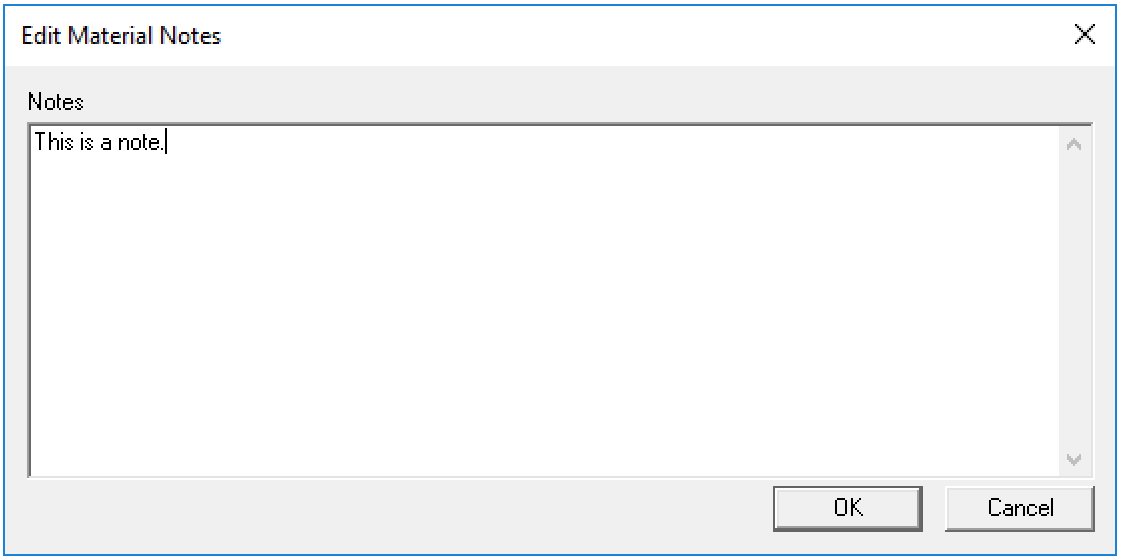


10. Clicking the Transparency box opens a Transparency dialog box with a text field and slider bar for selection.



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.



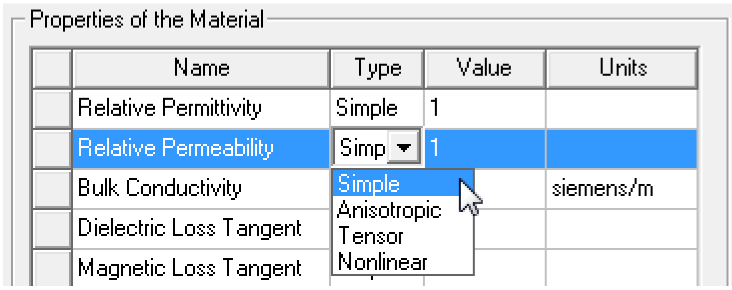
15. Click **OK** to add the Notes.
16. Click **OK** on the **View/Edit Materials** dialog to add the new material to the material library.

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.
Anisotropic	The material's characteristics vary with direction. This models only the diagonal.
Tensor	A full 3x3 matrix general tensor, which may or may not be symmetrical.
Nonlinear	A BH Curve.

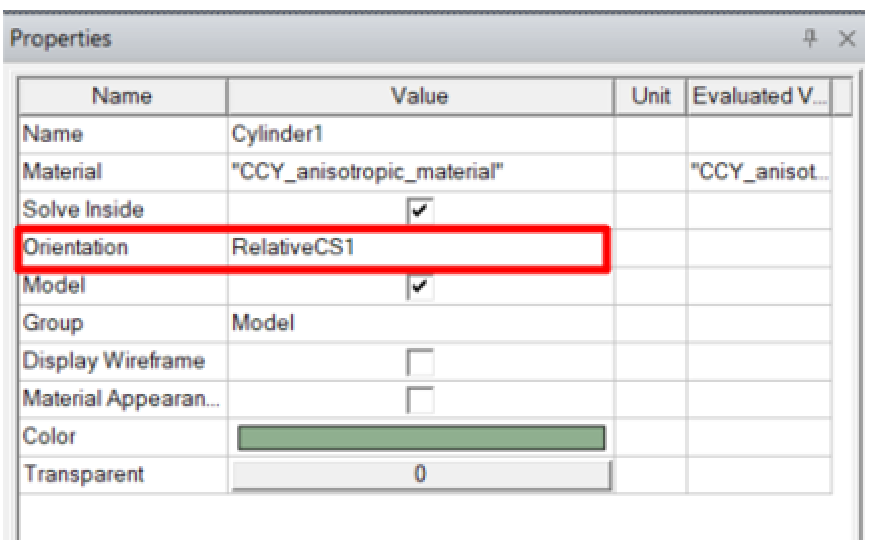
Select a material property type for each property from the **Type** drop-down menu. Of the four possibilities, only those applicable to the named material will be listed on the pull-down for type. Some properties only use the Simple type. Others include three or four potential types.



Defining Anisotropic Tensors

If a material property is anisotropic, its characteristics are defined by its anisotropy tensor. Each diagonal represents a tensor of your model along an axis. These tensors are relative to the coordinate system specified as the object's *Orientation* property (when supported). Otherwise, the tensors conform to the global Cartesian coordinate system. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently. Spherical and Cylindrical tensors must be converted to Cartesian coordinates, as described below. In such cases you must specify the coordinate system carefully.

NOTE:Spatial-dependent material properties are calculated according to coordinates in the 'Orientation CS'.This CS should be set carefully.

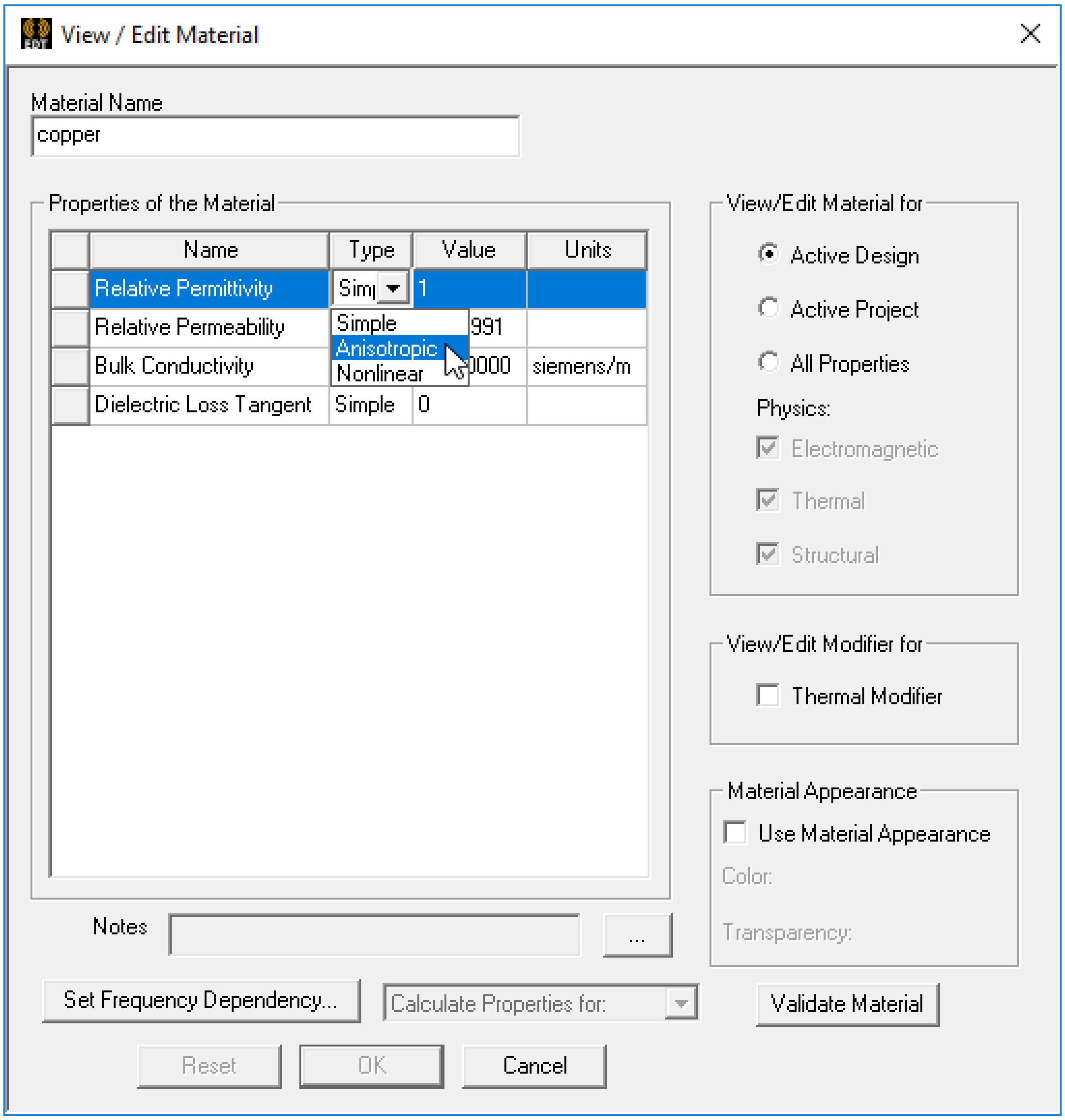


> **Important:**
>
> Anisotropic materials are supported in 2D models but are not available for Q3D.

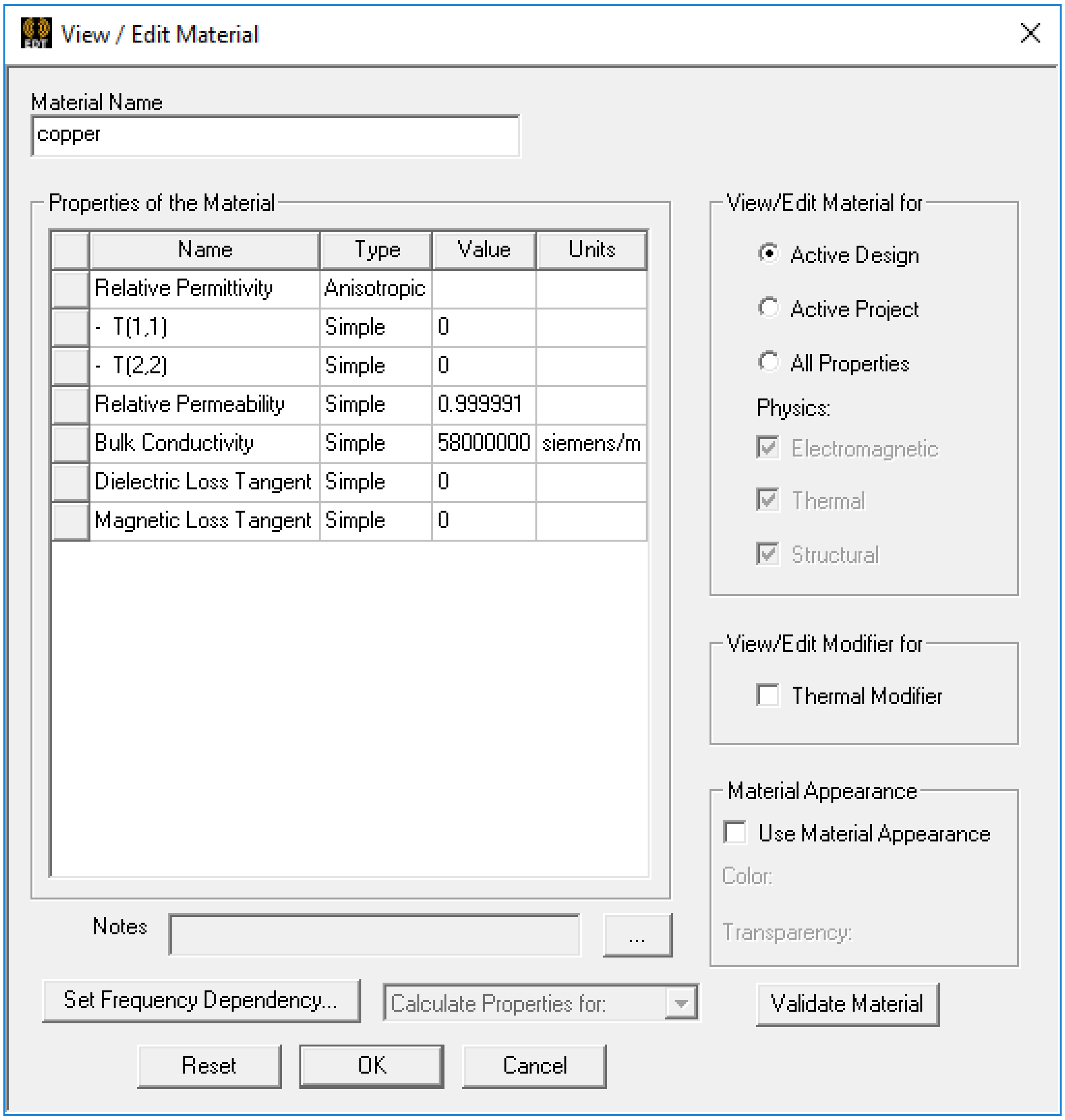
Assigning Anisotropic Tensors

To assign anisotropic tensors to permittivity, electric loss tangent, conductivity, permeability, and magnetic loss tangent:

1. From the **View/Edit Material** window, use the **Type** drop-down menu to select **Anisotropic**.



Two rows, labeled **T(1,1)** and **T(2,2)** appear, as shown below:



2. For each of the new anisotropic property rows, use the **Type** drop-down menu to select either **Simple** or **Nonlinear** (when supported). This setting determines the type of **Value** you can enter.
3. In the **Value** field, enter the permittivity, electric loss tangent, conductivity, permeability, and magnetic loss tangent. along each axis of the material's tensor.
   - For **Relative Permeability** – enter the relative permeability along each axis of the material's permeability tensor. This can be a simple value, a variable, a constant, or a Nonlinear BH Curve.

- For **Relative Permittivity** – enter the material’s relative permittivity along each tensor axis. This can be a simple value or a variable. If the relative permittivity is the same in all directions, use the same Simple values for each axis.
- For **Conductivity** – enter the material’s conductivity along each tensor axis. This can be a simple value or a variable.
- For **Dielectric Loss Tangent** – enter the ratio of the imaginary relative permittivity to the real relative permittivity in one direction. This can be a simple value or a variable.
- For **Magnetic Loss Tangent** – enter the ratio of the imaginary relative permeability to the real relative permeability in one direction. This can be a simple value or a variable.

4. Click **OK** to save the values and return to the **Select Definition** window.

Cylindrical Anisotropic Material Properties: Tensor Conversion from Cylindrical to Cartesian CS

HFSS accepts definitions for spatial-dependent martial properties only using definitions in a Cartesian coordinate system. Cylindrical and Cartesian bases are related as:

$$\begin{bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{bmatrix} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{e}_\rho \\ \hat{e}_\varphi \\ \hat{e}_z \end{bmatrix} = \boldsymbol{M}_{Cyl\to Cart} \begin{bmatrix} \hat{e}_\rho \\ \hat{e}_\varphi \\ \hat{e}_z \end{bmatrix}$$

where M is a transformation matrix.

Tensors, describing material properties, could be transformed accordingly as:

$$\bar{\bar{A}}_{xyz} = \boldsymbol{M}_{Cyl\to Cart} \bar{\bar{A}}_{\rho\varphi z} \boldsymbol{M}^T_{Cyl\to Cart} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{\bar{A}}_{\rho\varphi z} \begin{bmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If we assume only the diagonal components as non-zeros for the tensor in the Cylindrical basis

$$\bar{\bar{A}}_{\rho\varphi z} = \begin{bmatrix} A_{\rho\rho} & 0 & 0 \\ 0 & A_{\varphi\varphi} & 0 \\ 0 & 0 & A_{zz} \end{bmatrix}$$

the resulting tensor in the Cartesian basis could be derived as:

$$\bar{\bar{A}}_{xyz} = \begin{bmatrix} A_{\rho\rho}cos^2\varphi + A_{\varphi\varphi}sin^2\varphi & (A_{\rho\rho} - A_{\varphi\varphi})sin\varphi cos\varphi & 0 \\ (A_{\rho\rho} - A_{\varphi\varphi})sin\varphi cos\varphi & A_{\rho\rho}sin^2\varphi + A_{\varphi\varphi}cos^2\varphi & 0 \\ 0 & 0 & A_{zz} \end{bmatrix}$$

We should define and using X, Y and Z

$$\rho = \sqrt{X^2 + Y^2}$$

$$\varphi = \begin{cases} \operatorname{atan}\left(\frac{Y}{X}\right) + \begin{cases} 180^\circ \operatorname{sign}(Y) & if \quad X < 0 \\ 0 \quad if \; X \geq 0 \end{cases} & if \quad \rho > 0 \\ 0 \quad if \quad \rho = 0 \end{cases} \qquad -180^\circ \leq \varphi \leq 180^\circ$$

Related HFSS Project variables:

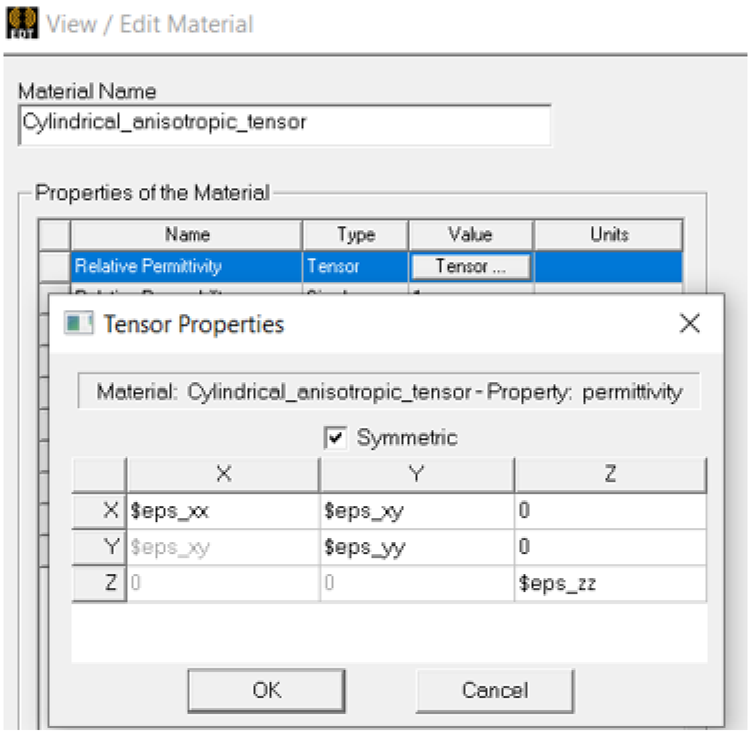
$rho	sqrt(X^2+Y^2)
$phi	if($rho>0, atan(Y/X)+180deg*if(X<0,sgn(Y),0),0)

Note: the material characteristics close to Z-axis could be slightly incorrect because of uncertainty of in the case of =0.

First you create all required Project variables.

Name	Value	Unit	Evaluated V...
$eps_rr	4		4
$eps_pp	8		8
$eps_zz	12		12
$rho	sqrt(X^2+Y^2)		*****
$phi	if($rho>0, atan(Y/X)+180deg*if(X<0,sgn(Y),0),0)		*****
$eps_xx	$eps_rr*cos($phi)^2+$eps_pp*sin($phi)^2		*****
$eps_xy	(eps_rr-eps_pp)*sin($phi)*cos($phi)		*****
$eps_yy	$eps_rr*sin($phi)^2+$eps_pp*cos($phi)^2		*****

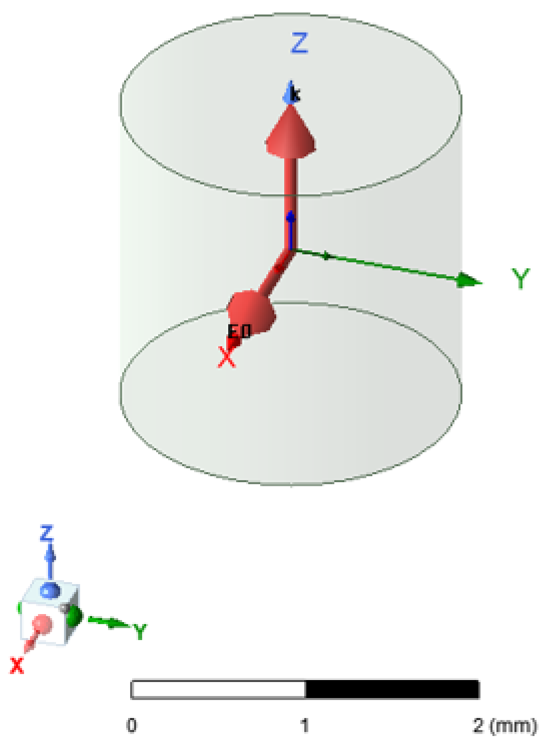
You can then use the variables to create the material.

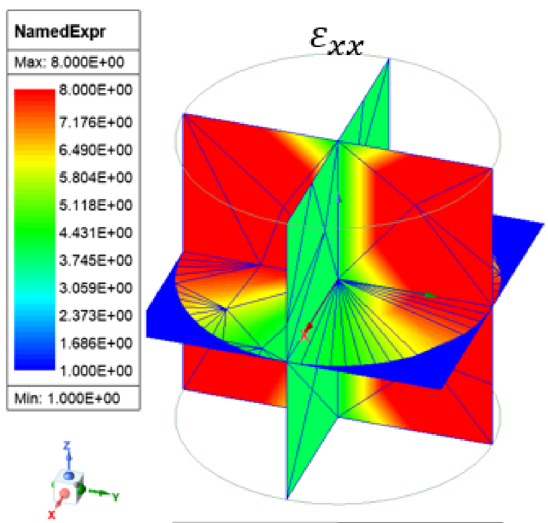


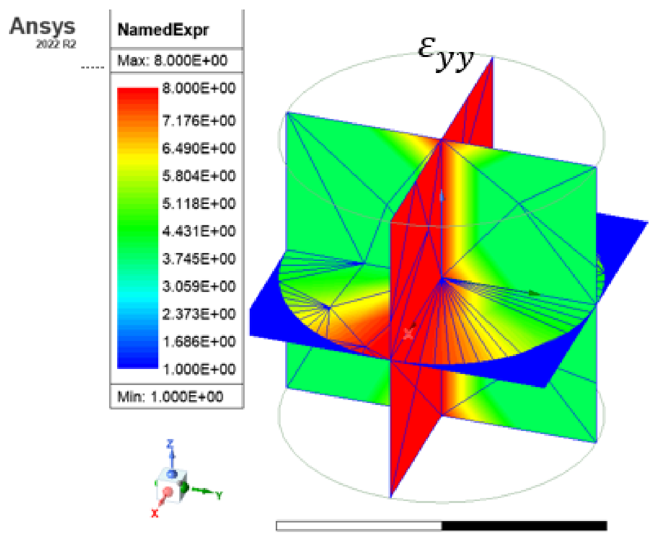
For the test case for cylindrical anisotropic material definition, using converted tensors based on variables:

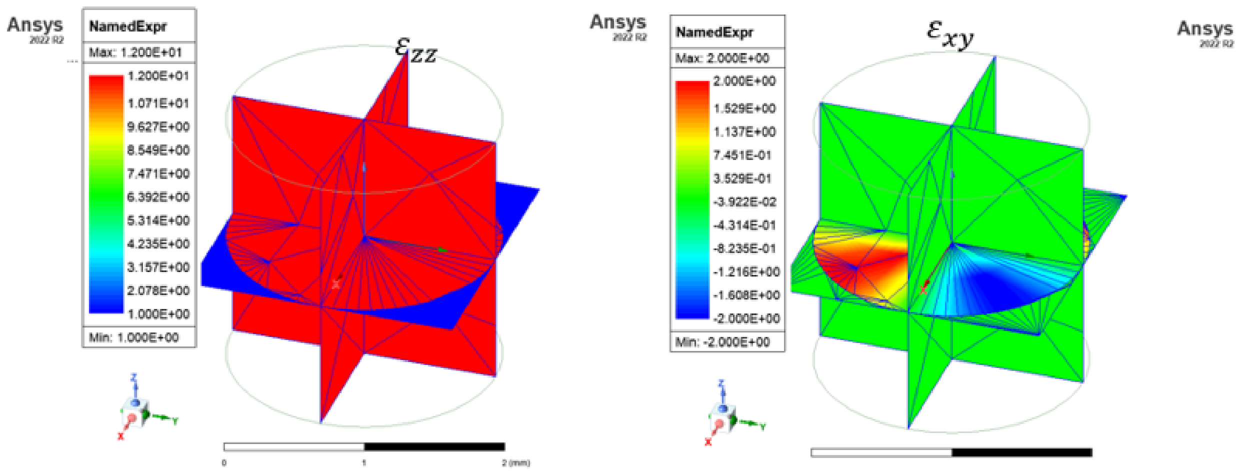
$$\varepsilon_{\rho\rho} = 4 \;\; \varepsilon_{\varphi\varphi} = 8 \;\;\; \varepsilon_{zz} = 12$$

For this problem type, consider that scattering on a cylinder where diameter and height are equal to 0.2 wavelength and a plane wave excitation along Z-axis.


Z
k
Y
E0
X
0
1
2 (mm)


NamedExpr
Max: 8.000E+00
8.000E+00
7.176E+00
6.490E+00
5.804E+00
5.118E+00
4.431E+00
3.745E+00
3.059E+00
2.373E+00
1.686E+00
1.000E+00
Min: 1.000E+00
ε_{xx}


Ansys
2022 R2
NamedExpr
Max: 8.000E+00
8.000E+00
7.176E+00
6.490E+00
5.804E+00
5.118E+00
4.431E+00
3.745E+00
3.059E+00
2.373E+00
1.686E+00
1.000E+00
Min: 1.000E+00
ε_{yy}



Spherical Anisotropic Material Properties: Tensor Conversion from Spherical to Cartesian CS

HFSS accepts definitions for spatial-dependent martial properties only using definitions in a Cartesian coordinate system. Spherical and Cartesian bases are related as

$$\begin{bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{bmatrix} = \begin{bmatrix} sin\theta cos\varphi & cos\theta sin\varphi & -sin\varphi \\ sin\theta sin\varphi & cos\theta cos\varphi & cos\varphi \\ cos\theta & -sin\theta & 1 \end{bmatrix} \begin{bmatrix} \hat{e}_\rho \\ \hat{e}_\theta \\ \hat{e}_\varphi \end{bmatrix} = \boldsymbol{M}_{Sph\to Cart} \begin{bmatrix} \hat{e}_\rho \\ \hat{e}_\theta \\ \hat{e}_\varphi \end{bmatrix}$$

where **M** is a transformation matrix

Tensors, describing material properties, could be transformed accordingly as

$$\bar{\bar{A}}_{xyz} = \boldsymbol{M}_{Sph\to Cart} \bar{\bar{A}}_{\rho\theta\varphi} \boldsymbol{M}^T_{Sph\to Cart}$$

We should define ρ,θ , and φ using X, Y and Z

$$\rho = \sqrt{X^2 + Y^2 + Z^2} \qquad \rho_{XY} = \sqrt{X^2 + Y^2}$$

$$\varphi = \begin{cases} \operatorname{atan}\left(\dfrac{Y}{X}\right) + \begin{cases} 180° \operatorname{sign}(Y) & if \quad X < 0 \\ 0 \quad if \; X \geq 0 \end{cases} & if \quad \rho_{XY} > 0 \\ 0 \quad if \quad \rho_{XY} = 0 \end{cases}$$

$$\theta = \begin{cases} \operatorname{acos}\left(\dfrac{Z}{\rho}\right) & if \quad \rho > 0 \\ 0 \quad if \quad \rho = 0 \end{cases}$$

$$0 \leq \theta \leq 180°$$
$$-180° \leq \varphi \leq 180°$$

Related HFSS Project variables

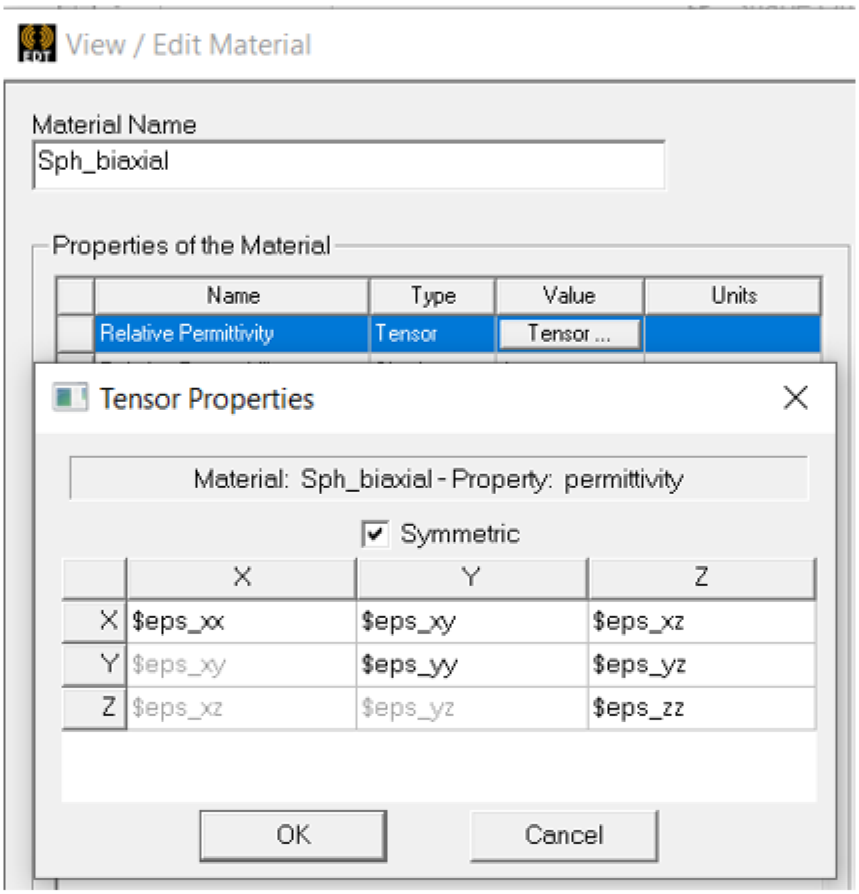
$rho	sqrt(X^2+Y^2+Z^2)
$rho_xy	sqrt(X^2+Y^2)
$theta	if($rho>0, acos(Z/$rho), 0)
$phi	if($rho_xy>0, atan(Y/X)+180deg*if(X<0,sgn(Y),0),0)

Note: the material characteristics close to the Z-axis could be slightly incorrect because of uncertainty of φ in the case of ρ_{xy}=0.

First you must create the necessary project variables.

Name	Value
$eps_rr	4
$eps_tt	8
$eps_pp	12
$rho	sqrt(X^2+Y^2+Z^2)
$rho_xy	sqrt(X^2+Y^2)
$theta	if($rho>0, acos(Z/$rho), 0)
$phi	if($rho_xy>0, atan(Y/X)+180deg*if(X<0,sgn(Y),0),0)
$eps_xx	($eps_rr*sin($theta)^2+$eps_tt*cos($theta)^2)*cos($phi)^2+$eps_pp*sin($phi)^2
$eps_xy	($eps_rr*sin($theta)^2+$eps_tt*cos($theta)^2-$eps_pp)*sin($phi)*cos($phi)
$eps_xz	(eps_rr-eps_tt)*sin($theta)*cos($theta)*cos($phi)
$eps_yy	($eps_rr*sin($theta)^2+$eps_tt*cos($theta)^2)*sin($phi)^2+$eps_pp*cos($phi)^2
$eps_yz	(eps_rr-eps_tt)*sin($theta)*cos($theta)*sin($phi)
$eps_zz	$eps_rr*cos($theta)^2+$eps_tt*sin($theta)^2

You can then use the variables to create the material.

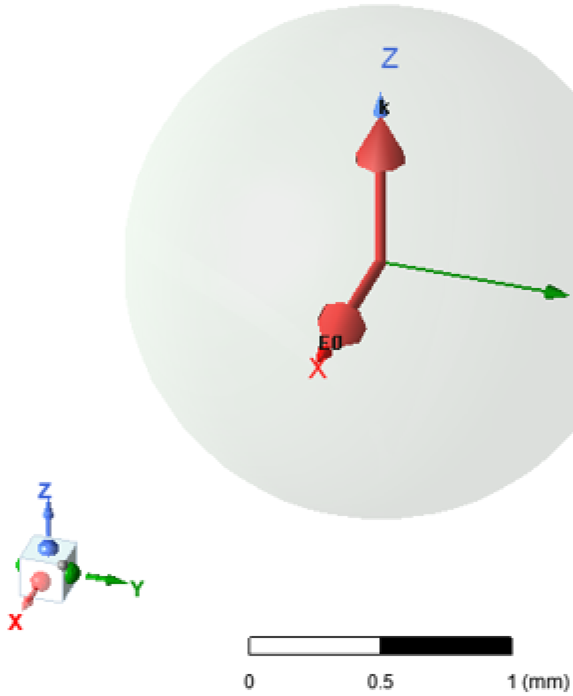


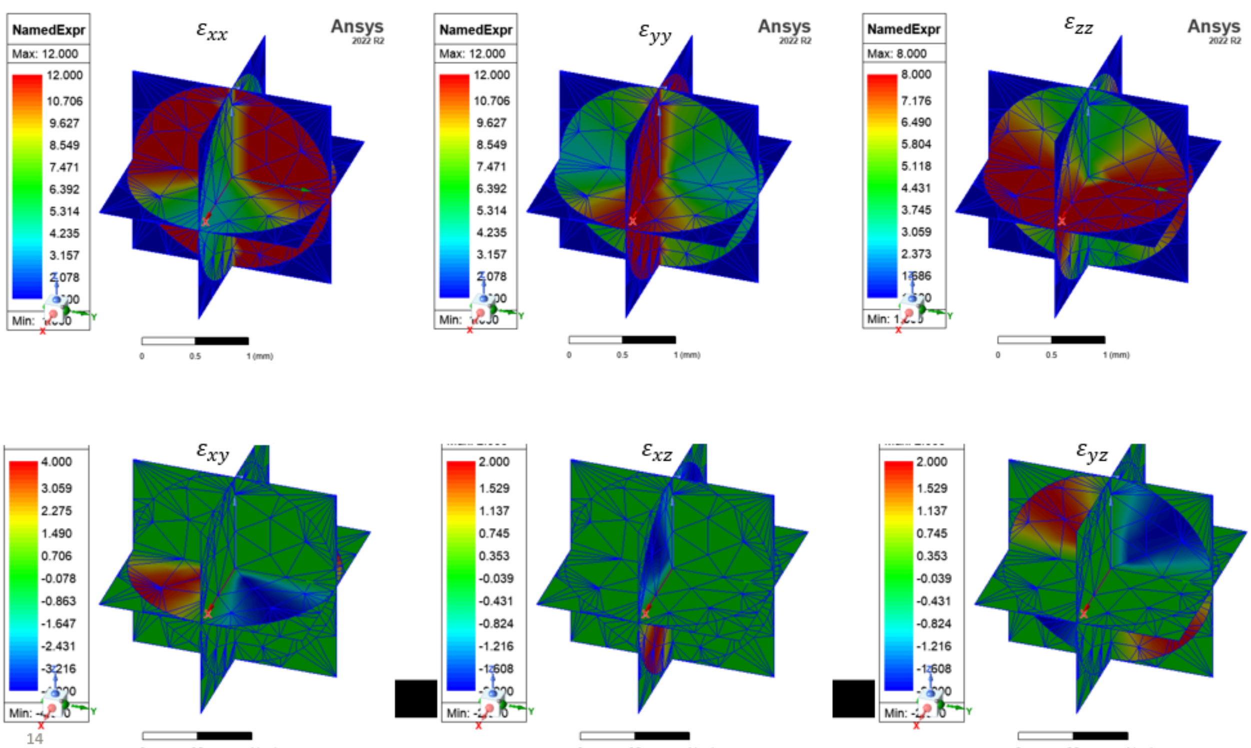
For the test case:

$$\varepsilon_{\rho\rho} = 4 \ \ \varepsilon_{\theta\theta} = 8 \ \ \varepsilon_{\varphi\varphi} = 12$$

Problem type: scattering on a sphere(diameter is equal to 0.2 wavelength).

Plane wave excitation along Z-axis


Z
k
Y
E0
X
Z
Y
X
0
0.5
1 (mm)


NamedExpr
Max: 12.000
ε_{xx}
Ansys
2022 R2
12.000
10.706
9.627
8.549
7.471
6.392
5.314
4.235
3.157
2.078
NamedExpr
Max: 12.000
ε_{yy}
Ansys
2022 R2
12.000
10.706
9.627
8.549
7.471
6.392
5.314
4.235
3.157
2.078
NamedExpr
Max: 8.000
ε_{zz}
Ansys
2022 R2
8.000
7.176
6.490
5.804
5.118
4.431
3.745
3.059
2.373
1.686
ε_{xy}
4.000
3.059
2.275
1.490
0.706
-0.078
-0.863
-1.647
-2.431
14
ε_{xz}
2.000
1.529
1.137
0.745
0.353
-0.039
-0.431
-0.824
-1.216
ε_{yz}
2.000
1.529
1.137
0.745
0.353
-0.039
-0.431
-0.824
-1.216

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

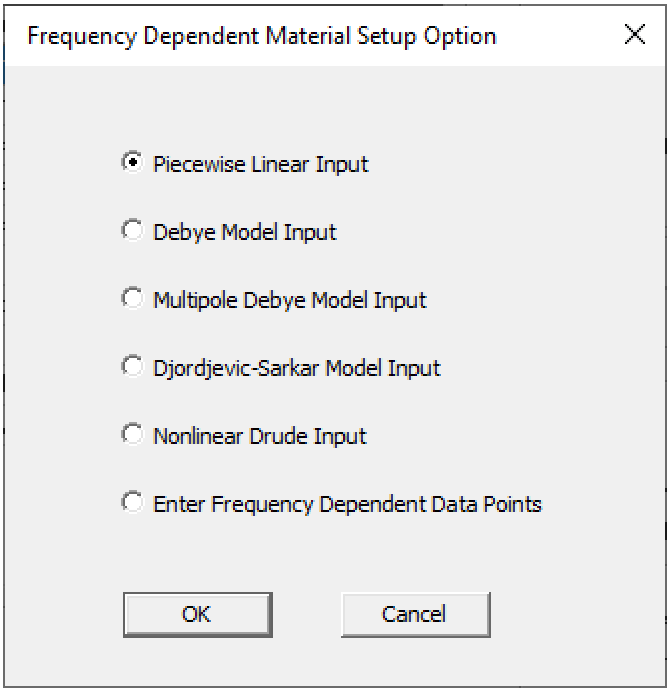
HFSS provides several frequency-dependent material models. The Piecewise Linear and Frequency Dependent Data Points models apply to both the electric and magnetic properties of the material. However, they do not guarantee that the material satisfies causality conditions, and so they should only be used 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, HFSS linearly interpolates the material properties; above and below the corner frequencies, HFSS 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. HFSS 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.

- **Nonlinear Drude Input** – Drude materials are now supported by the transient solvers, producing plasma density data within the material domains. This means that plasma density plots can be produced.
- **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. HFSS automatically creates a dataset for each material property. Based on a varying property's dataset, HFSS 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.

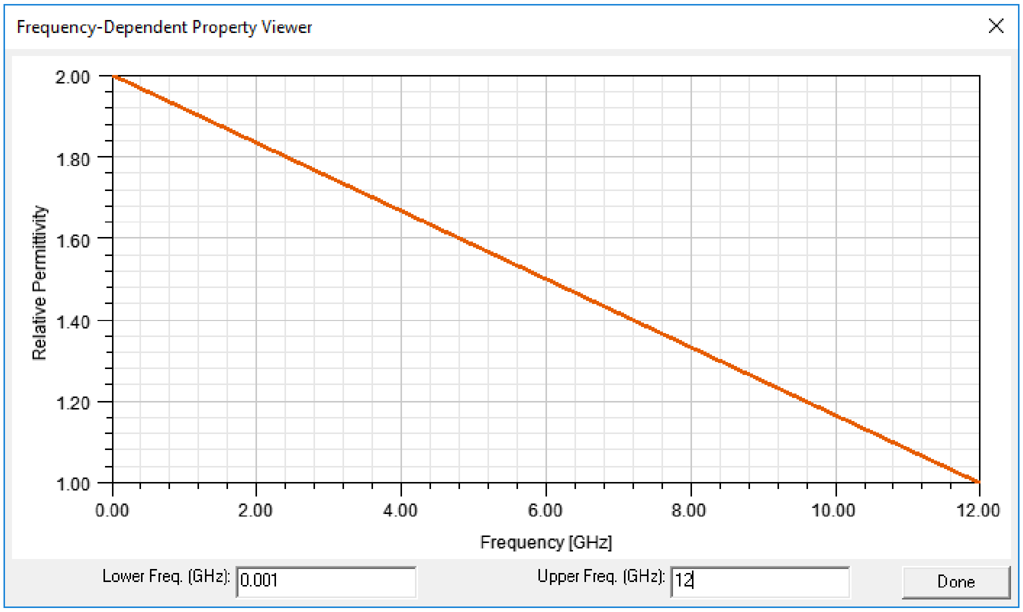
Frequency Dependence Visualization

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.



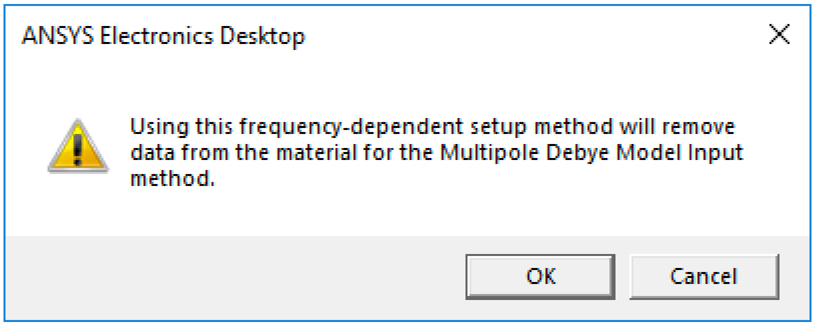
The window contains boxes where you can edit the frequency range for the plot. If the property was set by one of the nonlinear frequency setup methods, the frequency range will be derived from the data for that method, and edits to the lower/upper frequencies are not saved. Otherwise, the frequency range 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.

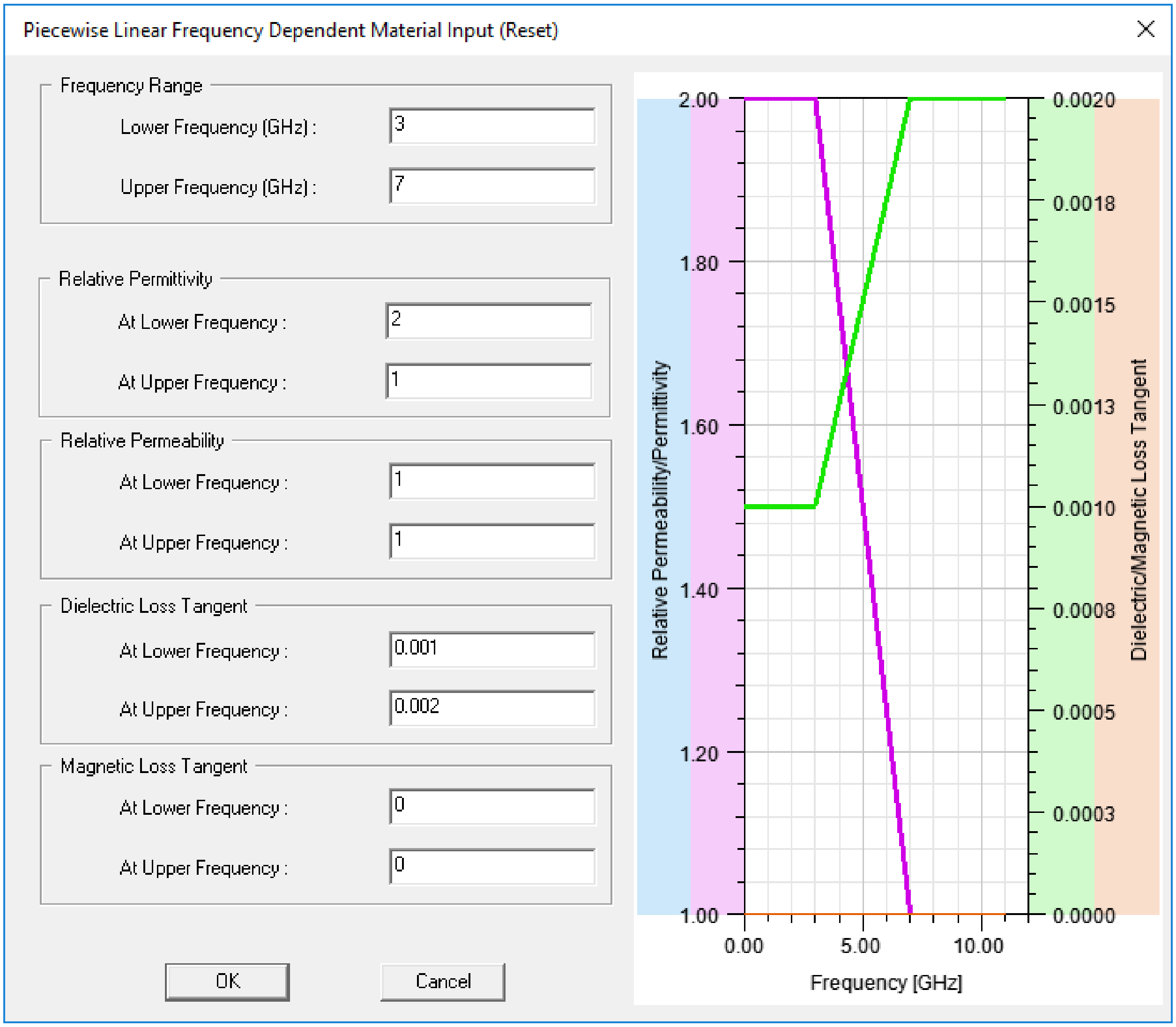
You can choose **OK** to continue with the edit and remove the invalidated setup data, or choose **Cancel** to discard any changes.



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** – HFSS 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:

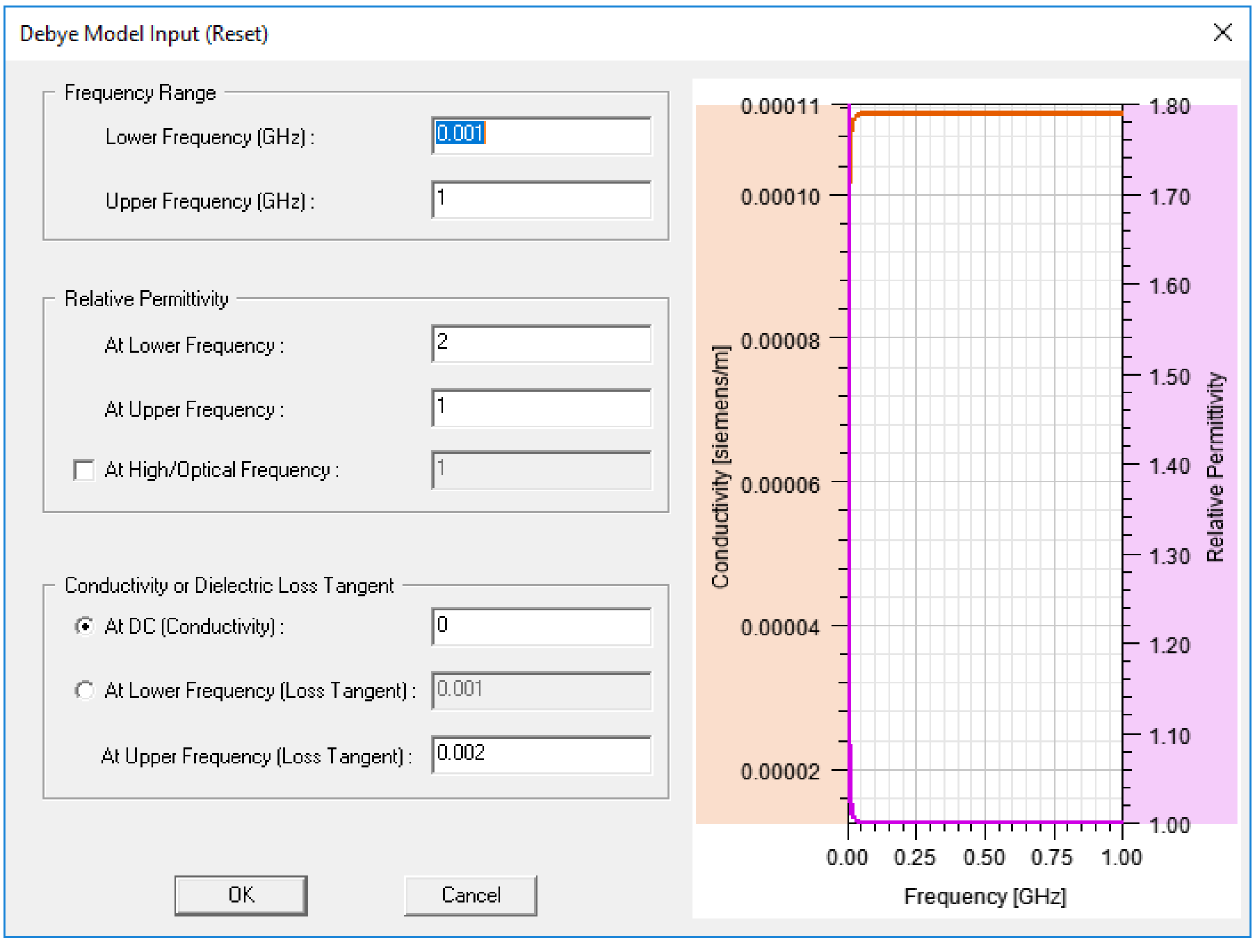
Neither the piecewise or the loss models ask for frequency dependent conductivity because there the constant sigma represents the DC loss and the frequency dependent loss tangent represents the polarization losses.

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** – HFSS 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, HFSS can interpolate the property’s values at the desired frequencies during solution generation.

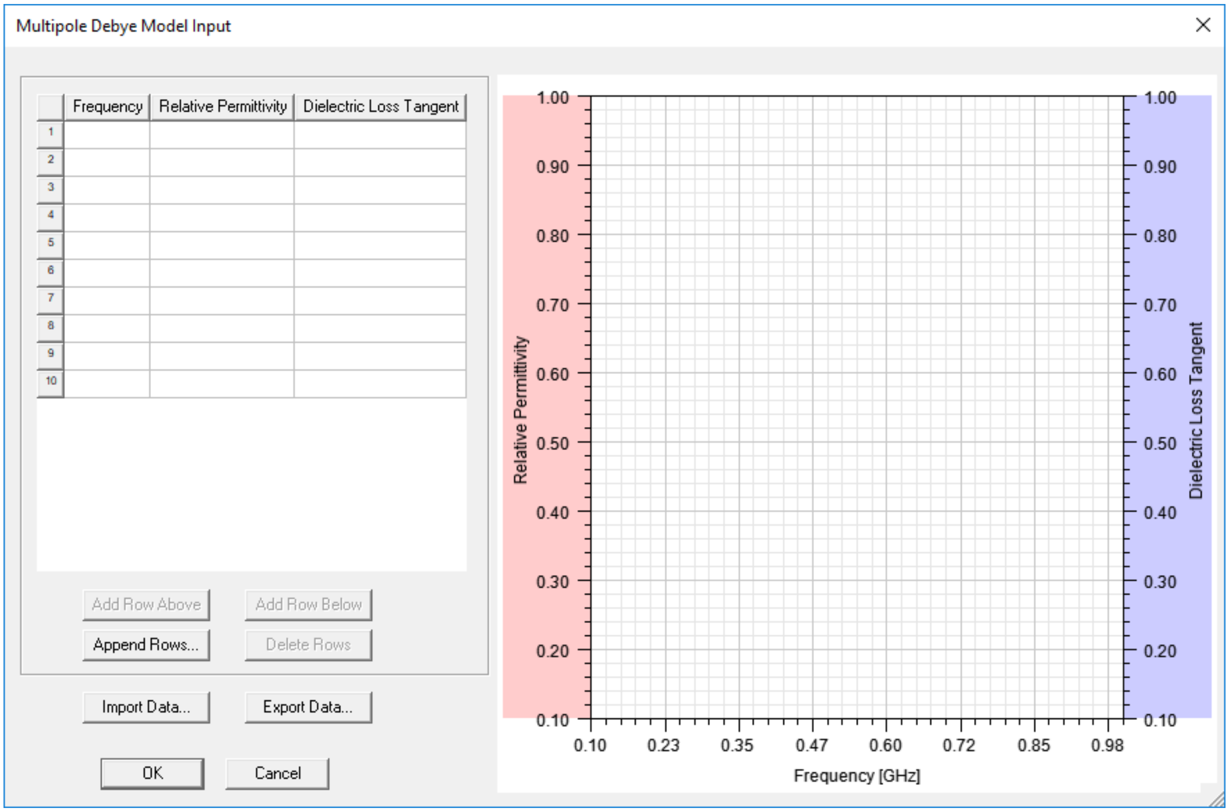
To modify the dataset with additional points, see Editing Datasets.

> **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.

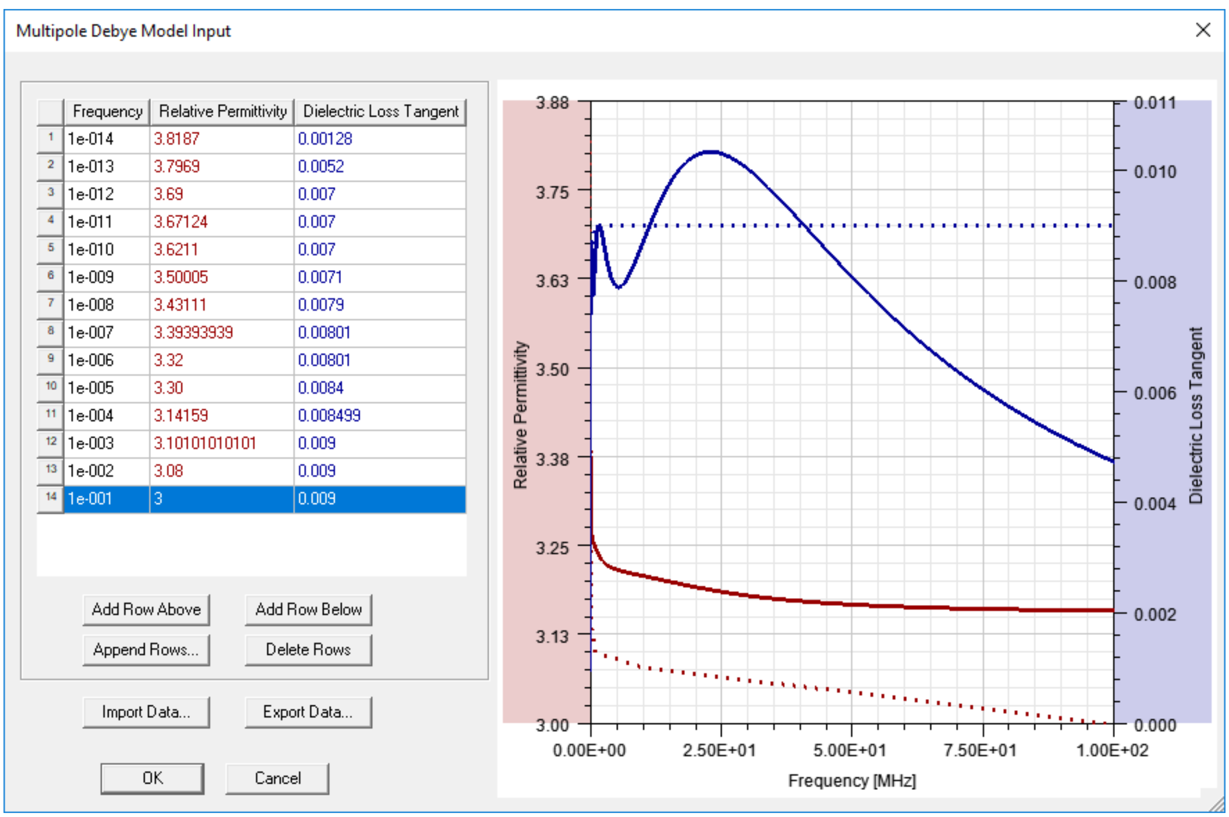
Enter data values manually, or click **Import Data** to import data from a *.tab file.

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.



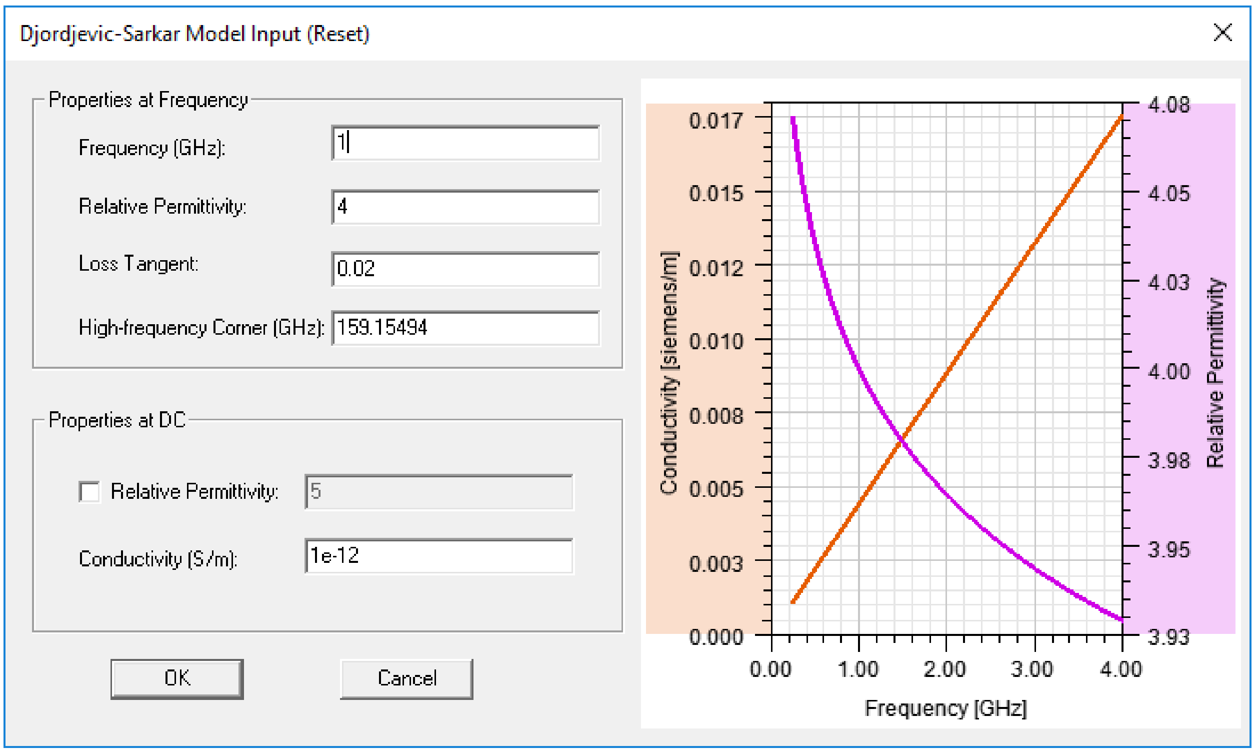
Double-click any plot element to change its properties (color, units, scaling, etc.).

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:

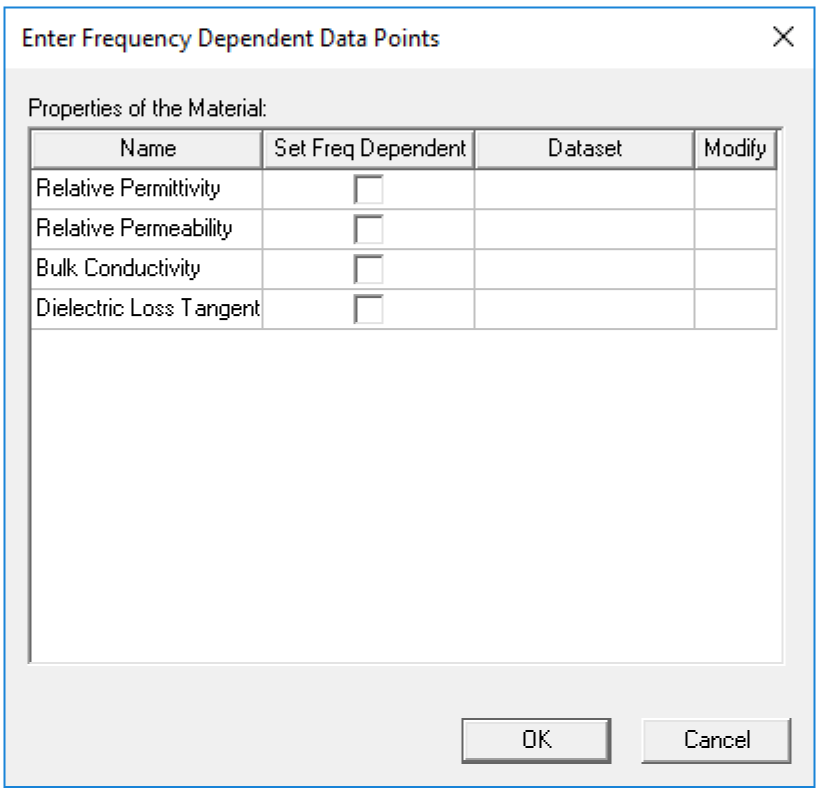
- **Relative Permittivity** – select the check box if you wish to specify relative permittivity.
- **Conductivity**

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

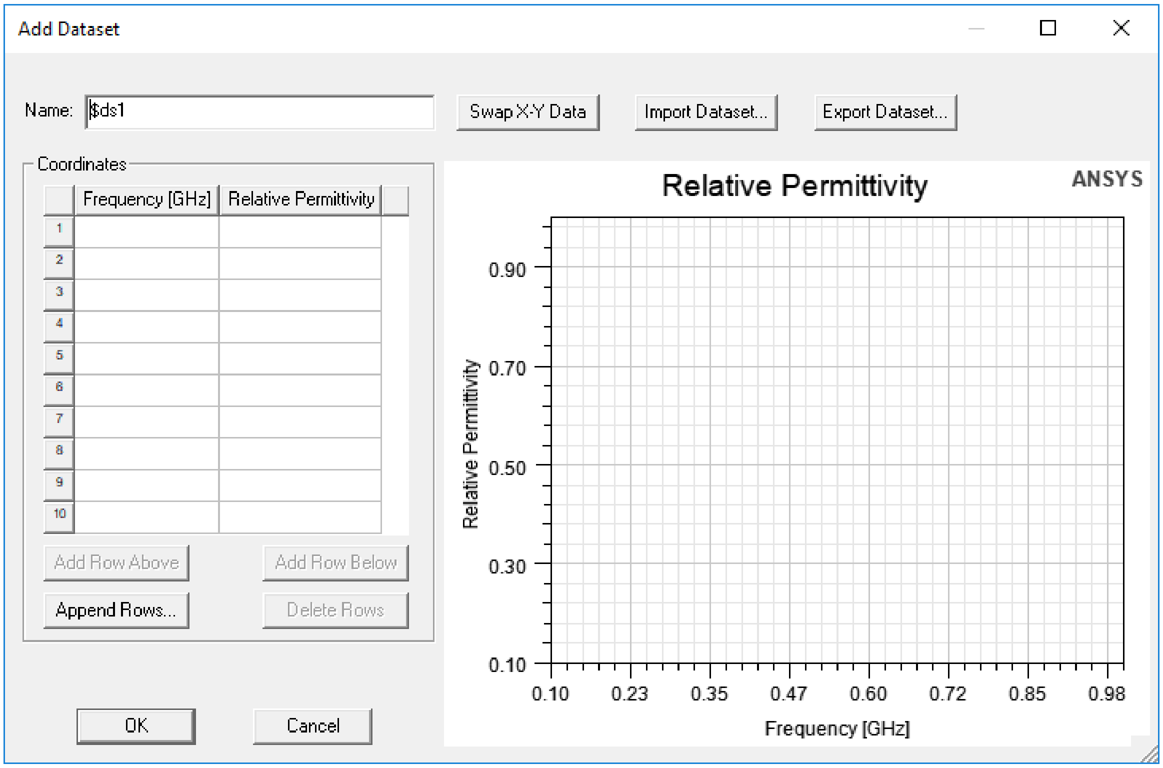
When you select **Enter Frequency Dependent Data Points** as the model for the frequency dependent material property, the **Enter Frequency Dependent Data Points** window appears.



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.

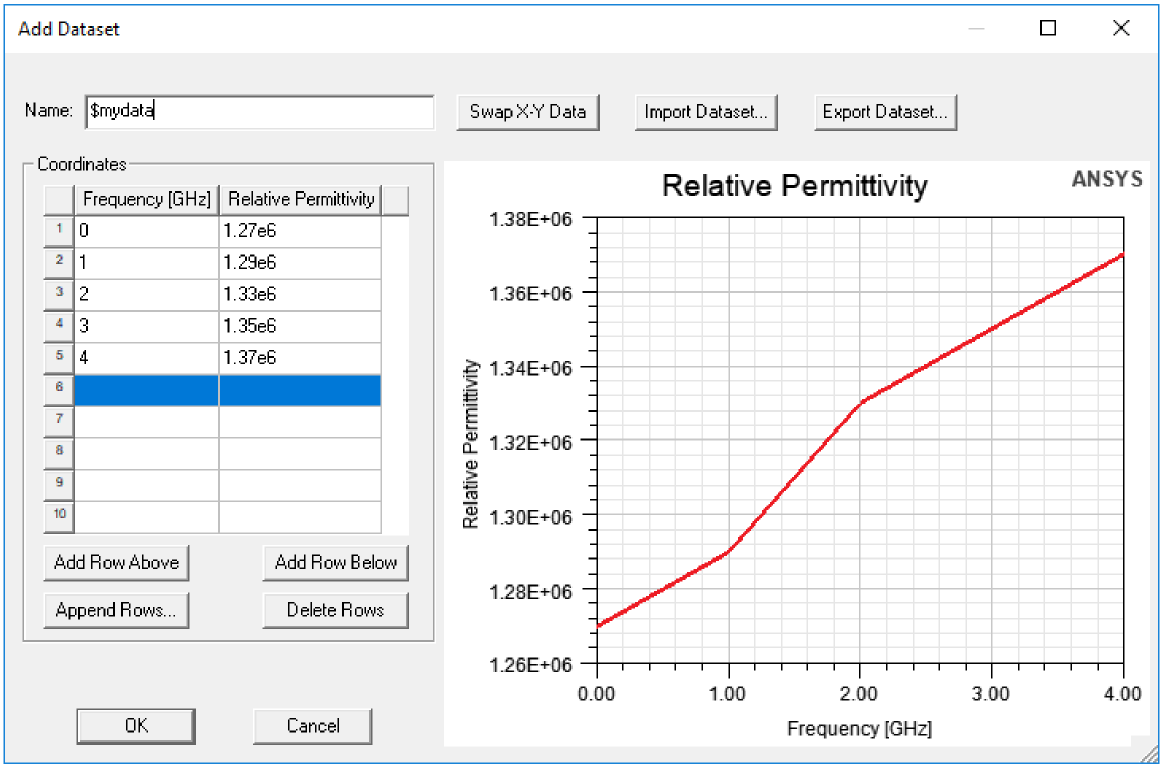


This window allows you to specify the following:

- **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.

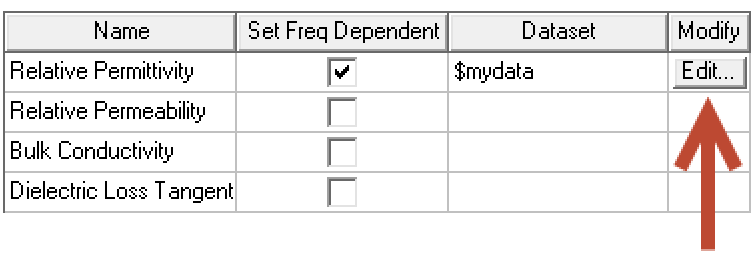
You can also:

- 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**.



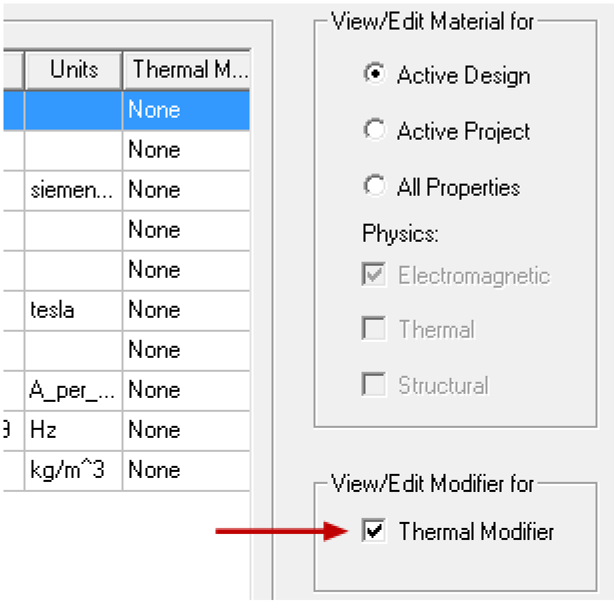
4. Click **OK** again to return to the **View/Edit Material** window.

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:

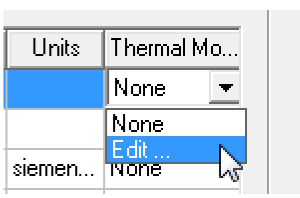
1. In the **View/Edit Material** dialog box, you must enable the **Thermal Modifier** check box:



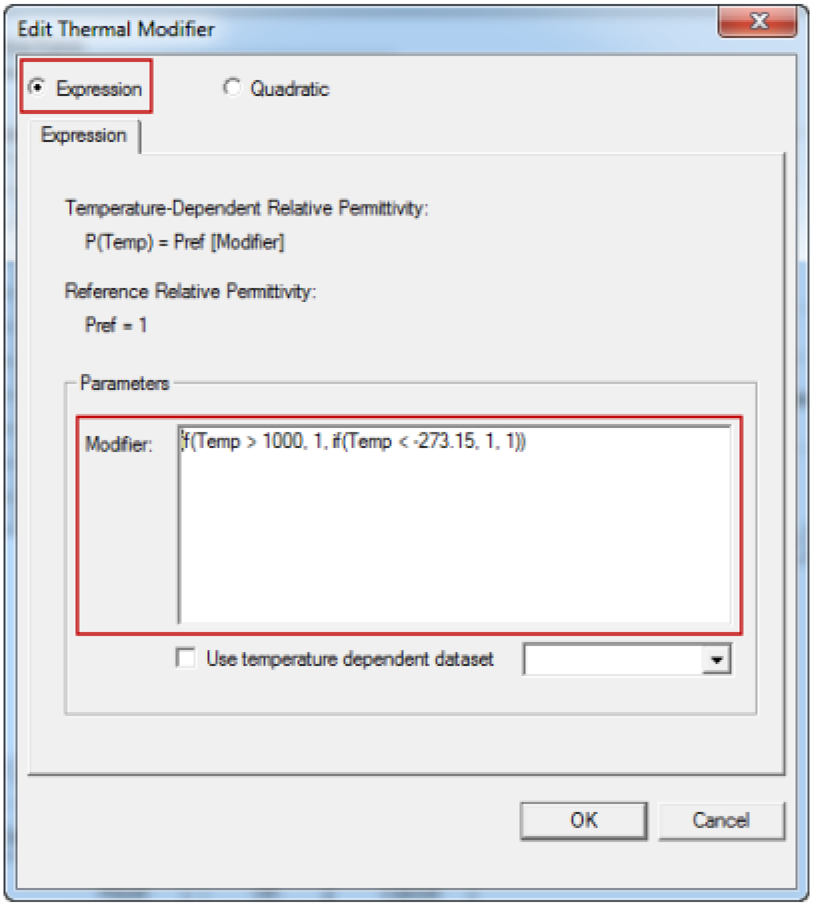
This option causes the properties table to expand to include a Thermal Modifier column.

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:



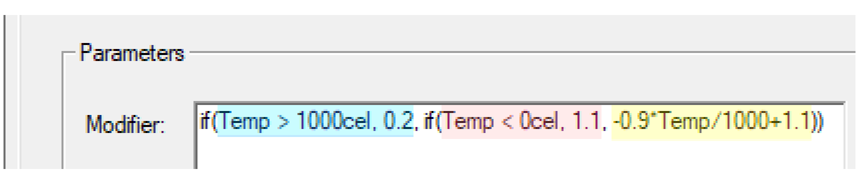
The **Edit Thermal Modifier** dialog box appears with the **Expression** option selected.



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.

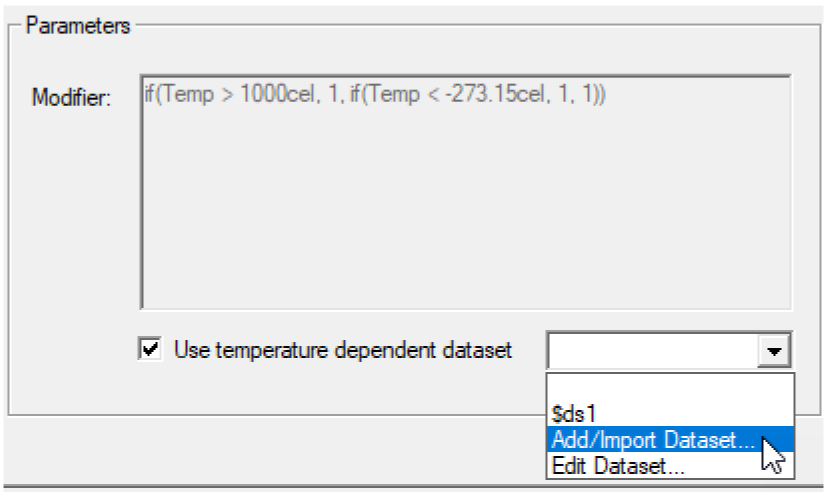
  The following example defines a multiplier that varies linearly from 1.1 (at 0° C) to 0.2 (at 1000° C). The multiplier is constant at 1.1 for temperatures below 0° C and constant at 0.2 for temperature above 1000° C:



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.

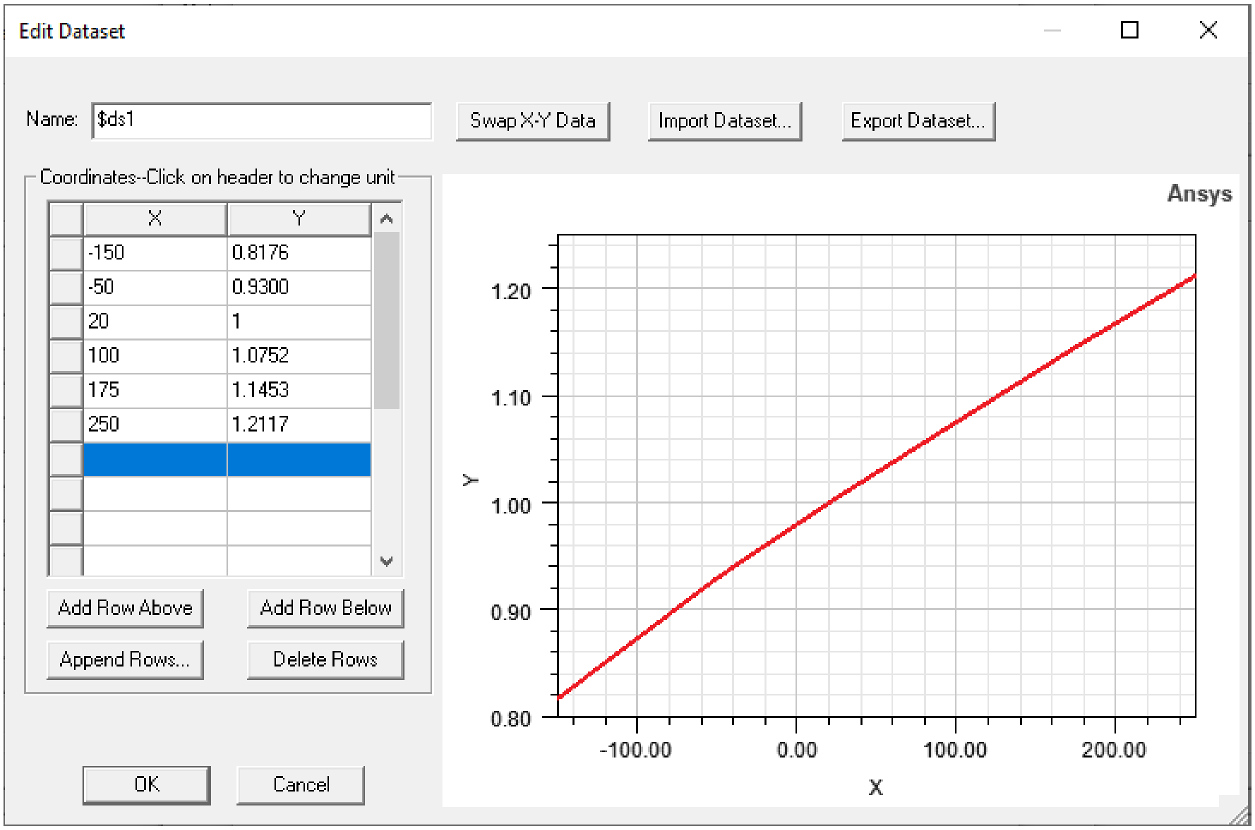
- Check **Use temperature dependent data set** to disable the Modifier text box. You can then use the drop-down menu at the bottom-right corner of the dialog box to select an existing project dataset or to create a new one by choosing **Add/Import Dataset**.



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.

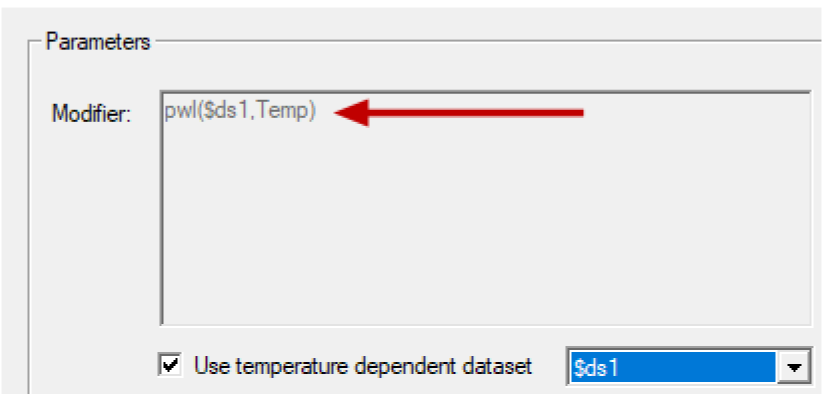
Important:

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:



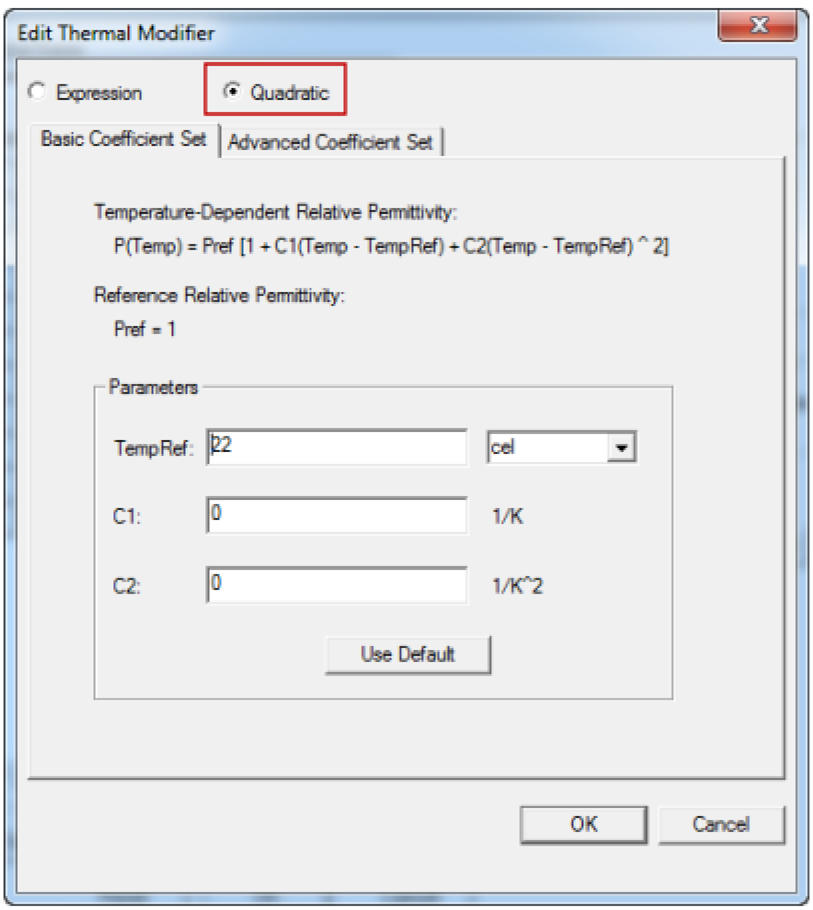
The syntax of the dataset expression is explained as follows:

- **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**.

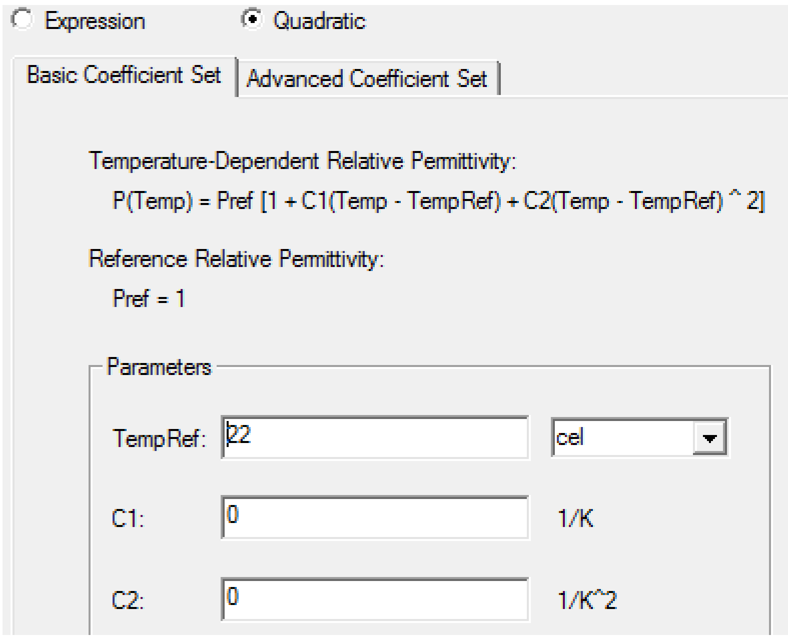
3. Alternatively, select the **Quadratic** option to display the tabs for **Basic Coefficient Set** and **Advanced Coefficient Set** (as shown below):



- 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:

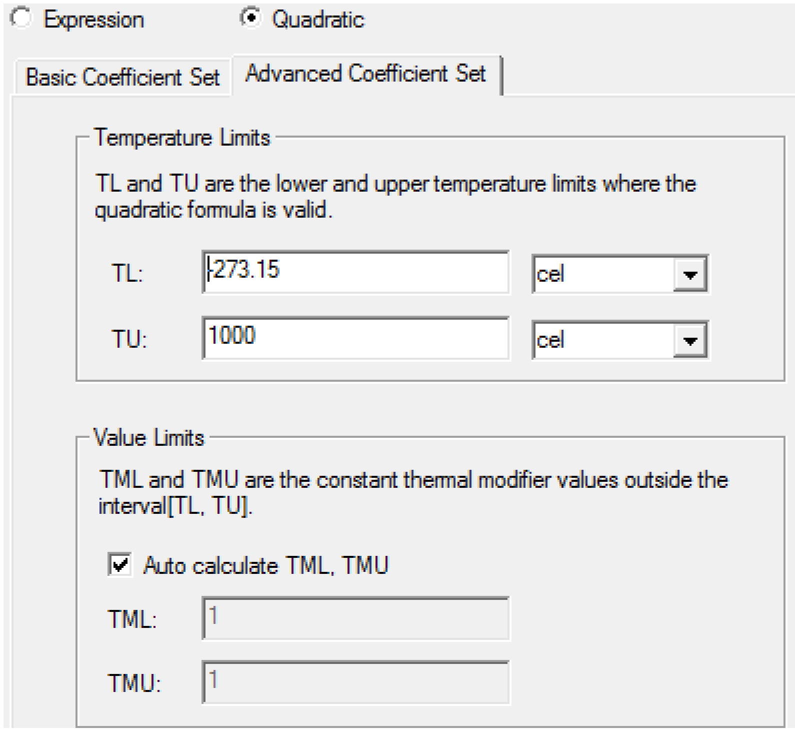
$$P(Temp) = Pref[1+ C1(Temp - TempRef) + C2(Temp - TempRef)^2]$$

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.

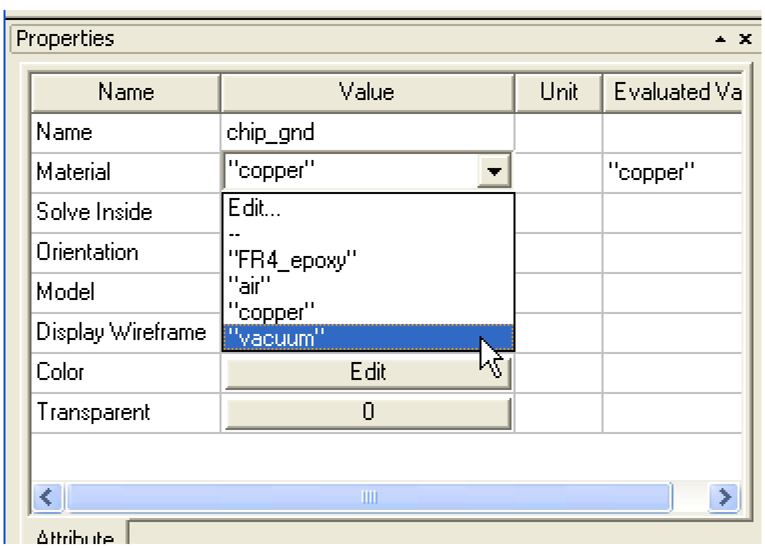
Defining Functional Material Properties

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

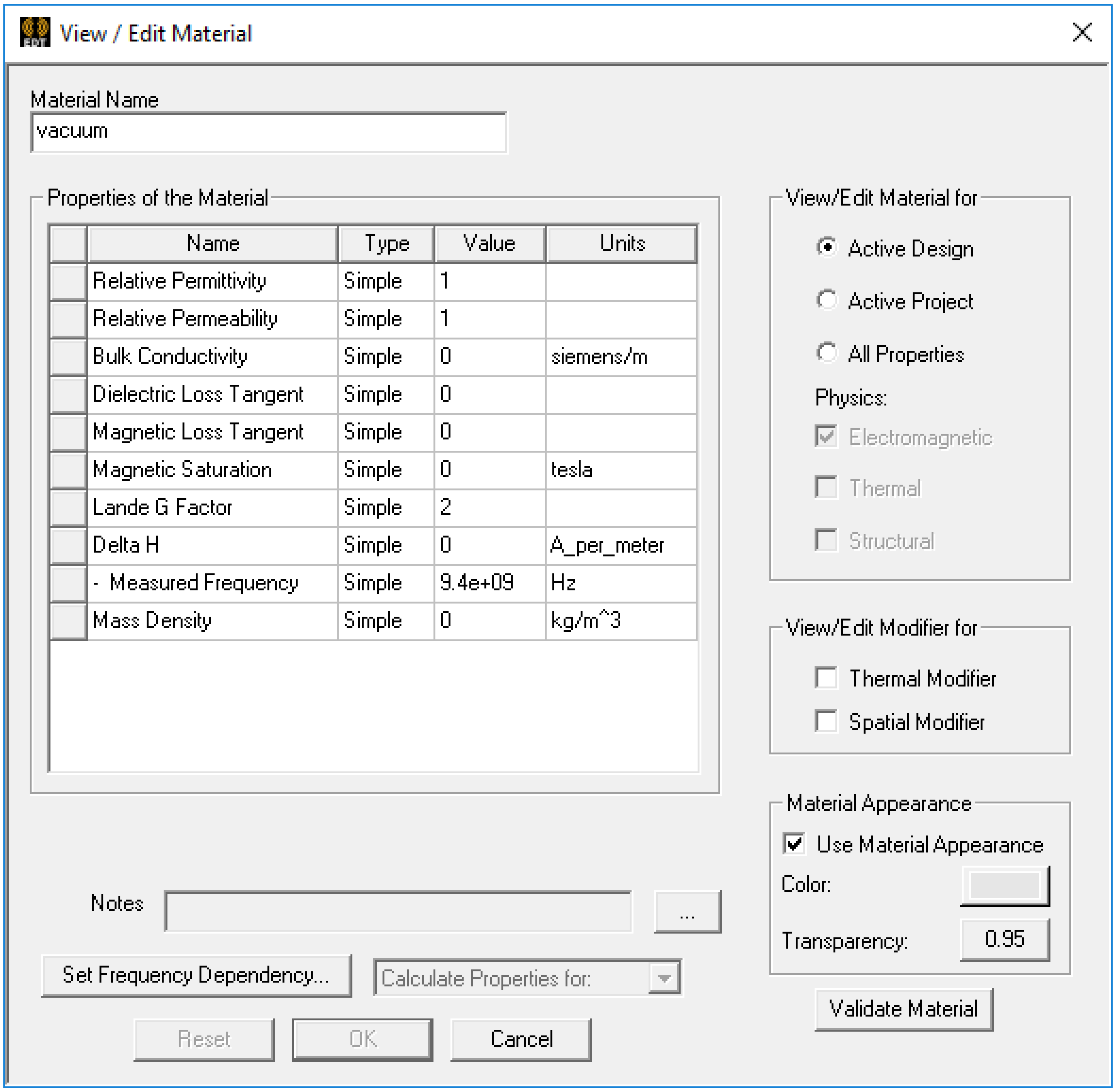
The Properties dialog for each object includes a materials property. If you click on the current material property you see a drop down list that includes an *Edit* command and a list of materials in the current project. You can select from the list of current materials to assign the selected material 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.

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** , **Anisotropic**, **Vector** and **Vector Mag**, or for Relative Permeability, **Non-linear**, as required for that property.

      For **Simple**, you provide a value or variable.

      For **Anisotropic**, you provide tensor values.

      For **Vector**, you provide a **Vector Mag**.

      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:

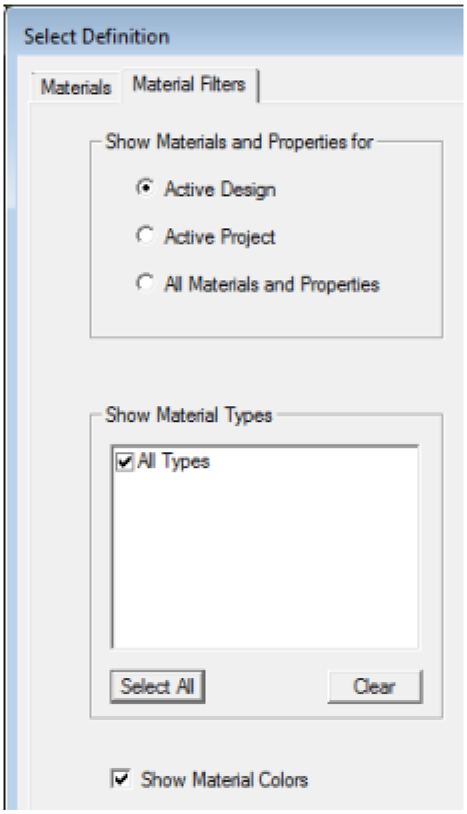
- Click the column heading by which you want to order the materials.

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.

Working with Material Libraries

There are three different materials libraries in HFSS, Q3D, Icepak, and Mechanical: 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.

7 - 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 Circuit. 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 Circuit 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.

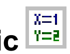
Setting Up a Parametric Analysis

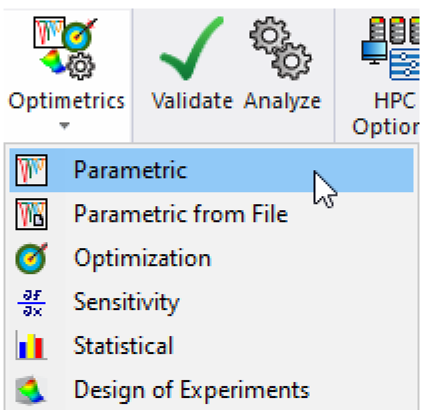
A *parametric setup* specifies all of the design variations that Optimetrics drives the software to solve. A parametric setup is made up of one or more variable sweep definitions, which are a set of variable values within a range that you want the software to solve when you run the parametric setup.

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 **Circuit**> **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.

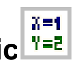
   After you define a parametric sweep, a shortcut menu becomes available when you right-click the setup name in the Project Manager.

> **Note:**
>
> Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Adding a 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 **Circuit** > **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.

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, Circuit 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:

      - Click **Output Variables**.

        The **Output Variables** dialog box appears.

      - Add the expression you want to evaluate, and then click **Done**.

        The recently created output variable appears in the **Quantity** list.

      - Click a new output variable in the **Quantity** 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 **Setup Calculations**.

   The **Add/Edit/Calculation** dialog box appears.

3. Select the **Calculation Range** tab.
4. In the **Variable** list, click an intrinsic variable.

   Single Value is selected by default.

5. In the **Value** box, click a value at which the solution quantity will be extracted.
6. Click **Update** and then click **Edit**.

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.

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:

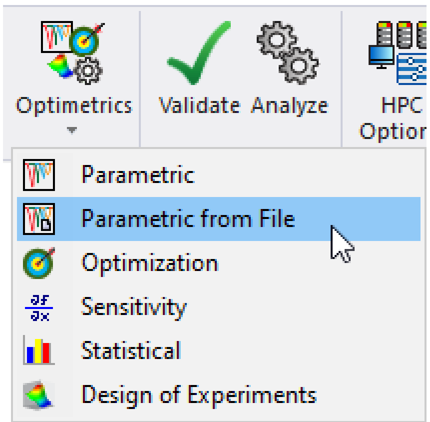
   - Right-click the progress window, and select **Show Subtask Progress Bars**.

     To hide the child progress bars:

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

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 **Circuit** > **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 [ ].

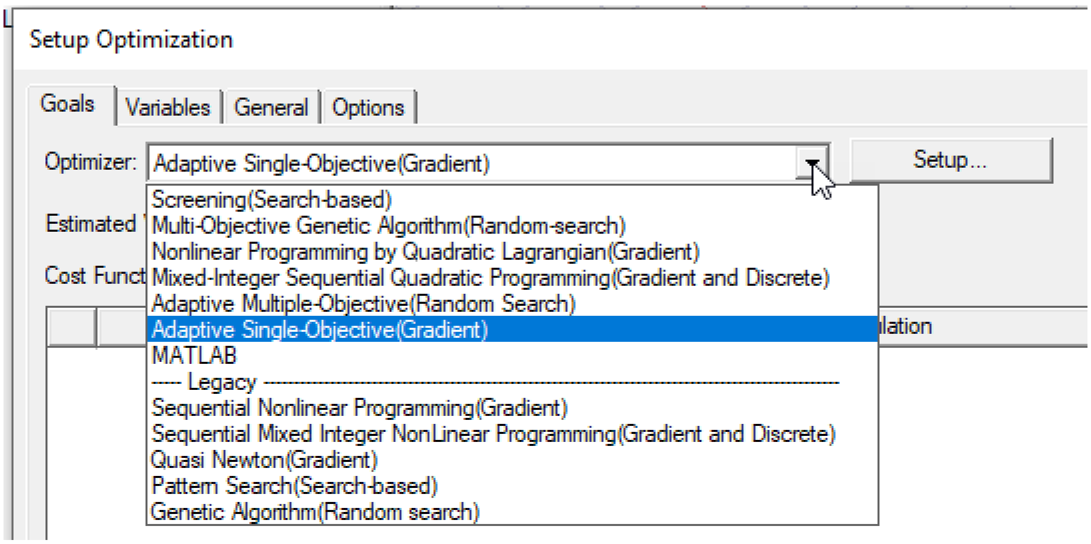
After you have imported a valid file, the detailed information regarding sweep values can be found at the lower portion of the Parametric **Setup Sweep Analysis** dialog**Sweep Definitions** tab (Operation/Description panel). It is treated as: first row as single point. then each time a new sweep point is added.

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.

- Nonliner 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)

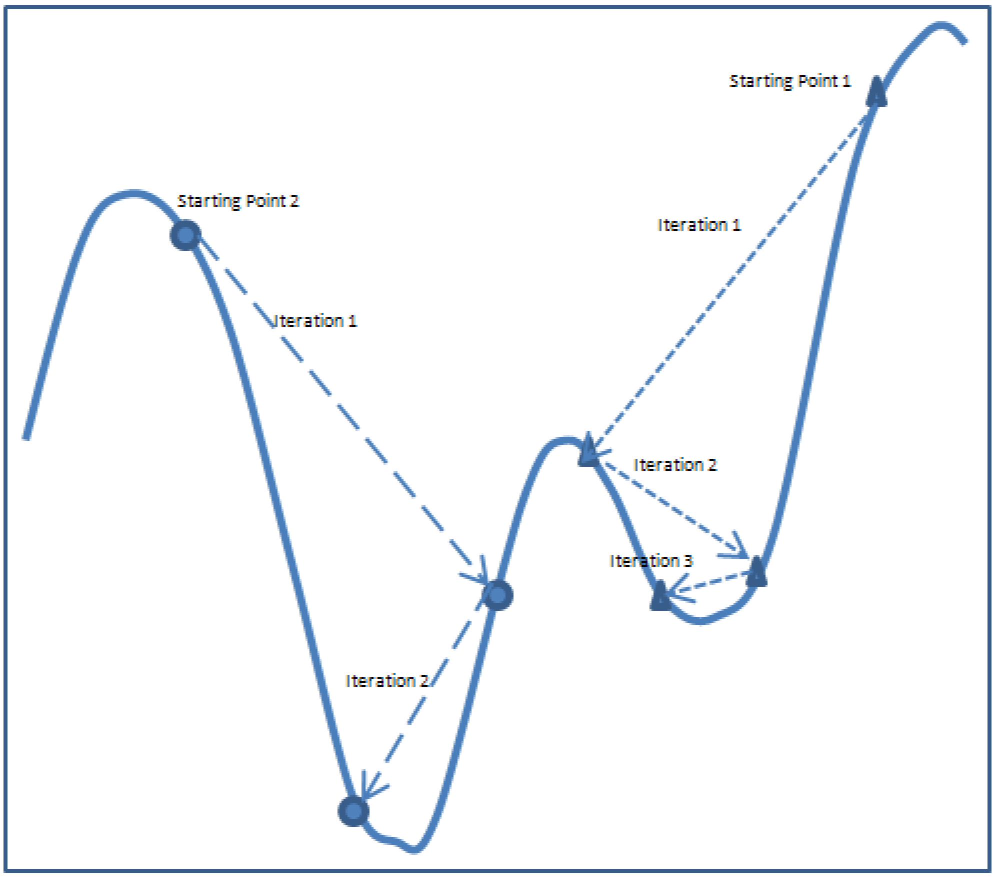
Quasi Newton (Gradient)

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.

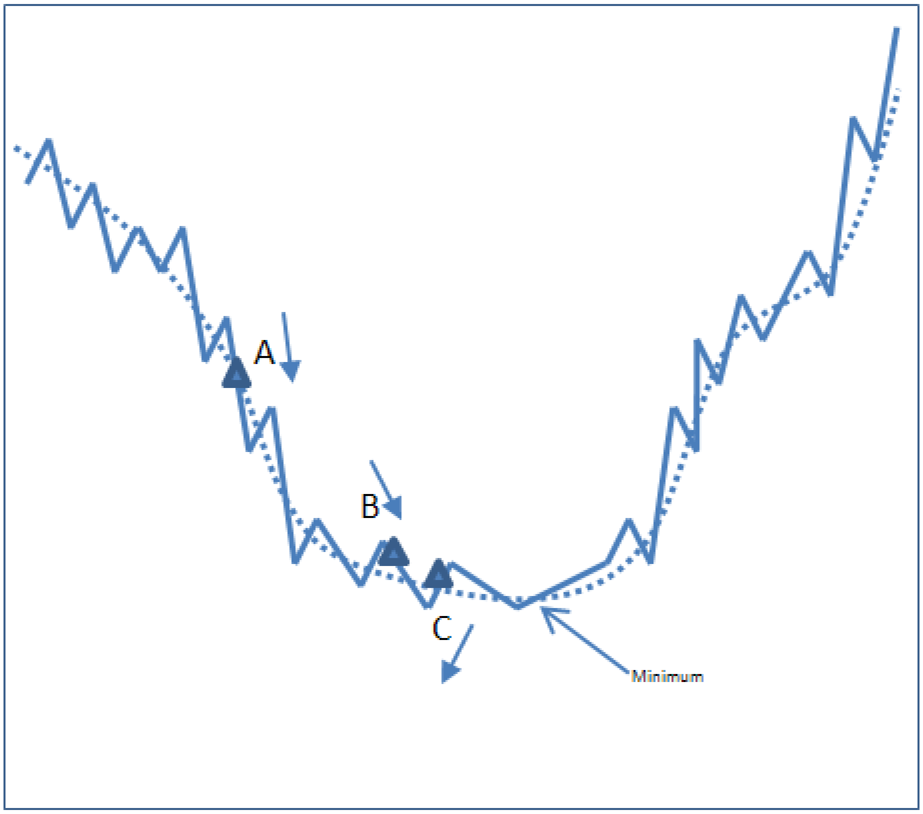
In the Quasi-Newton optimization procedure, the gradients and Hessian are calculated numerically. Essentially, the change in x and the change in the gradient are used to estimate the Hessian for the next iteration. The ratio of the change in cost to the change in the values of x provides the gradient, 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.

Pattern Search (Search-Based)

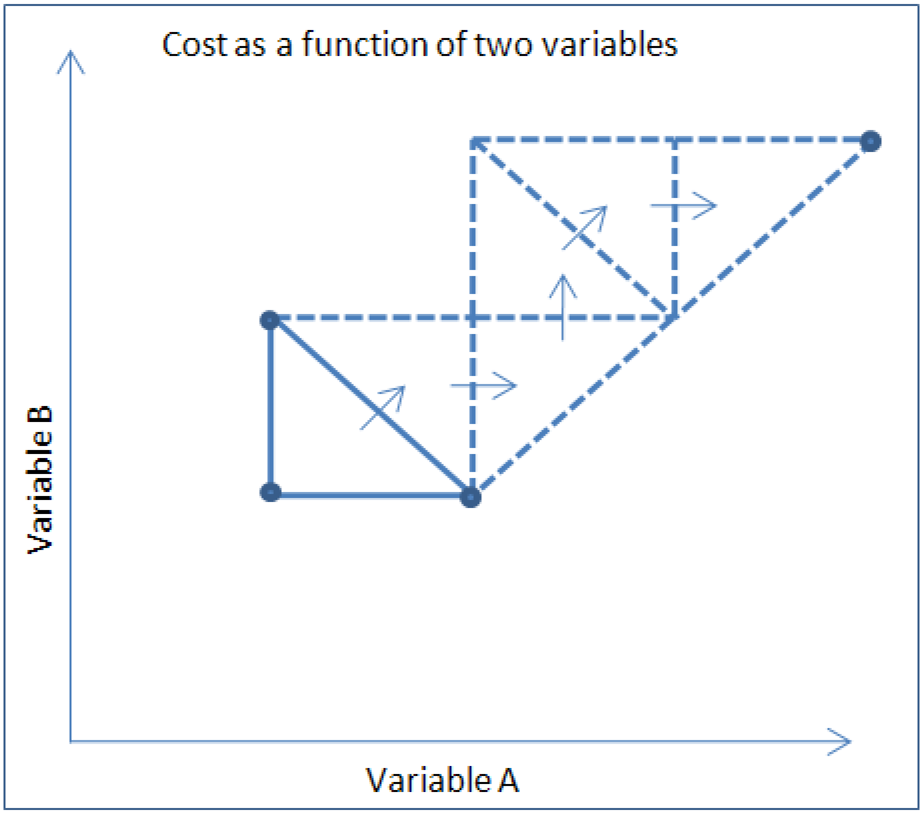
If the noise is significant in the nominal project, use the Pattern Search optimizer to obtain the results. It performs a grid-based simplex search, which makes use of simplices: triangles in 2D space or tetrahedra in 3D space. A simplex is a Euclidean geometric spatial element having the minimum number 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.

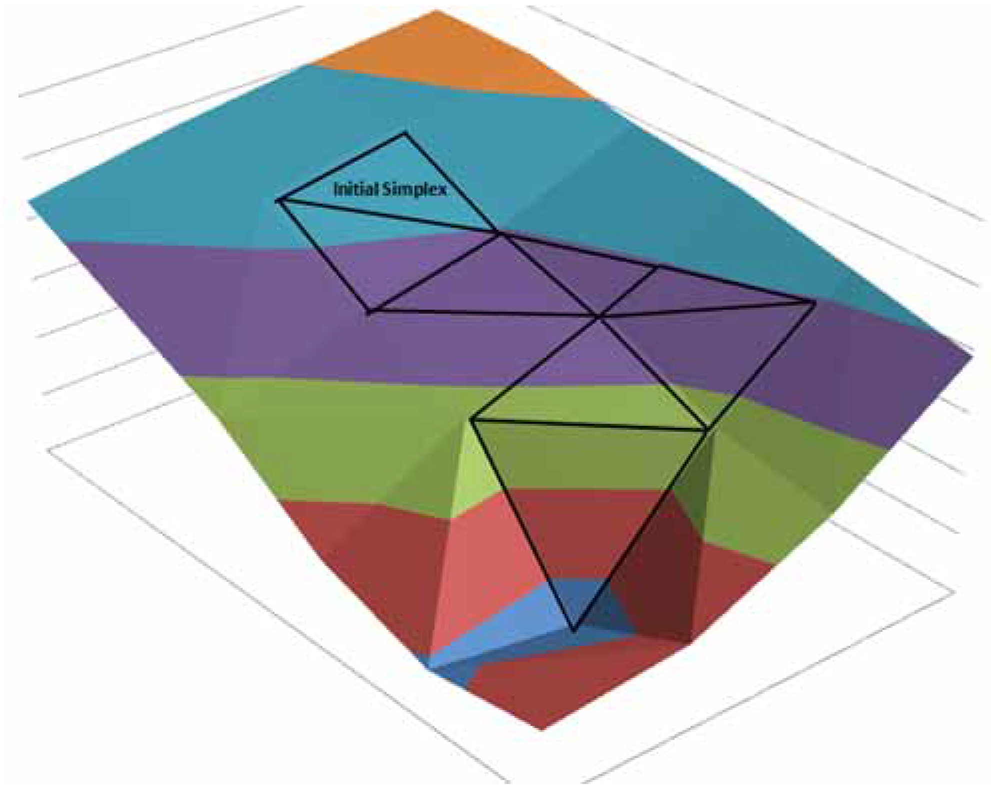
Cost functions can be quite nonlinear. As a result, during the function evaluations of the algorithm, the cost function can vary significantly. Also, it is important to understand the relationship between optimization function evaluation and iteration. Every iteration, depending on the number of parameters 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.

Sequential Non linear 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.

Sequential Mixed Integer NonLinear Programming (Gradient and Discrete)

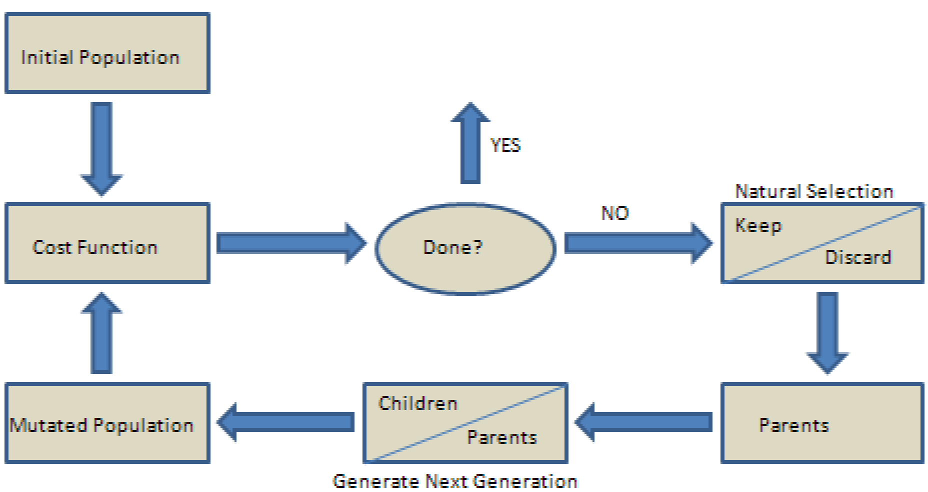
The Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) optimizer is equivalent to the SNLP (Gradient) optimizer with only one difference. Many problems require variables take only discrete values. One example might be to optimize on the number of turns in a coil. To be able to optimize 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.

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.

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
```

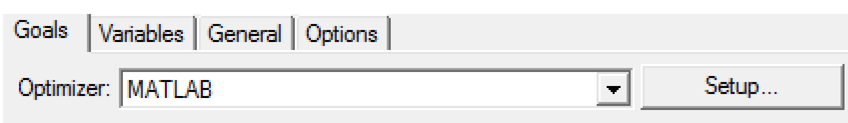
The answer will be 1 if the MATLAB Optimization Toolbox is licensed or 0 otherwise.

Specifying the MATLAB Location

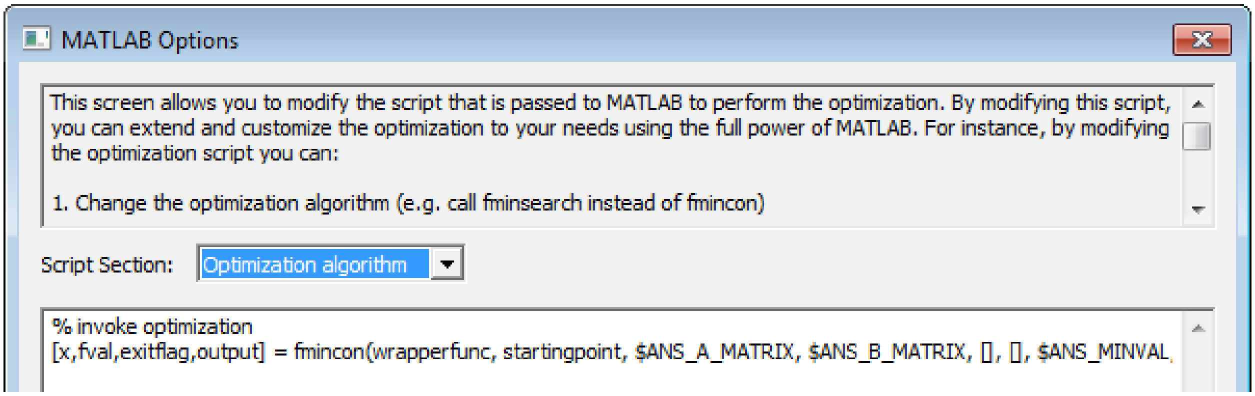
The Tools> General Options: Miscellaneous group contains a setting for the MATLAB location. This setting must to point to the version of MATLAB to be used for performing the optimization. The platform (32/64 bit or Linux) of the specified version of MATLAB must match the platform of this application.

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 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
```

Multi-Objective Genetic Algorithm (MOGA)

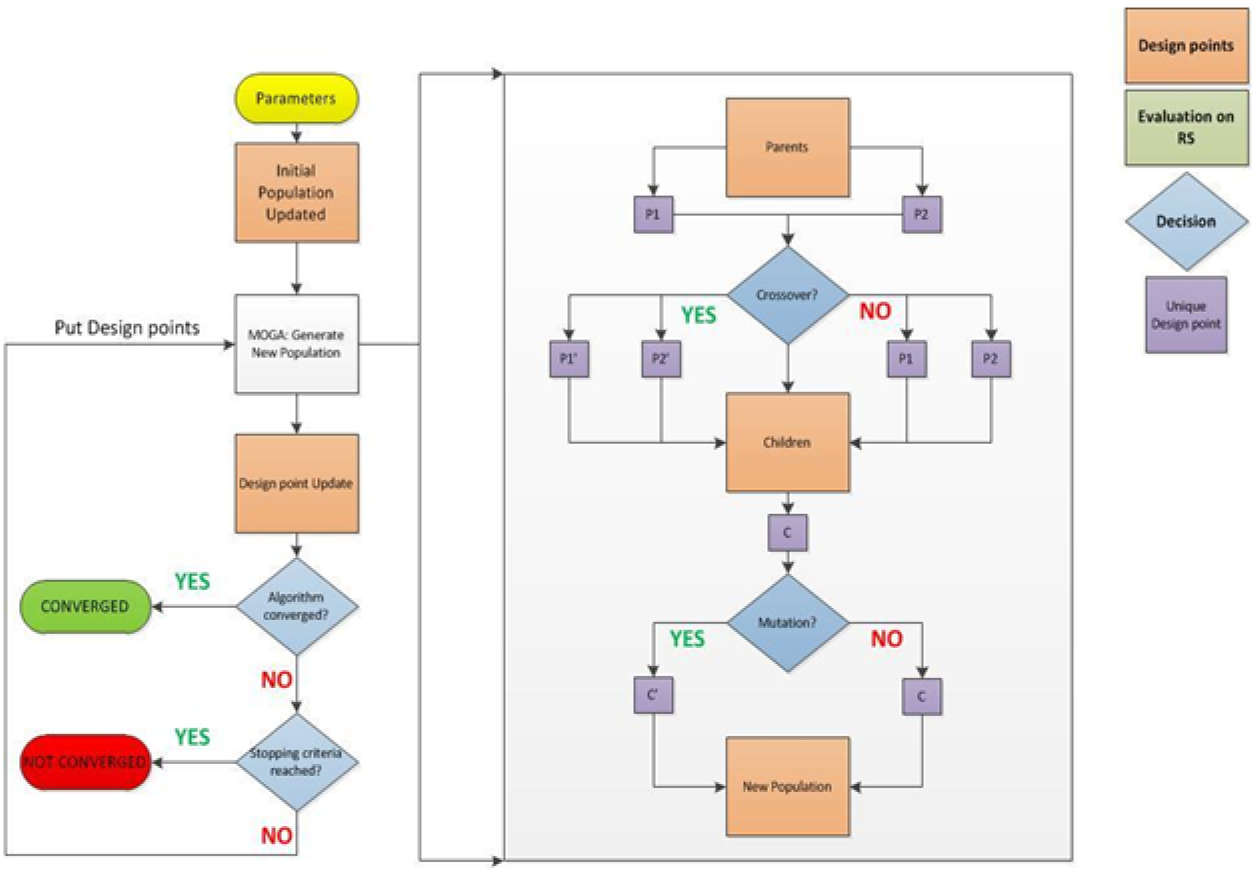
The Multi-Objective Genetic Algorithm (MOGA) used in GDO is a hybrid variant of the popular NSGA-II (Non-dominated Sorted Genetic Algorithm-II) based on controlled elitism concepts. It supports all types of input parameters. The Pareto ranking scheme is done by a fast, non-dominated sorting method that 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

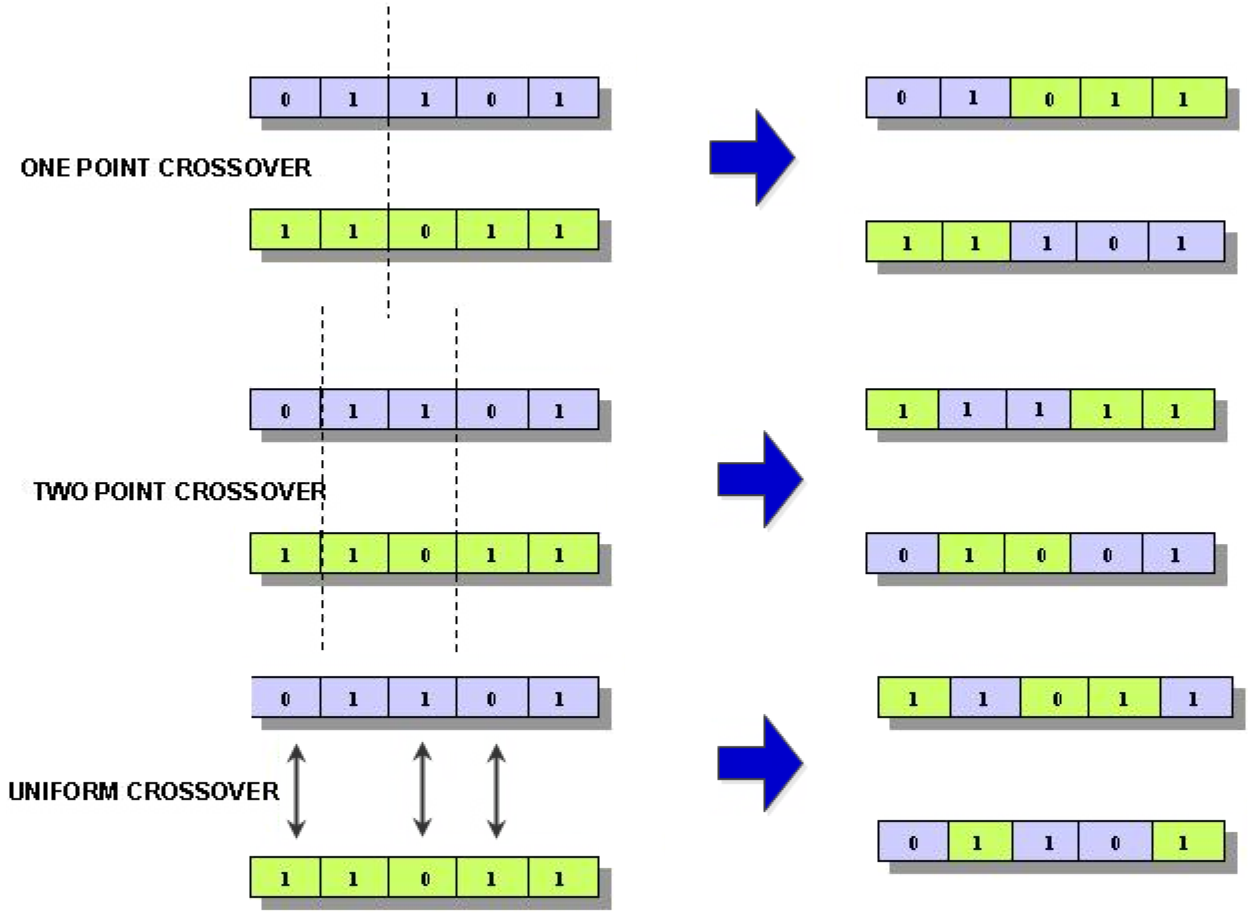
  A one-point cross-over operator that randomly selects a cross-over point within a chromosome then interchanges the two parent chromosomes at this point to produce two new offspring.

- 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.

$$C = P + (\mathrm{UpperBound} - \mathrm{LowerBound})\delta$$

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 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.

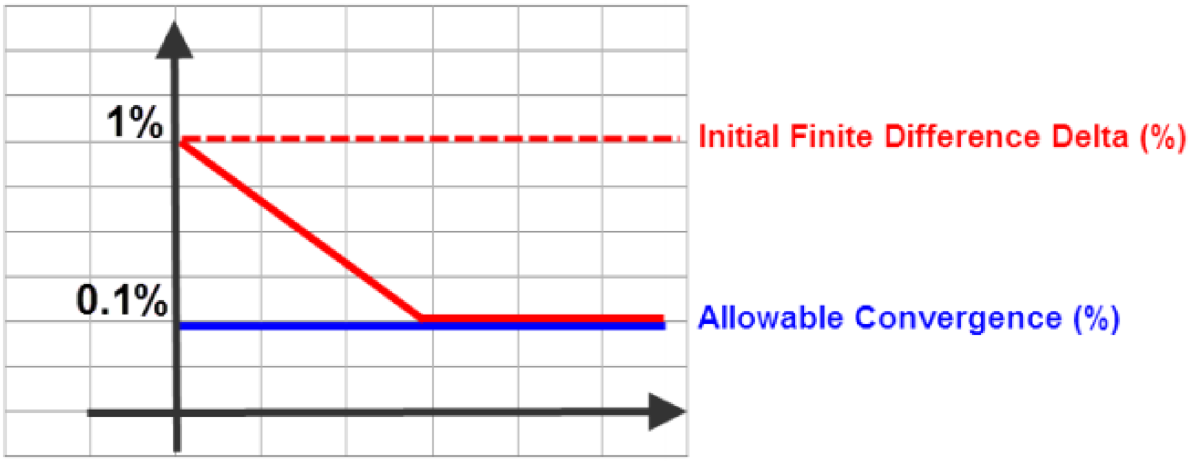
The main difference between these algorithms and Multi-Objective Genetic Algorithm (MOGA) is that MOGA is designed to work with multiple objectives and does not require full continuity of the output parameters. However, for continuous single objective problems, the use of NLPQL or MISQP gives greater 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.

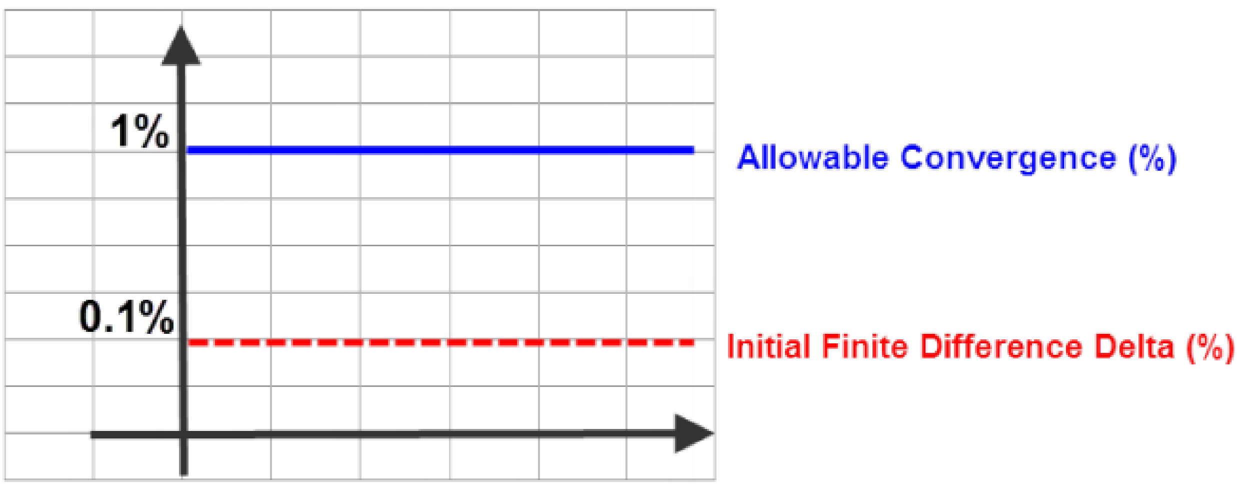
The default convergence rate is used in conjunction with the initial finite difference delta percentage value, which is specified by the Initial Finite Difference Delta (%) advanced property. This property defaults to 1% for a Direct Optimization system. You use this property to specify a percentage 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.

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:

$$g_j(x,y)=0, \quad j=1,\ldots,m_e,$$

$$g_j(x,y)\geq 0, \quad j=m_e+1,\ldots,m$$

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.

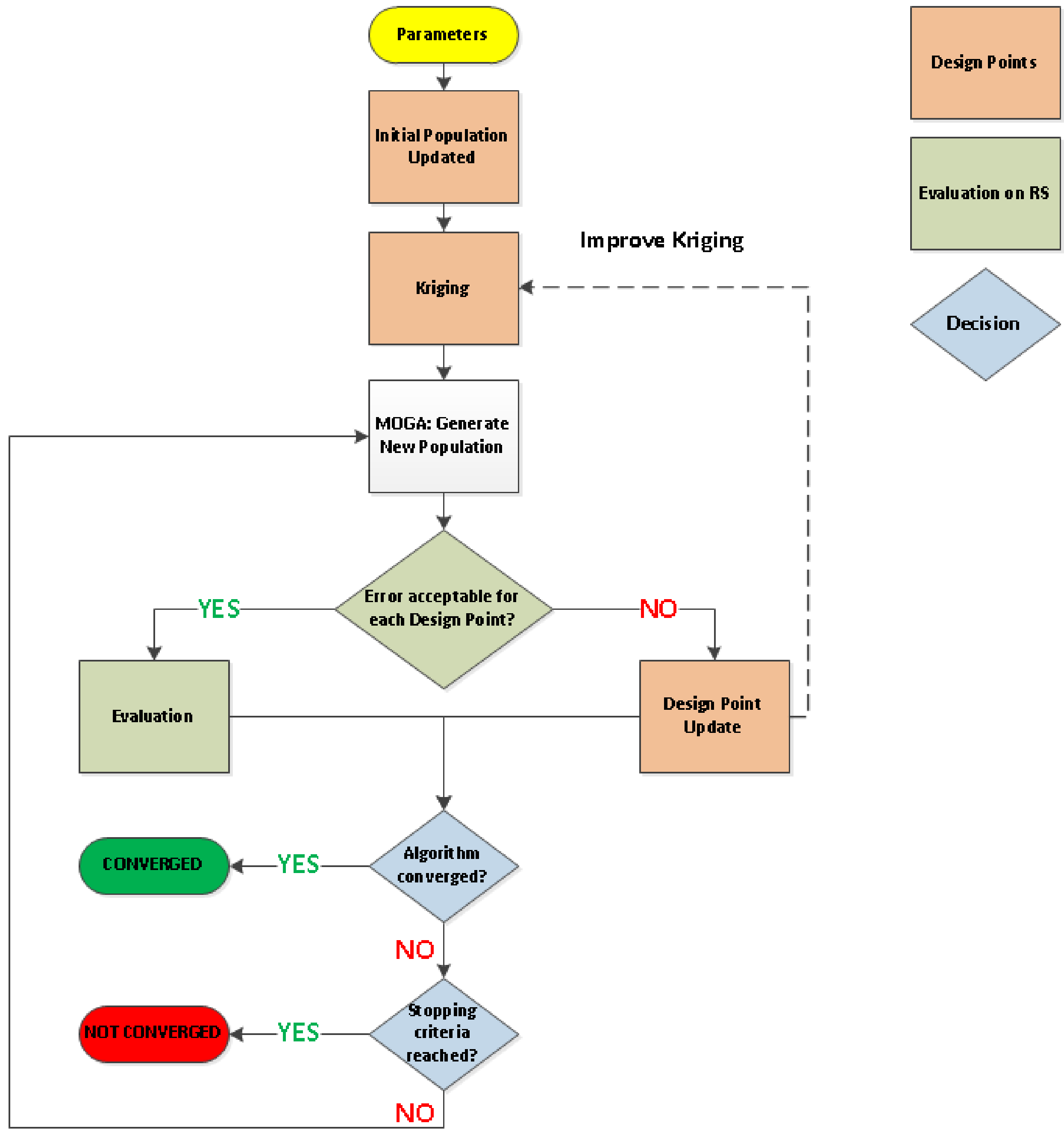
Adaptive Multiple Objective Optimization

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.

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:

$$g_k(\{x\})\leq 0, \forall k=1,2,\ldots,K$$

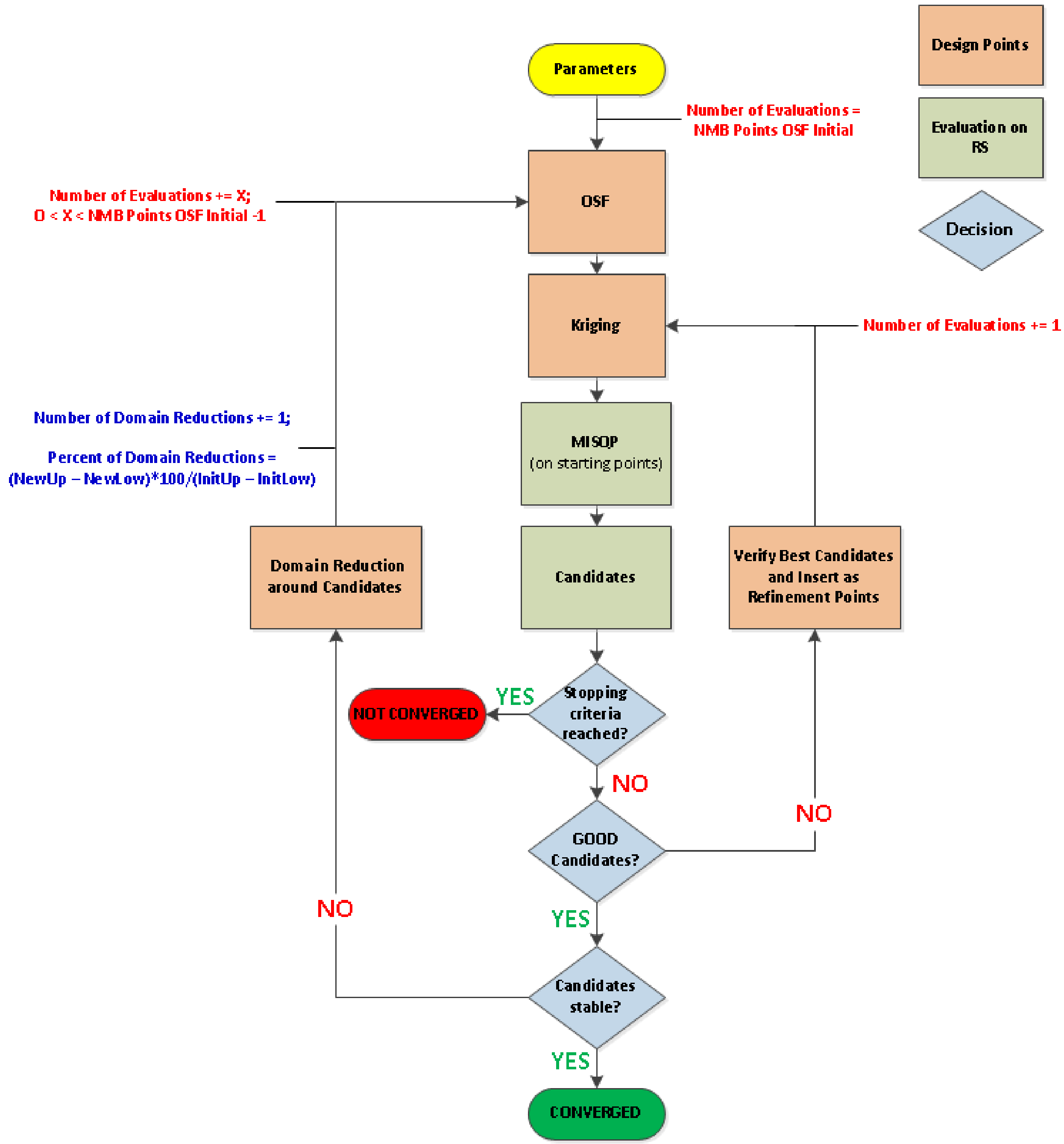
$$h_l(\{x\})=0, \forall I=1,2\ldots,L$$

Where:

$$\{x_L\}\leq\{x\}\leq\{x_U\}$$

ASO Workflow

The workflow of ASO follows:


Design Points
Evaluation on RS
Decision
Parameters
Number of Evaluations = NMB Points OSF Initial
OSF
Number of Evaluations += X;
0 < X < NMB Points OSF Initial -1
Kriging
Number of Evaluations += 1
Number of Domain Reductions += 1;
Percent of Domain Reductions = (NewUp – NewLow)*100/(InitUp – InitLow)
MISQP
(on starting points)
Domain Reduction around Candidates
Candidates
Verify Best Candidates and Insert as Refinement Points
NOT CONVERGED
YES
Stopping criteria reached?
NO
NO
GOOD Candidates?
NO
YES
Candidates stable?
YES
CONVERGED

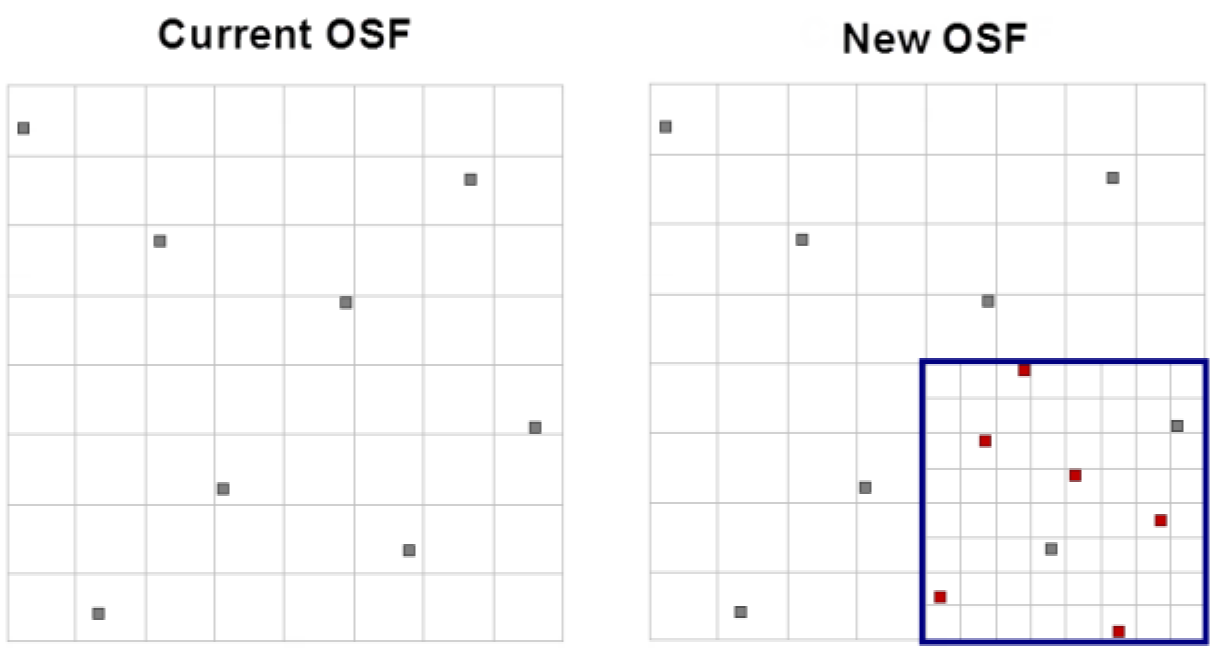
AMO Steps

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.

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 (cos*t*(*x*): $R^n \to R$, 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.

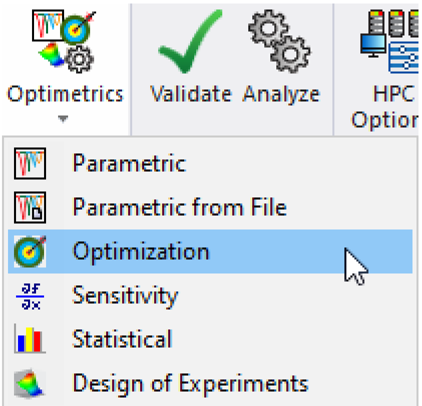
To provide a broad range of capability, Optimetrics incorporates the following types of numerical optimizers:

- Screening (Search based)
- Multi-Objective Genetic Algorithm
- Non-linear 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.

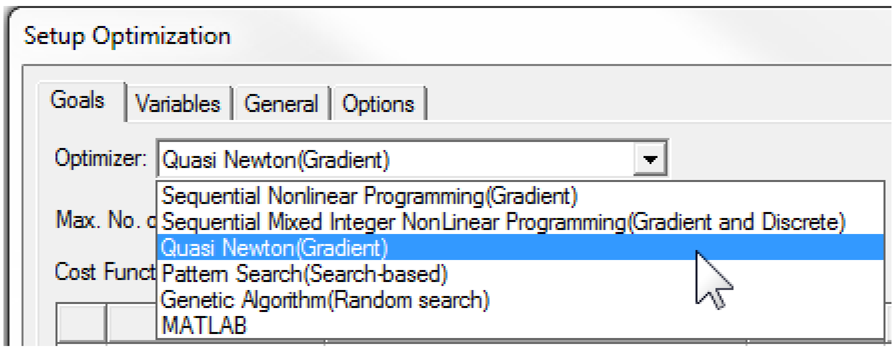
> **Note:**
>
> Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

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 **Circuit**> **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.

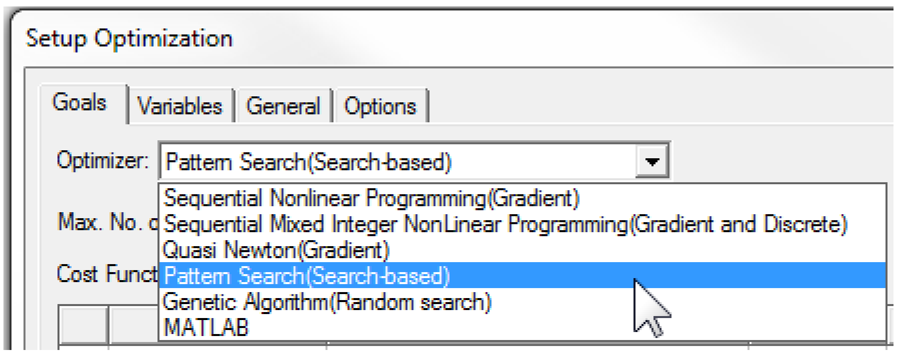
Optimization Setup for the Pattern Search (Search-based) Optimizer

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 **Circuit**> **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.

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 **Circuit**> **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.

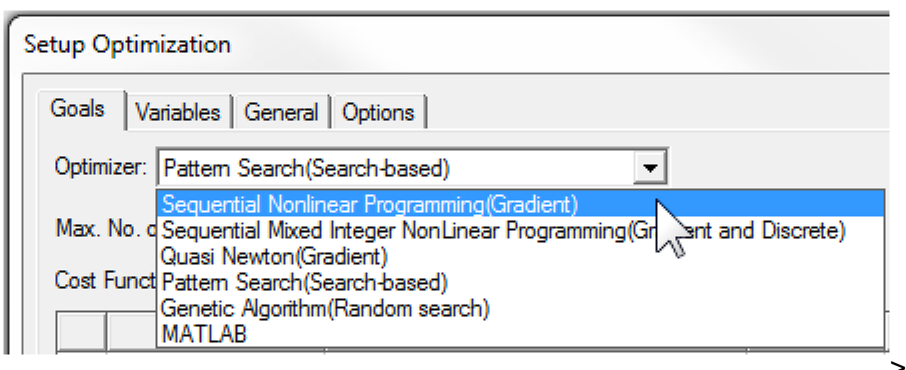
Optimization Setup for the Sequential Nonlinear Programming (Gradient) Optimizer

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 **Circuit** > **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.

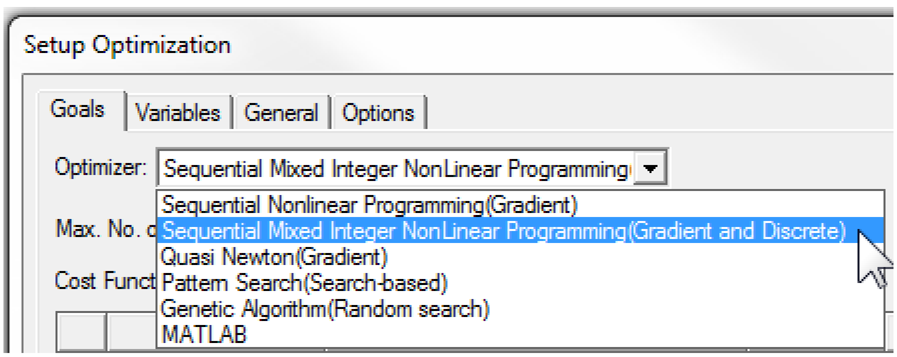
Optimization Setup for the Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) Optimizer

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 **Circuit** > **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.

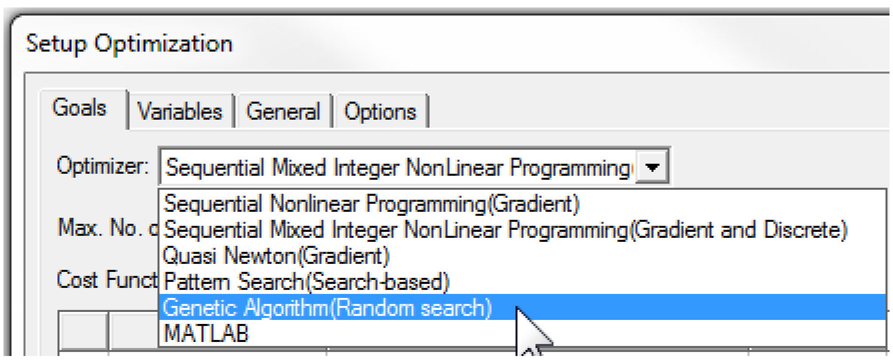
Optimization Setup for the Genetic Algorithm (Random search) Optimizer

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 **Circuit** > **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.

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 **Circuit**> **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.

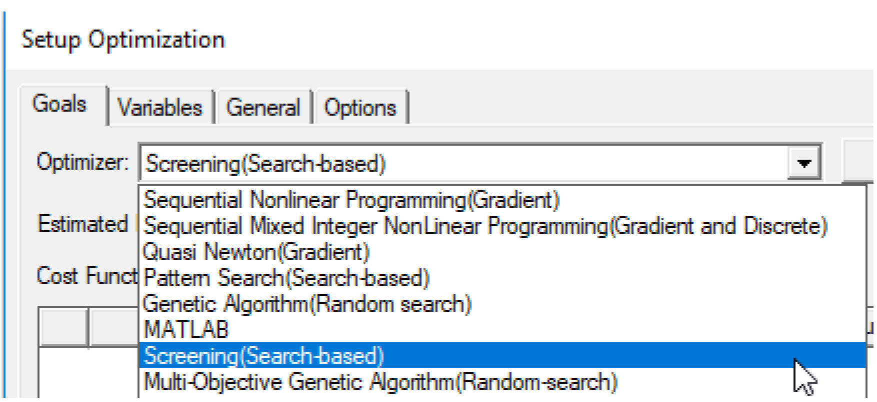
You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setting Up Screening (Search Based) Optimizer

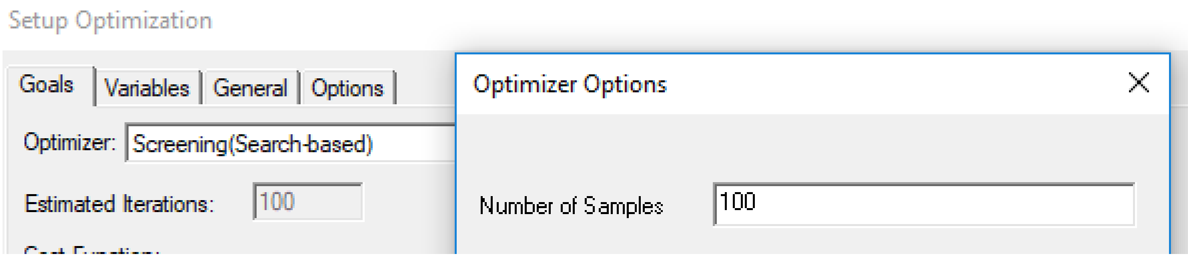
Following is the procedure for setting up an optimization analysis using the Screening Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

This is a non-iterative direct sampling method that uses a quasi-random number generator based on the Hammersley algorithm. You can start with Screening to locate the multiple tentative optima and then refine with NLPQL or MISQP to zoom in on the individual local maximum or minimum value. Usually Screening is used for preliminary design. Then, if you want to refine, these candidate points are used as starting points for gradient methods.

1. Set up the variables you want to optimize in the *Design Properties* dialog box. The variables must be swept in a Parametric setup.
2. Click **Circuit** > **Optimetrics Analysis** > **Add Screening & Optimization**. The *Setup Optimization* dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Screening (Search Based)** from the **Optimizer** drop-down menu.

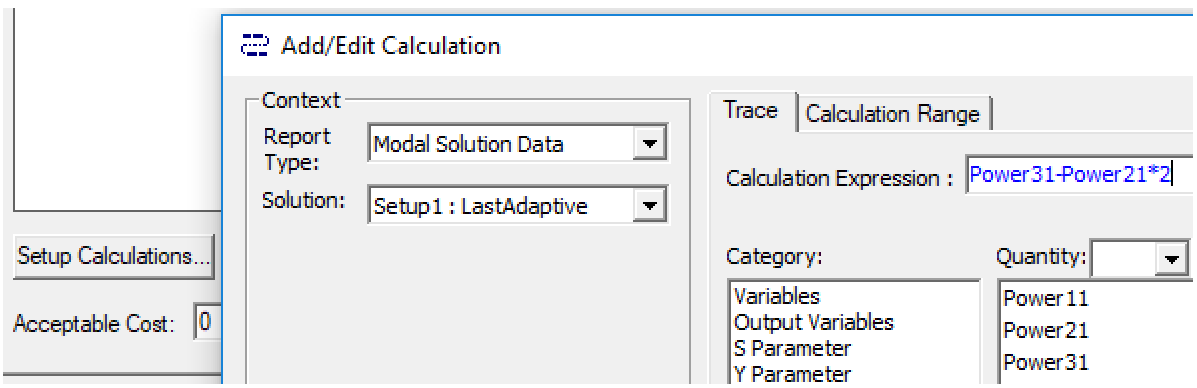


4. Optionally press the **Setup** button to open the **Optimizer Options window** to change the default number of samples from 100.



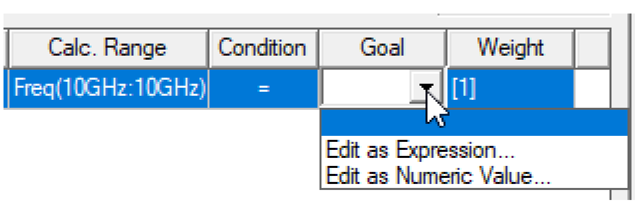
The number of samples must be greater than the number of enabled input parameters. The number of enabled input parameters is also the minimum number of samples required to generate the Sensitivities chart. You can enter a minimum of 2 and a maximum of 10,000. The default is 100 for a Direct Optimization system.

5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.



When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either Edit as Expression or Edit as Numeric Value...



7. This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to

the **Goal** column.

Calc. Range	Condition	Goal	Weight
Freq(10GHz:10GHz)	=	Power21-Power...	[1]

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.

8. Optionally, set the Acceptable Cost and Cost Function Noise.
9. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
10. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

12. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

> **Note:**
>
> Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

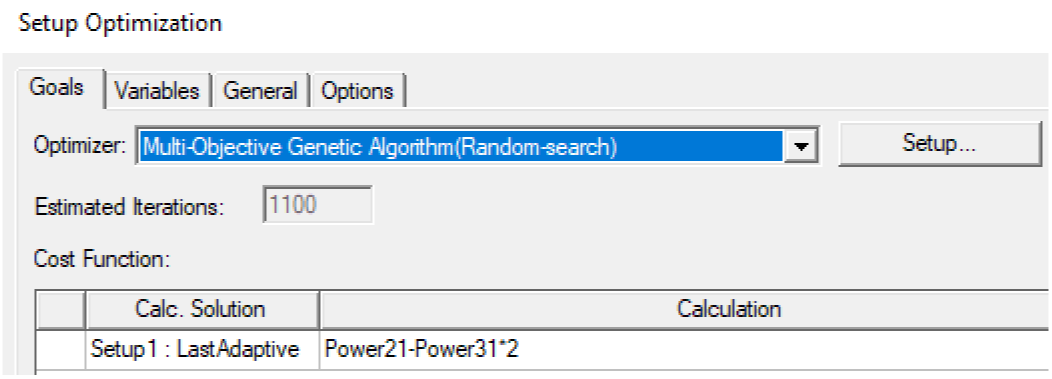
Setting Up Multi-Objective Genetic Algorithm(Random Search) Optimizer

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.

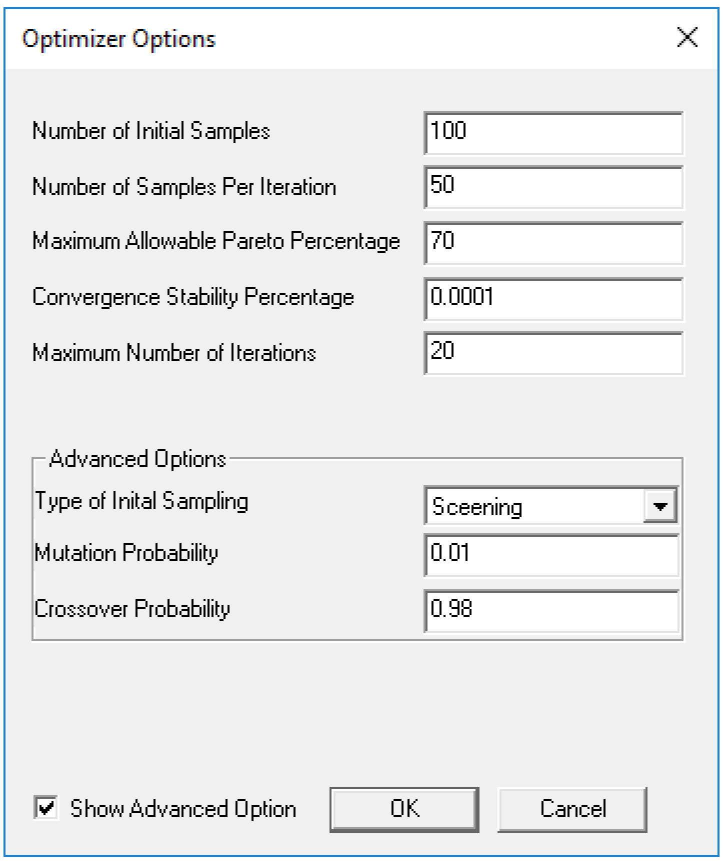
The Multi-Objective Genetic Algorithm (MOGA) is a hybrid variant of the popular NSGA-II (Non-dominated Sorted Genetic Algorithm-II) based on controlled elitism concepts. It supports all types of input parameters. The Pareto ranking scheme is done by a fast, non-dominated sorting method that is an order 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 **Circuit** > **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.

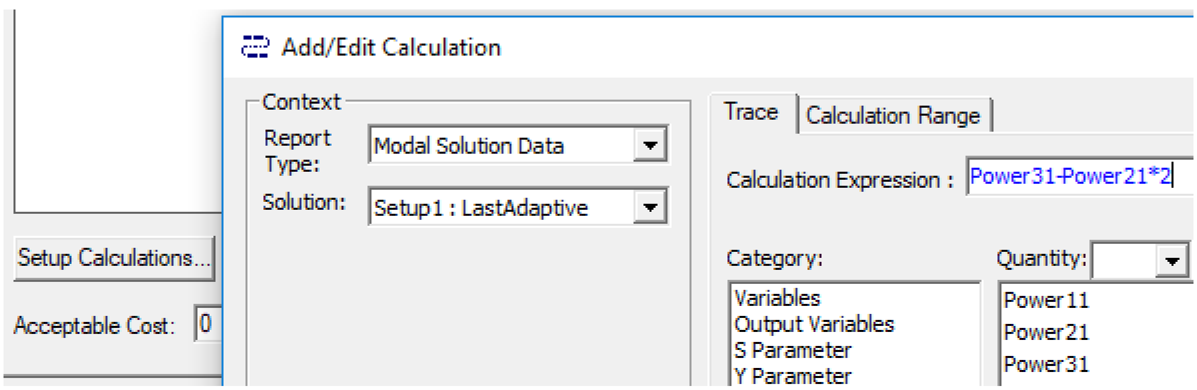
  You can enter a minimum of 2 and a maximum of 10000.

- **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)

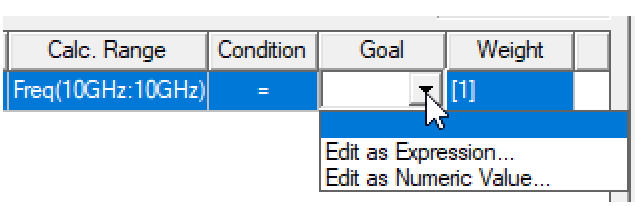
- **Type of Initial Sampling**: Advanced option for generating different kinds of sampling. If you do not have any parameter relationships defined, set to Screening (default) or Optimal Space-Filling. If you have parameter relationships defined, the initial sampling must be performed by the constrained 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.

Calc. Range	Condition	Goal	Weight
Freq(10GHz:10GHz)	=	Power21-Power...	[1]

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.

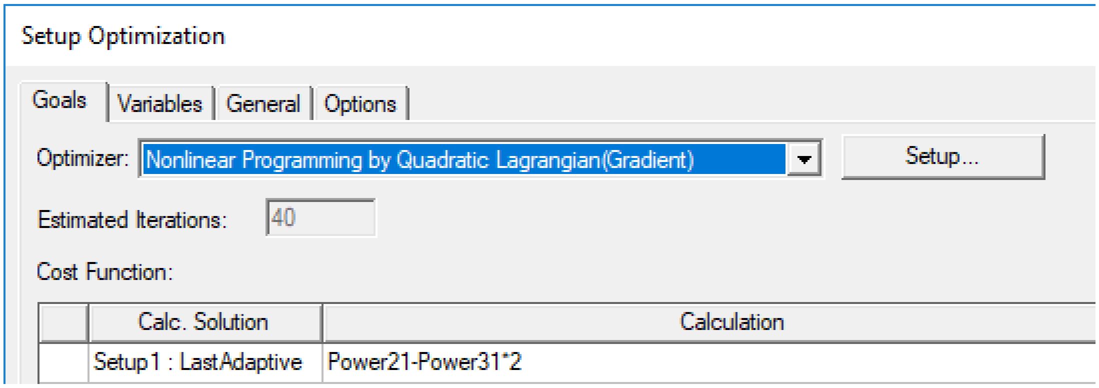
Setting Up Nonlinear Programming by Quadratic Lagrangian (Gradient) Optimizer

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.

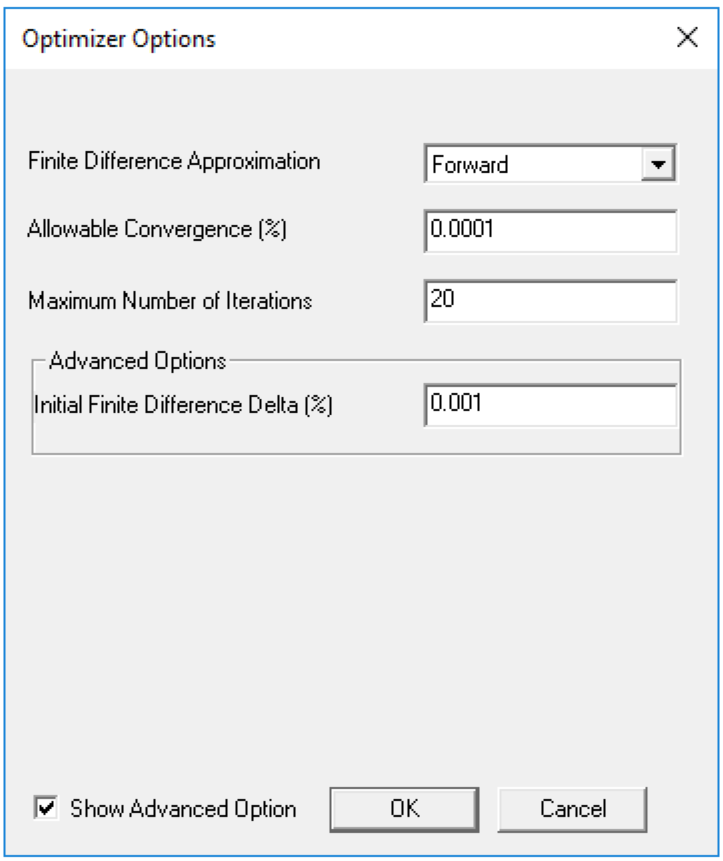
The NLPQL (Nonlinear Programming by Quadratic Lagrangian) method can be used for Direct Optimization systems. It allows you to generate a new sample set to provide a more refined approach than the Screening method. Available for continuous input parameters only, NLPQL can handle only one output parameter 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 **Circuit** > **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:

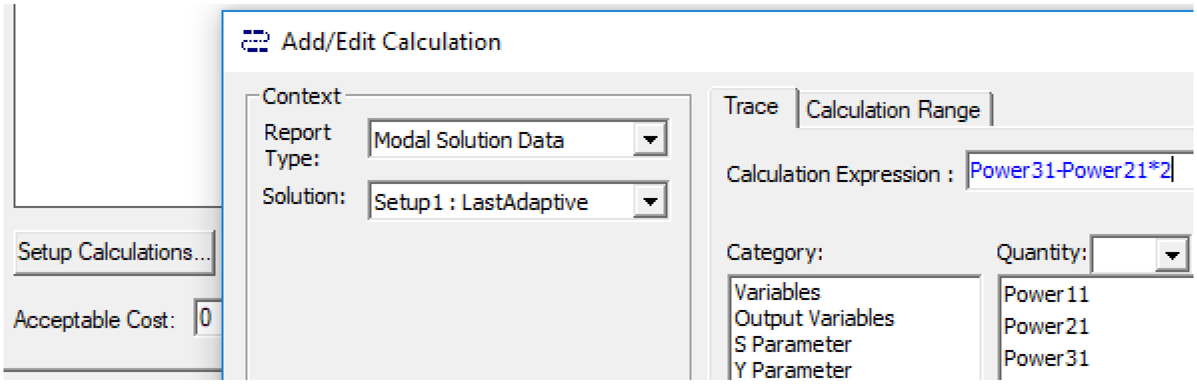
- **Central**: Increases the accuracy of the gradient calculations by sampling from both sides of the sample point but increases the number of design point evaluations by 50%. This method makes use of the initial point, as well as the forward point and rear point.
- >**Forward**: Uses fewer design point 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.

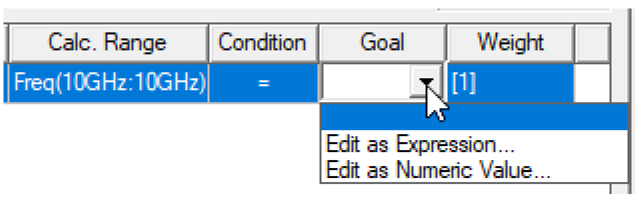
The default percentage value is 0.001. The minimum is 0.0001, and the maximum is 1. For parameters with Allowed Values set to **Manufacturable Values** or Snap to Grid, the value for Initial Finite Difference Delta (%) is ignored. In such cases, the closest allowed value is used to determine the finite difference delta.

5. Add a cost function by selecting the **Setup Calculations** button to open the *Add/Edit Calculation* dialog box.



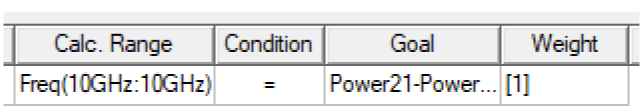
When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the *Add/EditCalculation* dialog box.

6. In the Optimization setup, in the 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 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.

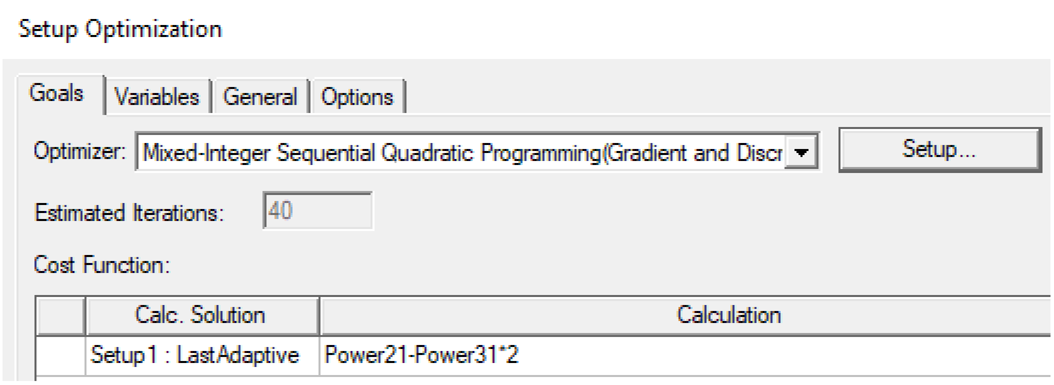
Setting Up Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete) Optimizer

Following is the procedure for setting up an optimization analysis using the Mixed-Integer Sequential Quadratic Programming optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

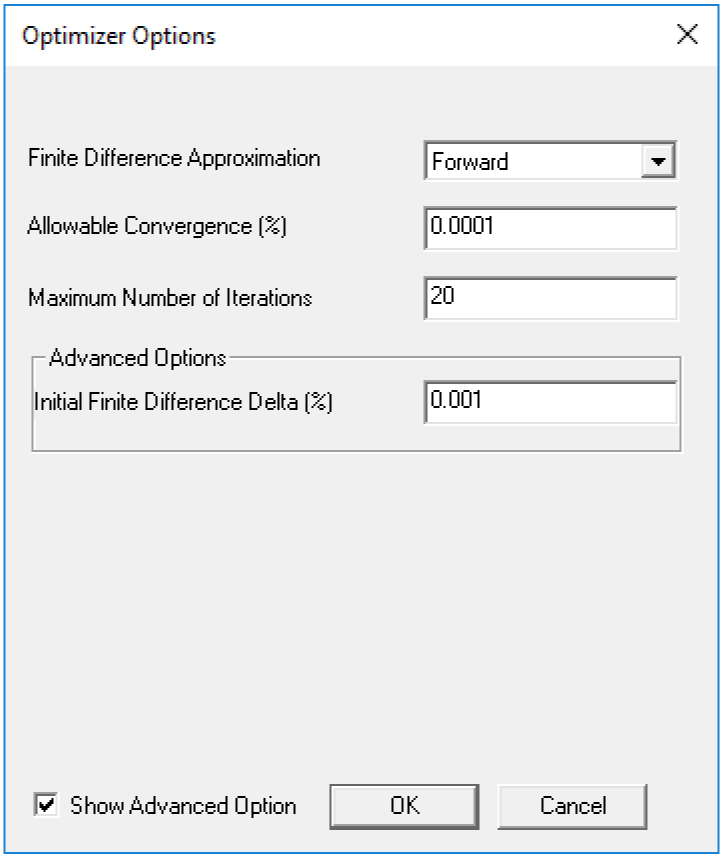
The Mixed-Integer Sequential Quadratic Programming (Nonlinear Programming by Quadratic Lagrangian) method (MISQP)can be used for Direct Optimization systems. It allows you to generate a new sample set to provide a more refined approach than the Screening method. MISQP is available for both continuous and discrete input parameters, which is why mixed is in its name. MISQP can handle only one output parameter goal.. Other output parameters can be defined as constraints. For more information, see Convergence Rate % and Initial Finite Difference Delta % in NLPQL and MISQP and Nonlinear Programming by Quadratic Lagrangian (NLPQL).

To generate samples and perform an NLPQL optimization:

1. Set up the variables you want to optimize in the *Design Properties* dialog box. The variables must be swept in a Parametric setup.
2. Click **Circuit** > **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:

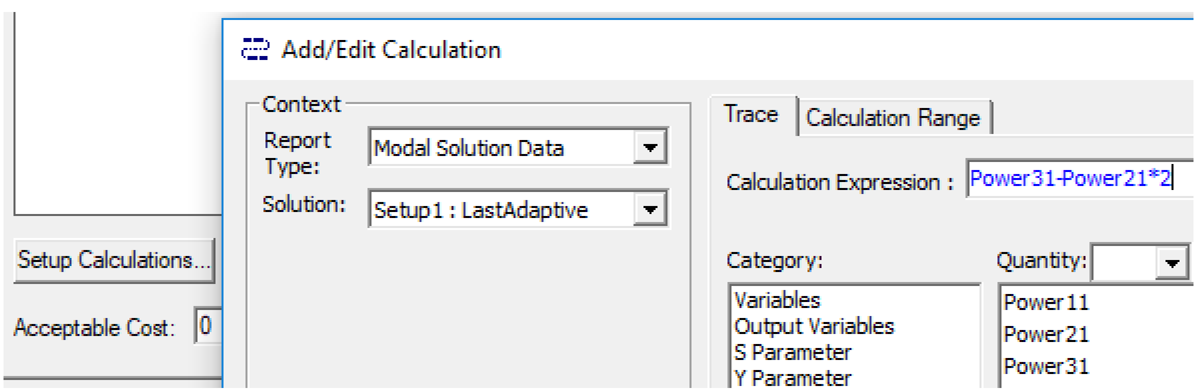
For **Central**: number of iterations * (2*number of inputs +1)

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.

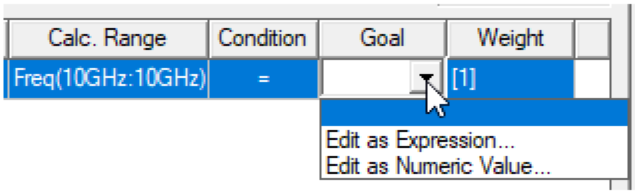
The default percentage value is 0.001. The minimum is 0.0001, and the maximum is 1. For parameters with Allowed Values set to **Manufacturable Values** or Snap to Grid, the value for Initial Finite Difference Delta (%) is ignored. In such cases, the closest allowed value is used to determine the finite difference delta.

5. Add a cost function by selecting the **Setup Calculations** button to open the *Add/Edit Calculation* dialog box.



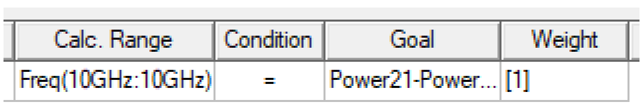
When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the *Add/EditCalculation* dialog box.

6. In the Optimization setup, in the 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 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.

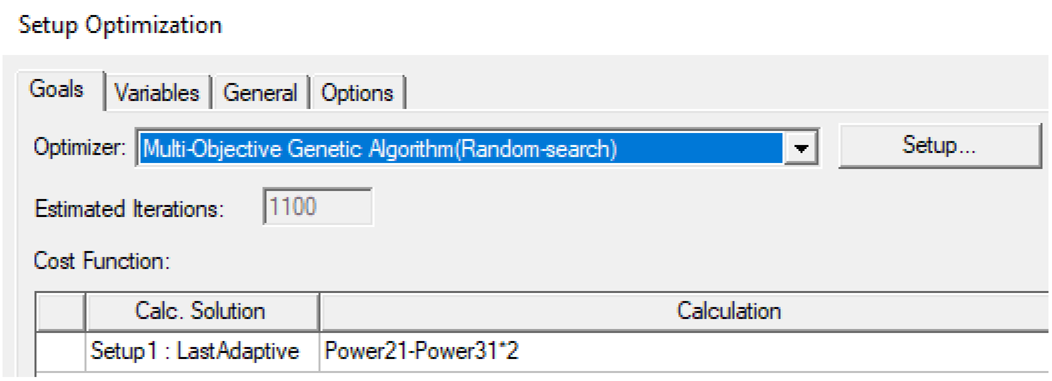
Setting Up Multi-Objective Genetic Algorithm(Random Search) Optimizer

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.

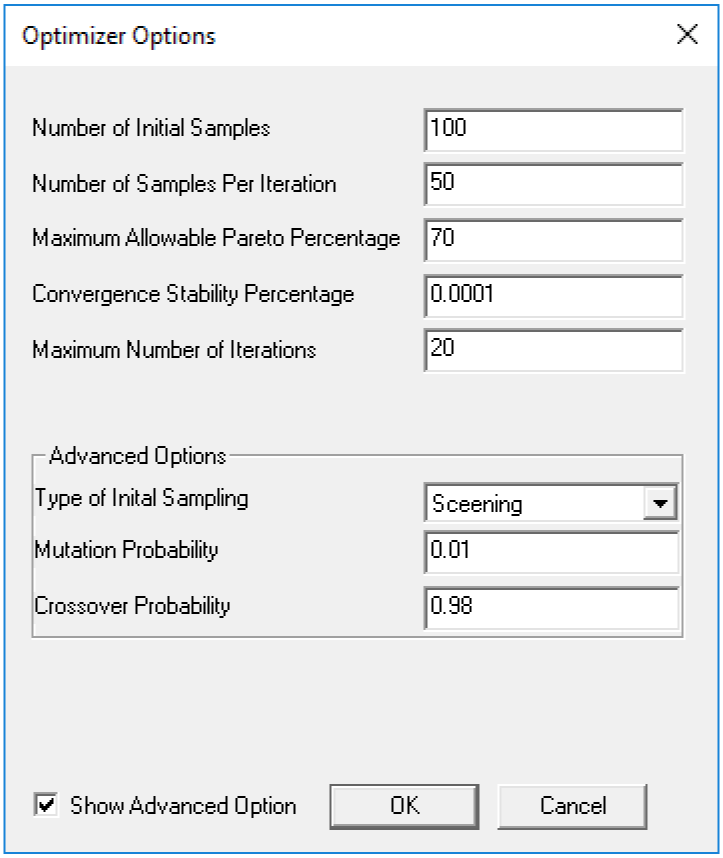
The Multi-Objective Genetic Algorithm (MOGA) is a hybrid variant of the popular NSGA-II (Non-dominated Sorted Genetic Algorithm-II) based on controlled elitism concepts. It supports all types of input parameters. The Pareto ranking scheme is done by a fast, non-dominated sorting method that is an order 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 **Circuit** > **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.

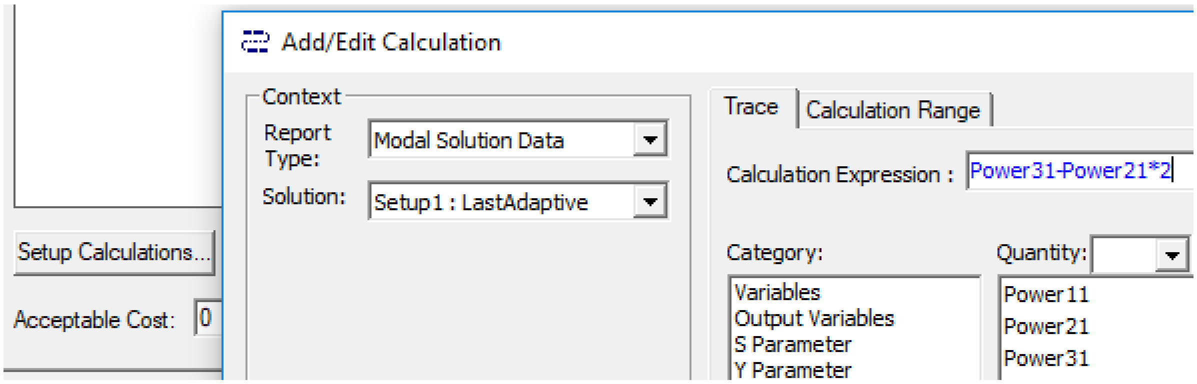
- **Maximum Allowable Pareto Percentage**: Convergence criterion. Percentage value that represents the ratio of the number of desired Pareto points to the number of samples per iteration. When this percentage is reached, the optimization is converged. For example, a value of 70 with Number of Samples 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.

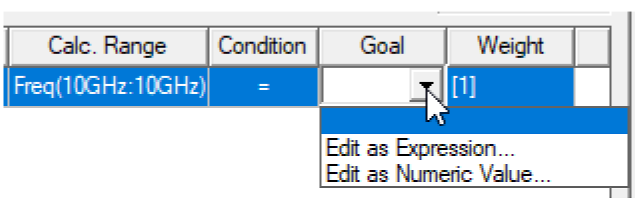
- **Crossover Probability**: Advanced option for specifying the probability with which parent solutions are recombined to generate offspring solutions. The value must be between 0 and 1. A smaller value indicates a more stable population and a faster (but less accurate) solution. If the value is 0, the 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.

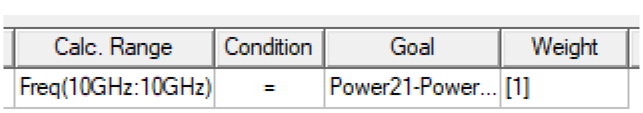
6. In the Optimization setup, in the dropdown for the Goal column, select either **Edit as Expression...** or **Edit as Numeric Value...**.



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.

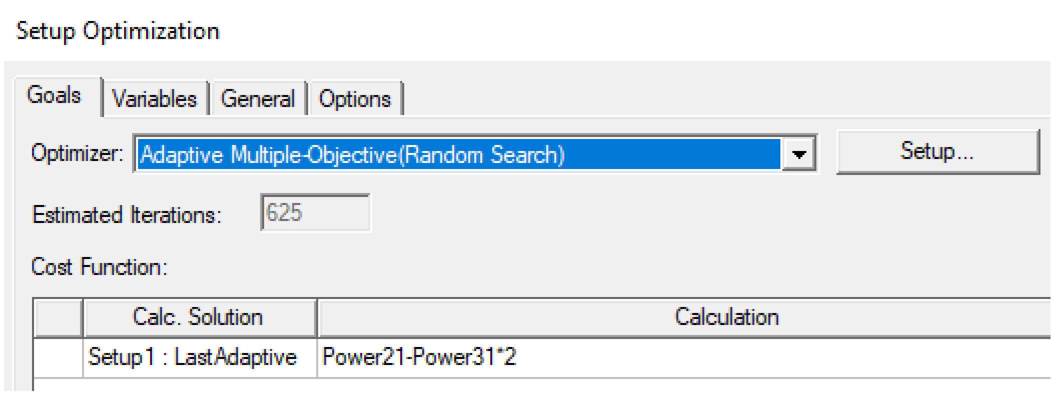
You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setting Up Adaptive Multiple-Objective (Random Search) Optimizer

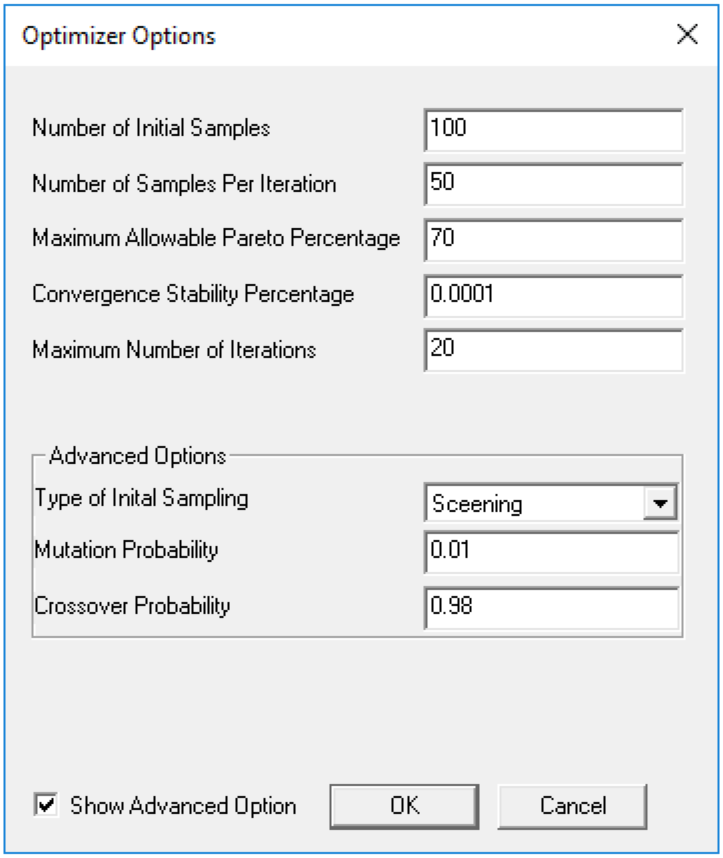
Following is the procedure for setting up an optimization analysis using the Adaptive Multiple Objective (Random Search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

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 **Circuit** > **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.

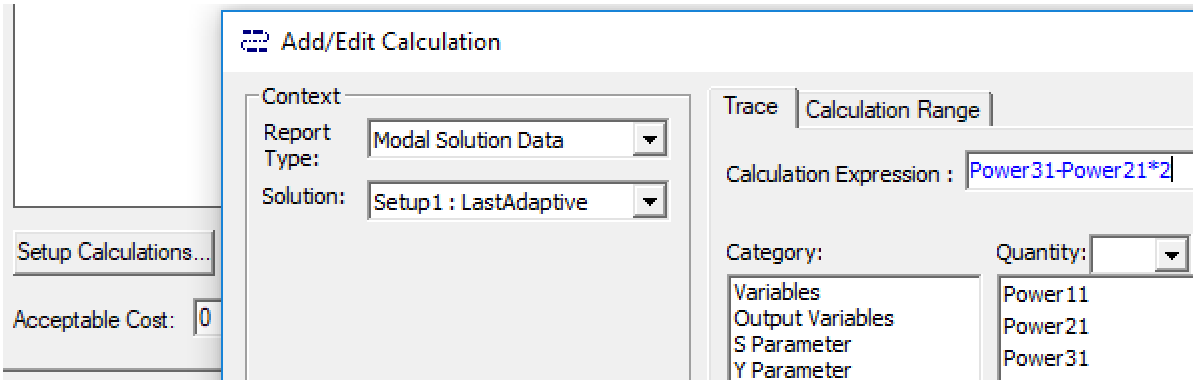
- **Maximum Allowable Pareto Percentage**: Convergence criterion. Percentage value that represents the ratio of the number of desired Pareto points to the number of samples per iteration. When this percentage is reached, the optimization is converged. For example, a value of 70 with Number of Samples 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.

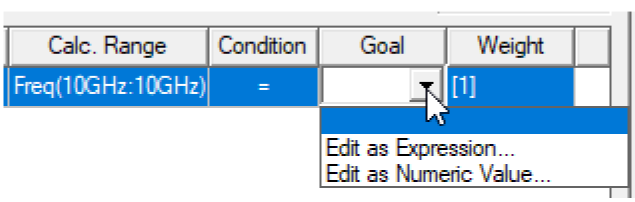
- **Crossover Probability**: Advanced option for specifying the probability with which parent solutions are recombined to generate offspring solutions. The value must be between 0 and 1. A smaller value indicates a more stable population and a faster (but less accurate) solution. If the value is 0, the 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.

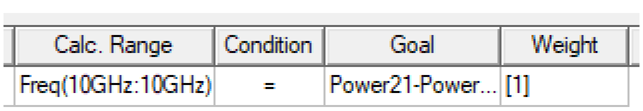


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the *Add/EditCalculation* dialog box.

6. In the Optimization setup, in the dropdown for the Goal column, select either **Edit as Expression...** or **Edit as Numeric Value...**.



7. This reopens the *Add/Edit Calculation* dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value 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.

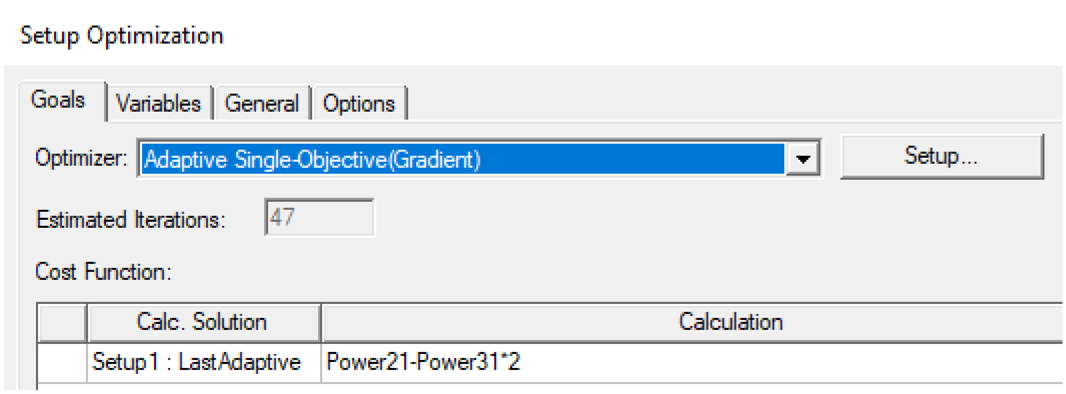
Setting Up Adaptive Single-Objective (Gradient) Optimizer

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.

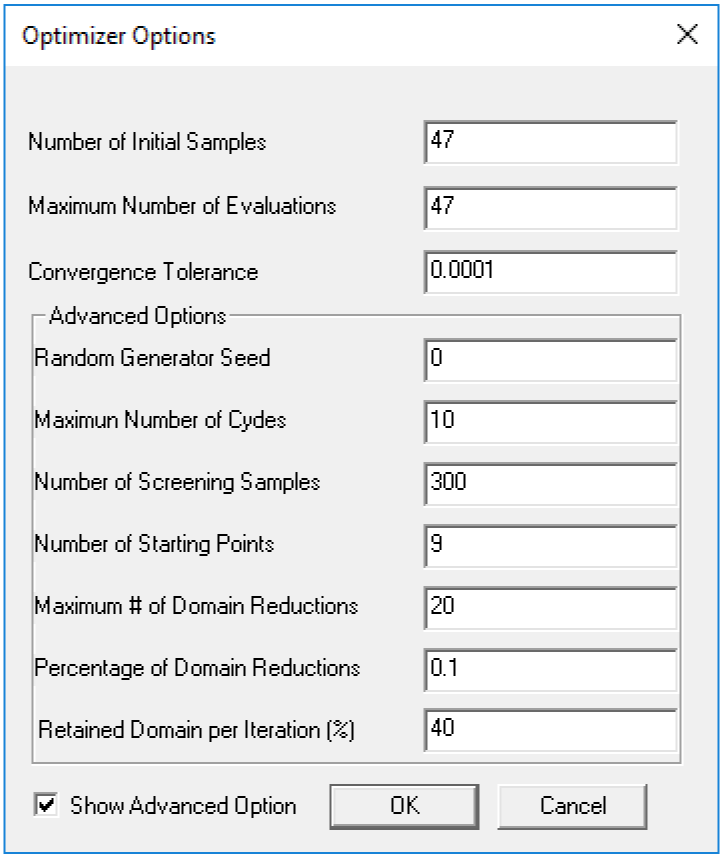
This gradient-based method employs automatic intelligent refinement to provide the global optima. It requires a minimum number of design points to build the Kriging response surface, but, in general, this method reduces the number of design points necessary for the optimization. Failed design points 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 **Circuit** > **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.

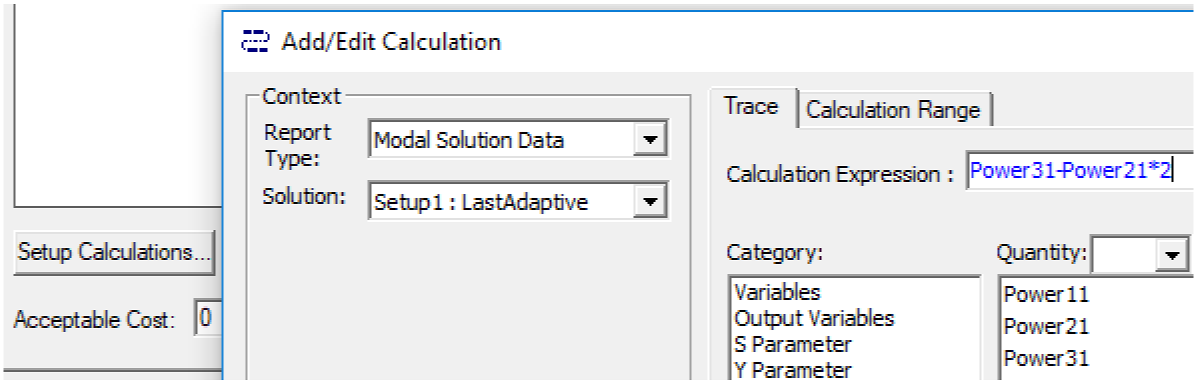
- **Random Generator Seed**: The value for initializing the random number generator invoked internally by OSF. The value must be a positive integer. This property allows you to generate different samplings by changing the value or to regenerate the same sampling by keeping the same value. The default 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.

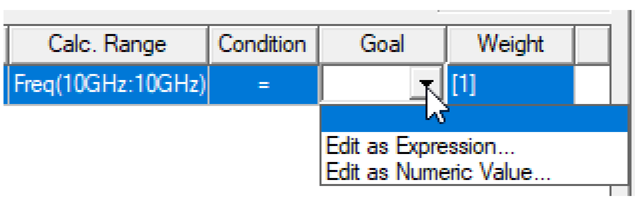
- **Number of Starting Points**: Determines the number of local optima to explore. The larger the number of starting points, the more local optima explored. In the case of a linear surface, for example, it is not necessary to use many points. This value must be less than the value for **Number of Screening 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.

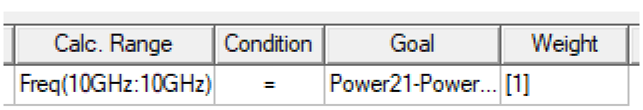


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog box.

6. In the Optimization setup, in the Goal column drop-down menu, select either **Edit as Expression...** or **Edit as Numeric Value...**.



7. This reopens the *Add/Edit Calculation* dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value 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.

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.

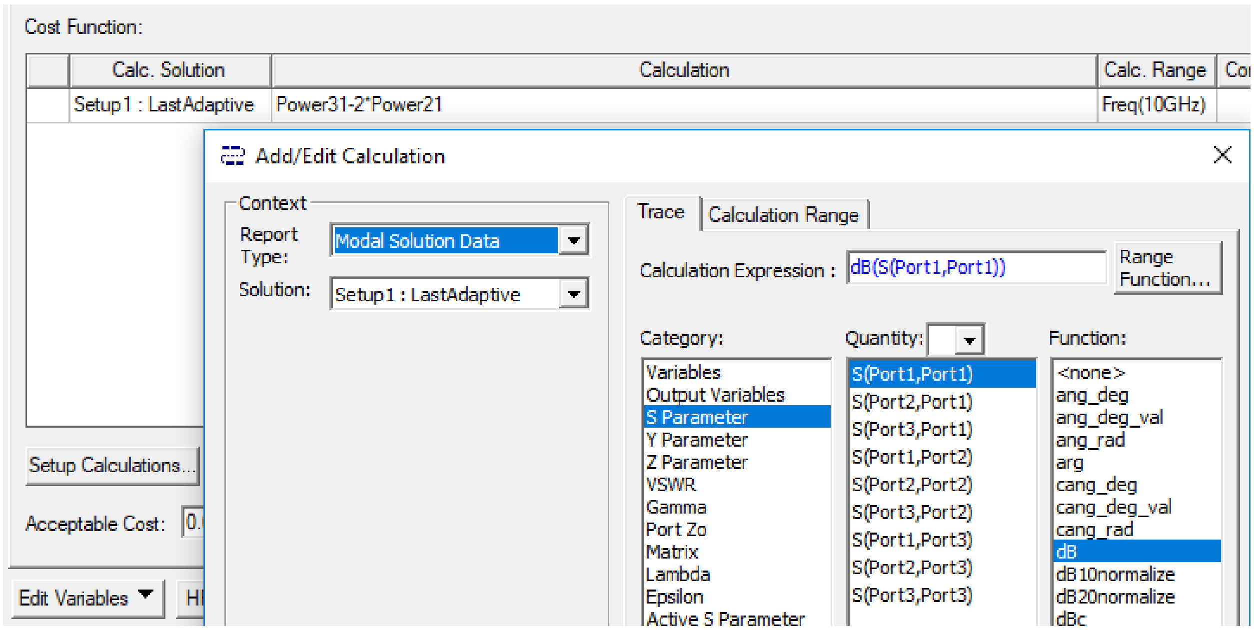
Cost Function

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.

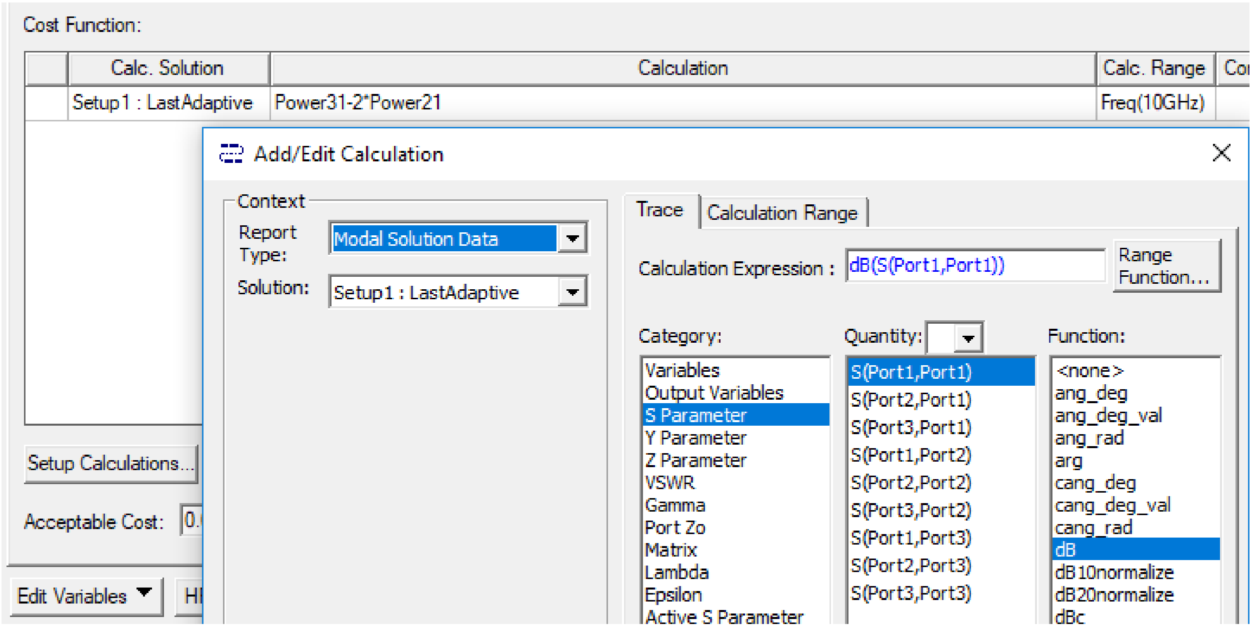
Adding a Cost Function

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 jth sub-goal.
- N_j is the number of frequencies for the jth sub-goal.
- e_i is the error contribution from the jth sub-goal at the ith 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.

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.

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.

Text Entry for Calc. Range or Edit Calculation Range Dialog

> **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)

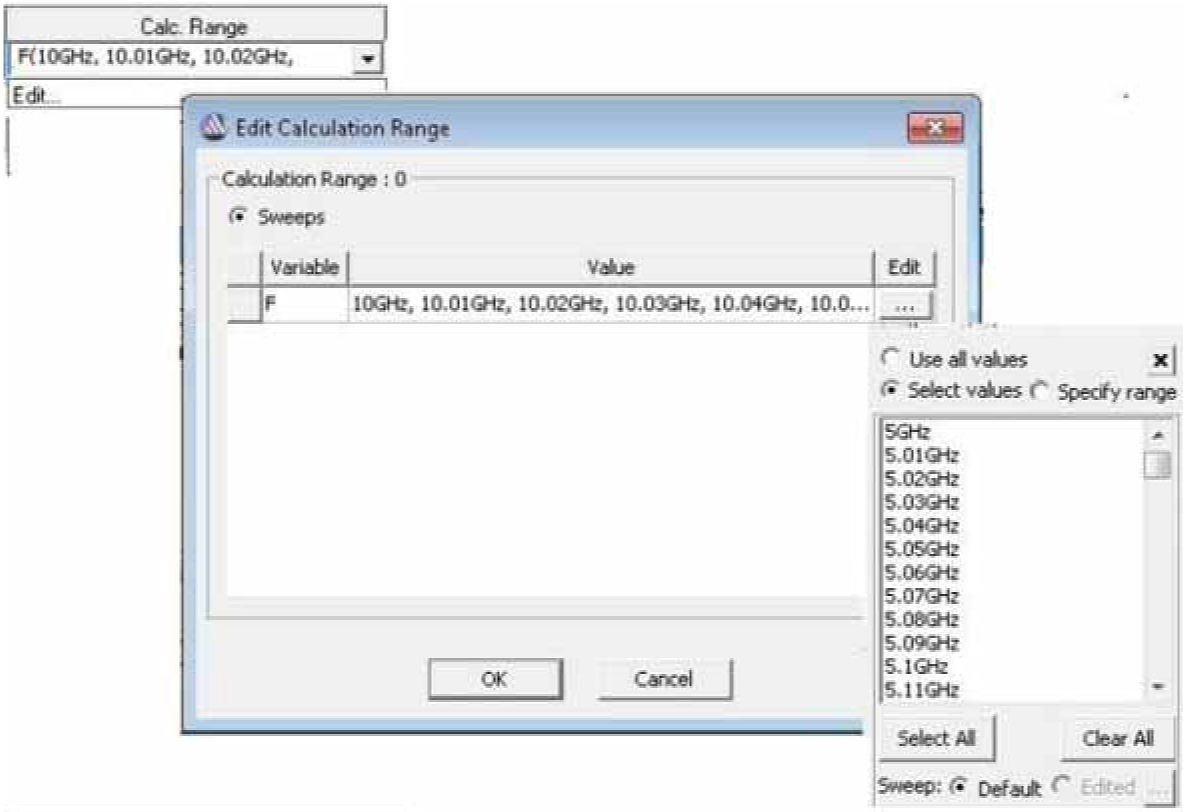
  min/max range, for example, Time([1ms, 2ms]) HFSS/MAXWELL/TWINBUILDER

- 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 ","

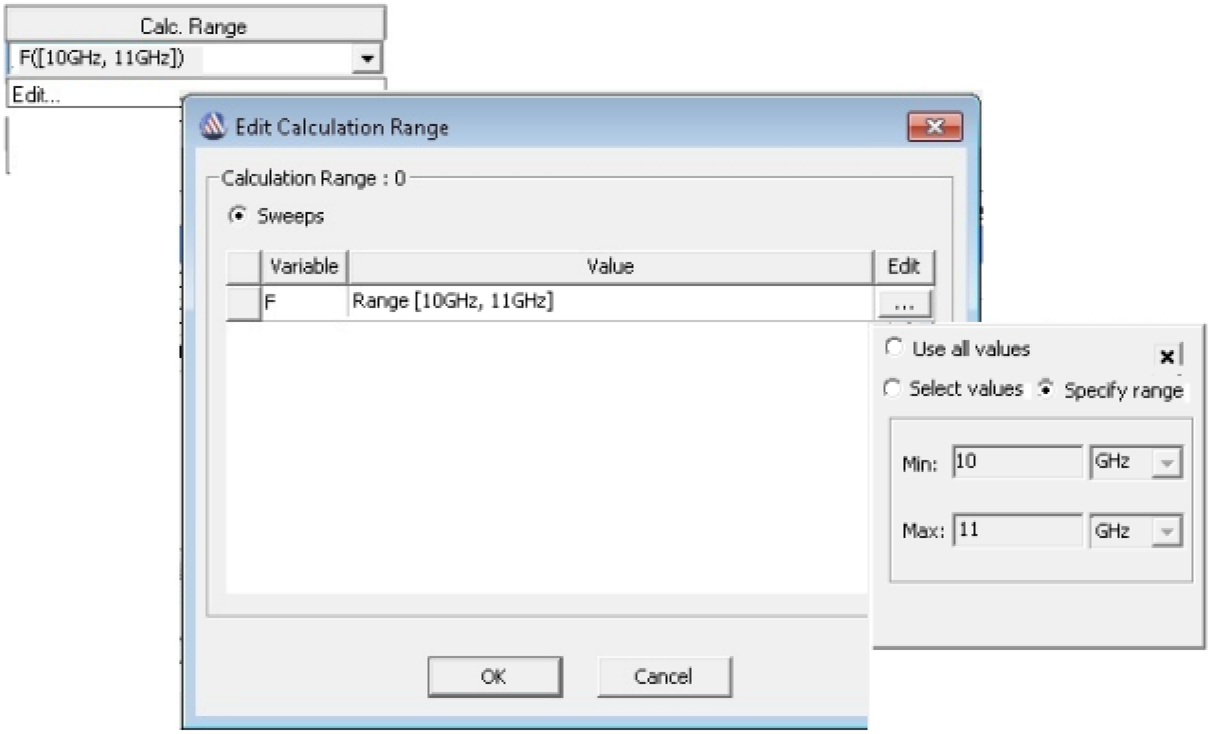
  For example, F(1GHz), cap(1pf, 1.2pf)

  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 **Circuit**> **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.

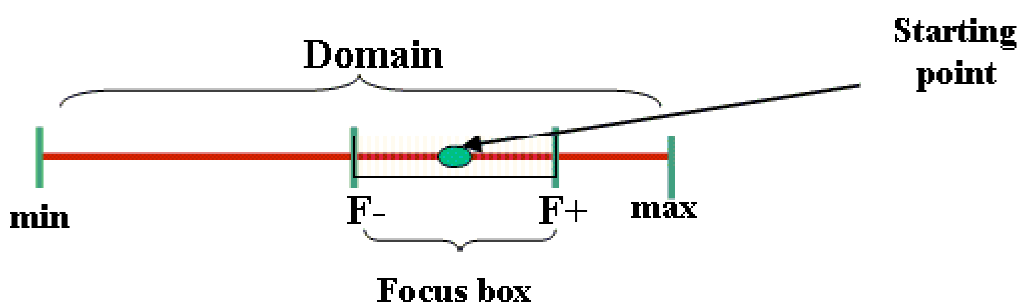
1. In the *Setup Optimization* dialog box, click the **Variables** tab.
2. Optimetrics displays **Min Step** and **Max Step** columns, with default values for each variable to be optimized.
3. In the **Min Step** text box, type the minimum step size value. Optionally, modify the unit system in the **Units** 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.

To Set the Min and Max Focus values

1. In the *Setup Optimization* dialog box, click the **Variables** tab.
2. Optimetrics displays **Min. Focus** and **Max. Focus** columns, with default values for each variable to be optimized.

   If you do not have an initial guess based on your knowledge of the problem, make the focus box equal to the domain; that is, the physical limits. This tells SNLP to search the entire decision space.

   - In the **Min. Focus** text box, type the minimum value of the focus range. Optionally, modify the unit system in the **Units** text box.
   - In the **Max. Focus** text box, type the maximum value of the focus range. Optionally, modify the unit system in the **Units** text box.
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.

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:

$\mathit{Calculation}_{(i)}\ \mathit{Condition}_{(i)}\ \mathit{Goal}_{(i)}\ \mathit{Weight}_{(i)}$

The cost function that the optimizer uses is built based on the norm setting as long as there are multiple goals and none of those use the "minimize" or "maximize" conditions. Thus, in this case the error associated with each individual goal (weighted) is combined in a way that is specific for each norm type chosen.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors:

$$Cost = \sum_{1}^{N} \left| w_i \varepsilon_i \right|$$

For **L2** norm the actual cost function uses the weighted sum of absolute values of the individual goal errors.

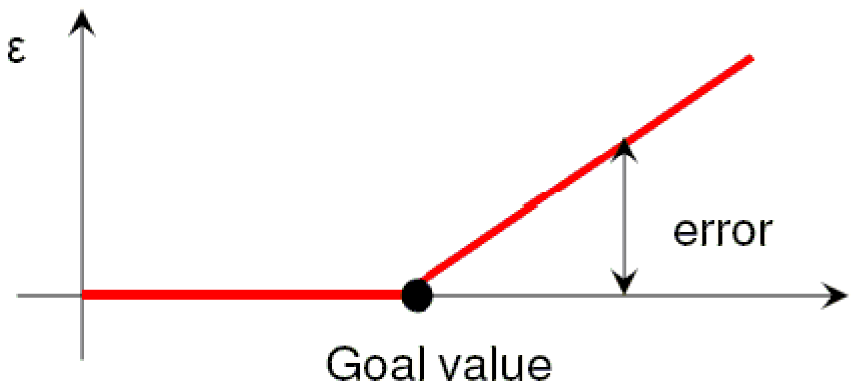
$$Cost = \sum_{1}^{N} w_i \varepsilon_i^2$$

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:

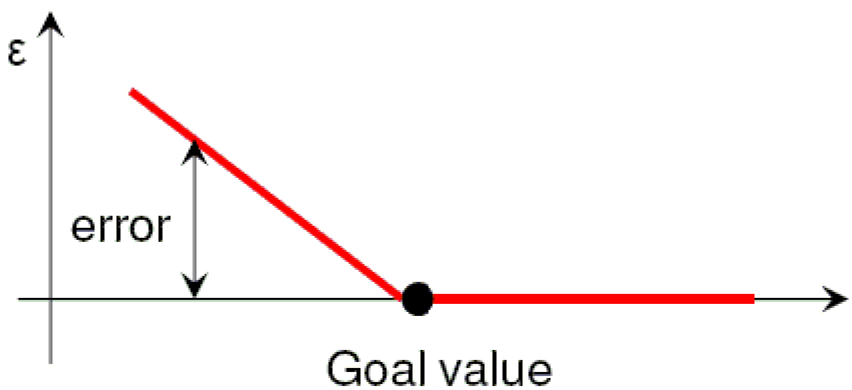
$$Cost = \underset{1}{\overset{N}{Max}} W_i \cdot \varepsilon_i$$

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.

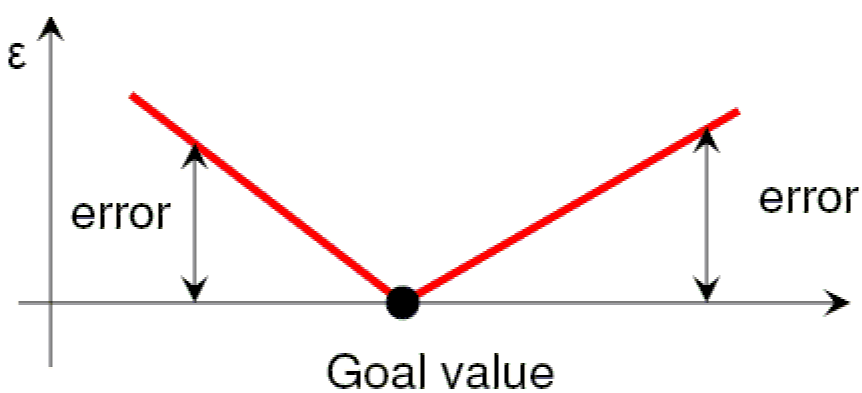
The graphical representation of the error is possible and depends upon the actual condition being used. If a "<" condition is used, the error can be represented as below:



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:

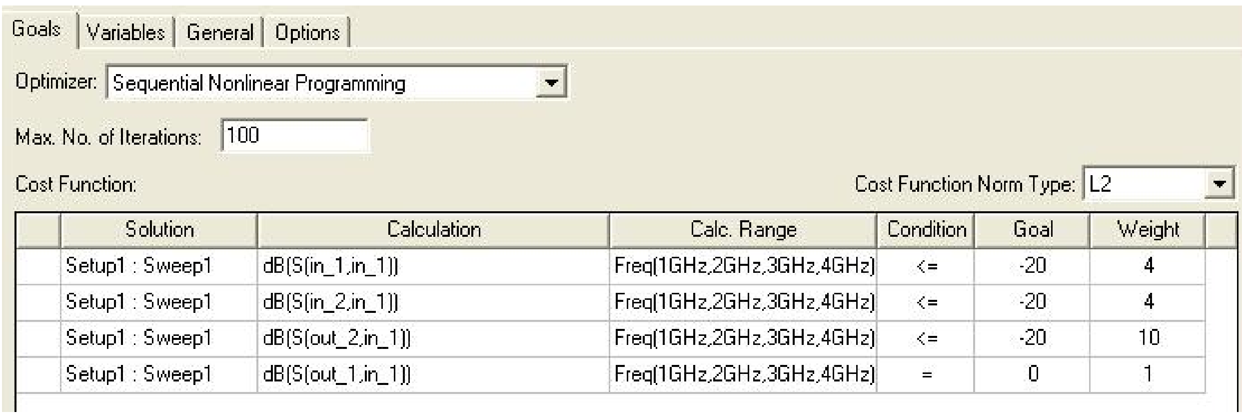


The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios. Note that when using "minimize" or "maximize" settings for the condition there should be a single goal setting which in this case coincides with the cost function.

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.

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.

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.

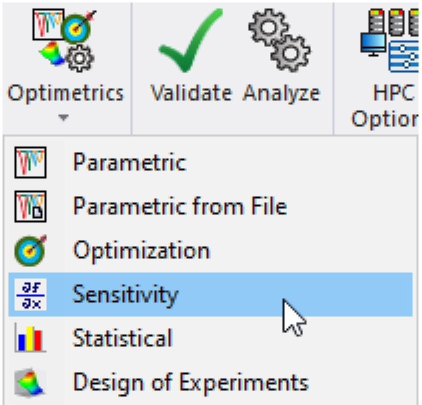
Selecting a Primary Output

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 **Circuit**> **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 **Circuit**> **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.

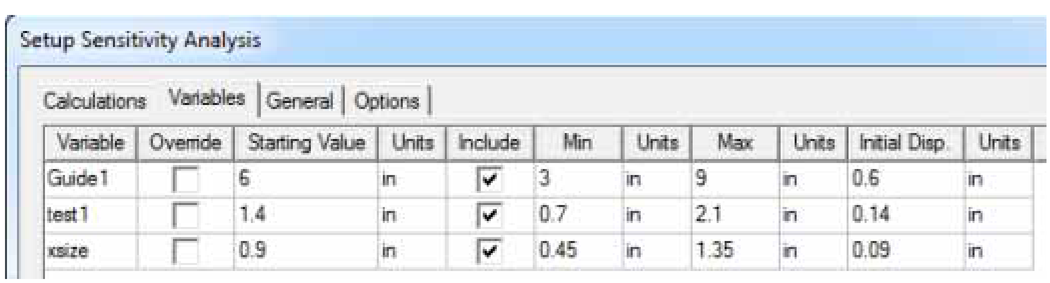
1. **Extreme value analysis:**

   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.**

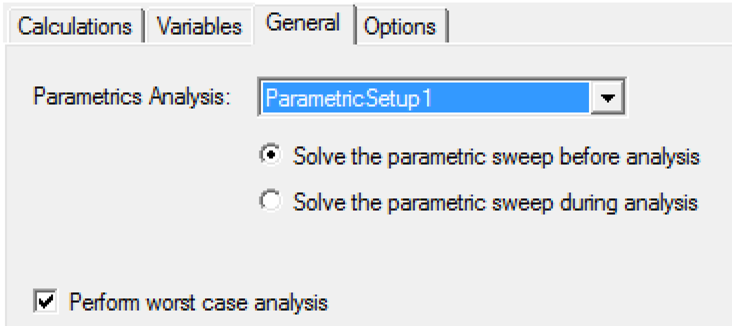
   - Setup:

     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.



Variable	Override	Starting Value	Units	Include	Min	Units	Max	Units	Initial Disp.	Units
Guide1		6	in	✔	3	in	9	in	0.6	in
test1		1.4	in	✔	0.7	in	2.1	in	0.14	in
xsize		0.9	in	✔	0.45	in	1.35	in	0.09	in

     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

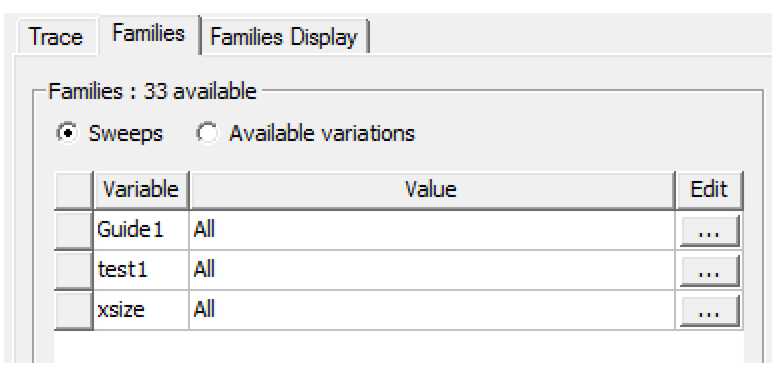
and Var1@maximum, Var2@minimum, Var3@minimum

- After setup, analyze your Sensitivity Setup under Optimetrics.
- To view worst-case result (upper bound only):

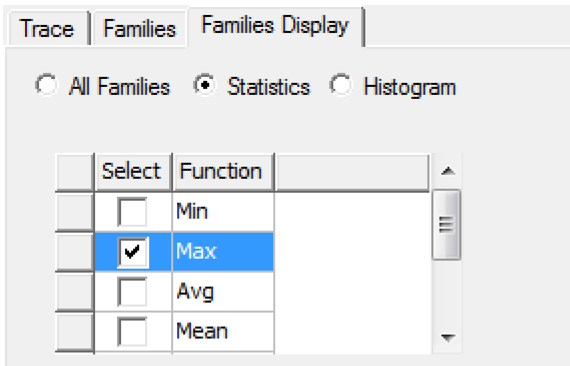
  Create a Data Table report.

  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**.



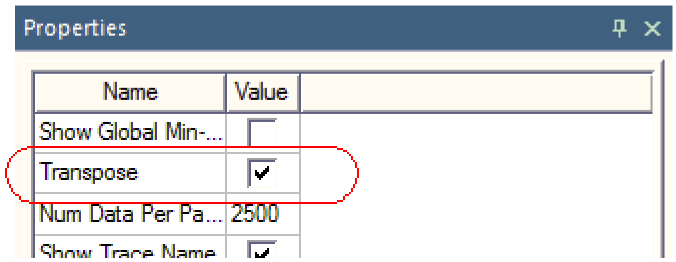
The associated data table shows the Max values.

	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.

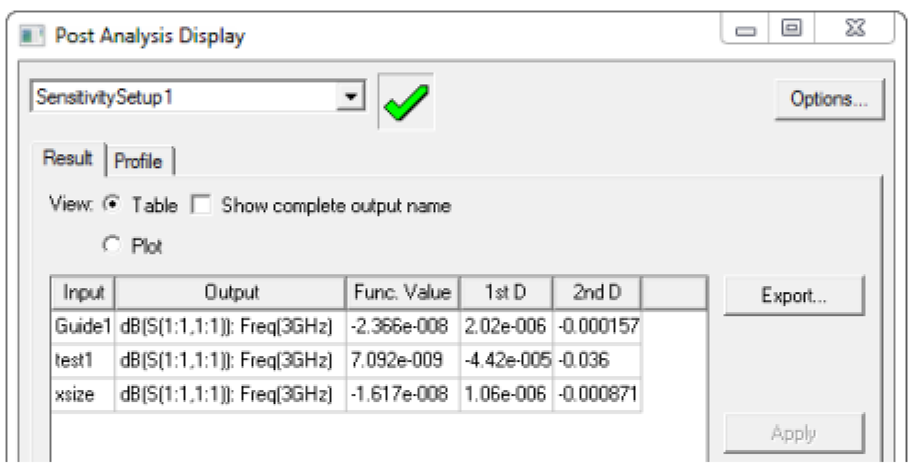
	Freq [GHz]	dB(S(1:1,1:1)) Setup1 : Sweep Guide1='3in' test1='2.1in' xsize='0.45in'
1	3.000000	-0.545922
2	4.000000	-2.300003
3	5.000000	-10.634331
4	6.000000	-4.586505
5	7.000000	-5.950253
6	8.000000	-9.019574
7	9.000000	-3.210097

> **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

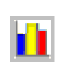
Statistical analysis allows you to explore the effects of random combinations of values of selected variables on selected global or local available analysis results. Therefore, before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical 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. The statistical analysis is currently not able to handle more than 30 variables during statistical analysis.

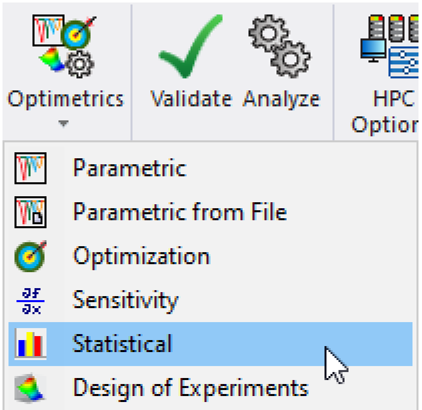
In addition to specifying the variables to be used in the statistical analysis and the parameters of the chosen distribution, the output quantities of interest also need to be specified. These quantities can be global ones such as previously defined parameters (Force/torque, inductance / capacitance, 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**.

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 **Circuit**> **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 Circuit to perform in the **Maximum Iterations** text box.
4. Specify a solution quantity to evaluate.
5. In the **Calculation** text box, set the value at which the solution quantity is to be computed.
6. Optionally, modify the distribution criteria to be used.
7. 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.

8. 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:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the **Geometry** from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
   - When selecting a geometry, you may also be required to specify a point within the geometry where the calculation is to be performed.
3. The **Output Variables** button opens the *Output Variables* dialog box allowing you to create special output variables to be used in the output parameter.
4. The **Calculation Expression** field in the **Trace** tab is used to enter the equation to be used for the solution quantities. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the output parameter.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculation Expression field.
   - Select a **Function** to apply to the value in the calculated expression.
   - For swept variables, the Range Function button opens the **Set 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.

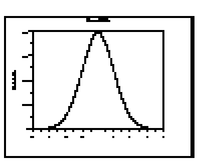
   If you changed the distribution type, the **Override** option is now selected. This indicates that the distribution type you selected is to be used for this optimization analysis; the current distribution type selected for the variable in the nominal design is ignored in this statistical analysis.

   - 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**, do the following:
   - Enter 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 **Circuit**> **Design Properties**, select **Statistics**.

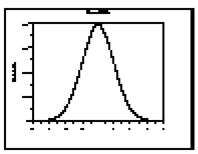
   If the variable is a project variable, do the following: Click **Project**> **Project Variables**.

   The *Properties* dialog box appears. Then, select **Statistics**.
3. Click in the **Distribution** column for the variable you want to change, and then select **Uniform**, **Gaussian**, **Lognormal**, or **User Defined**.
4. 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.

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.

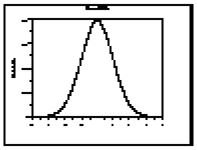


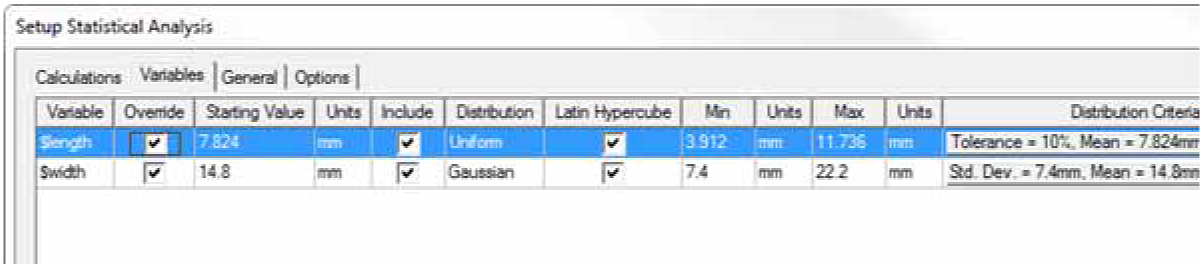
6. If the distribution type is **Uniform**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Type mean and tolerance values in the corresponding text boxes.

   The design variations are solved within the tolerance range of the starting value, using an even distribution.
7. 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.
8. 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.
9. 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.





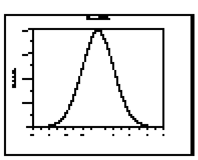
Uniform distributions such as variable "length" above use only the Tolerance value, and do not have a cutoff probability.

Edit Distribution

When setting the distribution type for a variable, you have the option of changing the distribution parameters from the default values.

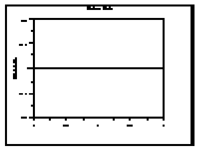
1. If the distribution type is **Gaussian**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Type the standard deviation of the distribution in the **Std Dev** text box.

   The design variations are solved using a Gaussian distribution within the specified mean and standard deviation values.

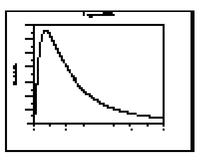
2. If the distribution type is **Uniform**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Enter the tolerance in the **Tolerance** text box.

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:

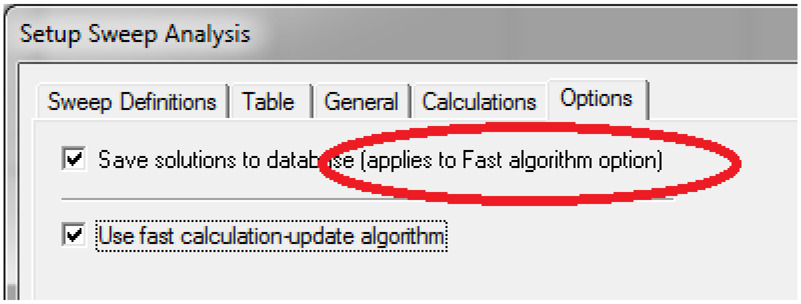
1. In the *Setup Statistical Analysis* dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve during the statistical analysis from the **Parametric Analysis** drop-down menu.
3. Select **Solve the parametric sweep during analysis**.

Using the Fast Calculation-Update Algorithm

A fast calculation-update algorithm is available to speed up Optimetrics and report updates during Optimetrics analyses. The fast calculation-update algorithm will generate the same Optimetrics results, only faster, and is available for all Optimetrics analyses *except* a Tuning analysis. By default, the fast calculation-update option is automatically enabled whenever it is applicable, but you can configure it manually using the **Options** tab of the applicable Optimetrics *Setup* dialog box. To enable the fast calculation-update algorithm:

1. From the menu bar, click your product and then **Circuit> Optimetrics Analysis> Add<***Optimetrics Type*>.

   A typical dialog box to *Setup Sweep Analysis* is shown below with the **Options** tab selected.

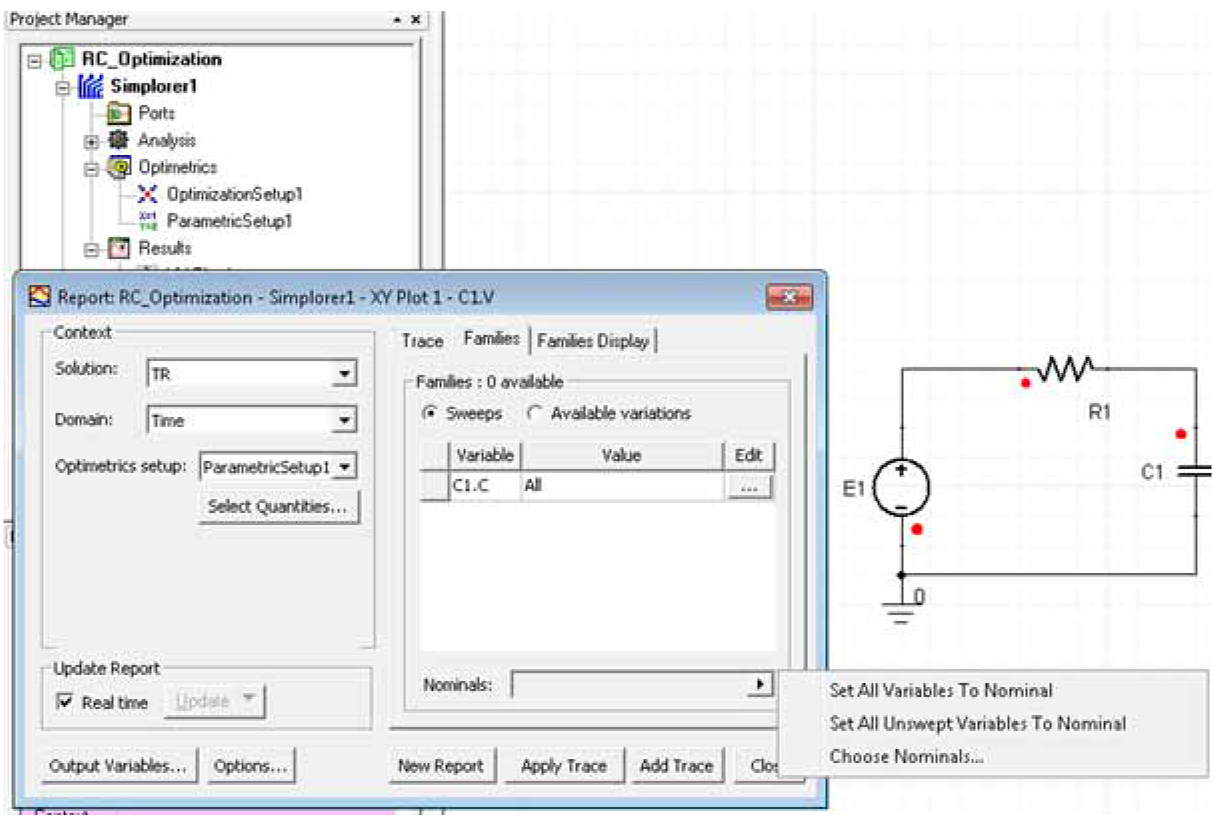


The *Options* tab is the same for all optimization type setups.

2. Select **Use fast calculation-update algorithm** to enable use of the algorithm. (See also, *Fast Calculation-Update Algorithm Limitations*.)

   When the fast calculation-update algorithm is enabled:

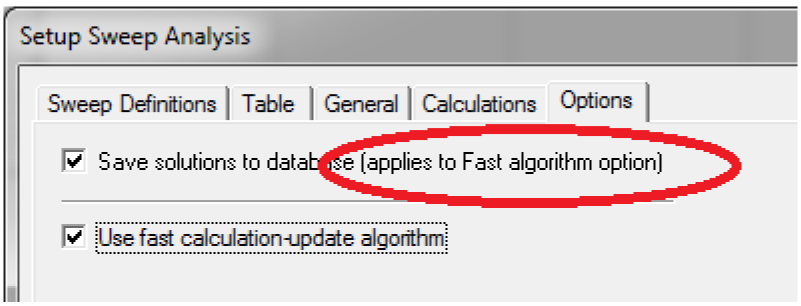
   - You can enable some reports to be updated automatically during the optimetrics analysis if you **Set All Variables To Nominal** in the Report/Trace setup dialog box:



- You can see each trace (overwriting the previous), by setting the **Optimetrics setup** in the Report dialog box to **None** in addition to having all variables set to Nominal. At the end of the analysis, the user will see the last calculated value.

3. If you have enabled **Use fast calculation-update algorithm** and want to save the solution data for every solved design variation in the Optimetrics analysis, select **Save solutions to database** as shown below. Selecting this option has no effect without enabling **Use fast calculation-update algorithm**.

> **Note:**
>
> Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.



- When the **Save solutions to database** option is checked, a plot with traces based on the Optimetrics Setup just run can be updated through a menu command. Right-click the desired report under *Results* in the Project Manager and select **Update Report**) to show results as appropriate (including a Family of results, if chosen).
- Without the **Save solutions to database** option checked, you can examine analysis data in the *Post Analysis Display* dialog box, which is available by right-clicking the Optimetrics Setup and choosing **View Analysis Result**.

Fast Calculation-Update Algorithm Limitations

The fast calculation-update algorithm **cannot** be used if **any** optimetrics calculation uses:

- **Project/Design variables** – If the project/design variable is ***not*** swept in the Optimetrics analysis, and you would like to use it in the expression: You can create an output variable for the Project/Design variable. Assign the value of the Project/Design variable to the output variable and use the output variable in the expression instead of a constant numerical value.
- **More than one range function** – For example, when range function is not the outermost function or when range function takes arguments.
- **More than one calculation range**, or the calculation range is not for primary sweep.

Similarly, when the fast calculation-update algorithm is enabled, a trace cannot be updated during an analysis if the trace expression uses:

- **Project/Design variables**. (However, you can use the same workaround described above.)
- **Any range function.**

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.

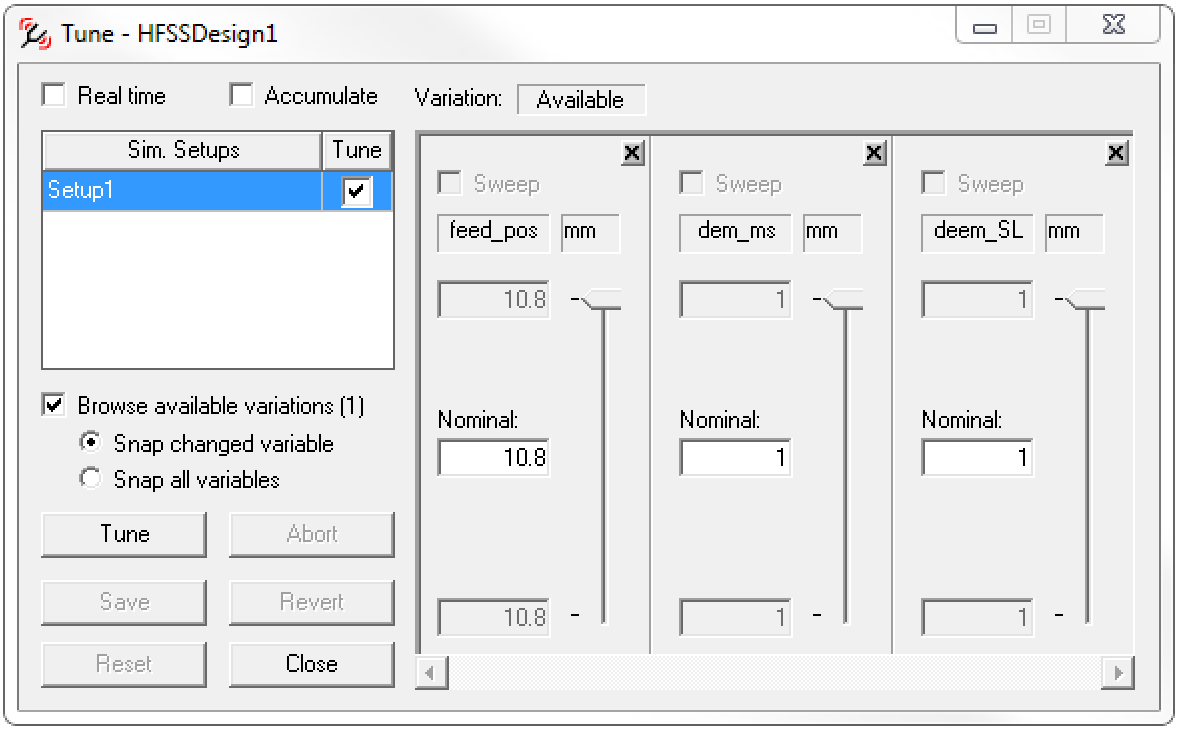
Tuning a Variable

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.

Applying a Tuned State to a Design

You can apply the variable values solved during a tuning analysis to the nominal design in one of the following three ways:

- When closing the **Tune** dialog box:
  1. Click **Close** to exit the **Tune** dialog box.

     The **Apply Tuned Variation** dialog box appears.
  2. Click the design variation you want to apply and then click **OK**.

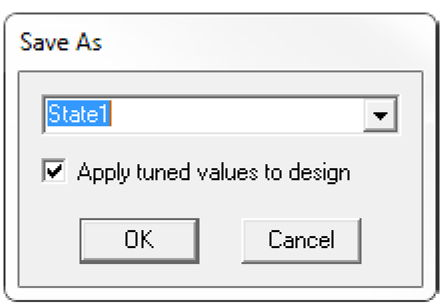
  The variable values from the solved design variation become the current variable values for the nominal design. If you have applied variant values, you should see the new values listed in the relevant Design or Project Properties lists of variables and values, and if the changes affect plots or physical features of a model, those changes should also be apparent.
- When saving a tuned state.
- When reverting to a tuned state.

Saving a Tuned State

You can save the settings in the **Tune** dialog box, including the variable values you specified for a tuning analysis. Saved states are only available during the current session of the **Tune** dialog box; they are not stored for the next session.

1. After tuning a variable, click **Save** in the **Tune** dialog box.

   A **Save As** dialog box appears.

2. Type a name for the tuned state in the text box.

3. Select **Apply tuned values to design** if you want to update the model to the new variable values.
4. Click **OK** to return to the **Tune** dialog box.

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.

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.

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 pre-defined 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.

Excluding a Variable from an Optimetrics Analysis

To exclude a variable from being optimized or included in a sensitivity or statistical analysis:

1. Do one of the following:
   - In the **Setup Optimization** dialog box, click the **Variables** tab.
   - In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
   - In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the independent variables that were selected for the optimization analysis are listed.
2. Clear the **Include** option for the variable you want to exclude from the analysis.

   The **Override** option is now selected. This indicates that, for this optimization analysis, the variable is not included.

   > **Note:**
   >
   > Alternatively, you can select the **Override** option first, and then clear the **Include** option for the variable you want to exclude.
3. Click **OK**.

Modifying the Value of a Fixed Variable

If you are not including a variable in an optimization, sensitivity, or statistical analysis, Optimetrics uses that variable's current value during the analysis.

To override the current value of a fixed variable for an Optimetrics setup:

1. Do one of the following:
   - In the **Setup Optimization** dialog box, click the **Variables** tab.
   - In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
   - In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.
2. Click **Set Fixed Variables**.

   The **Setup Fixed Variables** dialog box appears. Under **Fixed Variables**, all of the current independent variable values are listed.
3. Click the **Value** text box of the variable with the value you want to override.
4. Type a new value in the **Value** text box, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this Optimetrics setup; the current variable value set for the nominal design is ignored.

> **Note:**
>
> Alternatively, you can select the **Override** option first, and then type a new value in the **Value** text box.

5. Optionally, click a new unit system in the **Units** text box.
6. Click **OK**.

To revert to a default variable value, clear the **Override** option.

Linear Constraints

Once the optimization variables are specified, the optimizer handles each of them as an *n*-dimensional vector *x*. Any point in the design space corresponds to a particular *x*-vector and to a design instance. Each design instance may be evaluated via Finite Element Analysis and assigned a cost value; therefore, the cost function is defined over the design space:

$$\mathrm{cost}(\mathrm{x}) : R^n \rightarrow R$$

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.

Running an Optimetrics Analysis

Once you have created all necessary Optimetrics based analyses, you have several options for running the simulations.

- To use the **Analyze All** command at the Project or design level to simulate the nominal problem and subsequently run all Optimetrics setups, do the following:
  1. In the Project Manager window, right-click on the **project** or **design** name.
  2. Click **Analyze All** from the shortcut menu.
- To use the **Analyze All** command from the Optimetrics menu to simulate only the Optimetrics based setups, do the following:
  1. In the Project Manager window, right-click on **Optimetrics**.
  2. Click **Analyze** > **All** from the shortcut menu.
- You can choose to analyze only the setups related to a specific Optimetrics type of analysis. In order to simulate setups of a specific type, do the following:
  1. In the Project Manager window, right-click on **Optimetrics**.
  2. Click **Analyze** > **All{*TYPE*}** from the shortcut menu, where *TYPE* is the specific analysis type of interest, Parametric, Optimization, Sensitivity, or Statistical.

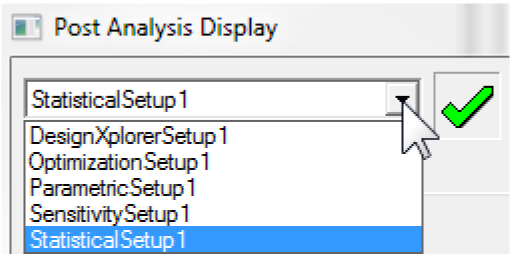
Viewing Analysis Results for Optimetrics Solutions

To view data specific to an Optimetrics solution:

- In 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.

- Use the drop-down menu to select from available setups.



- 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.

- 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.
- 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 **Circuit** 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 Circuit 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.

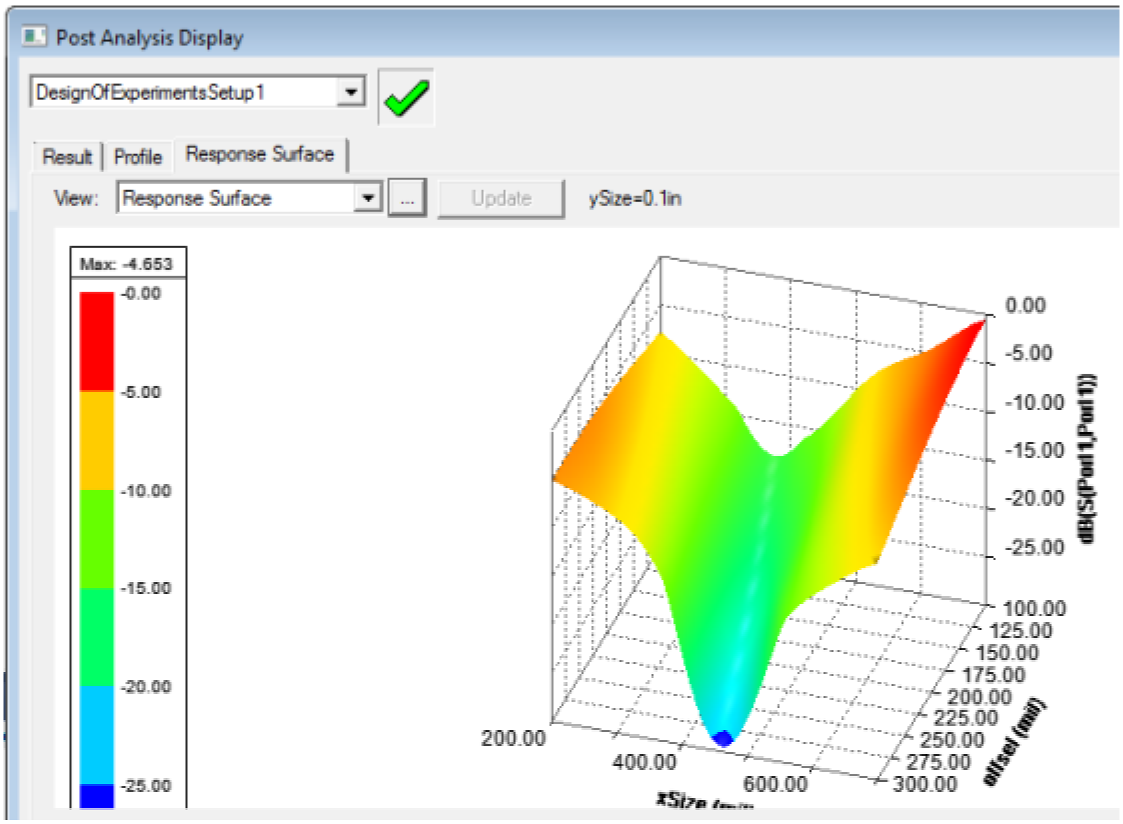
Using Design of Experiments

Design of Experiments (DOE) is a technique used to scientifically determine the location of sampling points and is included as part of the Response Surface, Goal Driven Optimization, and Analysis systems. Design of Experiments plus a mathematical approximation of output parameters lets you:

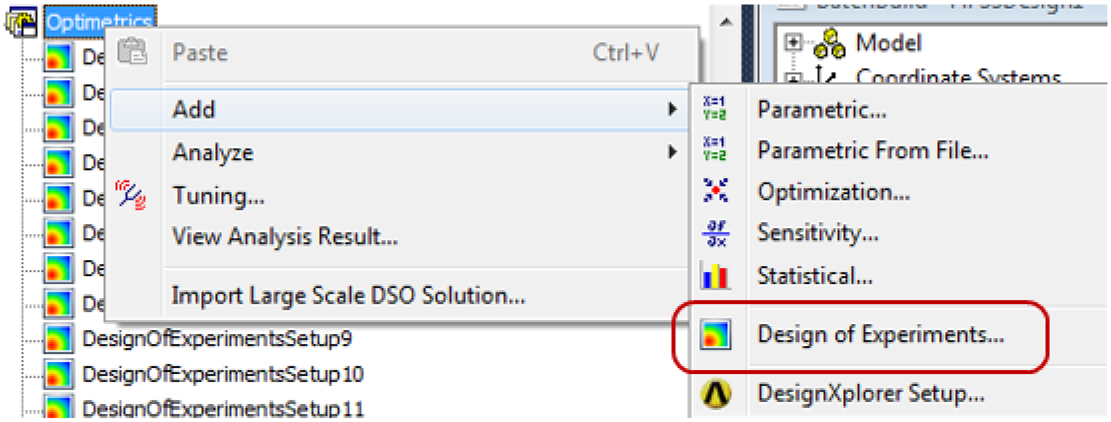
- Reduce the number of simulations
- Interactively explore the design space before running optimization

Design of Experiments describes the relationship between the design variables and the performance of the product by using Design of Experiments (DOE), combined with response surfaces. DOE and response surfaces provide all of the information required to achieve Simulation Driven Product Development. Once the variation of the performance with respect to the design variables is known, it becomes easy to understand and identify all changes required to meet the requirements for the product.

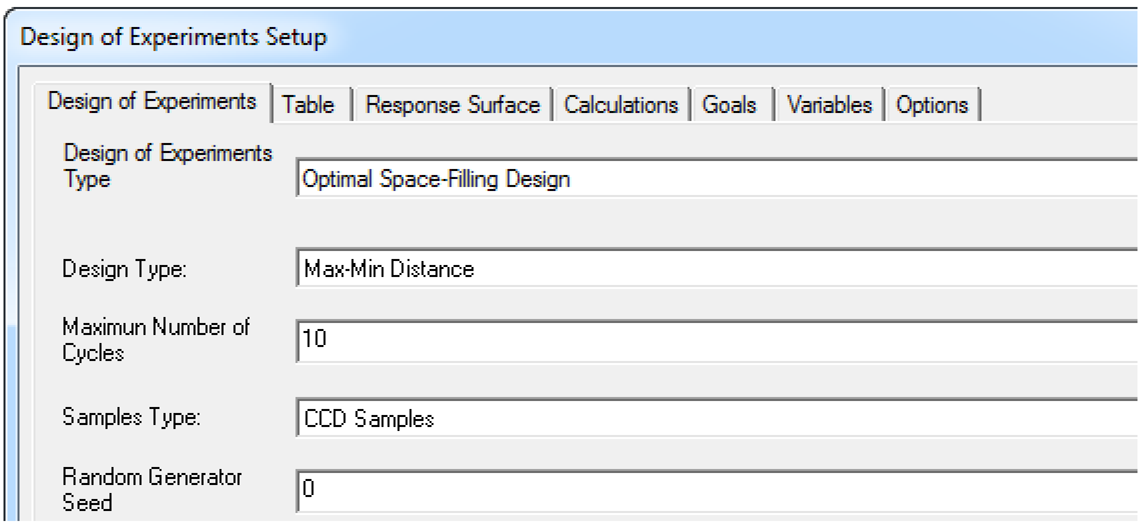
The goal is to create a response surface by interpolating through calculated points (a best curve fit). For each design, you can create a response surface for each output parameter. Once the response surfaces are created, you can share the information can in easily understandable terms: curves, surfaces, sensitivities, etc. They can be used at any time during the development of the product without requiring additional simulations to test a new configuration.



The Design of Experiments feature is integrated inside Electronics Desktop. Combined with Electronics Desktop's distributed solve feature, you can build the response surfaces from the DOE variation table much faster.



Selecting a Design of Experiments under Optimization opens a dialog with several tabs:



In the Design of Experiments setup, you select the DOE type, select the Response Surface, specify goals, view and include variables.

There are a wide range of DOE algorithms or methods available in engineering literature. These techniques all have one common characteristic: they try to locate the sampling points such that the space of random input parameters is explored in the most efficient way, or obtain the required information with a minimum of sampling points. Sample points in efficient locations only reduce the required number of sampling points and increases the accuracy of the response surface generated. For more information on the available types of DOE, see Design of Experiments Types.

Once you have set up your input parameters, you can update the DOE, which submits the generated design points to the analysis system for solution. Design points are solved simultaneously if the analysis system is set up to do so; sequentially, if not. After the solution is complete, you can update the Response Surface cell, which generates response surfaces for each output parameter based on the data in the generated design points.

> **Note:**
>
> Requirements and recommendations regarding the number of input parameters vary according to DOE type. For more information, see Number of Input Parameters for DOE Types.

If you change the Design of Experiments type after doing an initial analysis and preview the Design of Experiments Table, any design points generated for the new algorithm that are the same as design points solved for a previous algorithm will appear as up-to-date. Only the design points that are different from any previously submitted design points need to be solved.

You should set up your DOE Properties before generating your DOE Design Point matrix. The following topics describe setting up and solving your Design of Experiments, and viewing the results.

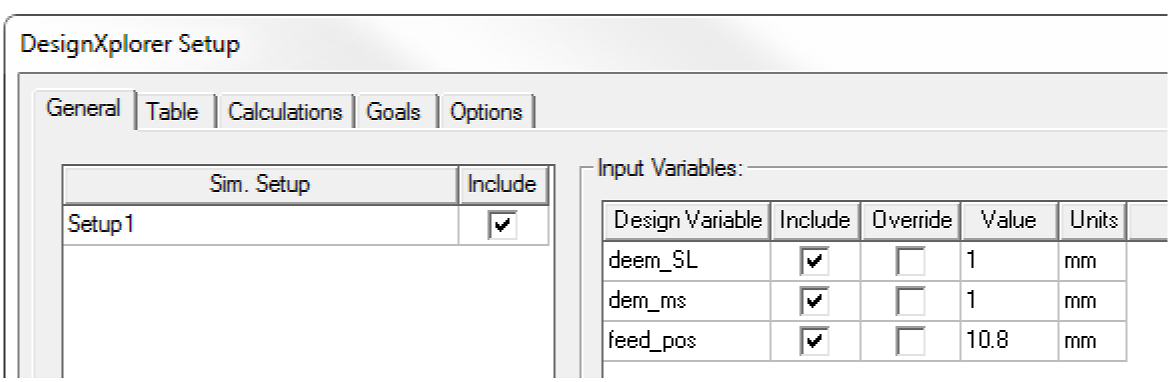
Link to DesignXplorer

You can export a .xml file containing information on a Circuit 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.

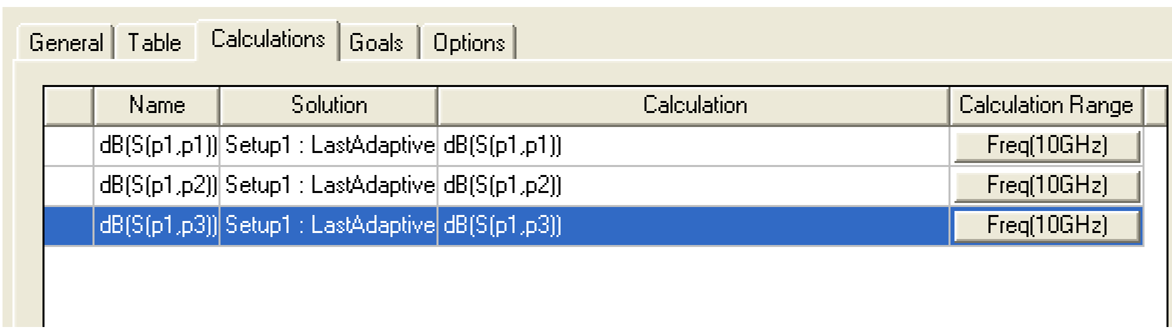
   This opens the **Design Xplorer** dialog box with the **General** tab selected. it lists the setups available in the current project, and the input variables it contains.



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.

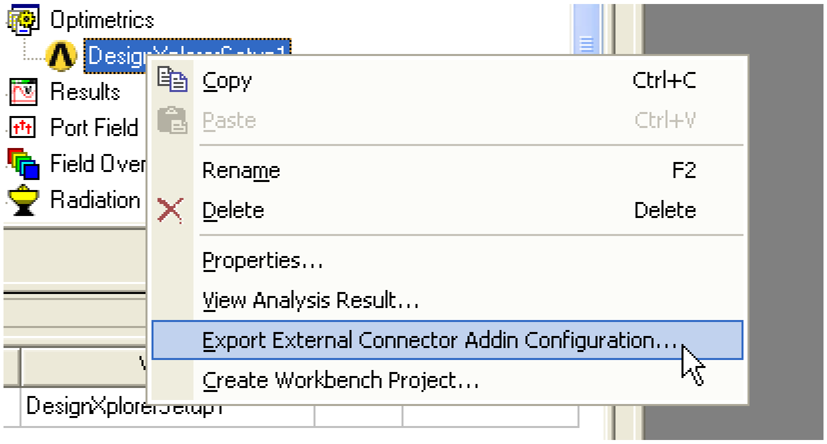


5. When you have added the calculations of interest, click **OK** to save the setup.

   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**.



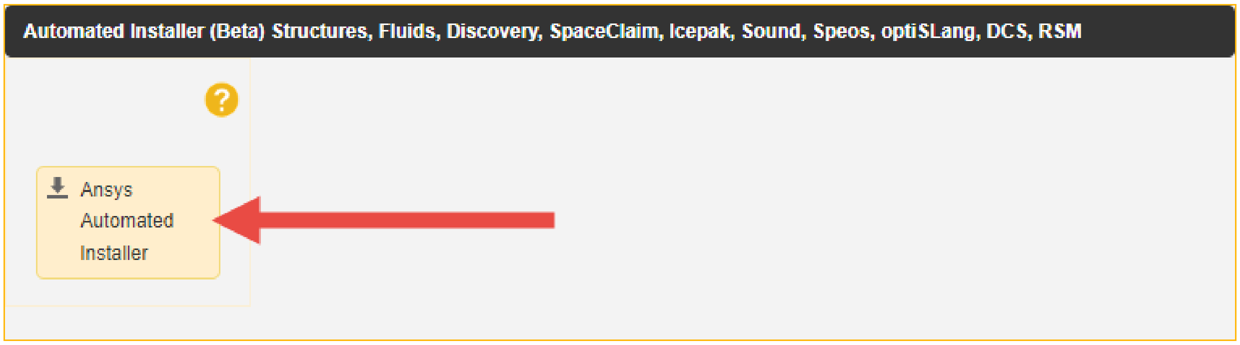
This displays a browser dialog that you can use to navigate your file system and name and saves the .xml file. This file contains information regarding the path along with the setup, variables, and simulation results that you specified.

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.

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/v242/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:

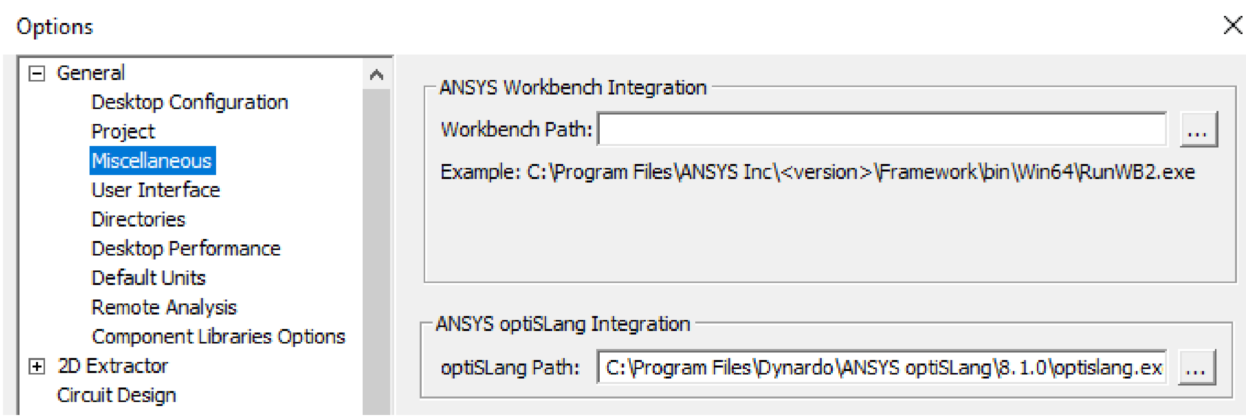
If the original Ansys EDT node and its copies are all kept in the optiSLang Setup run mode, they will remain linked to the specific optiSLang setup existing in the referenced AEDT project. The optiSLang project can contain and execute multiple algorithm systems, which means that the linked optiSLang 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)

8 - 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. Circuit 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.

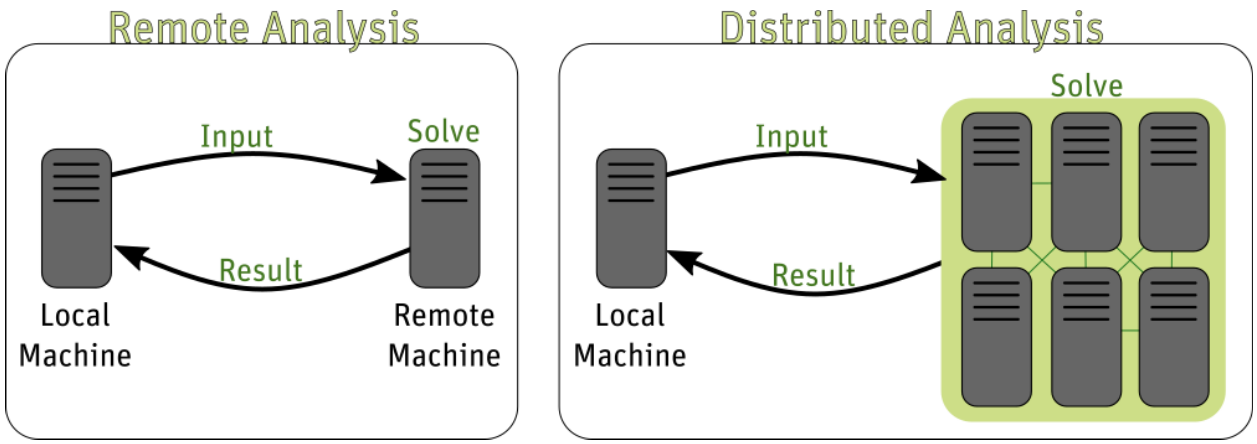
See the following topics for more information and configuration tips:

- 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).

   The list of currently supported HPC software includes:

   - Platform's Load Sharing Facility or LSF
   - Altair's PBS
   - Sun GridEngine

- Microsoft® Windows® Compute Cluster Server 2003
- Microsoft® Windows® HPC Server 2012 R2

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 2024 R2** > **Register withRSM**. You can also run **RegisterEnginesWithRSM.exe**, located in the product subdirectory (for example, C:\Program Files\AnsysEM\v242\Win64\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, /apps/AnsysEM/v242/RegisterEnginesWithRSM.pl).

     If the RSM service cannot run due to permission issues for the configuration file, it issues an error message and exits.

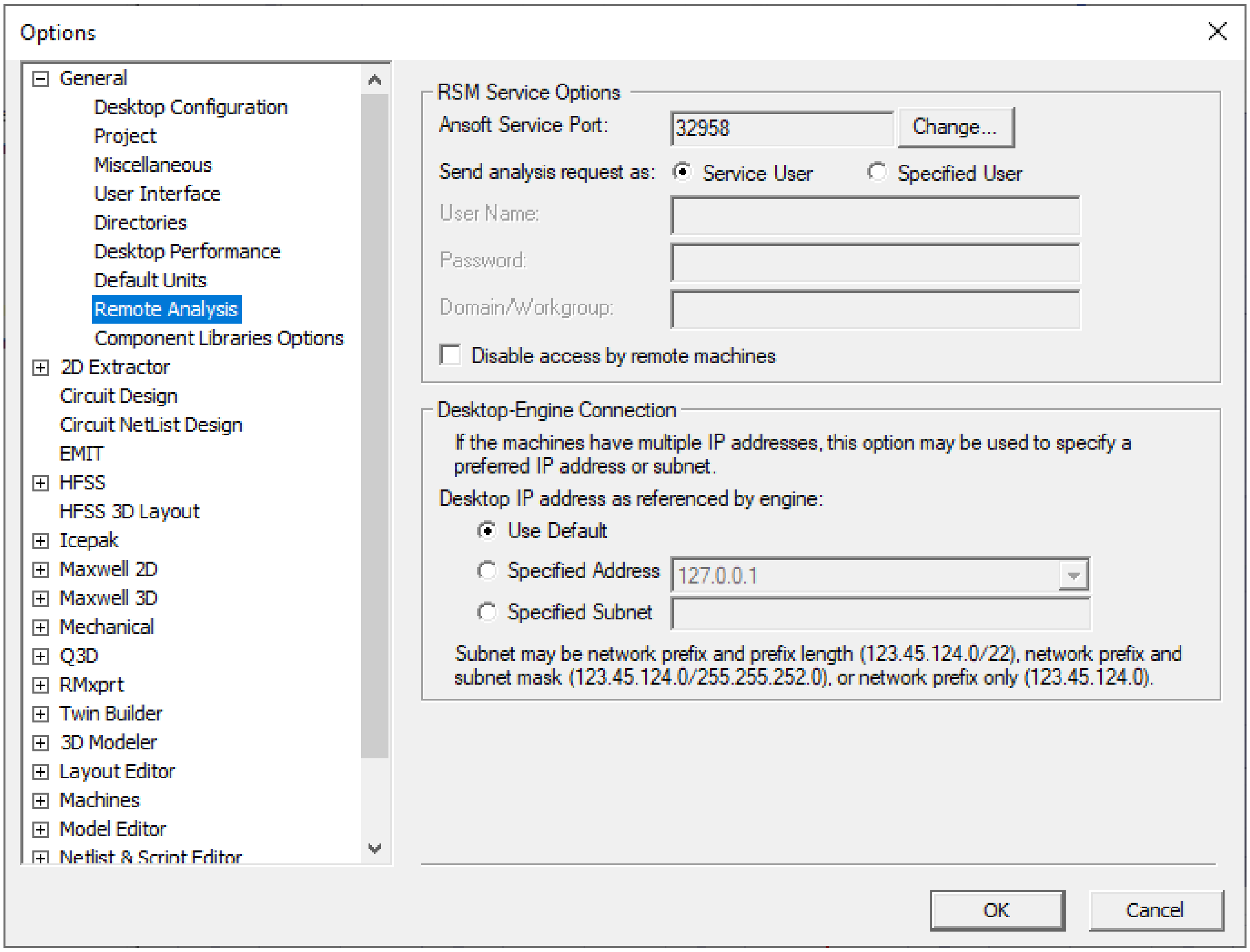
     If your product is not registered with RSM, the analysis will run locally.

Configuring the Local Machine to Solve Remotely

To set Circuit 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 2024 R2** > **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 2024 R2** > **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.

   To do this:

   1. Go to the rsm subdirectory of the Remote Simulation Manager installation directory, <RSM installdir>/rsm.
   2. Type
      `./ansoftrsmservice status`
   3. 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.

Circuit 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:

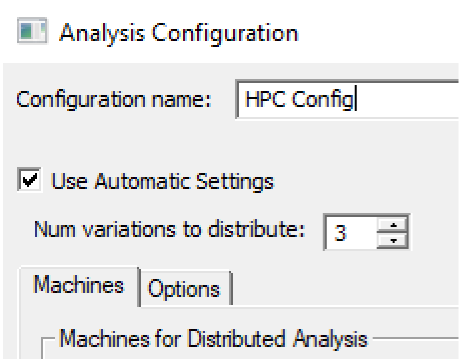
Communication between machines in remote analysis and distributed analysis can drastically affect performance. Use of a high-speed network system, like Gigabit or Infiniband, is recommended for optimal performance.

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 *Ansys Electronics Desktop Installation Guide* for details on installing MPI.

> **Note:**
>
> MPI is not supported currently for Circuit designs. Therefore, distributed solutions are limited to parametric analyses, which do not use MPI. This limitation is not to be confused with solving multiple threads of a process on a single multi-core computer. Parallel processing of this type is supported by the Circuit solvers.

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.

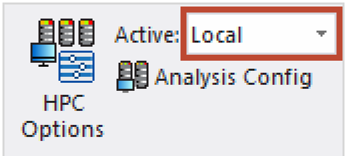
Configure a Distributed Analysis

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

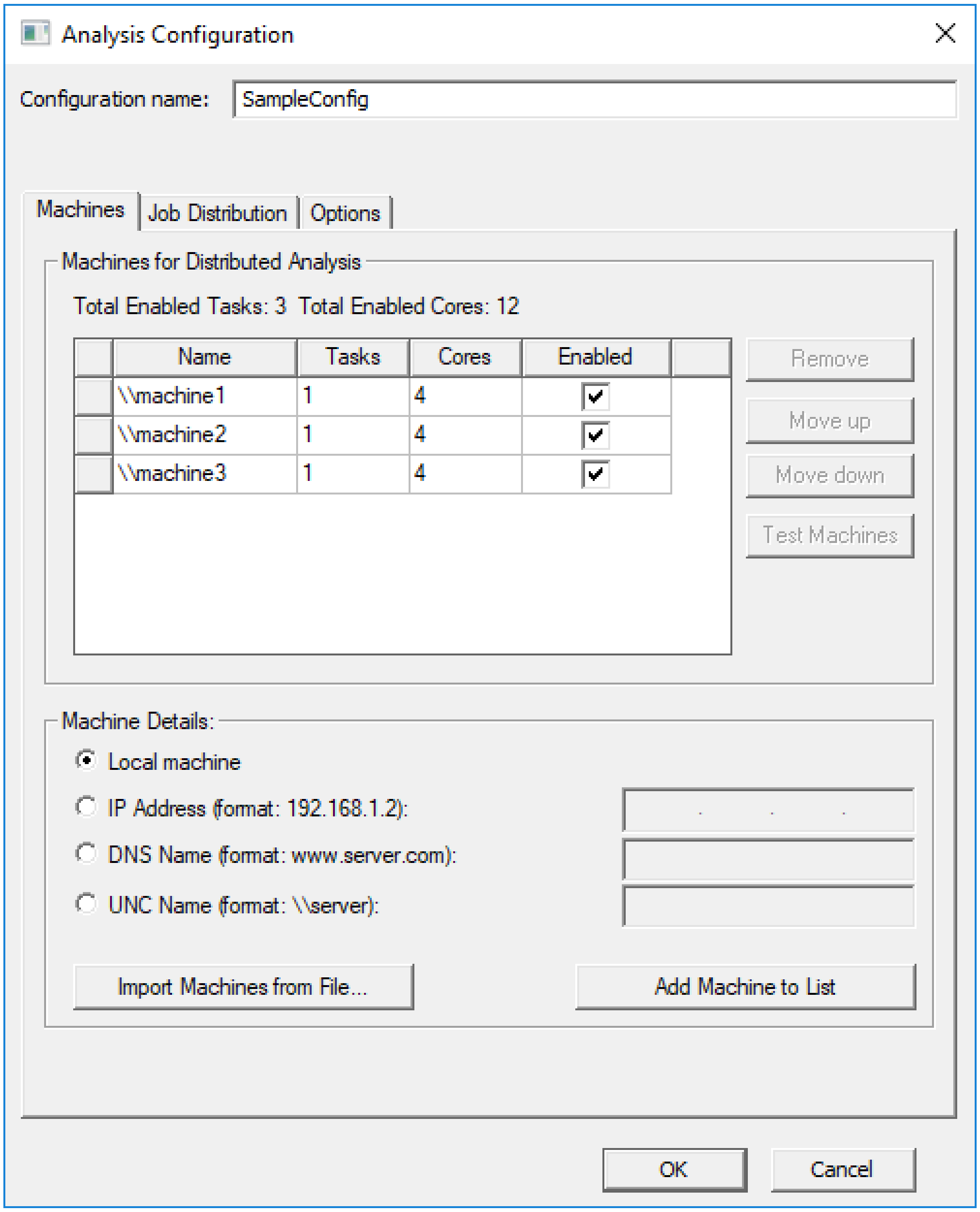
After you have selected the active distributed machine configuration, you can edit its configuration in the **Analysis Configuration** window.

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, 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:

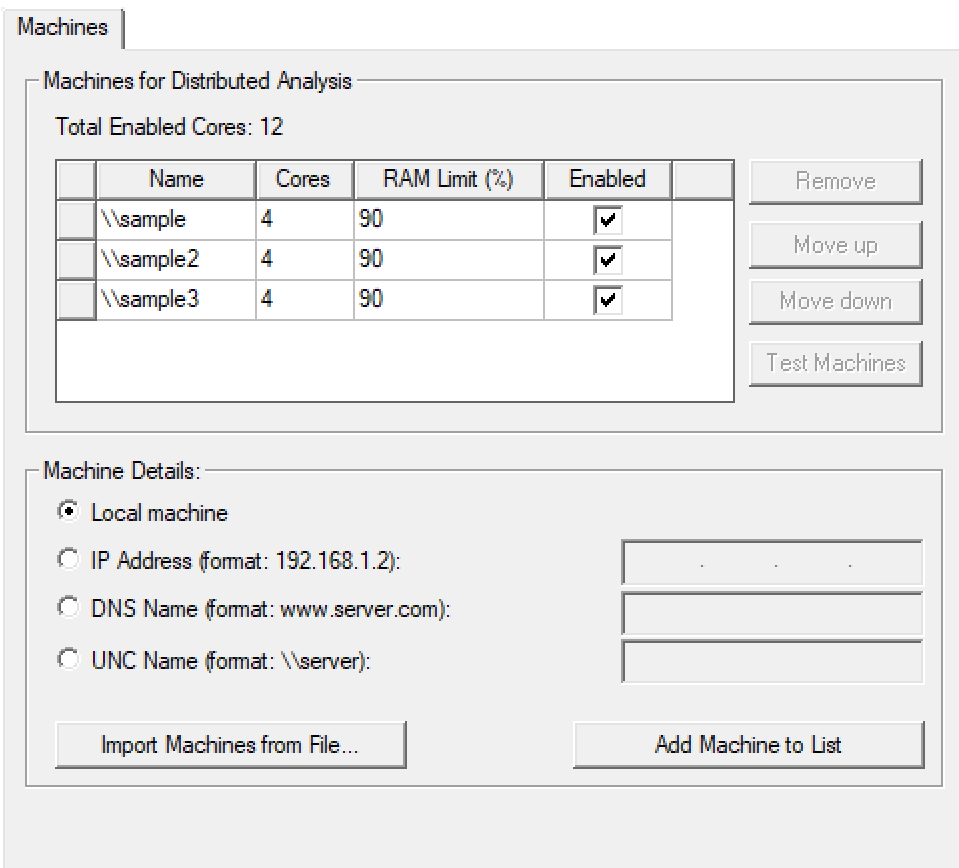
Available options may vary slightly depending on the solver and design.

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**.

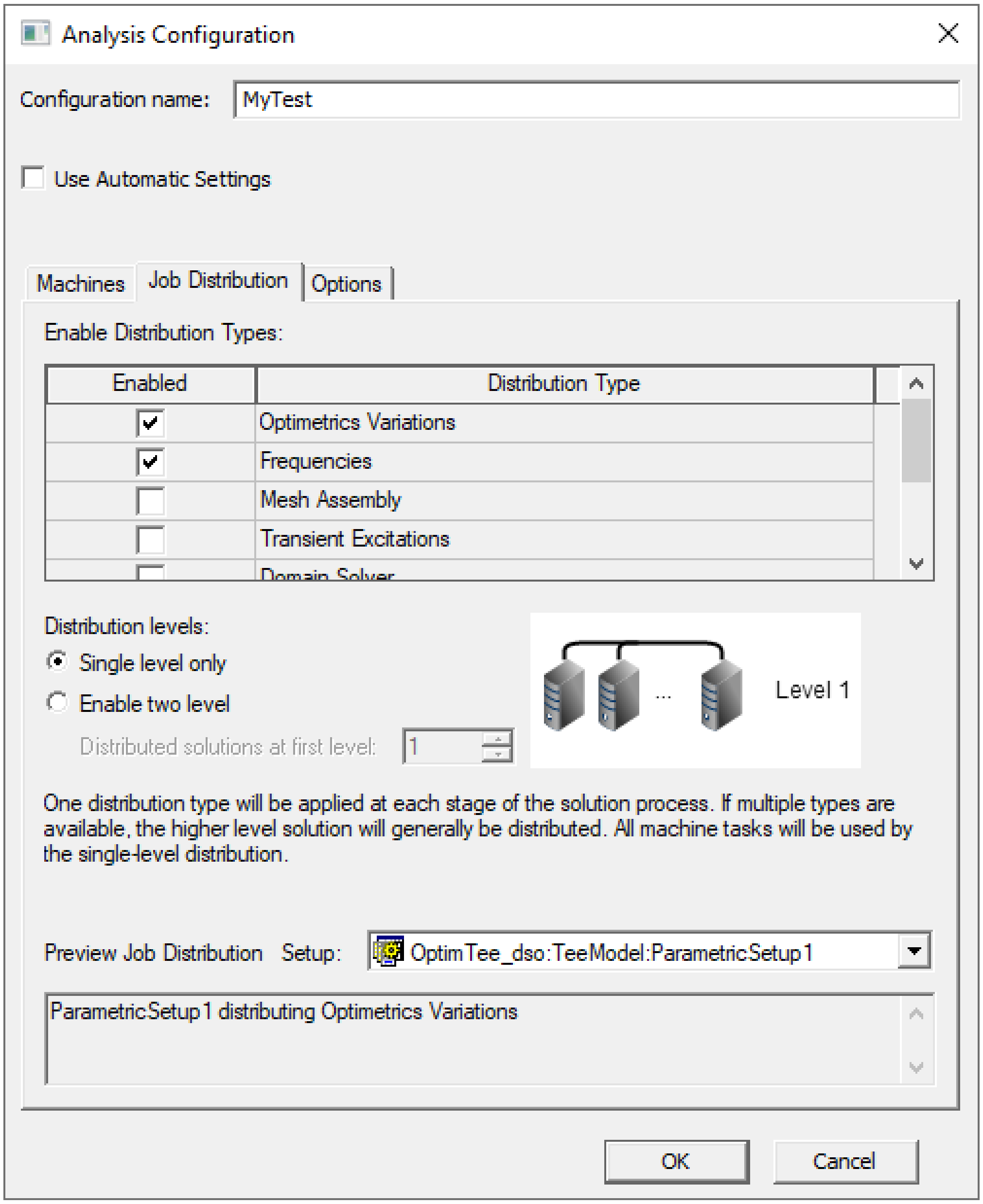
The **Test Machines** dialog box appears. The test goes through the current machine list and gives a report on the status of each machine. A progress bar shows testing progress. When the test is complete, click **OK** to close the dialog box. If you need to disable or enable machines based on the report, 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, Circuit automatically selects the best distribution type from the enabled types.

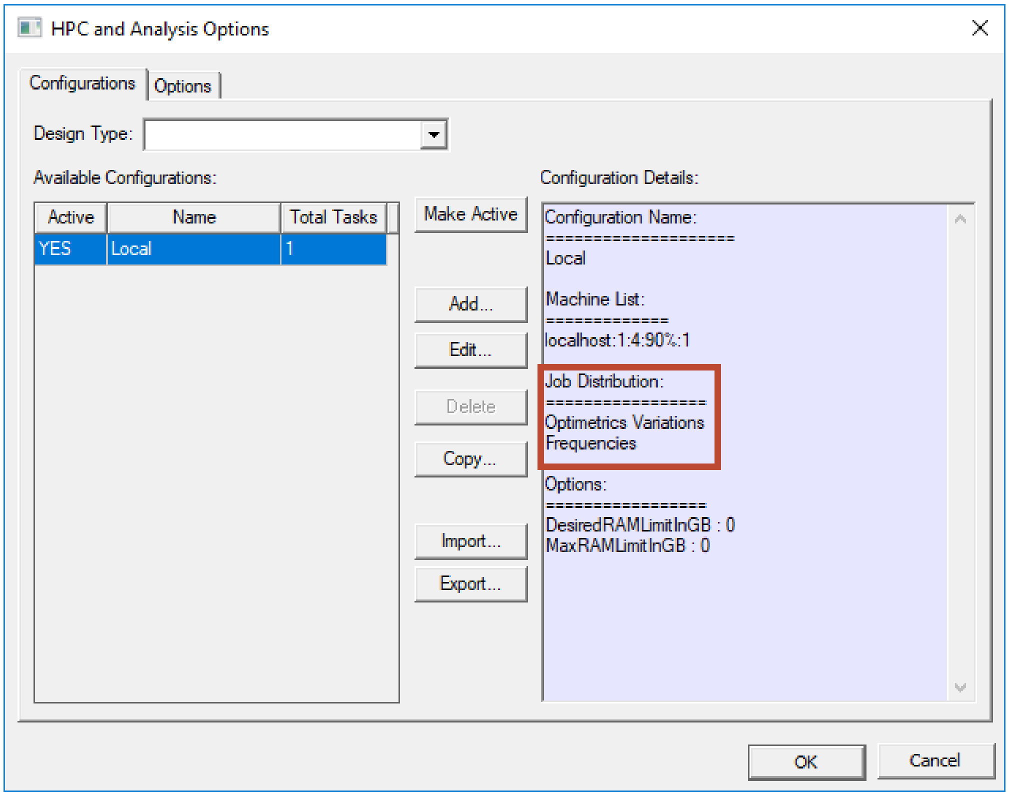
Enabling a distribution type does not mean it will be used. It must be also allowed by the solve setup. If you enable a distribution type for a given setup, and distribution is allowed, the preview window updates to describe the distributions. Note that enabled distribution types apply to all setups 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.

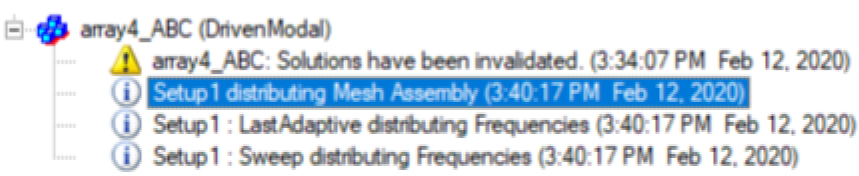
- If there's only one single mesh or if curvilinear contact exists: then normal initial mesh workflow with single G3dmesher will be launched.
- If there are more than one individual mesh and no curvilinear contact exists:
  - If Use Automatic Setting is not selected, under the **Job Distribution** tab, 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.

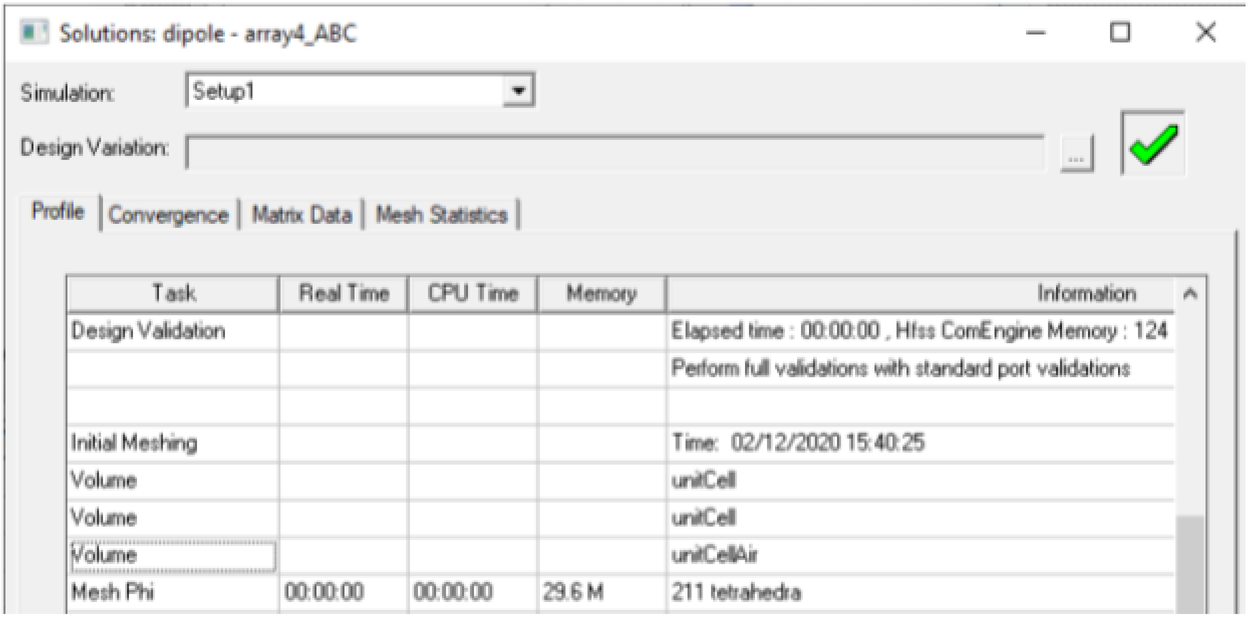
Enabled distribution types are listed when you select a configuration in the **HPC and Analysis Options** window.



When you run a simulation, the **Messages** window describes distributions.



When you view the Solution Profile, distributions for unitCell show as parallel Volume Tasks.



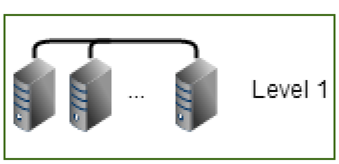
Distribution Levels

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(for Technical details, see Multiprocessing and the Iterative Solver)
- 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.

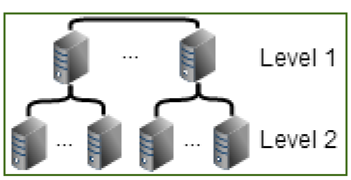
When multiple distribution types are available, the higher-level solution will generally be distributed. For example, when both Optimetrics Variations and Iterative Solver Excitations are enabled, Optimetrics Variations will be distributed. When both Optimetrics and Mesh Assembly, are selected Optimetrics 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.



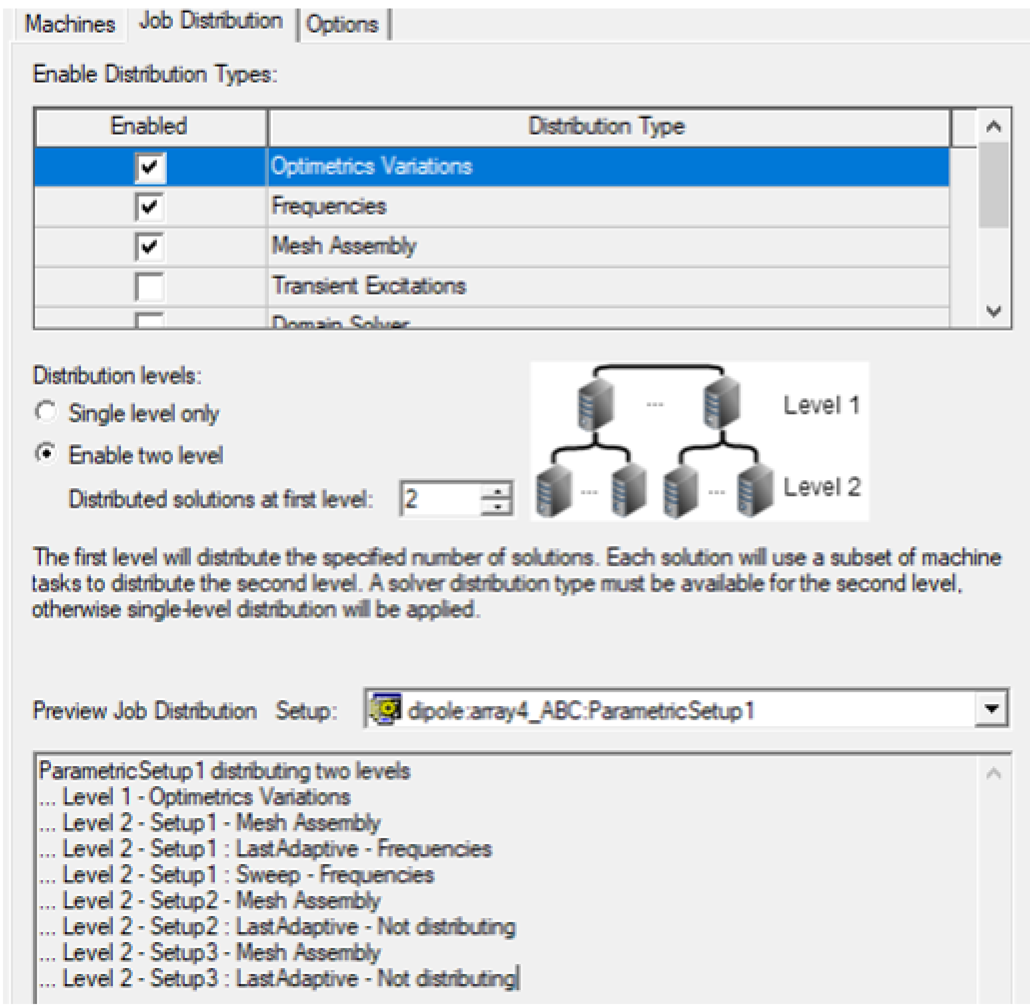
The following are examples of two-level distributions:

- 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:

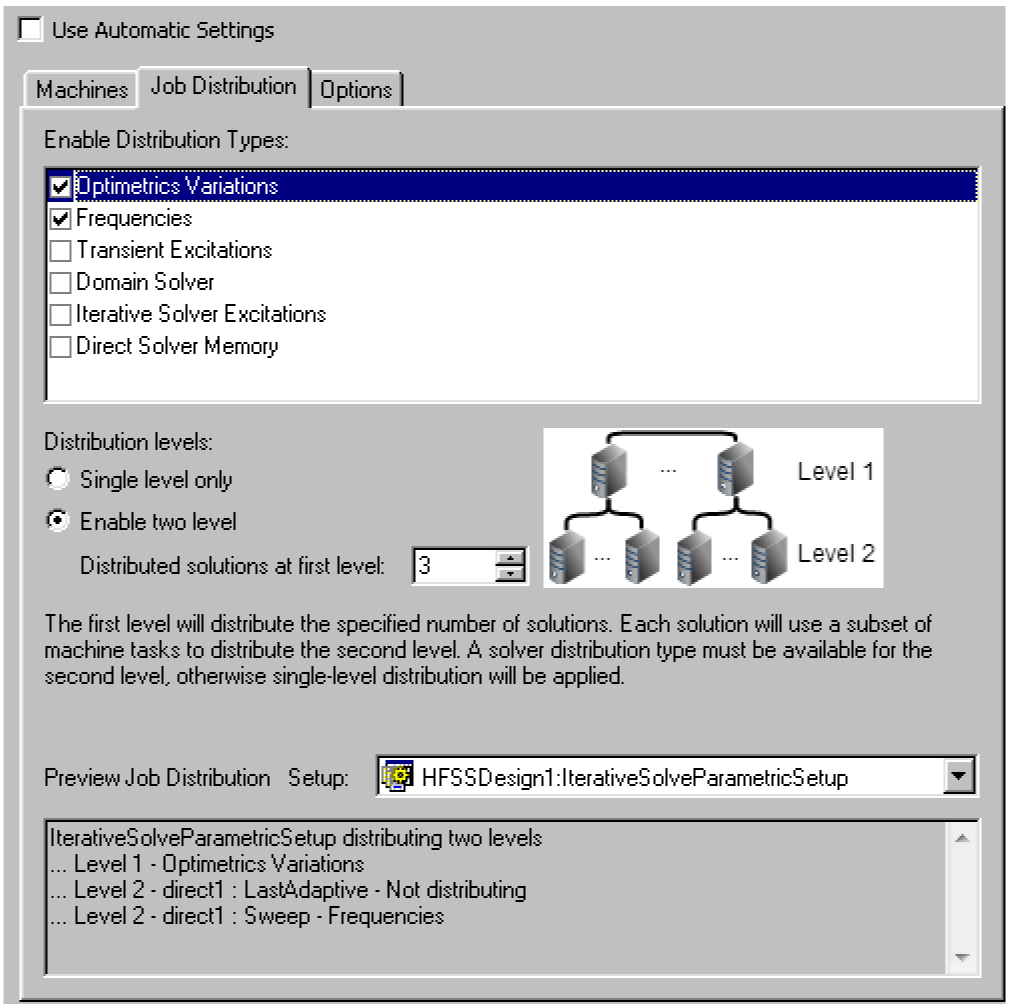


For more information, see: Two-Level Distribution Guidelines.

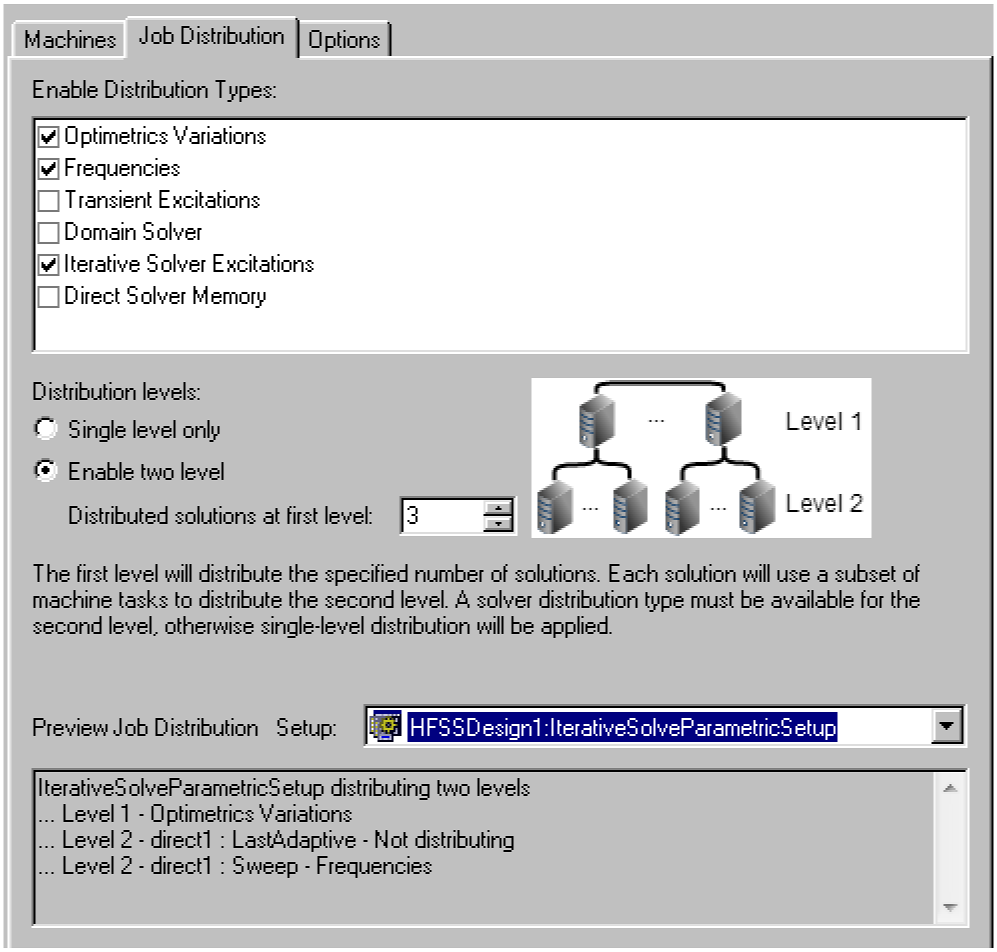
HFSS Frequency Distribution

HFSS Frequency Distribution can be treated as both of first and second level distribution. The following bullets described the cases for an HFSS design with frequencies sweep setup and parametric solve setup (for non transient solution).

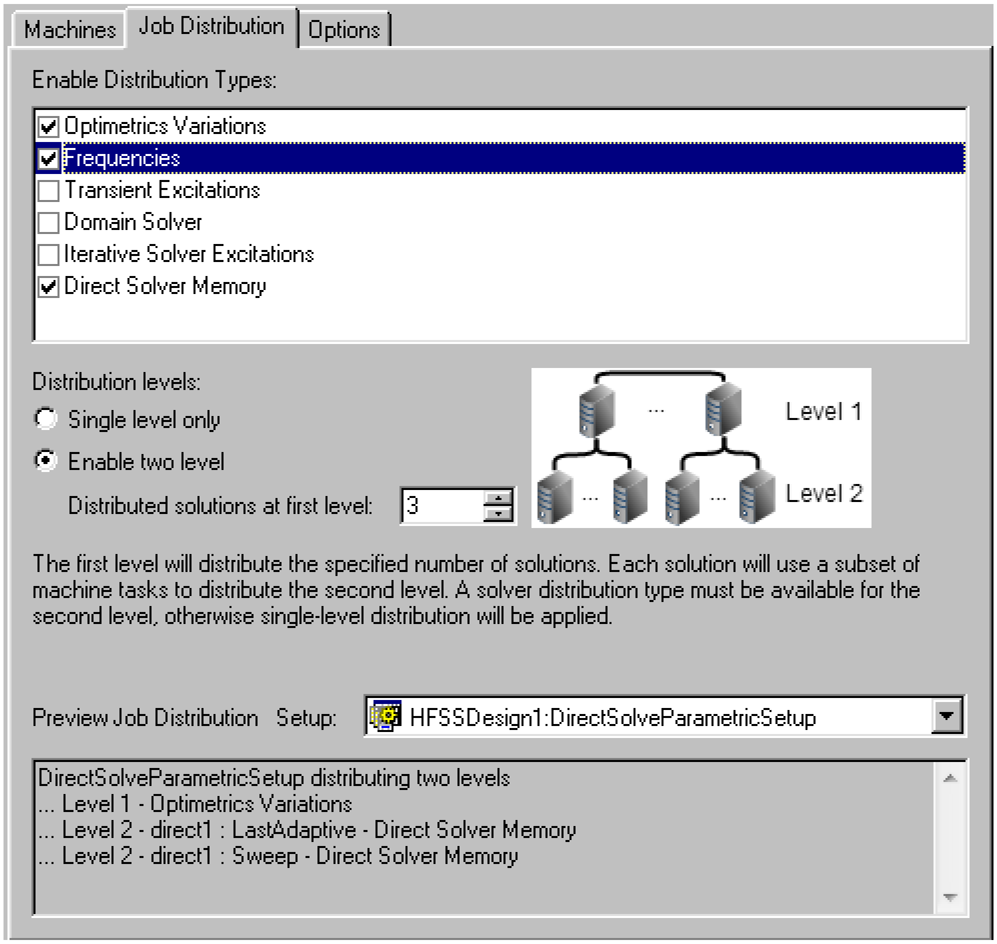
- In the **Analysis Configuration** window, if you select “Optimetrics Variations” and “Frequencies” with two-level distribution enabled, HFSS does two level distribution: first level, “Optimetrics Variations”; second level “Frequencies”. We can see this described in the “Preview Job Distribution” area:



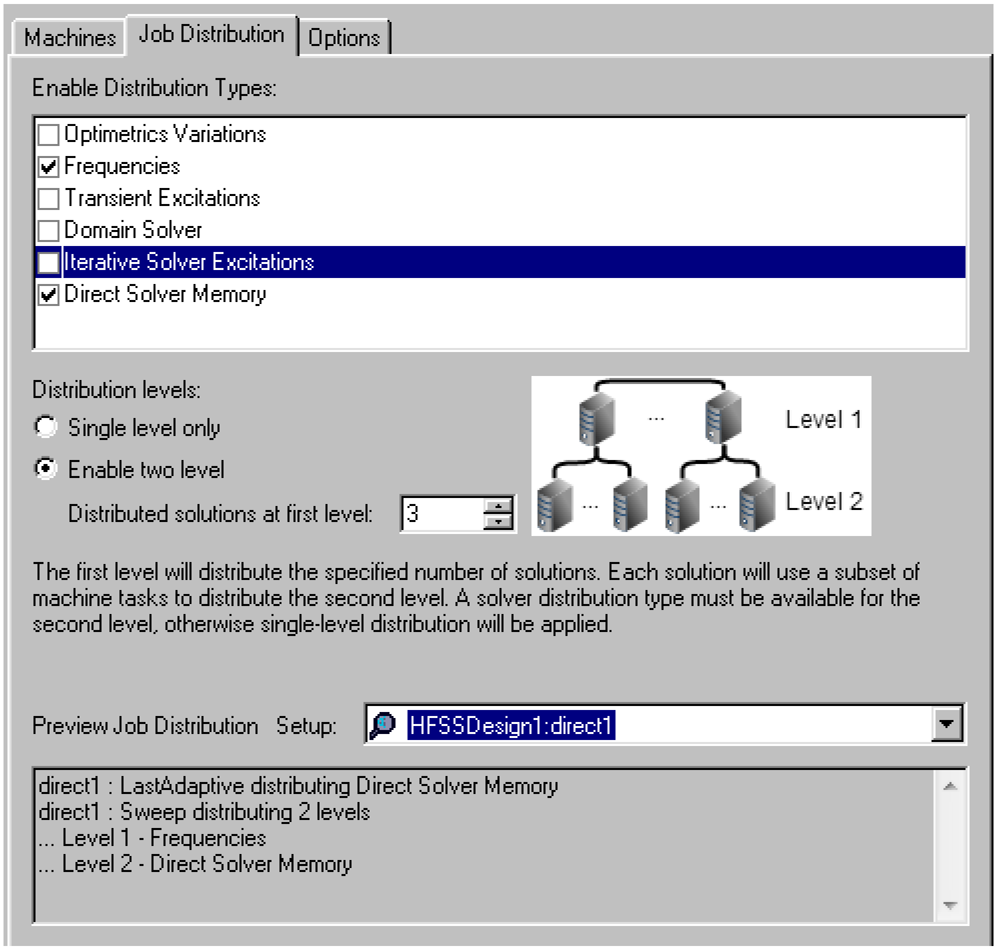
- In the **Analysis Configuration** window, if you select “Optimetrics Variations”, “Frequencies”, and “Iterative Solver Excitations“ with two-level distribution enabled, HFSS does two level distribution: first level, “Optimetrics Variations”; second level “Frequencies”. There is no “Iterative Solver Excitations“ distribution. We can see this in the “Preview Job Distribution” area:



- In the **Analysis Configuration** window, if you select “Optimetrics Variations”, “Frequencies”, and “Direct Solver Memory“ (or “Domain Solver”) with two-level distribution enabled, and if “Direct Solver Memory“ option is selected in the solve setup, HFSS does two level distribution: first level, “Optimetrics Variations”; second level, “Direct Solver Memory“ (or “Domain Solver”). There is no “Frequencies” distribution. We can see this in the “Preview Job Distribution” area:



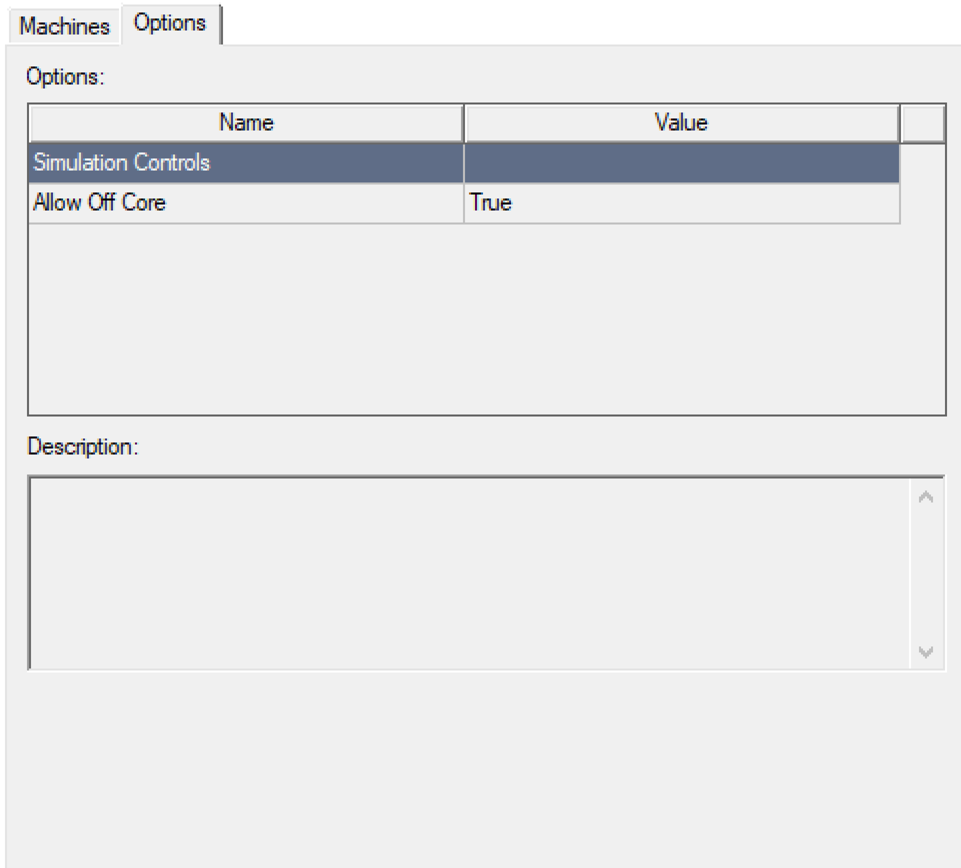
- In the **Analysis Configuration** dialog box, if you select “Frequencies”, and “Iterative Solver Excitations“(or “Direct Solver Memory“, or “Domain Solver”) with two-level distribution enabled and if you select “Direct Solver Memory“ option (or corresponding solver option) in the solve setup, HFSS only does two level distribution: first level, “Frequencies”; second level, “Iterative Solver Excitations“ (or corresponding solver. We can see this in the “Preview Job Distribution” area. See the following two examples. The first shows Enable Job Distribution Types for “Frequencies” and “Direct Solver Memory”.



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 **Circuit** > **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.

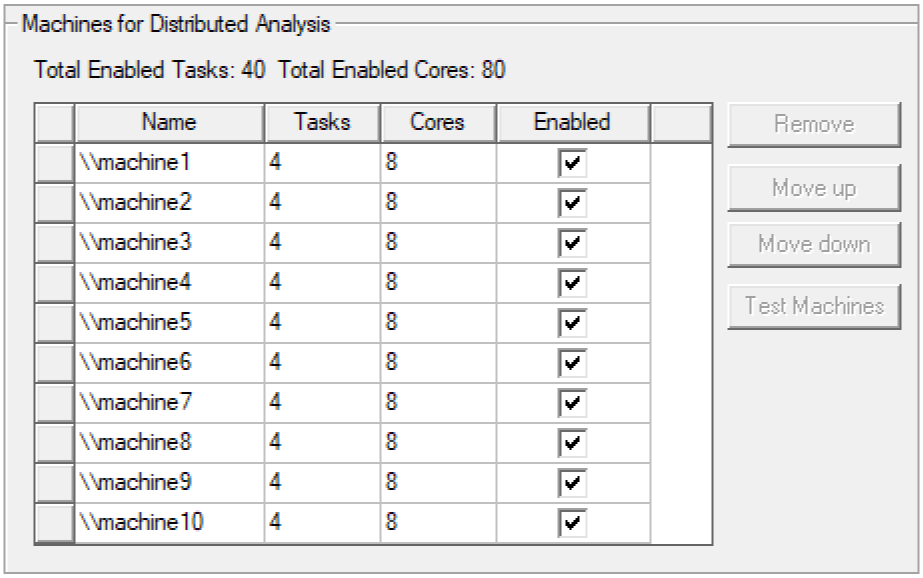
In the majority of situations, configuring a Distributed Solution Option (DSO) in combination with the multiprocessing option to take optimal advantage of the available hardware results in improved speed.

> **Note:**
>
> This 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.

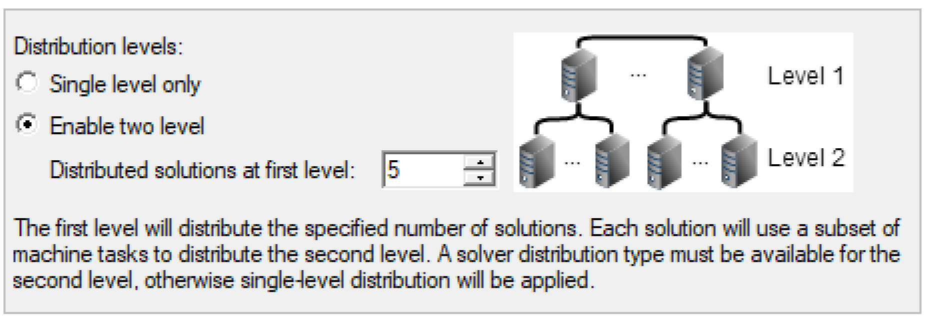
Two-Level Distribution Guidelines

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.



Name	Tasks	Cores	Enabled
\\machine1	4	8	☑
\\machine2	4	8	☑
\\machine3	4	8	☑
\\machine4	4	8	☑
\\machine5	4	8	☑
\\machine6	4	8	☑
\\machine7	4	8	☑
\\machine8	4	8	☑
\\machine9	4	8	☑
\\machine10	4	8	☑



Distributed Direct Solver Guidelines for HPC Configuration

The dependence of Distributed Direct solver on number of tasks is very similar to the Circuit 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. Circuit 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.

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 re-run 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. Post-processing 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 re-assigned to the running task, when the running task finish its original assignment. For more information, see Large Scale DSO theory.

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

LargeScaleDSO/VarRedistribution, where 0 disables redistribution (default), and 1 enables it.

LargeScaleDSO/RedistributionLimit, is a positive integer specifying the minimum estimated remaining time (in minutes) for variations to redistribute to another task. The default is 3.

Aborting a Large Scale DSO Simulation

To abort the whole Job, select the **Abort** button on the Job Monitor dialog.

To abort using the Job progress bar, click the button next to the Job progress bar, Click the **Abort** menu item in the popup.

To abort all tasks in a Node(host), click the button next to the node progress bar., and click the **Abort** menu item in the popup. Aborted Variations will be redistributed to other running Nodes, if redistribution is turned on.

To abort an individual task, click the button next to the node progress bar, and click the “Detail” menu item in the popup. In the Task status dialog box, click the **Task** button in the grid. In the variation status dialog box, click **Abort** button. Aborted Variations will be redistributed to other running tasks, if redistribution is turned on.

To terminate a hanging EM-Desktop process, the hanging ansysedt.exe process won’t respond to the first abort command. Send a second abort command to terminate the hanging process. Remaining variations will be redistributed to other running tasks, if redistribution is turned on.

For an EM-Desktop process crashed or killed by Windows Task manager or other tools, the task status will be shown as aborted, Remaining variations will be redistributed to other running tasks, if redistribution is turned on.

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`

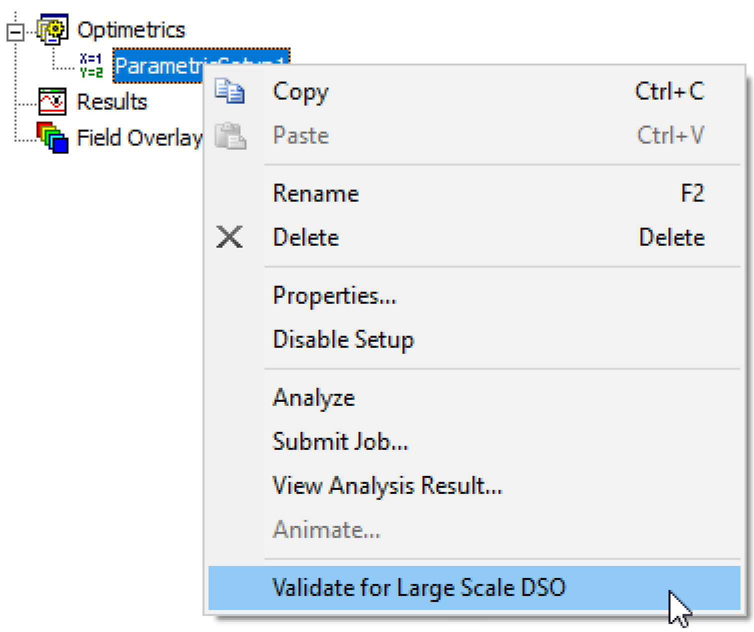
  LINUX: `<installation-directory>/<platform> desktopjob -regserver`

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.

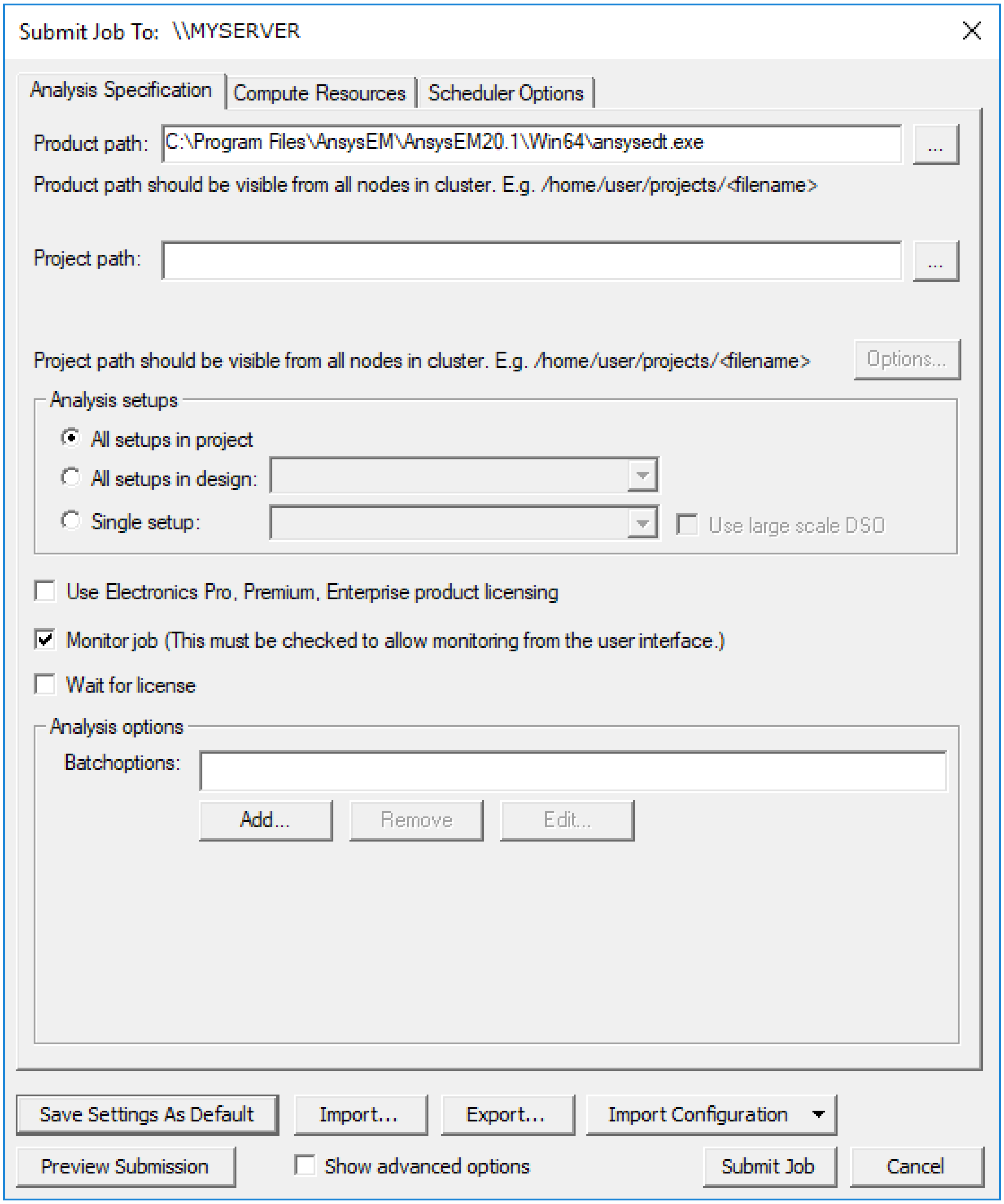
4. Copy the project (or folder, if the project references external files) from a personal workstation to a shared drive on cluster.

   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 **Circuit** > **Submit Job**.
   - Select the **Simulation** tab and click the **Submit** icon.

   The **Submit Job To** window appears, on the **Analysis Specification** tab.



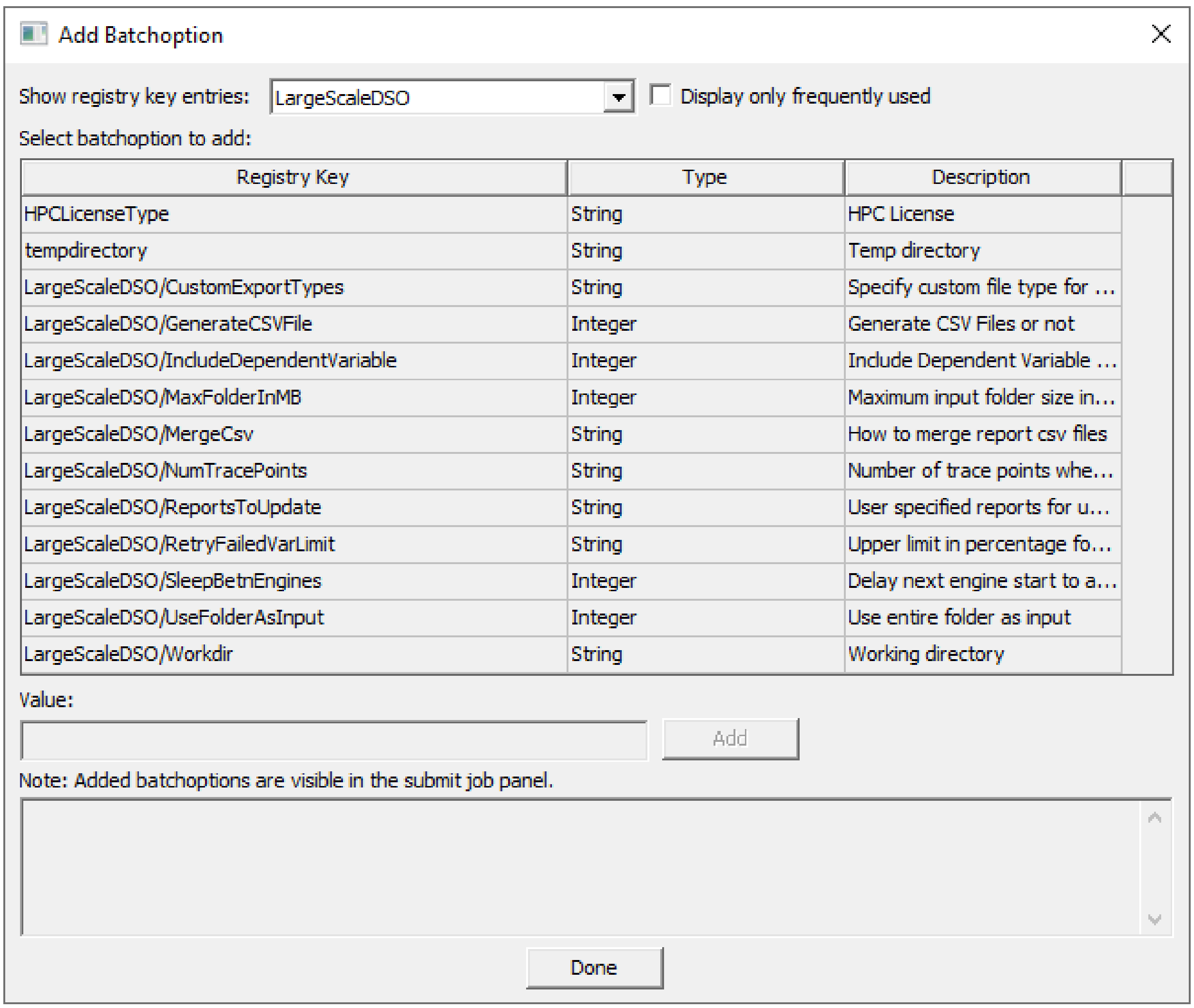
Note:

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.

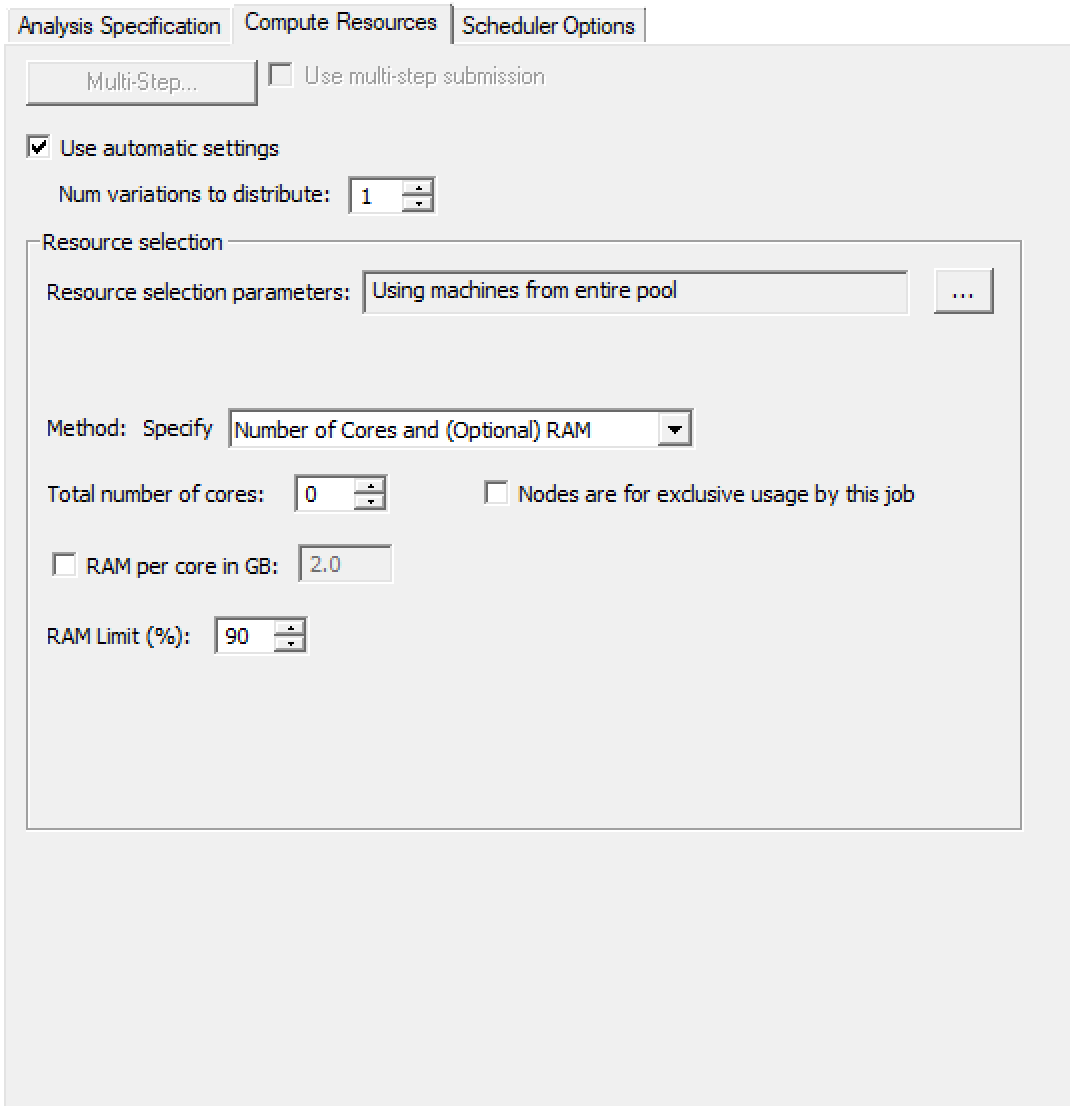
      For Large Scale DSO, select a **Single setup** from the drop-down menu and ensure the **Use large scale DSO** check box is selected.
    - Determine whether to **Use Electronics Pro, Premium, Enterprise product licensing**, whether to **Monitor job**, and whether to **Wait for license.** If you wish 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.



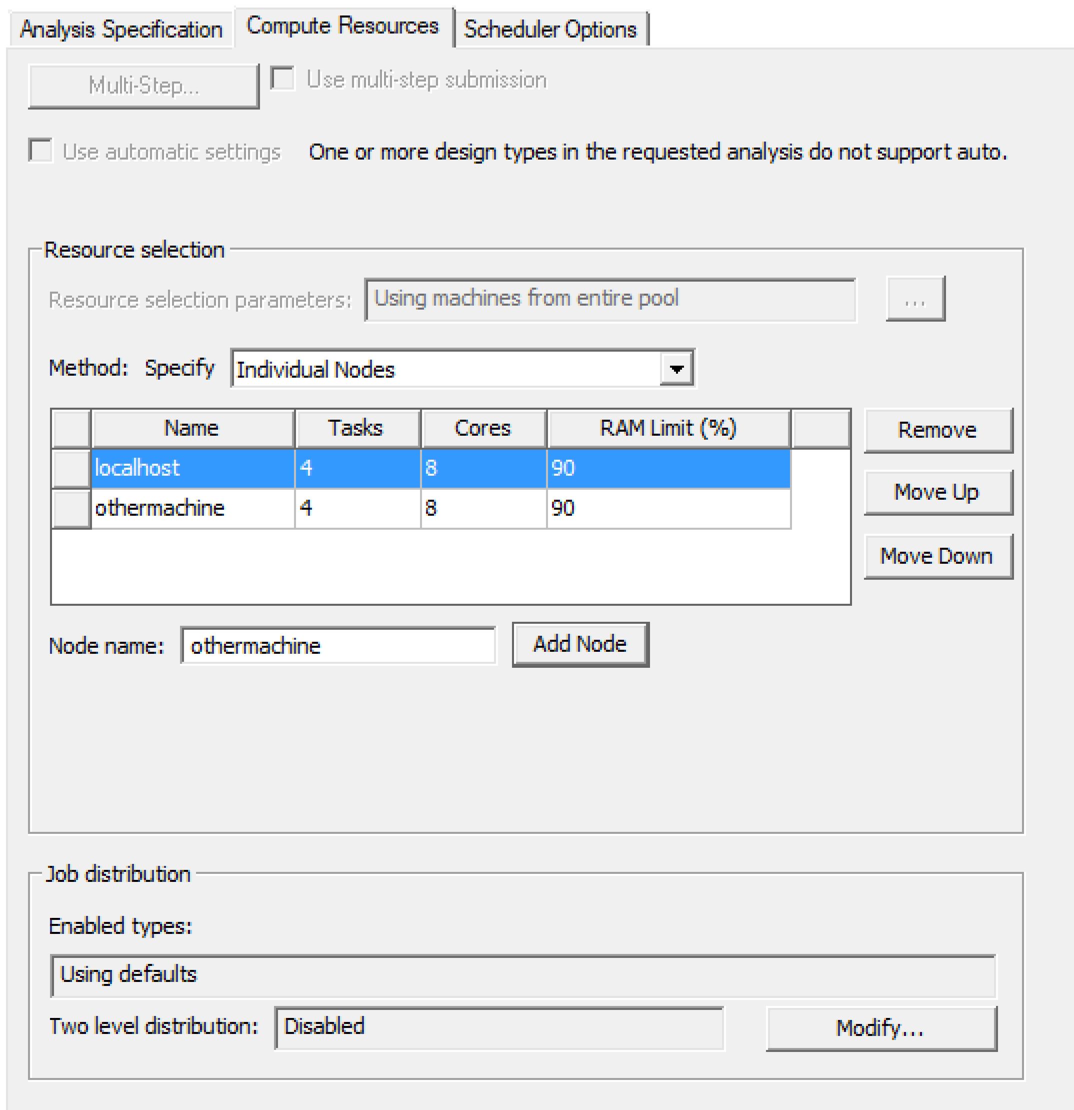
11. Select the **Compute Resources** tab.

    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"`

- **-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.

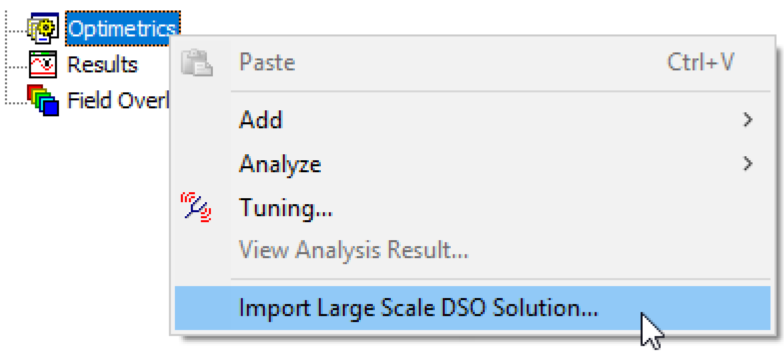
Post-Processing Large Scale DSO Dataset Solutions

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

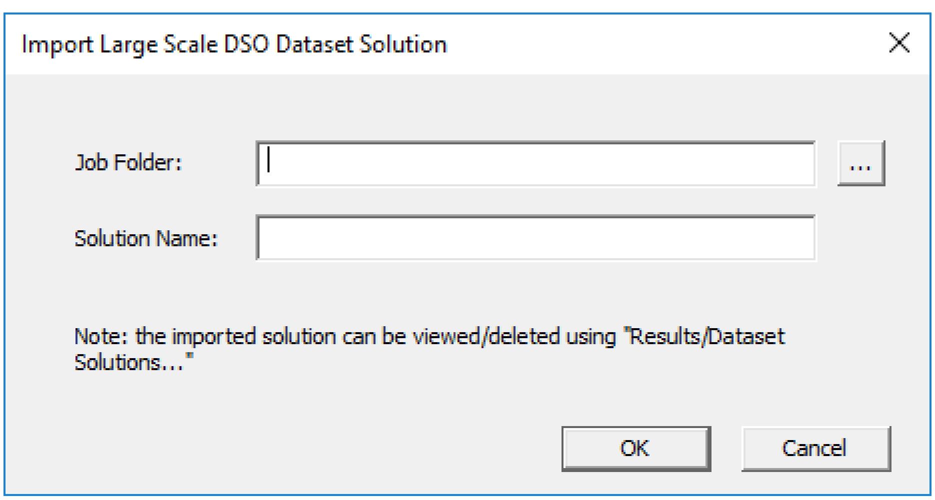
To import a Large Scale DSO solution:

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**.

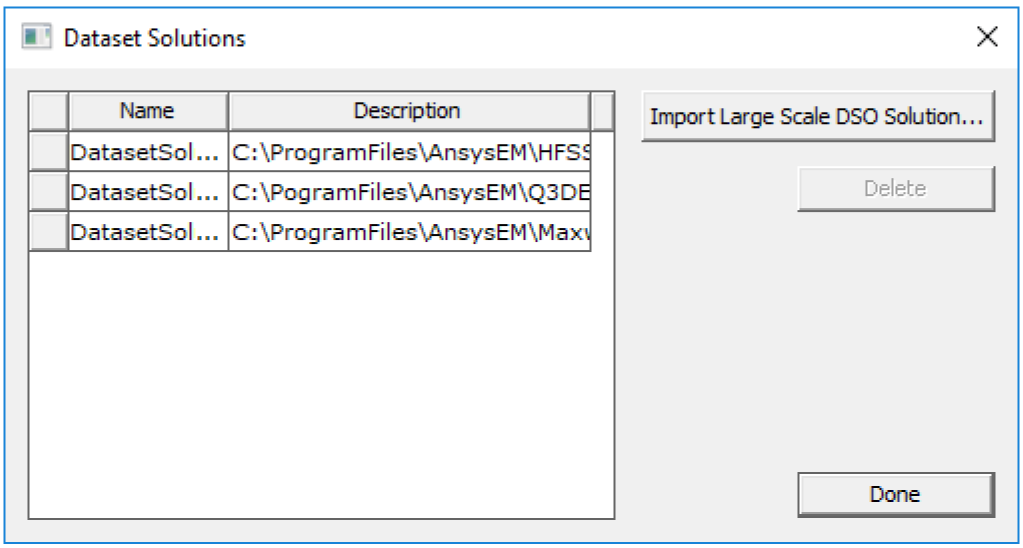
   The dataset is imported.

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

- Shared drive for projects: Cluster must provide a shared drive that hosts job inputs - the submitted project must be located on a shared drive (e.g., a sub-folder of user's home directory). The shared-drive must accessible using the same path on every node of cluster.
- 'Temp directory' configuration.

  Temp directory is either on 'local storage' or on storage that has equivalent speed characteristics. 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 Tutorial Example

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

This section provides an example of the use of Large Scale DSO to distribute parametric variations of an HFSS model across the nodes of a cluster or to multiple cores of a single machine.

This example presumes that your configuration satisfies the Prerequisites for Large Scale DSO. For this example, pre-suppose that we have a Windows cluster. Further suppose that the shared drive folder, which contains the input projects and computed results is at the location *\\sjo7na1\hfssprojs*.

Major steps for Large Scale DSO Example

1. Prepare the model for Large Scale DSO Analysis
2. Submit the Large Scale DSO Job
3. Post process the results

Prepare the Model for Large Scale DSO Analysis

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

1. Launch the desktop.
2. Create the input project on the shared drive.

   For this example, start with the standard HFSS OptimTee.aedt example and copy it to the shared drive.

   Copy "<installation-directory>\<platform>Examples\RF_Microwave\OptimTee.aedt" to "\\sjo7na1\hfssprojs\OptimTee.aedt"

3. set up the parametric table.

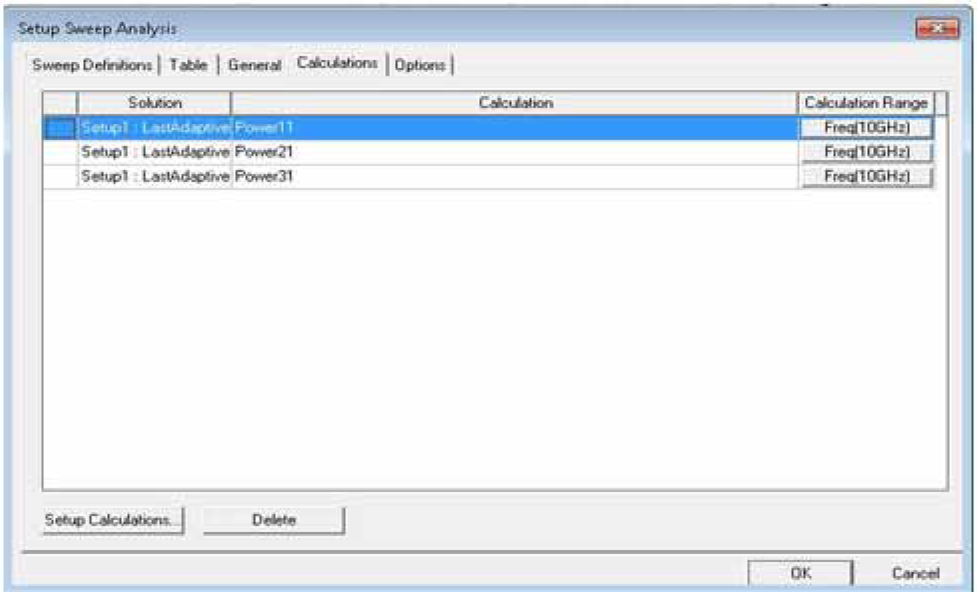
   For this example, use the existing 'ParametricSetup1' as the parametric setup to solve.

4. Prepare the outputs.

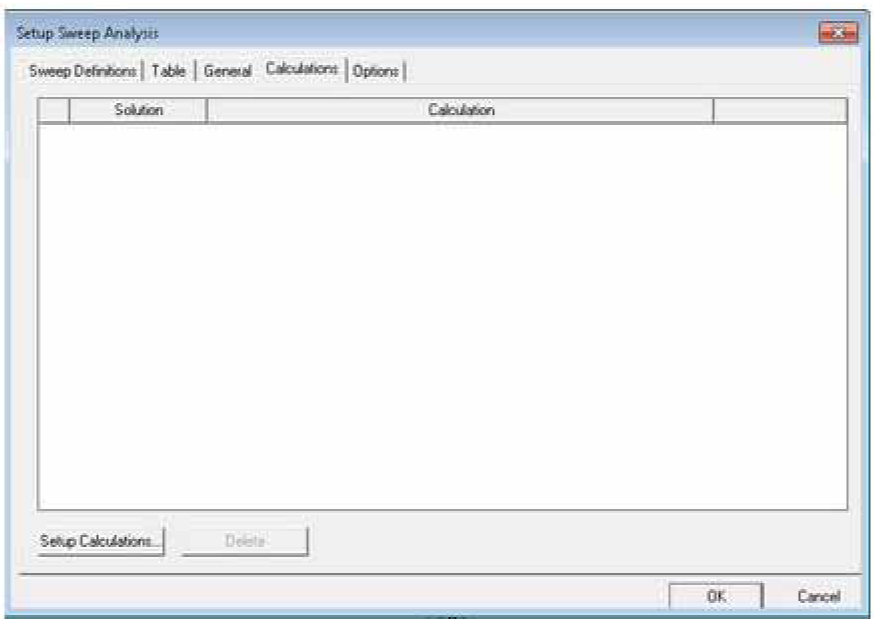
   Outputs from Large Scale DSO come from pre-defined rectangular plots that are created before the Analysis command is issued. Follow the steps below:

   a. Because these DSO outputs come solely from Rectangular Plots, delete all other postprocessing setups, and then turn off Save Fields And Mesh as shown below.

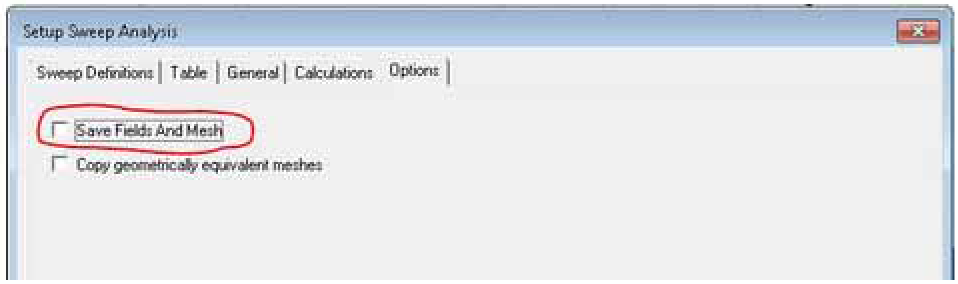
      Open ParametricSetup1 for editing. You will see that the OptimTee parametric setup contains three calculations, as shown below.



b. Delete all three calculations. When you have done so, the Setup Sweep Analysis Calculations tab looks like this:



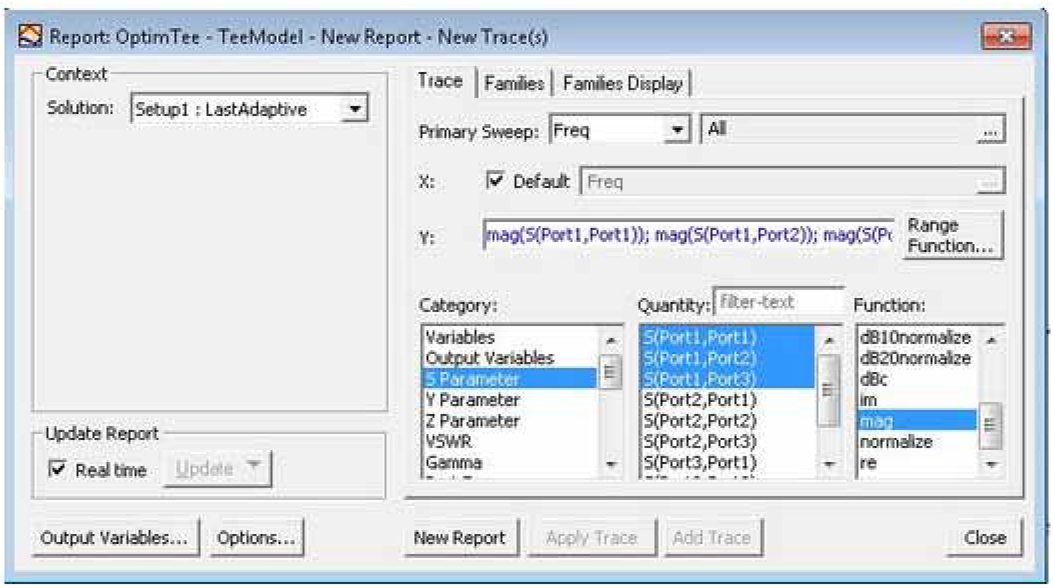
c. Click on the **Options** tab and uncheck Save Fields And Mesh, as shown below:



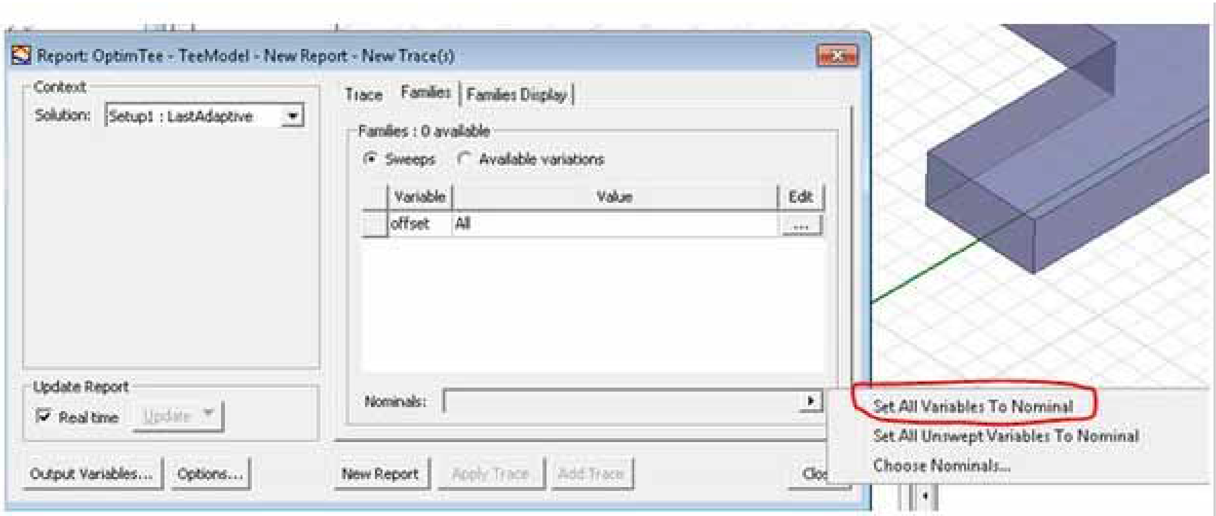
5. Use the Reporter to define outputs.

   For this example, you add six traces that correspond to six csv outputs of the Large Scale DSO job: mag(S11), mag(S12), mag(S13), Power11, Power12, and Power13.

   a. As shown below, select three quantities: mag(S11), mag(S12), and mag(S13).



   b. Click on the Families tab and ensure that all variables are set to Nominal, as shown below.



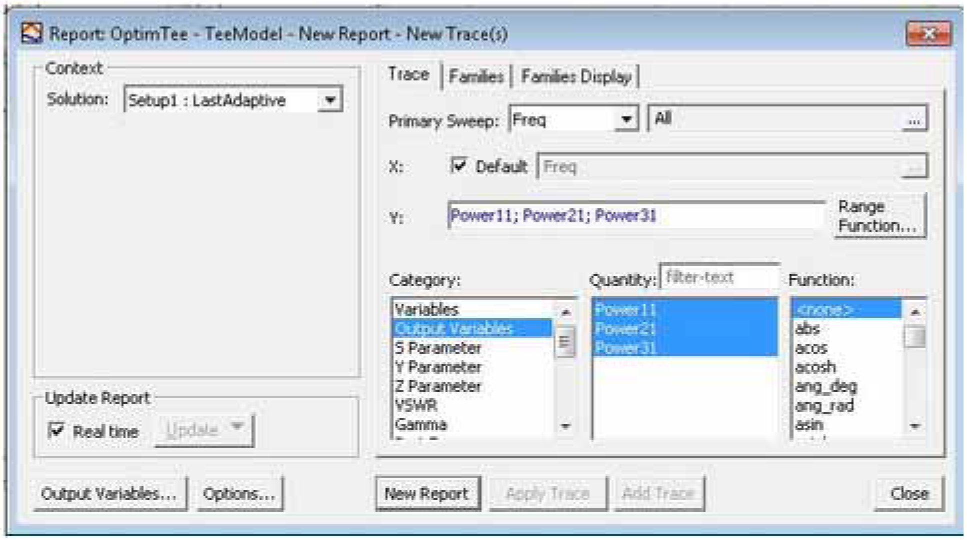
> **Note:**
>
> For Large Scale DSO, outputs are not extracted correctly unless all variables on the **Families** tab are set to nominal.

c. Click the **New Report** button.

   Use the Report to create a Power distribution plot that has three power distribution traces.

d. To create an S-parameter plot that has three traces, click **Results> Create Modal Solution Data Report> Rectangular Plot**.

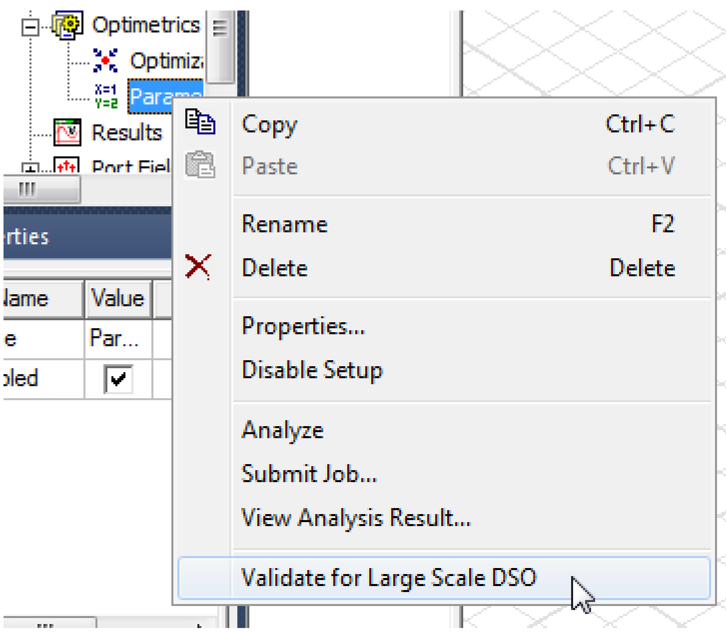
e. As shown below, select three quantities, Power11, Power21, and Power31.



f. Click on the Families tab and ensure that all variables are set to Nominal.

g. Click the **New Report** button.

6. Right-click the parametric sweep, and select Validate for Large Scale DSO.



7. A dialog reports any errors. These may occur if the steps just taken are neglected:

```
[error] Please remove all the calculations in Parametric
Setup/Calculation page.
```

```
[error] Please turn off "Save Fields And Mesh" in Parametric Setup/Options page.
[error] No rectangular plot exist in the design. Please create a rectangular report.
```

8. Correct any errors, if necessary to pass validation.
9. Save and Close the project.

Next

Submit the Large Scale DSO Job: Examples

Submit the Large Scale DSO Job: Examples

This section includes examples of submitting a Large Scale DSO job using Ansoft RSM and a Scheduler.

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

Using the Ansoft RSM Environment

1. Before submitting the job make sure that the input project is not open in any Ansys Electronics Desktop window.
2. For this example, suppose that there are two quad-core machines on your cluster with the names "m1" and "m2". Further suppose that there are two engines per machine, for a total of four parallel engines. Let the number of processors allocated to each engine be 1.
3. From a command prompt, issue the following command:

```
<installation-directory>\v<version>\<platform>\desktopjob.exe -cmd dso -machinelist "list=m1:2,m2:2" -batchoptions \\sjo7na\aedtprojs\hfssoptions.txt -batchsolve "TeeModel:Optimetrics:ParametricSetup1" \\sjo7na\aedtprojs\OptimTee.aedt
```

where the file \\sjo7na\hfssproj\hfssoptions.txt has the following contents:

```
$begin Config
  'Desktop/Settings/ProjectOptions/NumberOfProcessors'=1
  'HFSS/Preferences/SaveBeforeSolving'=0
  'HFSS/Preferences/HPCLicenceType'='pack'
#end 'Config'
```

4. Suppose the above job is assigned ID "jobID".

Using a Scheduler Environment (such as LSF)

1. Suppose you want to solve variations using four parallel engines, each engine being assigned a single core.
2. From a command prompt, run the following command:

   ```
   bsub -n 4 <installation-
   directory>\v<version>\<platform>\desktopjob.exe -cmd dso -
   batchoptions \\sjo7na\hfssprojs\hfssoptions.txt -batchsolve
   "TeeModel:Optimetrics:ParametricSetup1"\\sjo7na\hfssprojs\Opti
   mTee.aedt
   ```

   where the file \\sjo7na\hfssproj\hfssoptions.txt has the same contents as the RSM example above.

3. Suppose the above job is assigned an ID "jobid"

For example with 2 nodes of 44 cores each (with multiple distributed variations per machine):

MANUAL:
```
bsub -n 88 <installation-
directory>\v<version>\<platform>\desktopjob -cmd dso -machinelist
num=8 -monitor -useelectronicsppe=1 -ng -batchoptions "
'Circuit/NumCoresPerDistributedTask'=11
'Circuit/'HPCLicenseType'='Pack'" -batchsolve
TeeModel:Optimetrics:ParametricSetup1 OptimTee-v232-1.aedt
```

AUTO:
```
bsub -n 88 <installation-
directory>\v<version>\<platform>\desktopjob -cmd dso -machinelist
numcores=88 -auto -NumDistributedVariations 8 -monitor -
useelectronicsppe=1 -ng -batchoptions "
'Circuit/'HPCLicenseType'='Pack'" -batchsolve
TeeModel:Optimetrics:ParametricSetup1 OptimTee-v232-1.aedt
```

For example, with 16 nodes of 44 cores each (note each distributed variation will span 2 machines):

AUTO:
```
bsub -n 704 <installation-
directory>\v<version>\<platform>\desktopjob -cmd dso -machinelist
numcores=704 -auto -NumDistributedVariations 8 -monitor -
useelectronicsppe=1 -ng -batchoptions "
'Circuit/'HPCLicenseType'='Pack'" -batchsolve
TeeModel:Optimetrics:ParametricSetup1 OptimTee-v232-1.aedt
```

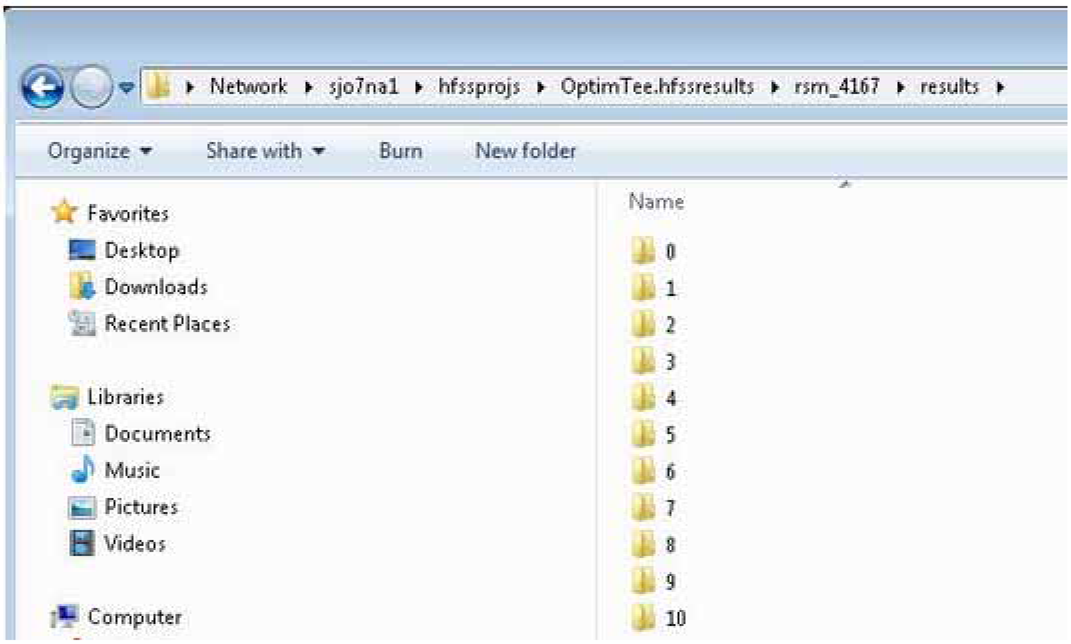
Next

Large Scale DSO Example: Post Process the Results

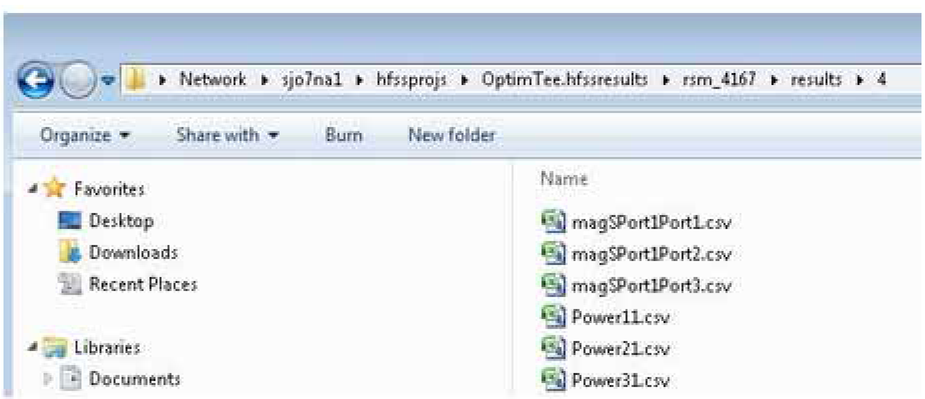
Large Scale DSO Example: Post Process the Results

Once the job is done, output is available in the ~\OptimTee.aedt\jobid\results folder. Each variation creates a subfolder, which in turn has one csv file per trace of each report. See the detailed information regarding job monitoring and the location of the analysis logs.

The following figure shows the results for 10 variations as located in 10 folders.

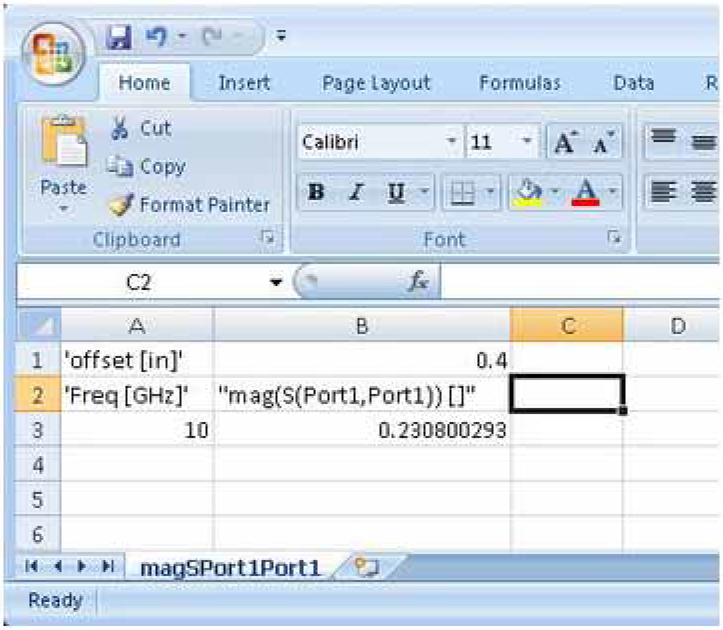


There are six csv files corresponding to three S-parameter traces and three power distribution traces, as shown below for the fourth variation.



You have three options for postprocessing csv files.

- Import Large Scale DSO Dataset Solution
- Use Microsoft Excel or any other application that has csv post processing functionality.
- Parse the csv output into your custom program, for any downstream flow.



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

Cluster Configuration Shared Drive Requirement

All input files (project, etc.) must be present on a shared drive that is accessible from every node of the cluster.

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.

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 which 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.

- **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:

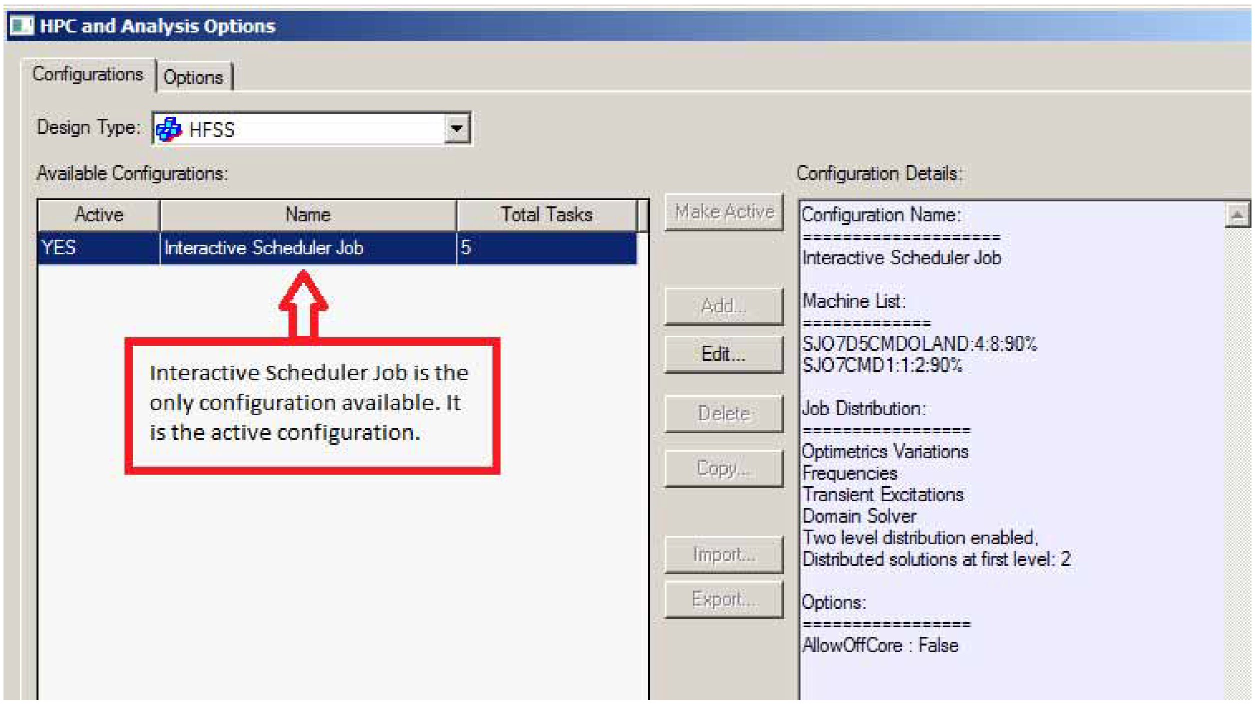
```
desktopjob.exe" -cmd dso -jobid RSM_27086 -machinelist
list=localHost:2:2:90%
-monitor -ng -batchoptions "
'LargeScaleDSO/MergeCsv'='acrossDPsAndTraces'"
-batchsolve TeeModel:Optimetrics:ParametricSetup2 E:\work\2018\LS_
DSO\OptimTee.aedt
desktopjob.exe" -cmd dso -jobid RSM_25248 -machinelist
list=localHost:2:2:90%
-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.

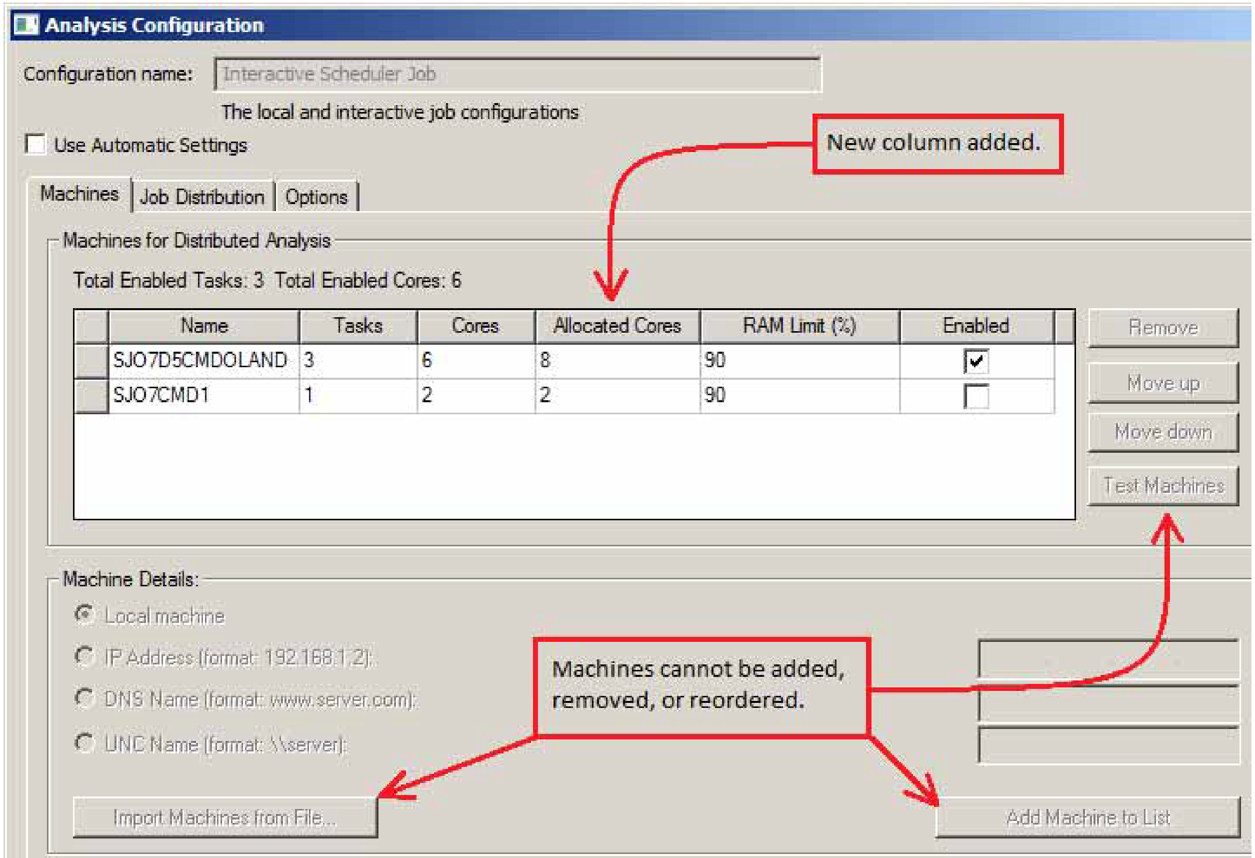


One major difference between interactive scheduler mode and batch mode or normal interactive mode is:

- 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 and cores 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

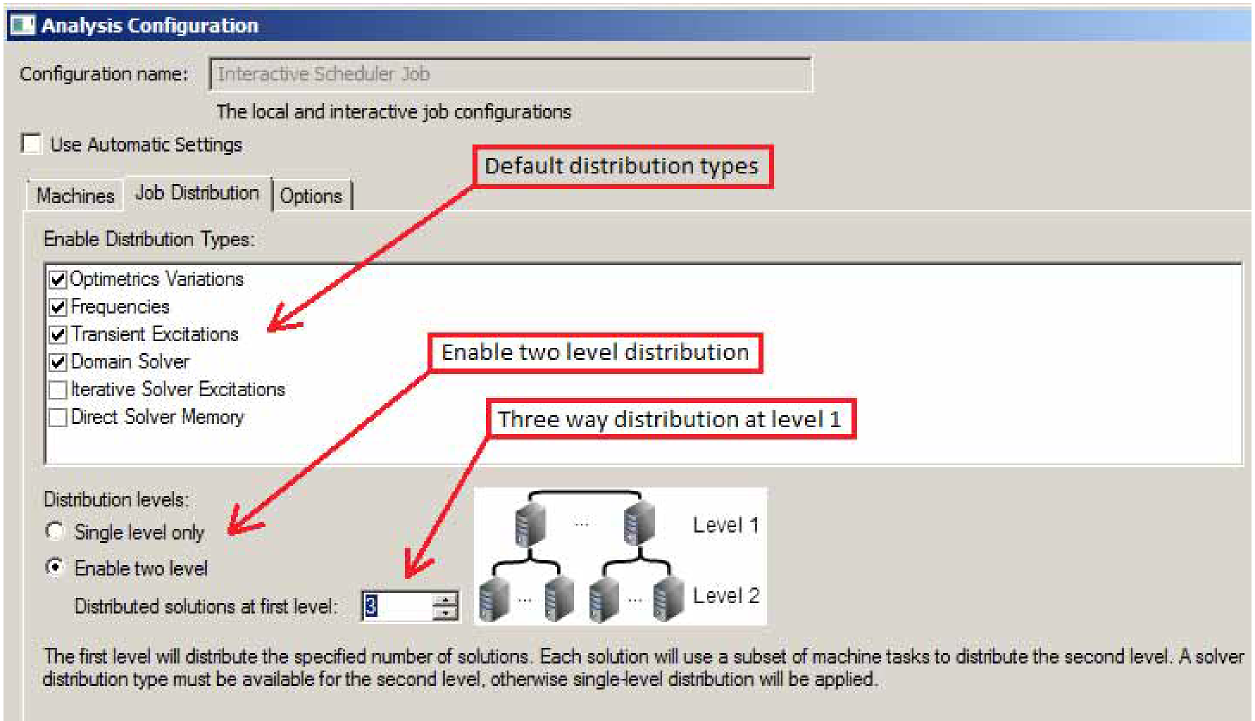
The registry contains the last value of these settings for an interactive scheduler job for the same user on the same host. Settings on the command line override settings from the registry. Any changes in the GUI will override the initial settings, even if the initial settings are from the command line. 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.

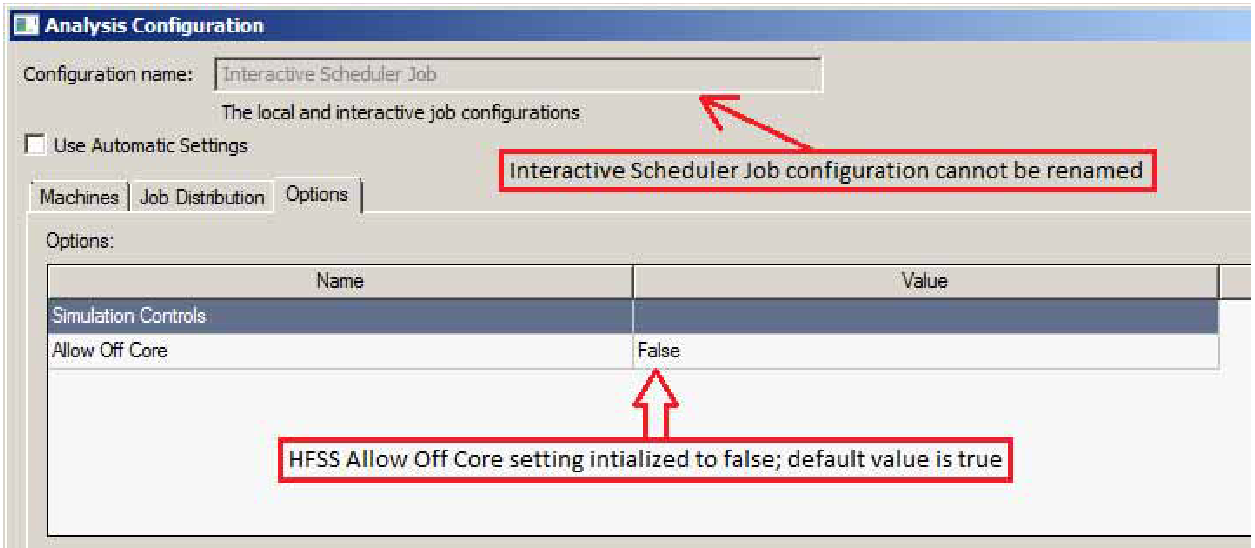
The settings on this tab are initialized from the command line (using the -batchoptions command line option) or from the registry. The registry contains the last value of these settings for an interactive scheduler job for the same user on the same host. Settings on the command line override settings 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'
```

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.
- **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.

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.

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
- 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.

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`

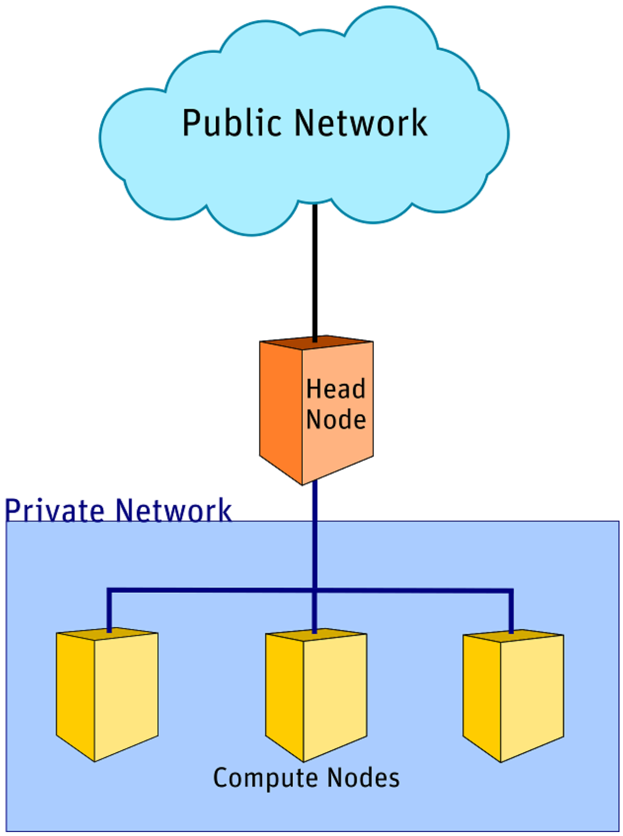
Term	Definition
Machine/Host/Node	Consists of one or more processors, memory, disk, etc.
Multiprocessing	A single engine uses multiple cores on the same machine.
Parallel Job	Job that runs on multiple cores belonging to the same or different machines.
Processor	Consists of one or more cores.
Resource	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\AnsysEM\v242\win64 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/AnsysEM/v242 on each execution node of the cluster.

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 2024 R2, 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.

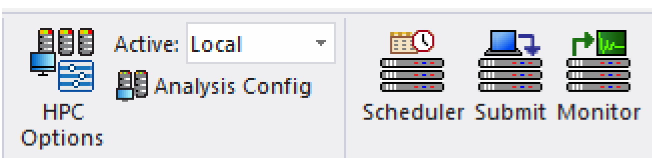
Job Submission User Profile on Cluster Compute Nodes

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.

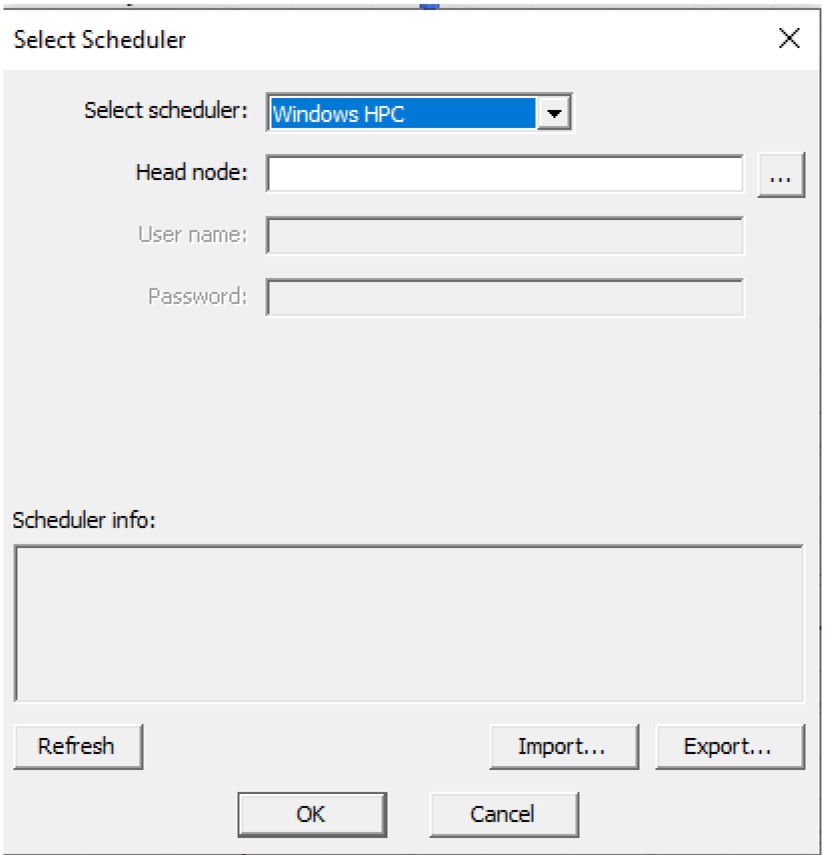
Submitting and Monitoring Jobs for Windows HPC

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:

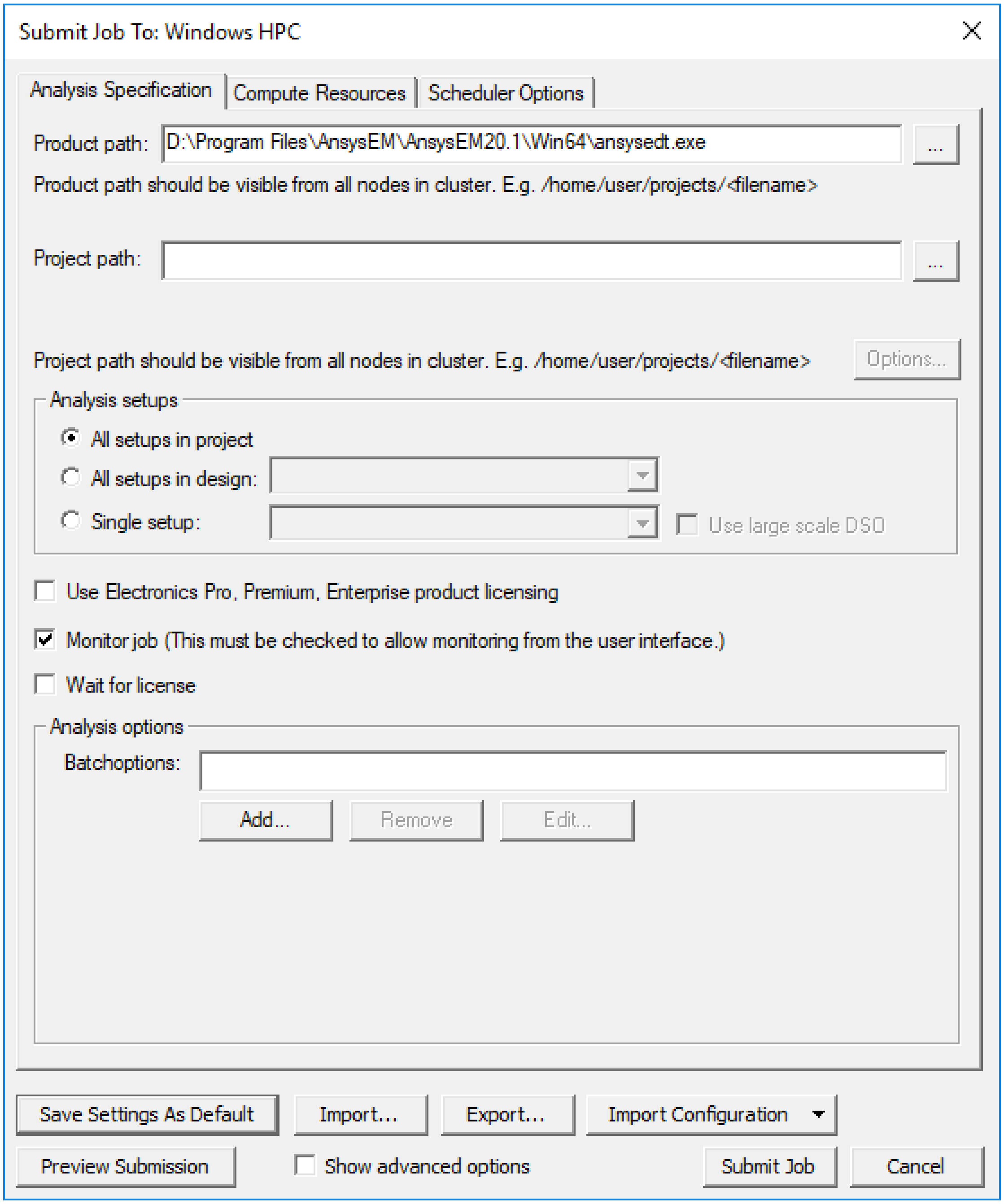
- Select **Tools** > **Job Management** > **Submit Job...**
- Click **Project** > **Submit Job...**
- Click **Circuit** > **Submit Job...**
- Select the **Simulation** tab and click the **Submit** icon.
- Use a shortcut menu to select **Submit Job**.

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)

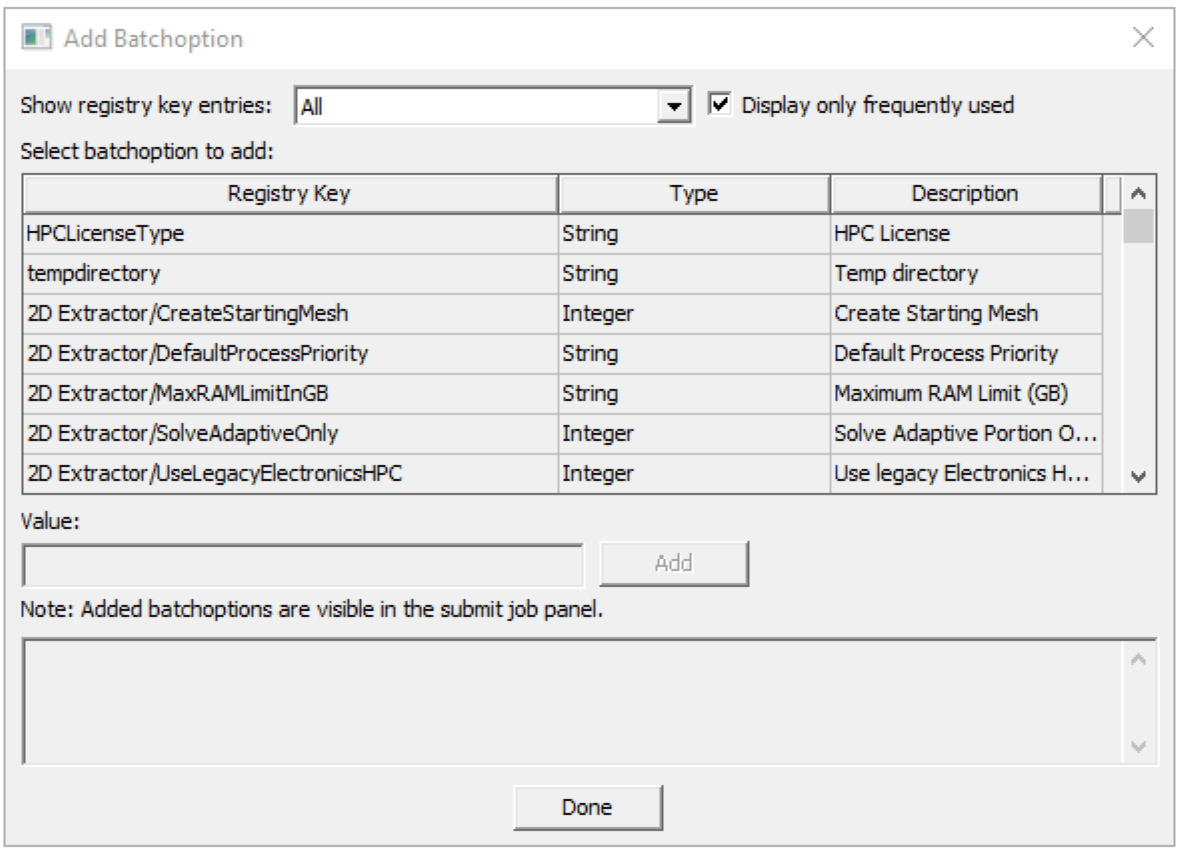
- Single setup (if you selected the Submit Job command from the shortcut menu, the setup name populates the field)

If you specify multiple setups, they will be processed sequentially in the order displayed in the edit box.

The Analysis options include:

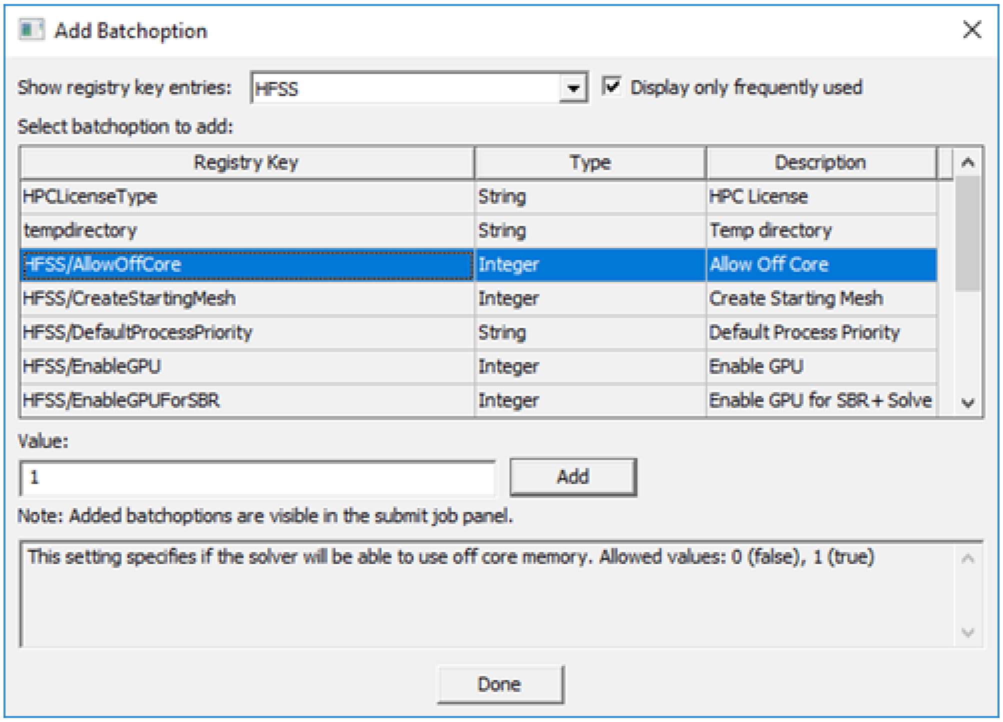
- Monitor job – you must enable this option to monitor the job from the user interface.
- Wait for license – specify whether to wait until a license if available before starting a simulation.
- Batch options – optionally specify -batchoptions in the text field. See detailed discussion of -batchoptions beginning under Running Ansys Electronics Desktop from a Command Line.

The **Add...** button opens the **Add Batchoption** dialog box.

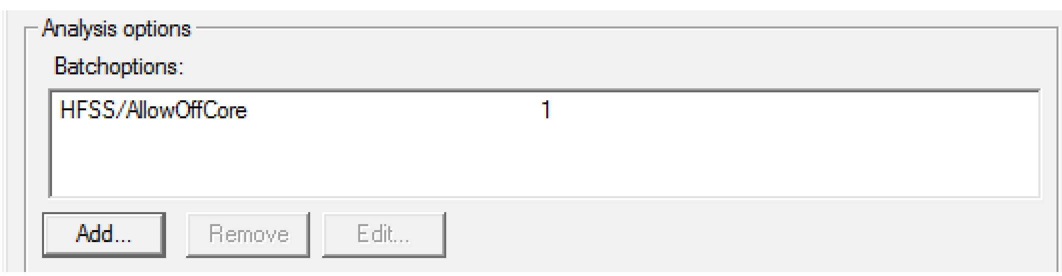


This dialog box provides access to all -batchoption commands. The drop-down menu lets you select specific categories, and you can choose to display only frequently used commands. You can edit and remove any batch options you specify.

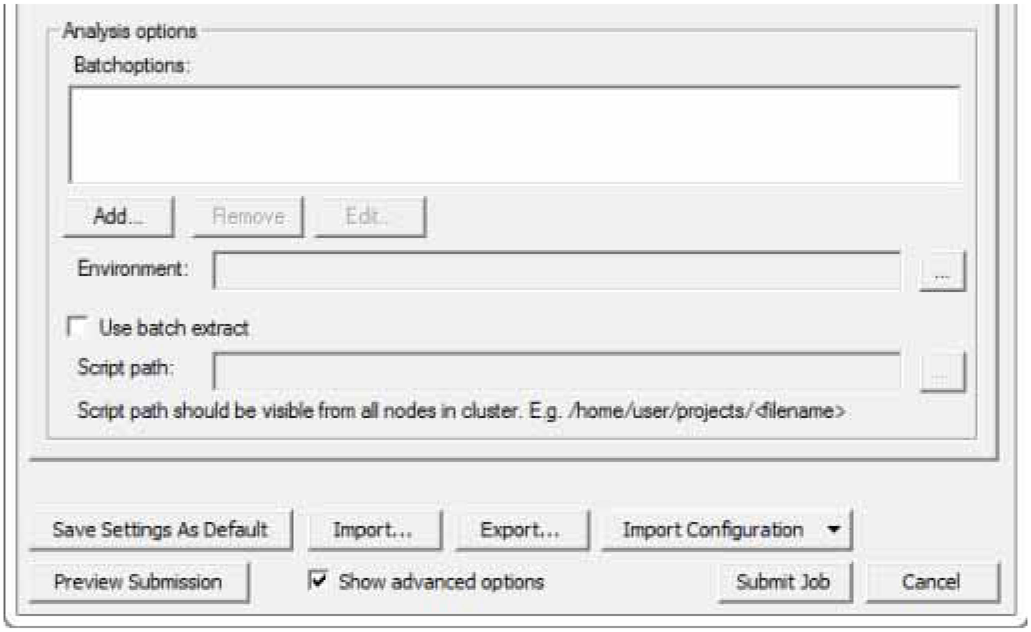
Select a Registry Key in order to show the current Value for the type. The lower field explains the meaning of the Type Value.



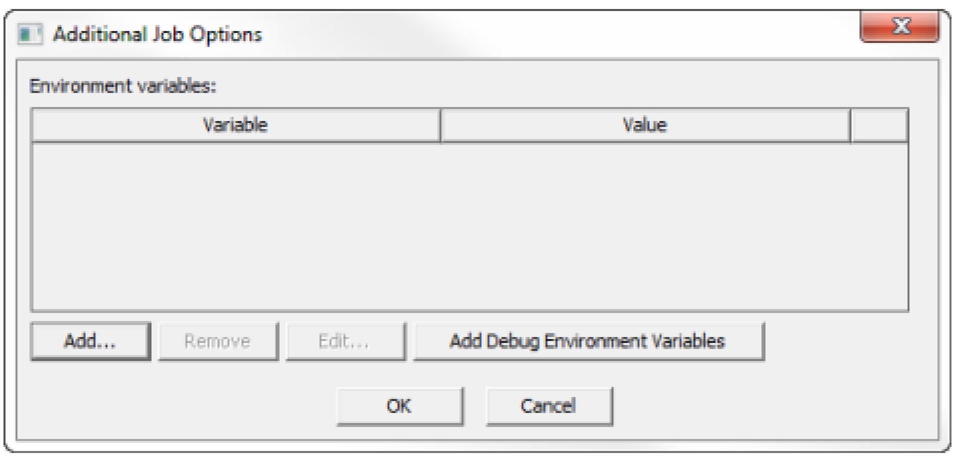
Any batchoptions for which you select **Add** will be visible in the **Submit Job** dialog box.



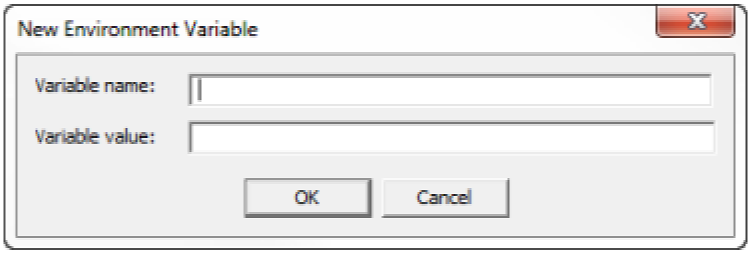
If you have the **Show advanced options** box checked in the **Submit Job** dialog box, the Environment field and the Use batch extract fields display.



The **Environment** field lets you specify any environment variables. Click the ellipsis button [...] to display the **Additional Job Options** dialog box.

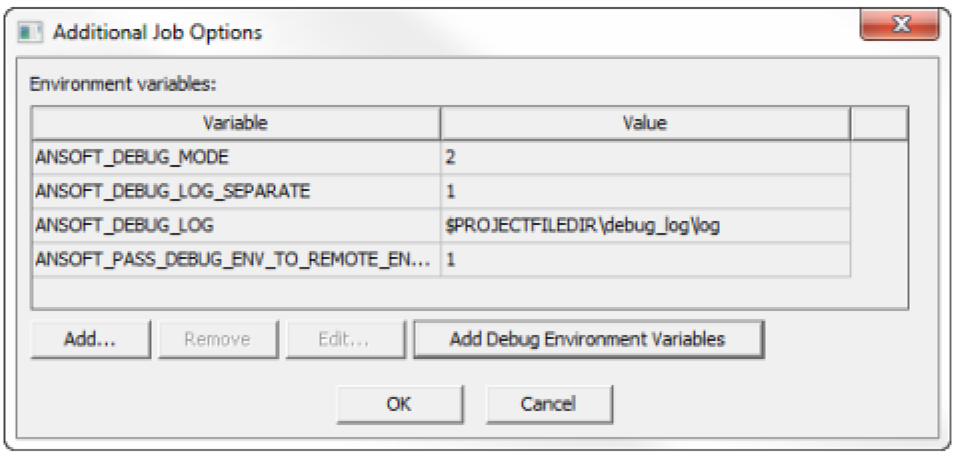


Click the **Add...** button to open the **New Environment Variable** dialog box.



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

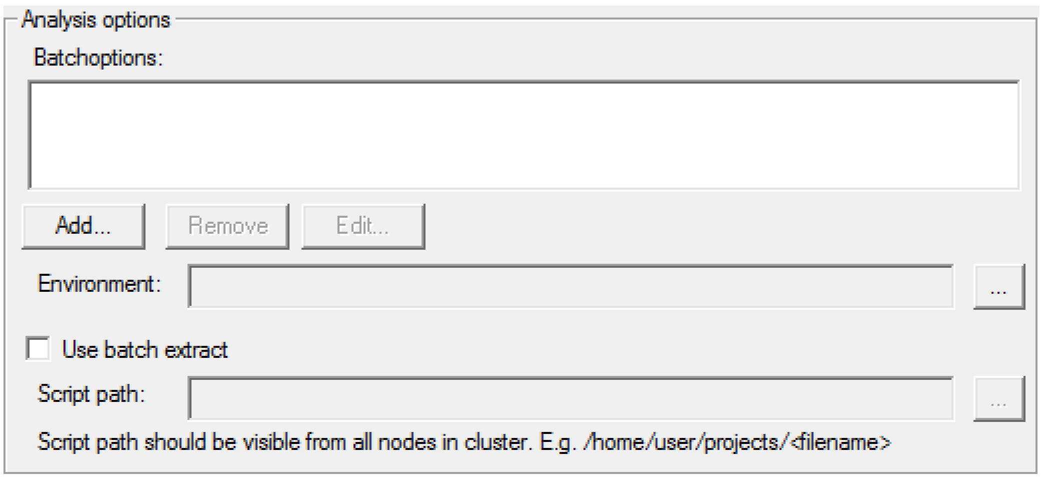
Set the variable value to 1 to disable the online check. Debug logging fully verifies what aspects of the cluster are causing this check to fail.

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 Options**, any variables that you add will be displayed in the **Environment** field of the **Submit Job** dialog box.

Use Batch Extract for Windows HPC

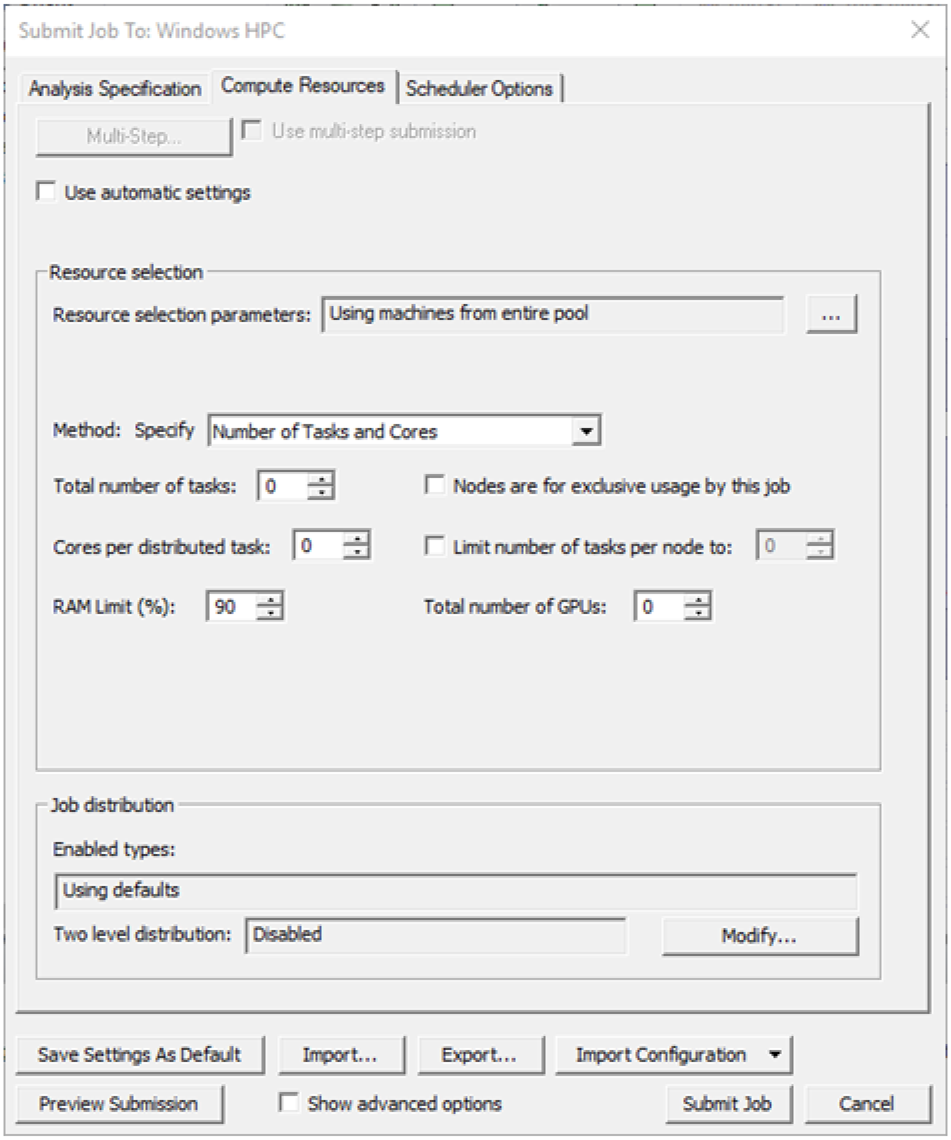
Enabling **Show Advanced Options** for Windows HPC also displays the **Use batch extract** check box and script path field.



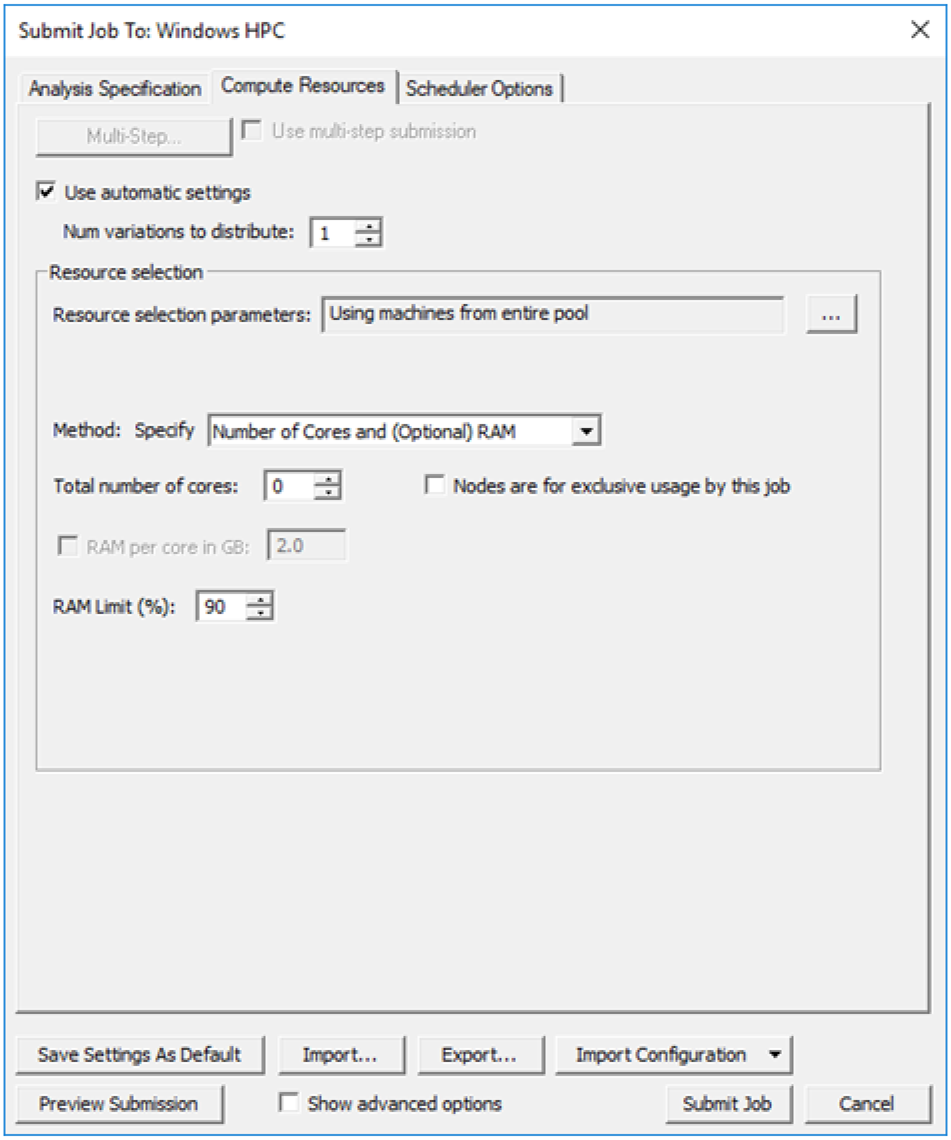
See Running Ansys Electronics Desktop from a Command line for an explanation of the solve information available through batch extract.

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.

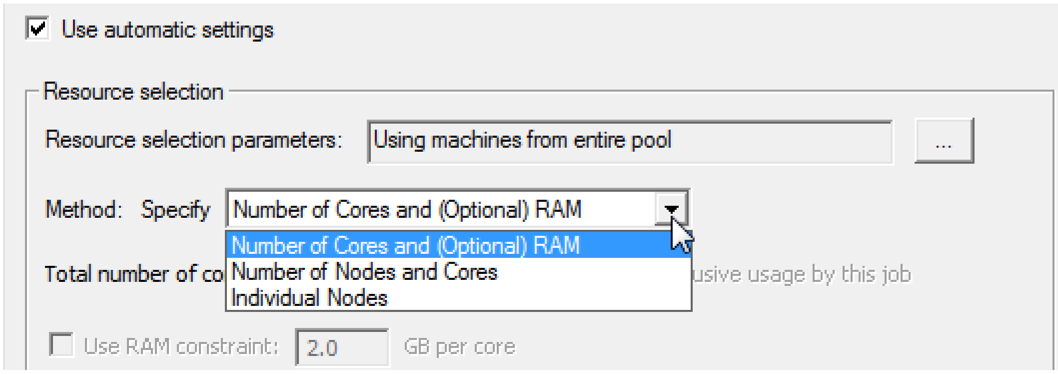


For Ansys Electronics Desktop configurations, the **Submit Job To** dialog box includes a **Use automatic settings** check box that simplifies the **Compute Resources** tab.



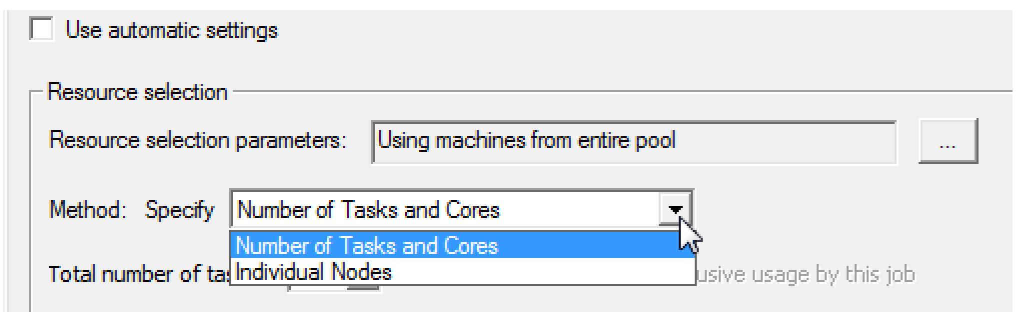
With **Use automatic settings** selected, the **Job distribution** field is removed. When using automatic settings, you can specify **Resource selection** parameters. The ellipsis button [...] opens the Compute Resource Selection Parameters dialog box. If you do not specify any parameters, the default 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**.



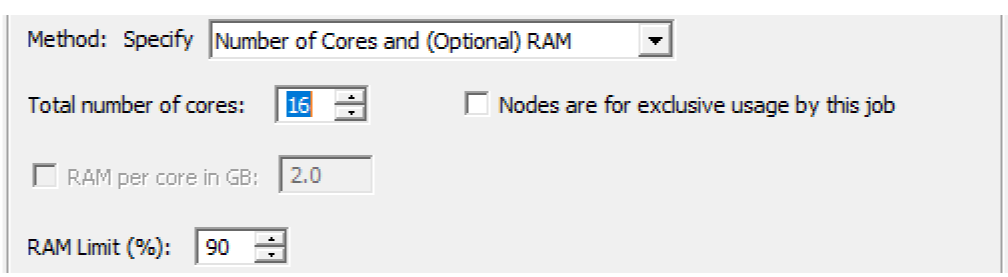
> **Note:**
>
> If you select Use automatic settings, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. It does distribute frequencies, domains, and use of multiple level domains.

If you uncheck or cannot access Use automatic settings, these two Methods are listed:

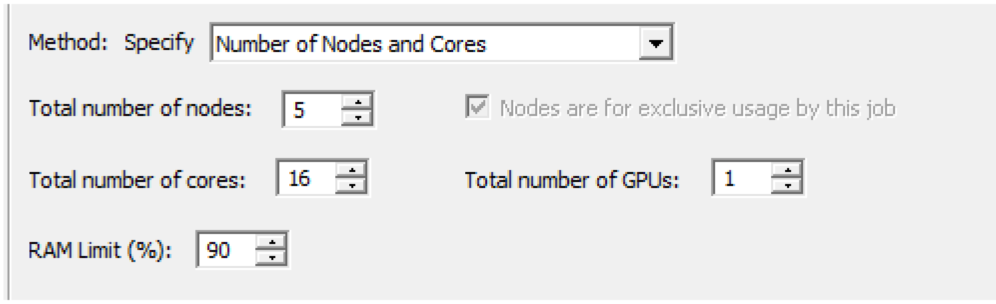


For Windows, there is a 64 core limit for a single machine. Each Method selection changes the available options listed:

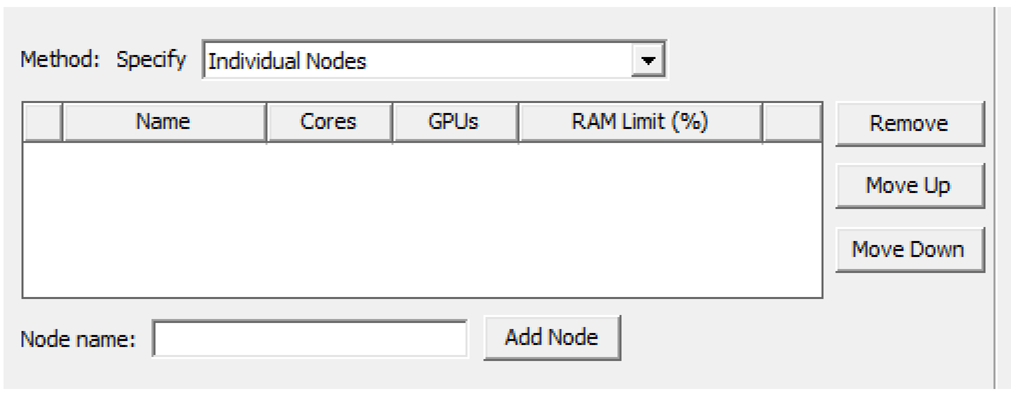
- Specify Number of Cores and (Optional) RAM



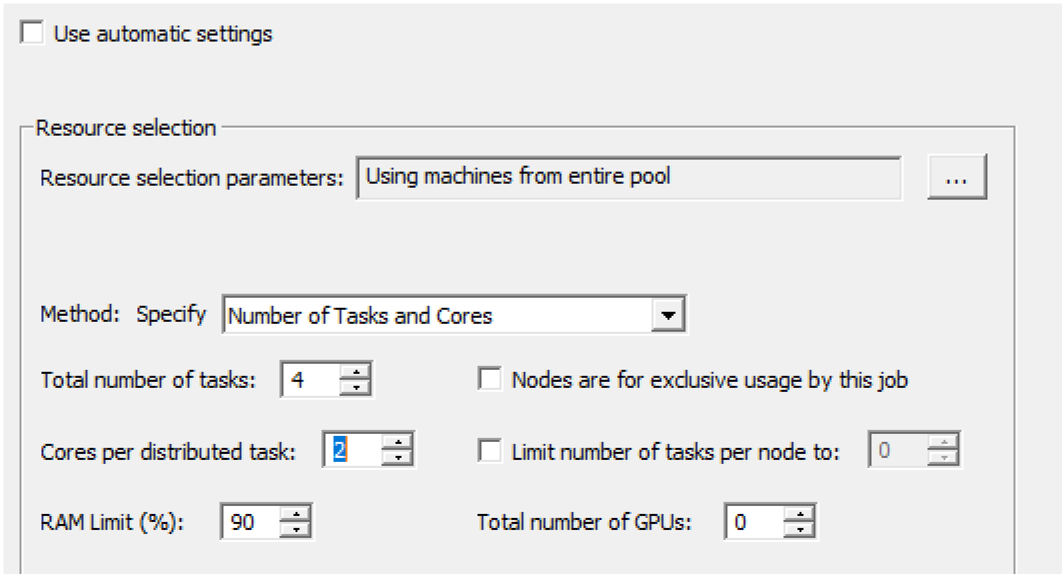
- Number of Nodes and Cores



- Individual Nodes



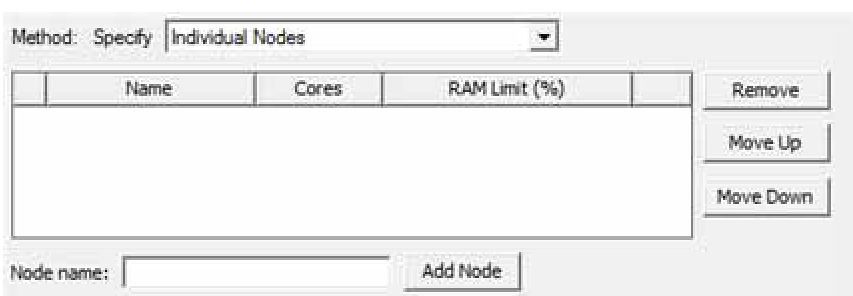
- Number of Tasks and Cores ("Use automatic settings" is unchecked for this option. Checking "Use automatic settings" means that you do not have to specify tasks or core parameters):



Individual Node List

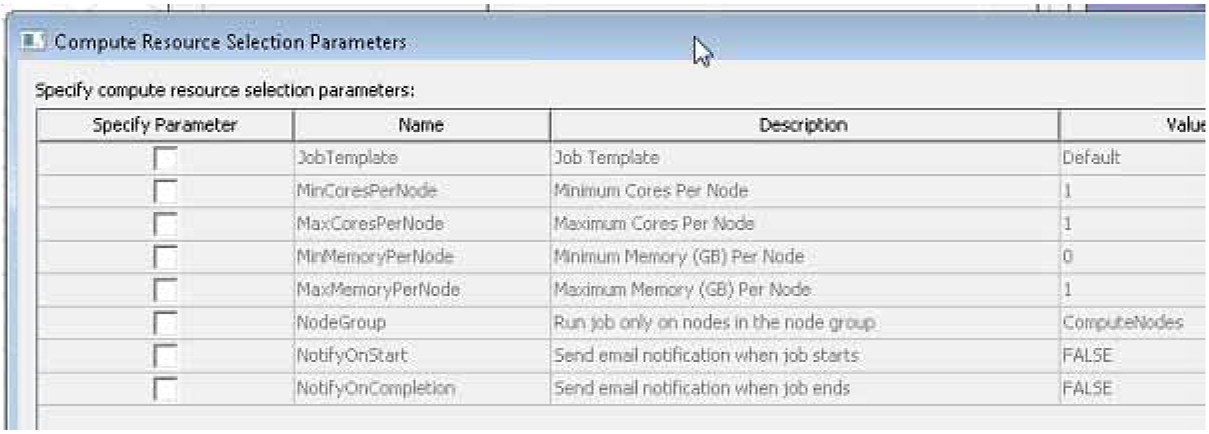
For Windows HPC jobs, you may either specify a node list, or specify job parallelization parameters, but not both.

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.



Specify Parameter	Name	Description	Value
☐	JobTemplate	Job Template	Default
☐	MinCoresPerNode	Minimum Cores Per Node	1
☐	MaxCoresPerNode	Maximum Cores Per Node	1
☐	MinMemoryPerNode	Minimum Memory (GB) Per Node	0
☐	MaxMemoryPerNode	Maximum Memory (GB) Per Node	1
☐	NodeGroup	Run job only on nodes in the node group	ComputeNodes
☐	NotifyOnStart	Send email notification when job starts	FALSE
☐	NotifyOnCompletion	Send email notification when job ends	FALSE

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.)

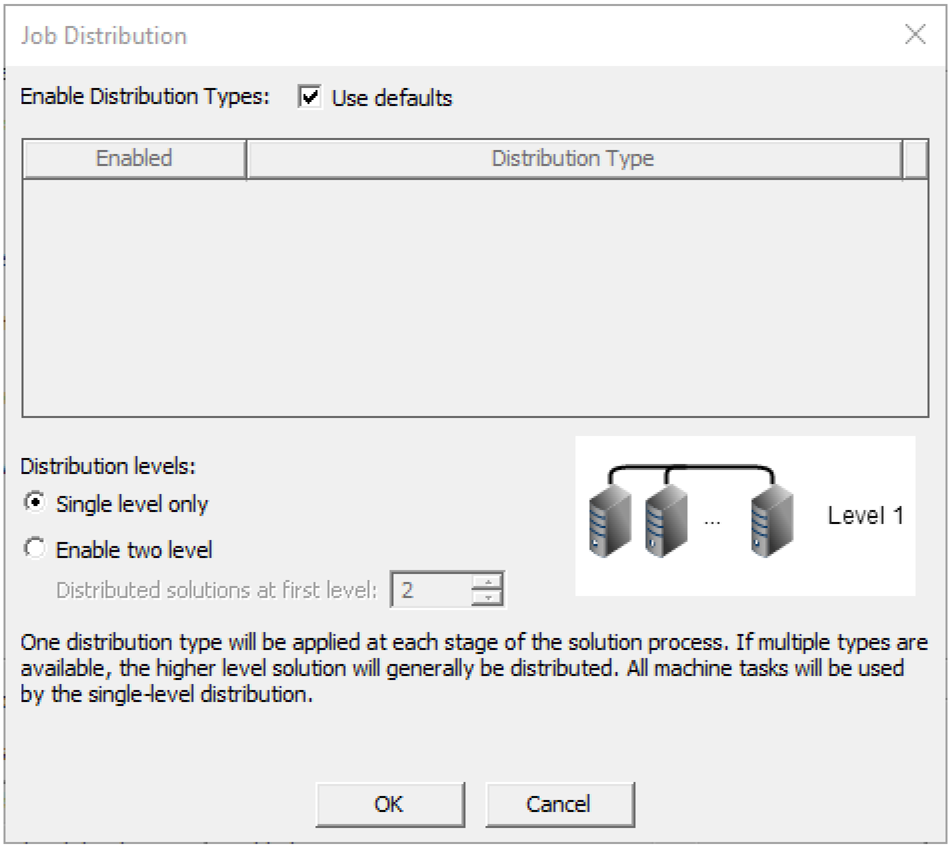
Job Parallelization

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:

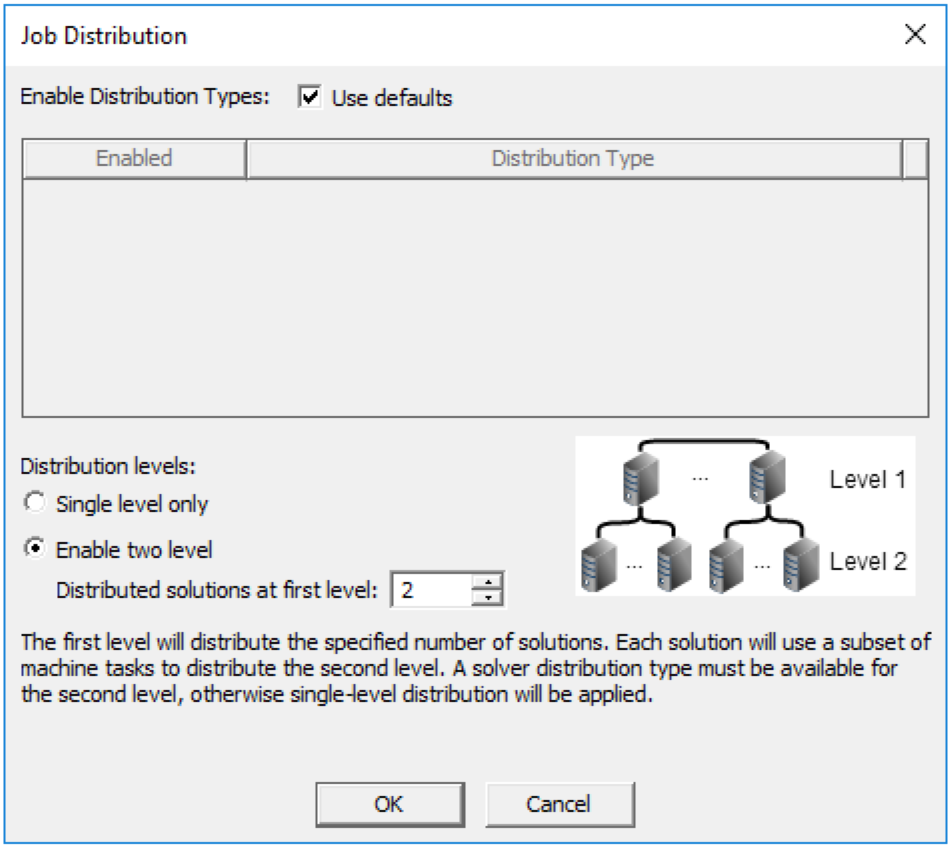
- Total number of tasks: The number of nodes requested for the job is the total number of tasks divided by limit on the number of tasks per node, rounded up if it is not an integer.
- Cores per distributed task. This determines the amount of multiprocessing per task.
- Whether nodes are for exclusive 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.



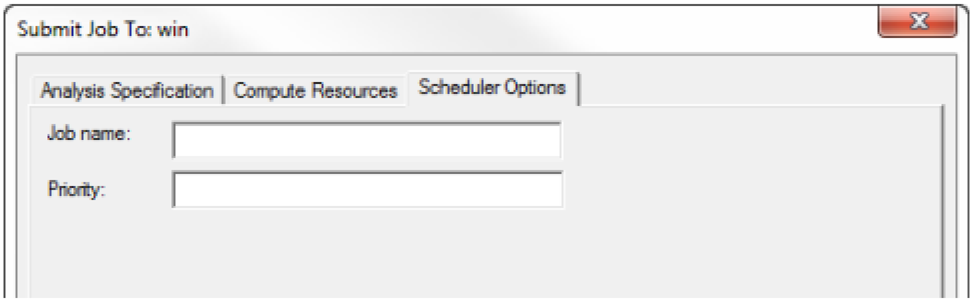
- Second level distribution operates within DSO. If available and enabled you can specify the number distributed solutions for level 1.



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 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

See the Ansys Electromagnetics Suite 2024 R2 Linux Installation Guide for additional information on supported schedulers.

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 2024 R2, 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/AnsysEM/v242/Linux64/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/AnsysEM/v242/Linux64/ansysedt
 -ng -BatchSolve ~/projects/OptimTee.aedt
```

- The -l h_rt=00:15:00 option indicates that this job has a "hard" runtime limit of 15 minutes.

Serial job using a script, with a runtime limit specified in the script:

```
qsub ~/sge/scripts/OptimTee.csh
```

- The **-b y** option is absent, so the script ~/sge/scripts/OptimTee.csh will be run when the job starts.
- The script file OptimTee.csh may contain SGE directives in addition to the command(s) to run. In this example, a directive with a hard runtime limit if 15 minutes is included in the script.

Script file contents:

```
#!/bin/csh
#$ -l h_rt=00:15:00
```

```
/opt/AnsysEM/v2421/Linux64/ansysedt -ng -BatchSolve
 ~/projects/OptimTee.aedt
```

- The SGE directive #$ -l h_rt=00:15:00 is equivalent to including -l h_rt=00:15:00 on the qsub command line.

Distributed processing job using 4 engines:

```
qsub -b y -pe pe1 4 /opt/AnsysEM/v2421/Linux64/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

Consumable Memory Limits

Serial Jobs in SGE

Parallel Jobs in SGE

Using Multithreading with Parallel Jobs

Submitting Exclusive Jobs

In many cases, clusters are used to run "large" Ansys Electromagnetics 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/AnsysEM/v242/Linux64/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/AnsysEM/v242/Linux64/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/AnsysEM/v242/Linux64/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 Platform's Load Sharing Facility (LSF)

The Load Sharing Facility (LSF) scheduler is only supported on Linux. Jobs may be submitted in any of the following ways:

- Job Submission GUI
- Using LSF commands (qsub, etc.)

See the Ansys Electromagnetics Suite 2024 R2 Linux Installation Guide for additional information on supported schedulers.

> **Note:**
>
> If a temp directory is setup by the LSF cluster administrator, analysis engines use this temp directory, overriding the setting in the Ansys EM product.

LSF Job Management

You can use Ansys Electronics Desktop to submit batch jobs to LSF and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select LSF as the scheduler.
2. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to LSF.

   LSF-specific Settings:

   - On the **Compute Resources** tab, click the **Resource Selection Parameters** ellipses button (**...**) to specify the following:
     - **Queue** – a drop-down menu lets you select Normal, chkpn_rerun_queue, idle, license, night, normal_allow_excl, owners, priorituy, 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, and a limit number of tasks per node.
3. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.

LSF Job Submission Guidelines

Before submitting an LSF job, ensure the following are true:

- The project is available in a shared drive that is accessible to all machines in the cluster.
- The project is available using the same path on all machines of cluster.

- There is sufficient space in the project directory and temp directories.
- There is sufficient memory per engine.
- The number of compute resources (Distributed Analysis machines and Multi Processing cores) will achieve the desired scale factor and effective resource utilization.

Integration of Ansys EM Products with LSF

With LSF you do not need to set 'Distributed Analysis Machines' or 'Remote Machine' options. Instead, you submit a job to LSF, requesting the appropriate resources for the job (number of processors, memory per processor, etc.).

For example:

```
bsub -n 1 ansysedt.exe -Batchsolve -ng -local -machinelist num=1
OptimTee.aedt
```

```
bsub -n 4 ansysedt.exe -Batchsolve -ng -Distributed -machinelist
num=4 OptimTee.adsn
```

The job is queued by LSF until the requested resources are available. Upon resource availability, LSF starts Electronics Desktop with the specified command line on one of the allocated machines. During analysis, Electronics Desktop dynamically obtains the allocated 'Distributed Analysis Machines' from LSF. Electronics Desktop interfaces with LSF to launch engines on remote machines without going through Ansys RSM.

Installing Ansys EM Tools on LSF Cluster

The LSF scheduler is supported on Linux only.

lsf.conf should contain this line:

```
LSF_UNIT_FOR_LIMITS=MB
```

The administrator should have this line in the lsb.params file to ensure that memory reservations are per-slot (per-core):

```
RESOURCE_RESERVE_PER_SLOT=Y
```

Jobs may be submitted in any of the following ways:

- Using LSF commands (bsub, etc.)
- Using the generic scheduler GUI in local mode
- Using the generic scheduler GUI in service mode

See the Ansys Electromagnetics Suite 2024 R2 Linux Installation Guide for additional information on supported schedulers.

Setup:

1. Install on a single node, on a shared drive.
2. Setup 'temp directory' to a path that is same on all nodes (for example, /tmp).
3. Ensure that the product is available using the same path on all nodes.

Permissions:

- All users of the cluster should have read/write permissions to temp directory.
- All users should have read/execute permissions to installation directory.
- Turn OFF firewall between cluster nodes.

When a desktop scheduler GUI is run the same node as the job submission node, no other configuration is necessary; installation is sufficient. Select the scheduler through the Electronics Desktop GUI. Ensure that scheduler commands are available in the path before you launch Electronics Desktop.

> **Note:**
>
> There is no need to install RSM unless you are using the scheduler GUI on a post-processing node that is different from the job submission node. In this case, RSM must be configured with the scheduler type and path.
>
> A post-processing node is a node in the cluster that can run Electronics Desktop in graphical mode. A job submission node is a node in the cluster in which job submission commands are available.

Per-slot Resource Reservation

Set the cluster for per-slot resource allocation if the automatic cores and RAM resource selection method is to be used. You can check the cluster to see if per-slot resource allocation is configured by using the "bparams -a" command. Search the output for "RESOURCE_RESERVE_PER_SLOT" to determine the setting (either "Y" or "N"). If set to "N" then consult the LSF administration guide on how to change this to "Y".

Scenario 1: The post-processing node and job-submission node roles are served by distinct machines.

In this case, perform the following configuration:

The job-submission node should be configured to run RSM service, which serves as a proxy to scheduler. The RSM Service should be running as 'root' in order to facilitate jobs running using the credentials of the job's owner. A **configuration file in the RSM installation folder should be edited** to specify information regarding the scheduler that manages jobs on this cluster. A block labeled 'Scheduler' must be included within the 'AnsoftCOMDaemon' block. This block contains two string entries:

- SchedulerName: this contains the unique part of the scheduler proxy library name
- ConfigString: this contains a scheduler specific configuration string

The case of the SchedulerName string is significant on Linux because Linux file names are case sensitive. In Ansys Electromagnetics Suite 2024 R2, 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 /Program Files/AnsysEM/v242/Win64/ansysedt -ng
-BatchSolve -machinelist num=4 ~/projects/OptimTee.aedt
```

The -n 1 option indicates that this job runs on one core.

Serial Job Requiring a Minimum of 4GB

```
bsub -n 1 -R "select[mem>4000]"
/Program Files/AnsysEM/v242/Win64/ansysedt -ng
-BatchSolve -machinelist num=4 ~/projects/OptimTee.aedt
```

The -R "select[mem>4000]" option indicates that this needs a minimum of 4 GB memory.

Multi-processing Job using 4 Cores

```
bsub -n 4 -R "span[ptile=4]"
"/Program Files/AnsysEM/v242/Win64/ansysedt -ng -BatchSolve
-batchoptions -machinelist num=4 ~/projects/OptimTee.aedt"
```

- The -R "span[ptile=4]" option indicates that the four cores need to be on the same machine.
- The -batchoptions option indicates that HFSS should use four cores for multi-processing.

- The entire hfss command is in double quotes, and the double quotes enclosing the -batchoptions value are escaped. Each of these double quotes is replaced by the sequence "\"".

Distributed Processing Job using 4 Engines

```
bsub -n 4 /Program Files/AnsysEM/v242/Win64/ansysedt -ng -BatchSolve
-Distributed ~/projects/OptimTee.aedt
```

- The -n 4 option indicates that the four cores are needed for the job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines. The -Distributed option can have additional options, such as includetypes=xxx, excludetyeps=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".

Distributed Processing and Multi-processing Job using 4 Cores, with 2 Cores for Multi-processing

```
bsub -n 4 -R "span[ptile=2]" ~/projects/OptimTee.csh
```

Shell Script (~/projects/OptimTee.csh):

If a command is included in the **bsub** command line, the entire command will be processed by the command shell two times. The command is processed when the **bsub** command is processed by the shell and is processed again when the command is started by the scheduler. This example shows how to use a shell script so that the command line will be processed only once. The command is placed in the shell script, and then the shell script pathname is placed in the **bsub** command line. Then, the command is only processed by the command processor when the job is started. When using this approach, the shell script should be accessible from all of the cluster hosts.

```
#!/bin/csh
/Program Files/AnsysEM/v242/Win64/ansysedt -ng -BatchSolve
-Distributed -machinelist num=2 -batchoptions
~/projects/OptimTee.aedt
```

- The **-n 4** option indicates that the four cores are needed for the job.
- The **-R "span[ptile=2]"** option indicates that the cores must be allocated in groups of two cores on the same machine.
- The **-machinelist num=2** option indicates that this is a DSO job and that a total of two engines will be started.
- The **hfss** command is placed in the shell script (~/projects/OptimTee.csh). In the **bsub** command line, the **hfss** command is replaced by the shell script pathname.

Monitoring LSF Batch Jobs

You can monitor jobs through the Electronics Desktop user interface, or through the command line.

The suggestions below are for batch jobs run under LSF.

Ansys Electronics Desktop -monitor Command Line Option

The -monitor command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

LSF bpeek Command

The LSF bpeek command may be used to monitor job progress. The command **pbeek [-f]** *job_id* displays the standard output and standard error produced by the job with id *job_id* from the job start to the current time (the time when the command is executed). This command is only valid for jobs that have not yet finished. When used with the **-f** option on Linux, the output of the job is displayed using the command tail **-f**, so that ongoing progress may be monitored.

In order to display messages to standard output and standard error, specify the **-monitor** command line option on the Ansys EM tool command line. Then, these messages can be seen using the LSF bpeek command.

Terminating LSF Batch Jobs

To cancel or terminate an Ansys EM LSF batch job, we recommend using the job monitoring UI to terminate jobs cleanly, rather than using the bkill commands. Using this approach will allow the batch job to shut down in an orderly fashion.

Using the LSF **bkill** command without the **-s SIGTERM** option or simply terminating the job processes may cause some of the following problems:

- Some engine processes are not shut down and continue to run.
- LSF job is not fully removed.
- Project .lock file is not removed.
- MainWin core service processes (watchdog, mwrpcss and/or regss) are not stopped.

Some of these may interfere with submission of additional LSF batch jobs. For example, it may be necessary to manually remove the project lock file to submit another batch job for the same project. MainWin core service processes may also interfere with starting subsequent Ansoft batch jobs. Normally, these processes should timeout and end 15 seconds after the Ansys Electromagnetics product shuts down. Any MainWin core service processes (watchdog, mwrpcss and/or regss) that continue to run for more than 15 seconds after the product has stopped may be hung. The hung processes may need to be manually killed, after ensuring that these processes are associated with an Ansys EM job that has finished or terminated.

Stop a job cleanly - ensures that the results obtained until now are preserved:

`bkill -s TERM <jobid>.`

Stop an job abruptly - results are most likely lost. You have to manually remove the project lock file:

`bkill <jobid>`

LSF Known Issues and Workarounds

The following are known issues. Workarounds are noted when available:

- Desktop or remote machine cannot have multiple IP addresses. This is unsupported.
- Core dump files may appear when a job has finished running. Results are still computed correctly. Workaround: Limit size of core dumps to 0 through the following job submit option: `bsub -C 0 -n <number-of-cores> -q <queue-name>`
- Firewall should be turned off on all machines in the cluster.
- Sometimes LSF ends a job (for example, a job may be preempted due to a high priority job). This may result in the presence of a .lock file in the project directory. You must manually delete the .lock file before continuing with further analysis.
- When an LSF job is ended, MainWin services (watchdog, regss, and mwrcpss) could keep running. The result is that later jobs cannot start on the machine. The fix is to end these processes before starting a new job.
- Analysis fails abruptly when running out of resources (cpu/memory/disk). Ensure sufficient resources are provided.

LSF Troubleshooting

The following are general troubleshooting steps:

1. Ensure the LSF `lsrun` command is enabled.
2. Look for user errors.

   For example:

   - Are the executable path and project path correct and complete?
   - Are there sufficient resources (CPU/Memory/Disk) allocated to the job?
   - Is the project available on the execution host?
   - Does the job submitter have read/writer permissions on the project directory and read/execute permissions on the installation directory?
   - Is the project locked?

3. Determine whether this is a standalone product issue.
   - Run Electronics Desktop on the machine outside of the scheduler and see if it opens and analyzes.
4. Examine outputs and logs.
   - Output of the LSF batch job. Obtain this using LSF commands: `"bacct -l <jobid>"`
   - Batch log (typically <projectname>.log, located in the project directory.
5. Enable additional debug logs using the steps below.

   In the job submission window, set the following environment variables:

   - ANSOFT_DEBUG_MODE = 1
   - ANSOFT_DEBUG_LOG = <*path to directory accessible by all machines in the cluster*>
   - ANSOFT_DEBUG_LOG_SEPARATE = 1
   - ANSOFT_LSF_LOG = <*path to a specific .log file in the directory set under ANSOFT_DEBUG_LOG*>
6. For each pair of machines between which remote analysis fails, run `ping remote-machine` and note the output.
7. For each machine in the network, dump network interfaces (for example, run `ifconfig -a`) and note the output.

Integration with PBS (Portable Batch System)

The 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)

See the Ansys Electromagnetics Suite 2024 R2 Unix/Linux Installation Guide for additional information on supported schedulers.

PBS Job Management

You can use Ansys Electronics Desktop to submit batch jobs to PBS and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select PBS as the scheduler.
2. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to PBS.

   PBS-specific Settings:

   - On the **Compute Resources** tab, click the **Resource Selection Parameters** ellipses button (**...**) to specify either the **Queue** or **QueueAtServer** parameter.

For **Queue**, you may select a queue for the job from the list of queues configured for the default server. Only queues that are enabled and that do not have the from_route_only attribute set to true are listed.

For **QueueAtServer**, you may specify a queue at the default server or at another server by entering text into this field in one of three formats:

- queue_name
- @server_name
- queue_name@server_name

The queue_name format submits with the name queue_name. The other formats submit with the name server_name. The *destination* value of the -q *destination* option on the qsub command line is the user-specified string. *This string will not be validated by the scheduler proxy library*.

Either **Queue** or **QueueAtServer** may be specified, but both may not be specified. If neither **Queue** nor **QueueAtServer** is specified, the job is submitted to the default queue at the default server.

3. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.

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/AnsysEM/v242/ansysedt -ng -BatchSolve
~/projects/OptimTee.aedt
```

Serial job that needs a minimum of 4GB memory and two hours of real (wallclock) time:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l walltime=2:00:00
#PBS -l mem=4gb
/opt/AnsysEM/v242/ansysedt -ng -BatchSolve
~/projects/OptimTee.aedt
```

Multi-processing job using 4 cores:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l ncpus=4
/opt/AnsysEM/v242/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/AnsysEM/v242/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/AnsysEM/v242/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.

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" below
- "Distributed Analysis and Multi-Processing in the Same Job" on the next page

Serial Job on a Single Processor

Suppose Ansys Electronics Desktop is installed at "C:\Program Files\AnsysEM\v242\Win64\ and you are using RSM for DSO:

```
C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -BatchSolve -
machinelist num=2
-monitor \\shared_drive\projs\OptimTee.aedt
```

User is using LSF for remote-analysis/DSO

```
bsub -n 1 C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -
BatchSolve -machinelist num=3 -monitor -local \\shared_
drive\projs\OptimTee.aedt
```

Distributed Job Using Four Processors

Ansoft RSM

```
C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -Batchsolve -
monitor -Distributed
-machinelist list="10.1.1.221, 10.1.1.222, 10.1.1.223, 10.1.1.224"
\\shared_drive\projs\OptimTee.aedt
```

LSF

```
bsub -n 4 C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -
Batchsolve -monitor
-Distributed -machinelist num=4
\\shared_drive\projs\OptimTee.aedt
```

Multiprocessing Job Using Four Cores

Multi-processing job using 4 cores

```
bsub -n 4 -R "span[ptile=4]" C:\Program
Files\AnsysEM\v242\win64\ansysedt.exe -ng -monitor
-Local -BatchSolve -machinelist num=4 -batchoptions \\shared_
drive\registry.txt \\shared_drive\projs\OptimTee.aedt
```

This requests 4 cores to come from the same machine, as multi-processing needs cores to be on the same machine

Distributed Analysis and Multi-Processing in the Same Job

Distributed-processing using 4 engines and multi-processing using 4 cores, using a total of 16 cores

```
bsub -n 16 -R "span[ptile=4]" c:Program
Files\AnsysEM\v242\win64\ansysedt.exe -ng
-BatchSolve -Distributed -machinelist num=4
-batchoptions \\shared_drive\registry.txt
\\shared_drive\projs\OptimTee.aedt
```

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.

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 2024 R2 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 2024 R2.

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\AnsysEM\v242\Win64, then the scheduler proxy library must be installed in directory C:\Program Files\AnsysEM\v242\Win64\schedulers.

The scheduler proxy library base name must match "libprefix_scheduler" on Windows and "liblibprefix_scheduler" on Linux. The extension must be a valid extension for a dynamically loaded library on the platform where it is used. The scheduler proxy library name prefix libprefix shall be unique, so it does not conflict with other scheduler proxy libraries in the same directory. To avoid confusion, the scheduler proxy library name should be all lower case on OSs where file names are case sensitive.

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]
- 32 bit code [MACHINE:X64]

Linux

The proxy library should be compiled and linked as shared library (*.so) file. The following compiler and linker options are recommended when building using gcc/g++:

Compiler Options

- Generate 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. See the GetMachineListAvailableForDistribution section above. If the hostname does not refer to the local host, then this function shall use the scheduler to launch the engine on the specified host. If the hostname refers to the local host, then the engine may be started as a child process, or it may be started using the scheduler.

Signature

```
extern "C" int FN_PREFIX_LaunchProcess(const char* hostName,
const char* exePathName, const char* arg1, const char* arg2);
```

Arguments

hostName: The name of the host where the process is to be launched.

exePathName: The pathname of the analysis engine executable to be started.

arg1: The first argument of the analysis engine command line.

arg2: The second argument of the analysis engine command line.

Return Value

Returns 0 on success. Returns a non-zero value if an error occurs.

> **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.

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.

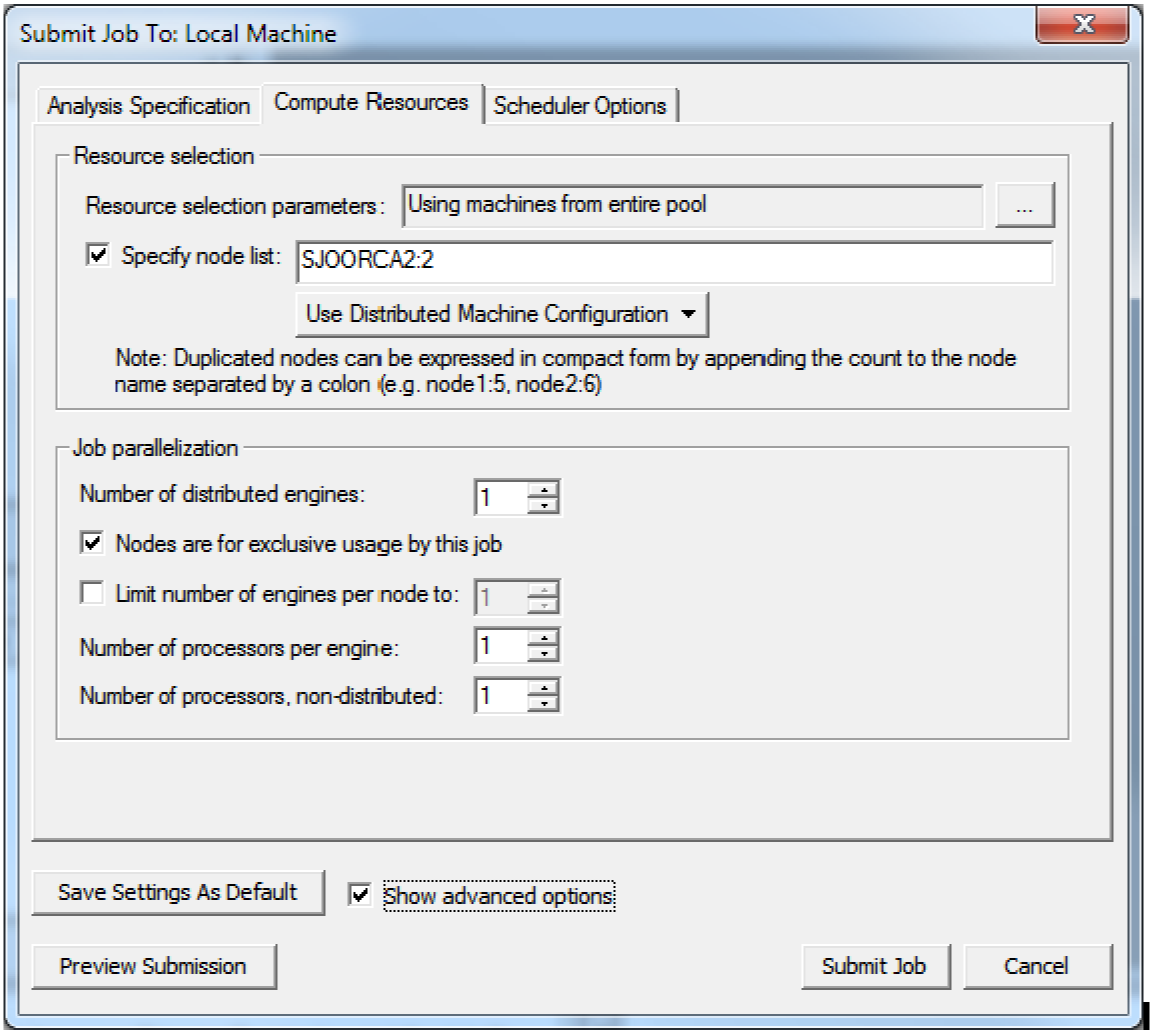
int GetSchedulerInfo(std::string& msg, std::string& schedulerName, std::string& schedulerDescription, std::string& schedulerVersion):

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:

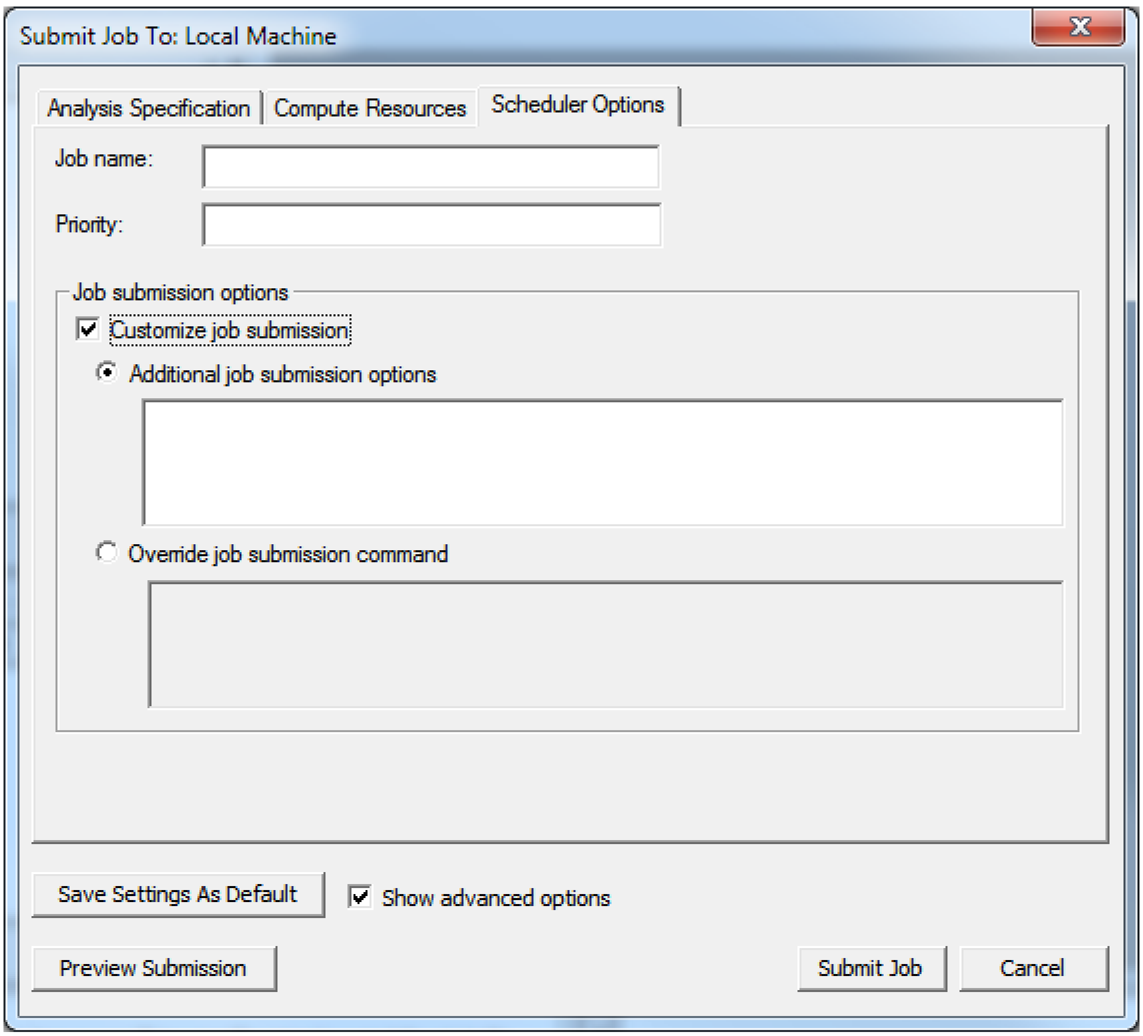
- the total number of distributed engines,
- the number of cores to allocate for each distributed engine,
- the maximum number of engines to allocate to a single node (optional), and
- the number of cores to allocated for the non-distributed portion of the analysis.
- It also contains a boolean parameter 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.

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

- None of the Proxy Functions are Called
- Troubleshooting IsProductLaunchedInYourEnvironment Function
- Troubleshooting GetSchedulerDisplayName

- Troubleshooting GetThisJobID
- Troubleshooting GetTempDirectory
- Troubleshooting GetMachineListAvailableForDistribution
- 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\AnsysEM\v242\Win64, then the IronPython program must be installed in directory C:\Program Files\AnsysEM\v242\Win64\schedulers.

The program file extension must be ".py". Select the program name so that it does not conflict with other IronPython programs in the same directory. If the Operating System or file system treat file names in a case sensitive manner, the file extension ".py" must be lower case.

Python Programming Notes

The scheduler program will be run in the IronPython environment both on Microsoft Windows and on Linux. There are some differences between IronPython and CPython. The version of IronPython in use 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

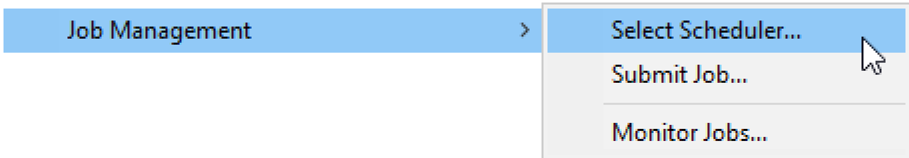
Returns 0 on success. Returns a non-zero value if an error occurs.

> **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.

Selecting a Scheduler

Access the **Select Scheduler** window one of three ways:

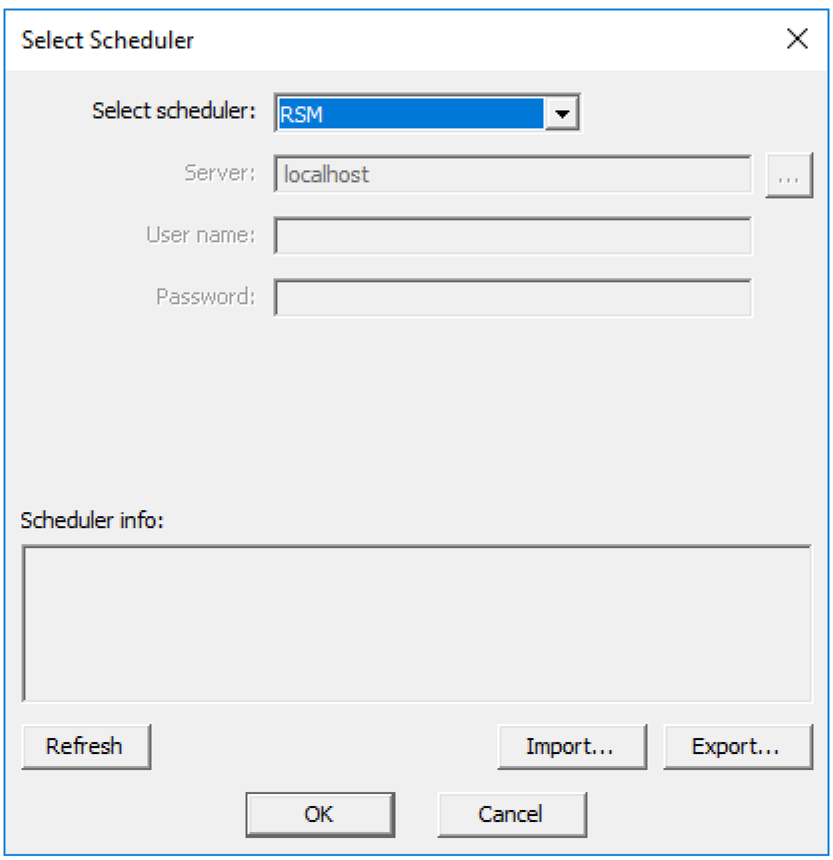
- Click **Tools** > **Job Management** > **Select Scheduler...**



- Select the **Simulation** tab and click the **Scheduler** icon.

- From a command window, use the -showselectscheduler command.

The **Select Scheduler** window appears:



From the **Select Scheduler** window:

- Use the **Select scheduler** drop-down menu to select a scheduler.

> **Note:**
>
> If you select a scheduler that is unsupported in your environment, you will receive a warning message.
>
> See HPC Integration for a list of currently supported schedulers.

- If applicable for the scheduler type, enter server and user information.
- Information about the selected scheduler appears in the **Scheduler info** field.
- Click **OK** to complete your selection.

Submitting a Job

Ansys Electromagnetics Desktop supports its own Remote Simulation Management (RSM) and Ansys Cloud 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.

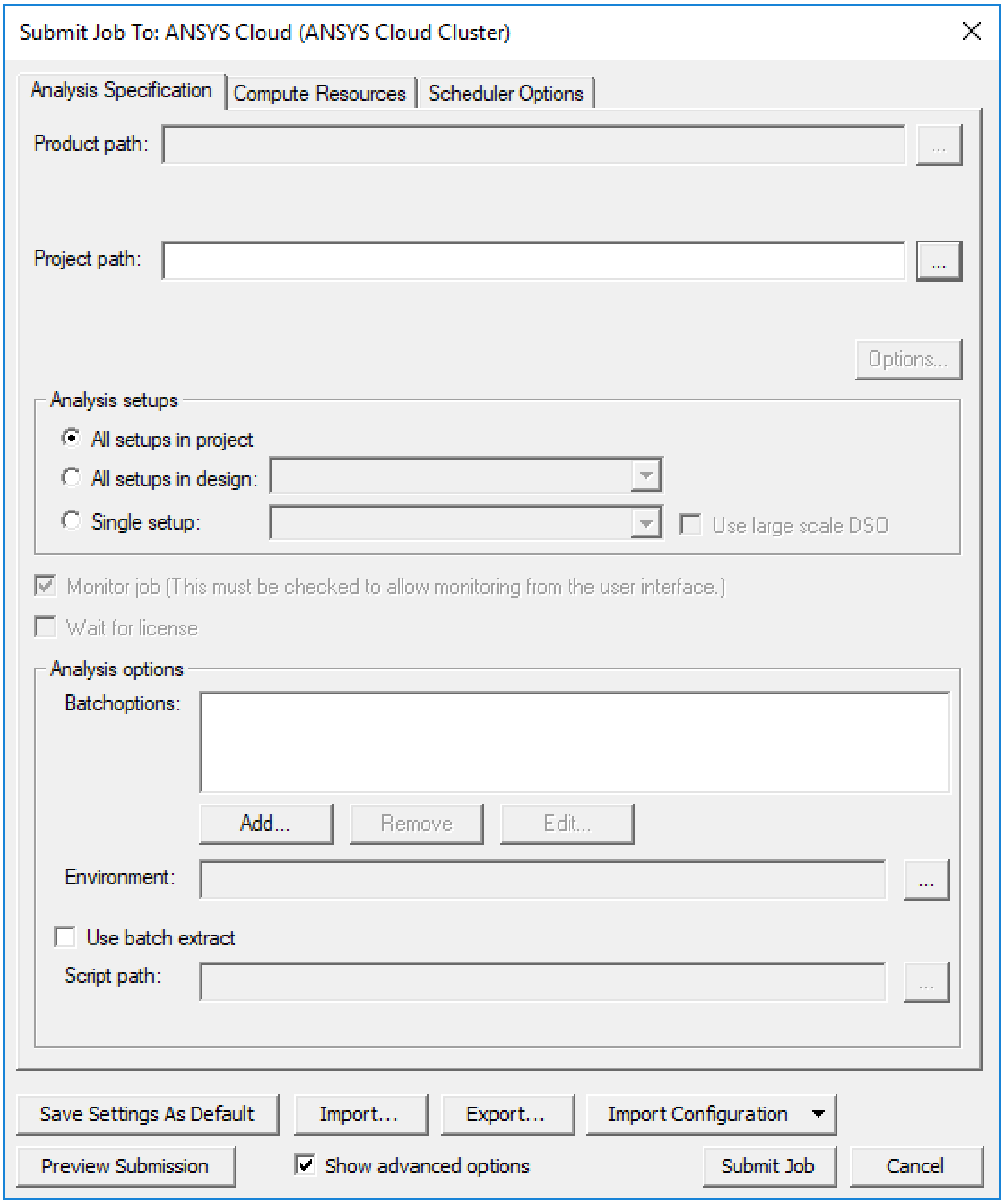
> **Note:**
>
> For certain methods of resource selection for Windows HPC job submission, Ansys Electronics Desktop checks for nodes being online. In an auto-scaling cluster, nodes can fail the online check because they're not online until a job is running. An environment variable allows customers to 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 **Circuit** > **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.

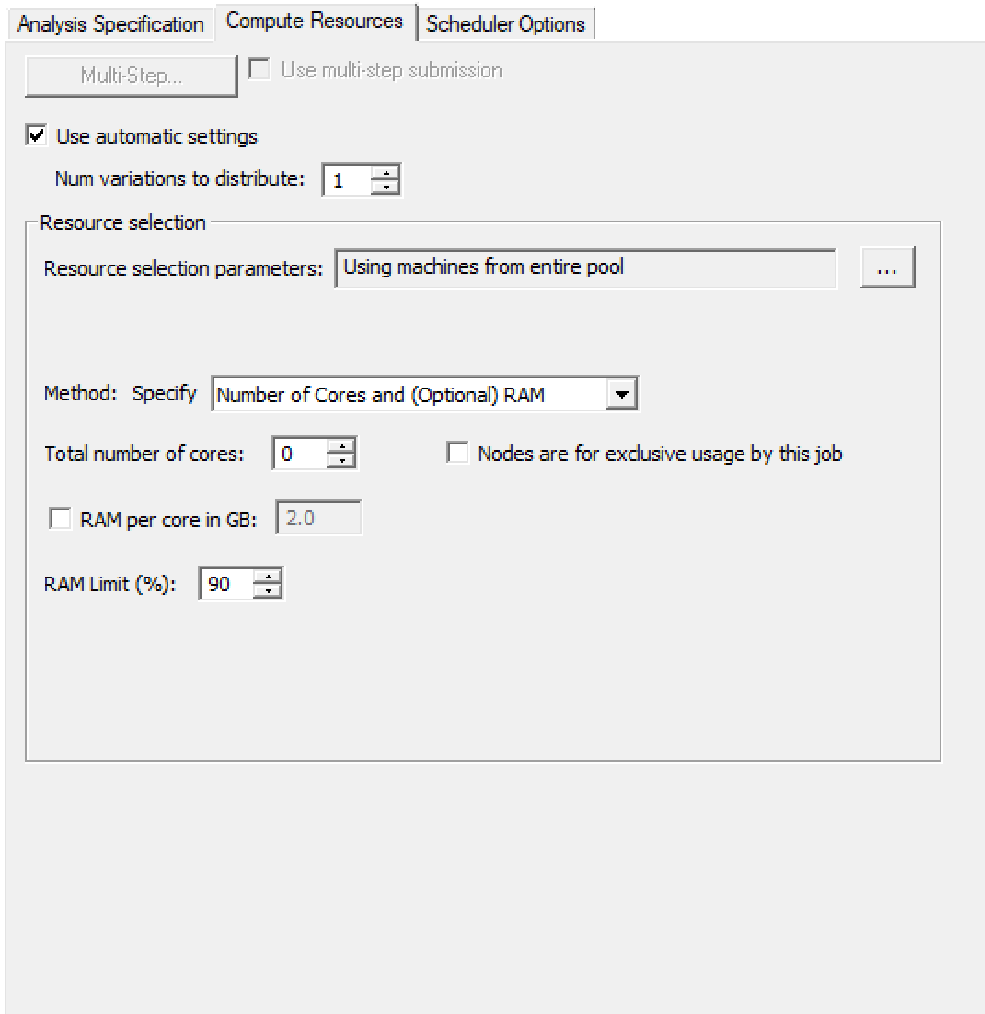
   The **Product path** and **Project** fields support mapped drives. Click the ellipses (**...**) to select files.

   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.



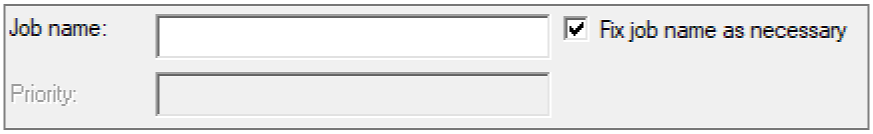
Options may vary slightly depending on your selected scheduler.

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.
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

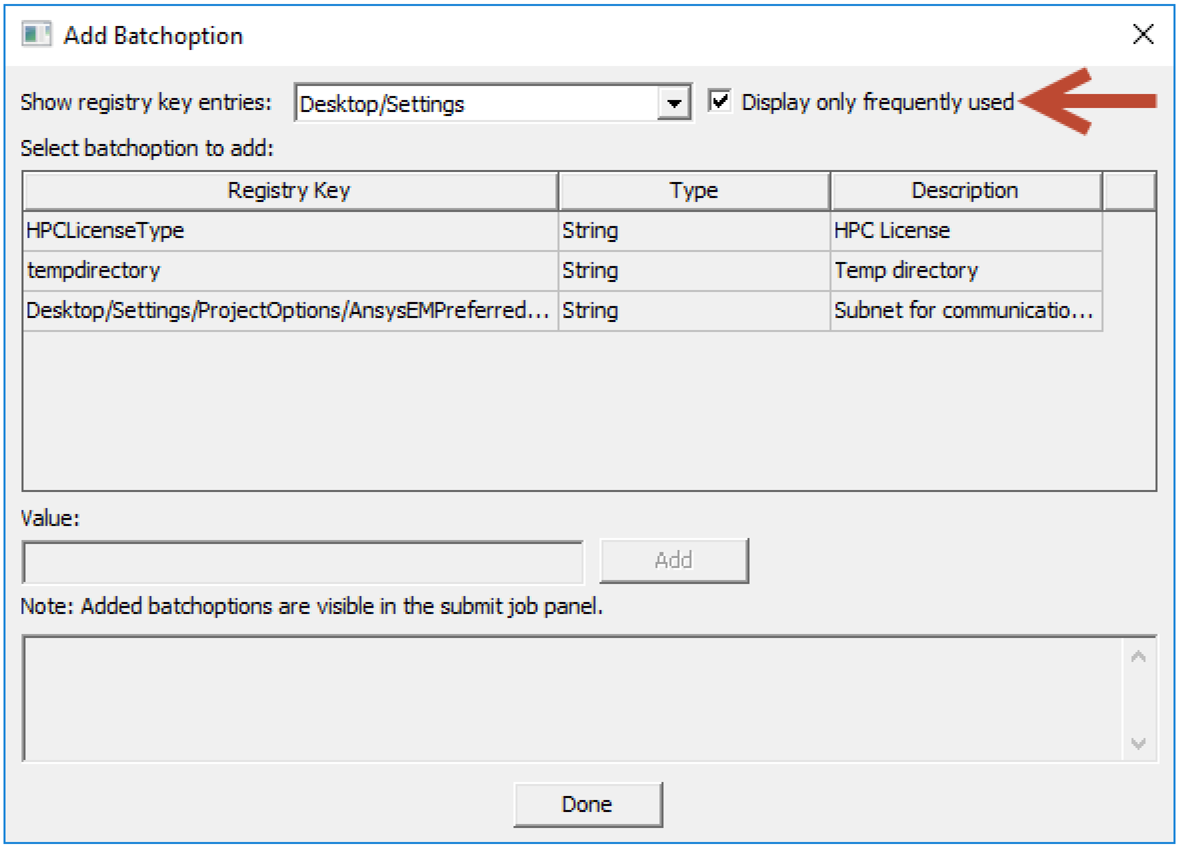
From the job submission window, you can select **Show advanced options** to enable additional analysis options and job submission options.

This topic covers the following:

- Batchoptions
- Environment Variables
- Batch Extract
- Customize Job Submission

Batchoptions

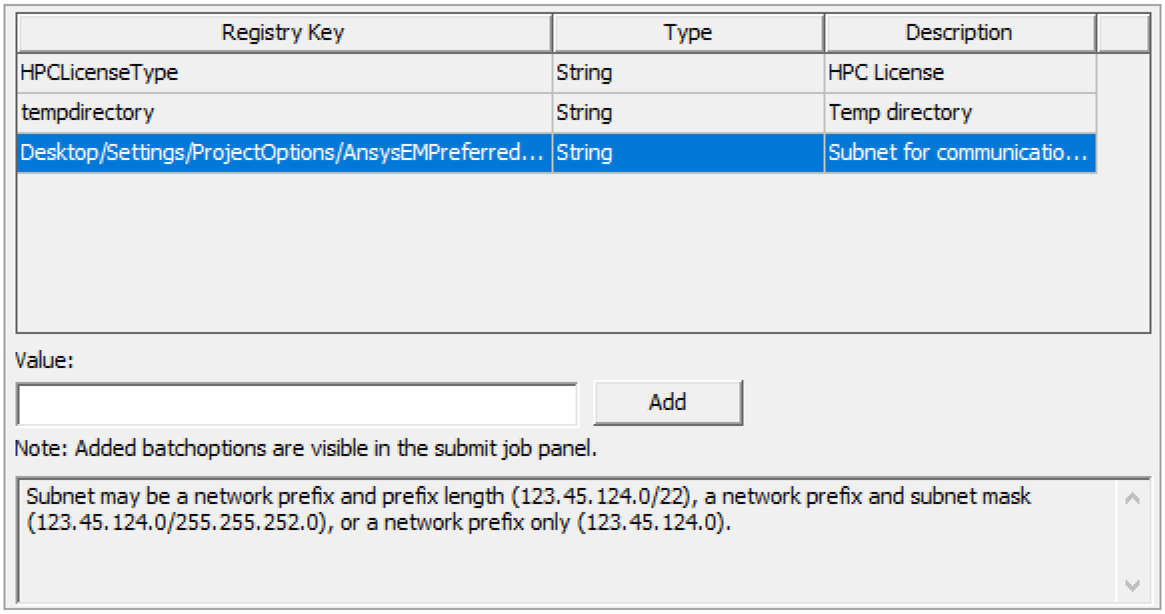
In the **Submit Job To** window, under **Batchoptions**, click **Add** to open the **Add Batchoption** window.



The **Show registry key entries** drop-down menu allows you to select categories of registry keys to display.

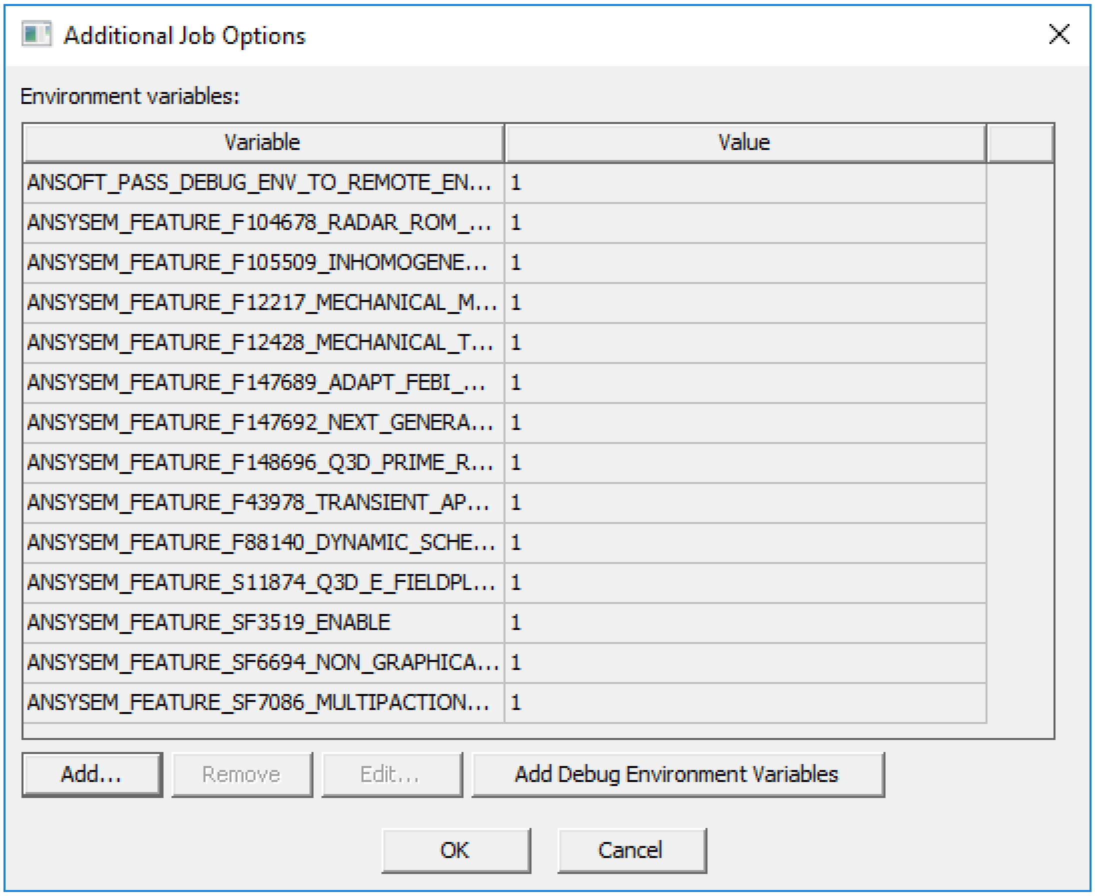
Note that **Display only frequently used** is selected by default. Deselect this to view all options for the selected category.

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 8-170

Serial Job on a Single Processor

Suppose Ansys Electronics Desktop is installed at "C:\Program Files\AnsysEM\v242\Win64\ and you are using RSM for DSO:

```
C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -BatchSolve -
machinelist num=2
-monitor \\shared_drive\projs\OptimTee.aedt
```

User is using LSF for remote-analysis/DSO

```
bsub -n 1 C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -
BatchSolve -machinelist num=3 -monitor -local \\shared_
drive\projs\OptimTee.aedt
```

Distributed Job Using Four Processors

Ansoft RSM

```
C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -Batchsolve -
monitor -Distributed
-machinelist list="10.1.1.221, 10.1.1.222, 10.1.1.223, 10.1.1.224"
\\shared_drive\projs\OptimTee.aedt
```

LSF

```
bsub -n 4 C:\Program Files\AnsysEM\v242\win64\ansysedt.exe -ng -
Batchsolve -monitor
-Distributed -machinelist num=4
\\shared_drive\projs\OptimTee.aedt
```

Multiprocessing Job Using Four Cores

Multi-processing job using 4 cores

```
bsub -n 4 -R "span[ptile=4]" C:\Program
Files\AnsysEM\v242\win64\ansysedt.exe -ng -monitor
-Local -BatchSolve -machinelist num=4 -batchoptions \\shared_
drive\registry.txt \\shared_drive\projs\OptimTee.aedt
```

This requests 4 cores to come from the same machine, as multi-processing needs cores to be on the same machine

Distributed Analysis and Multi-Processing in the Same Job

Distributed-processing using 4 engines and multi-processing using 4 cores, using a total of 16 cores

```
bsub -n 16 -R "span[ptile=4]" c:Program
Files\AnsysEM\v242\win64\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

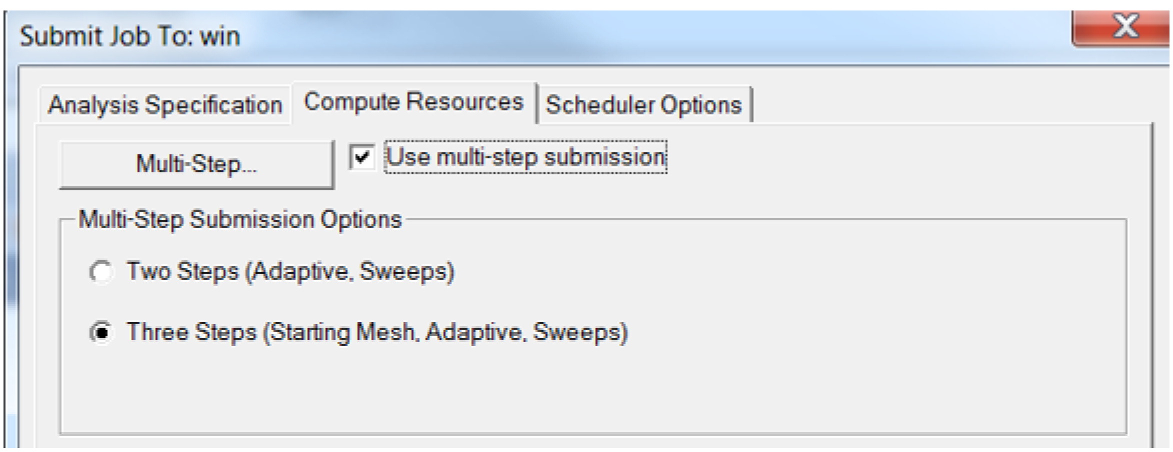
Multi-step job submission allows you to divide the simulation of a project on a cluster as two or more jobs, each of which has unique resource specifications. For example, creating the initial mesh and doing adaptive refinement can use a single machine, while frequency sweeps can easily be distributed 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

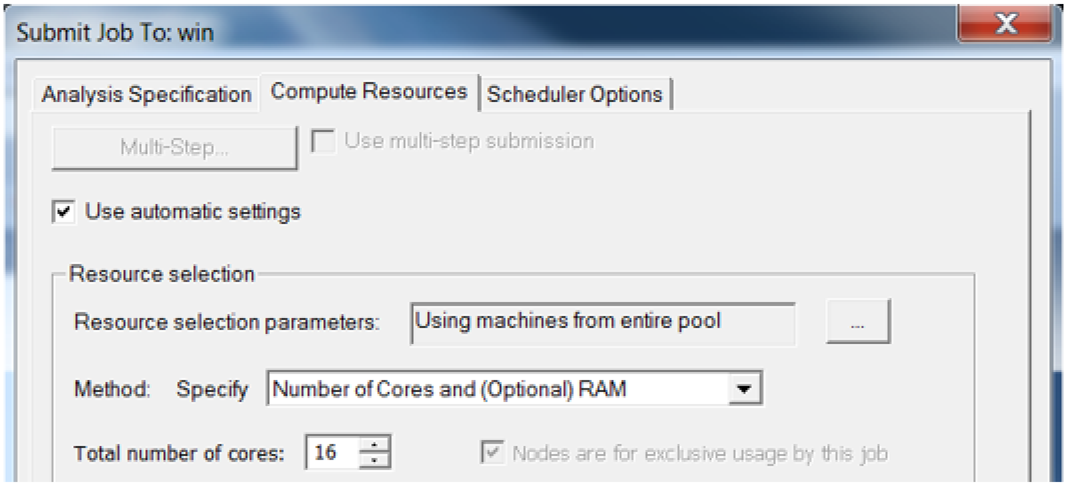
Multi-step job submission is only allowed for a single setup of a project. Only certain design types (or setup types for a given design type) offer this functionality. Ansoft RSM does not support multi-step jobs because it does not have queuing capabilities.

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.



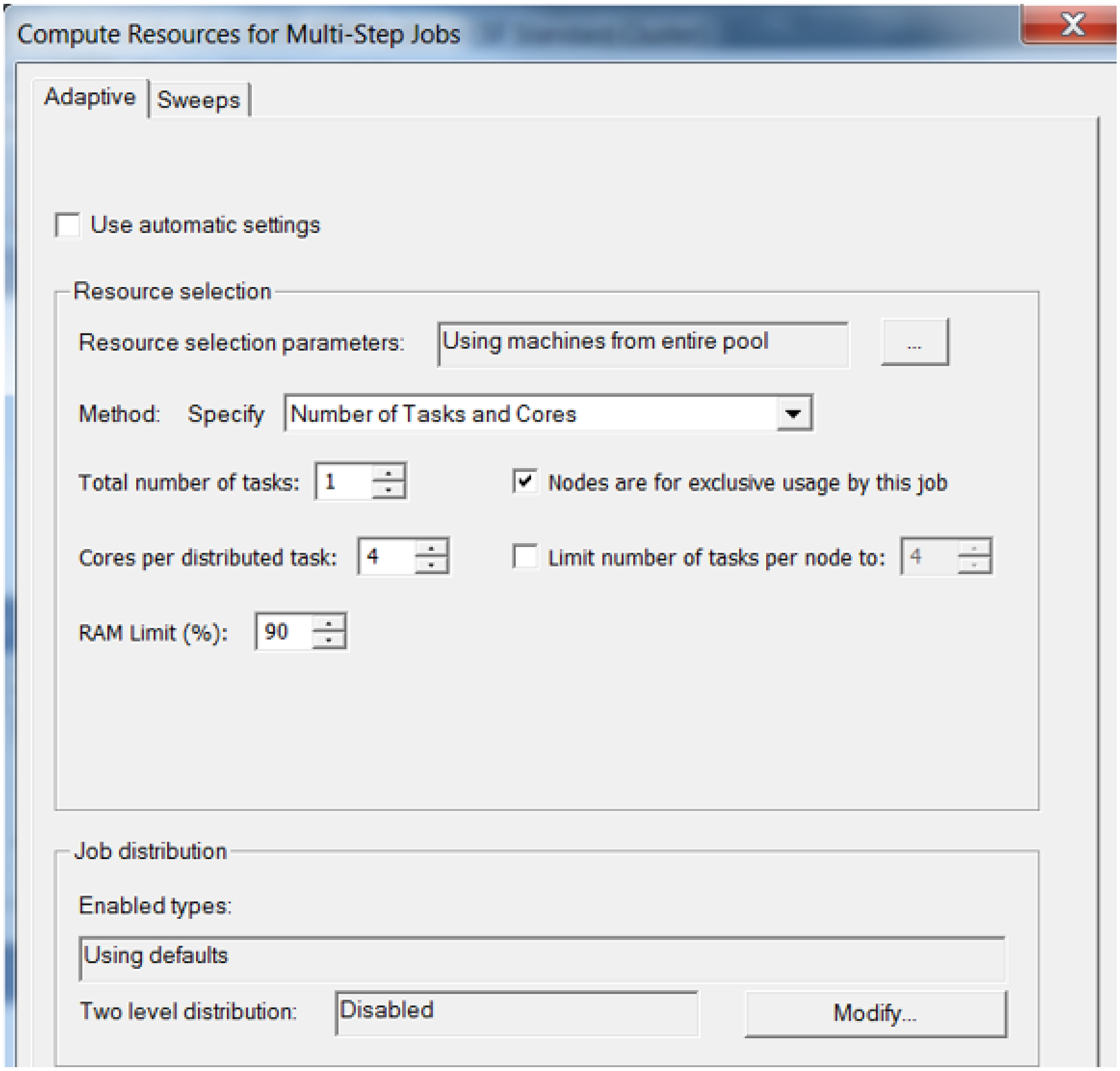
If the Multi-Step button and check box are not enabled it could be because you have not selected a single setup, the design type of the setup does not support Multi-Step, or the scheduler type (e.g., RSM) does not support it.



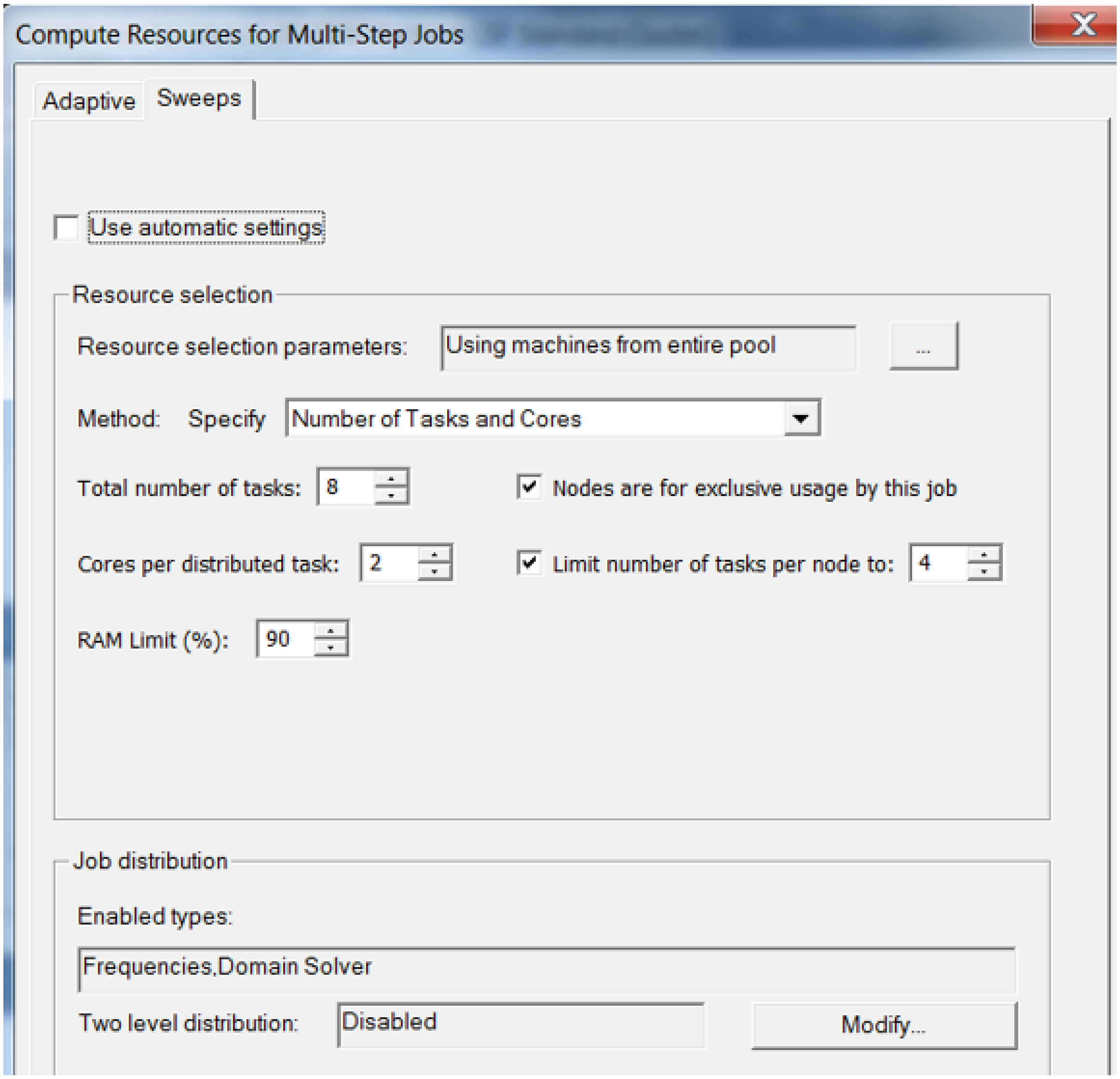
On the **Compute Resources** tab, check the **Use multi-step submission** box and select the appropriate submission option, that is:

- Two Steps for Adaptive and Sweeps
- Three Steps for Mesh, Adaptive, and Sweeps

Then click **Multi-Step…**, which will bring up a **Compute Resources for Multi-Step Jobs** dialog box where there is a tab for each step used, that is for potential for Mesh, Adaptive, and Sweeps.

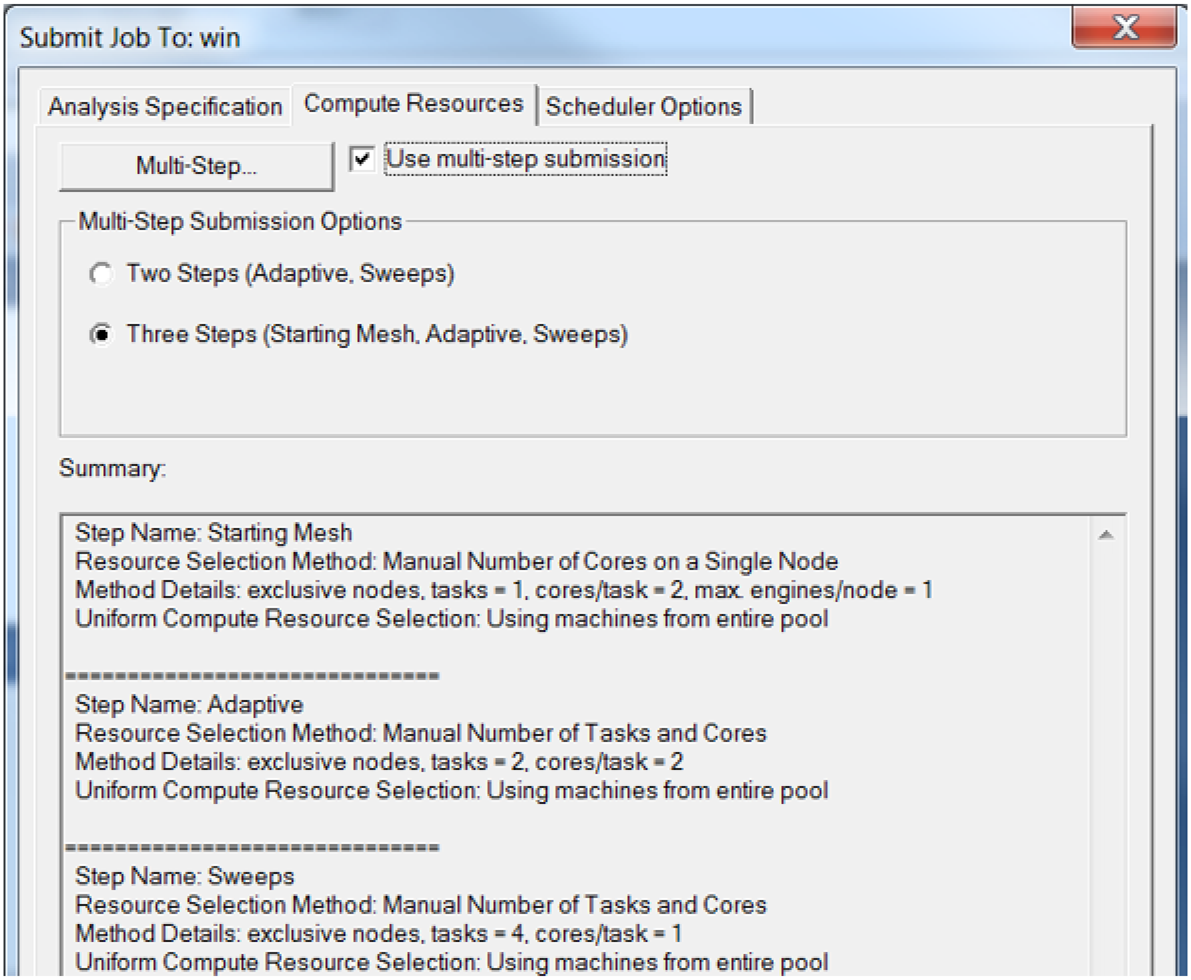


Notice, for example, that the **Sweeps** tab lists the same resources choices but they can be assigned differently.

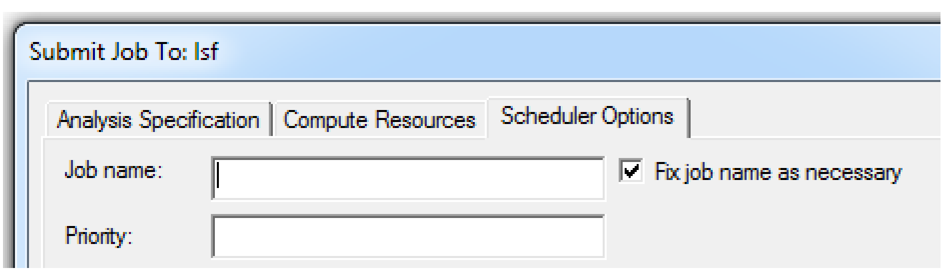


Set the desired compute resources for each step and click **OK**.

The Summary field of the **Compute Resources** tab shows a text summary of resource specifications used for each step.

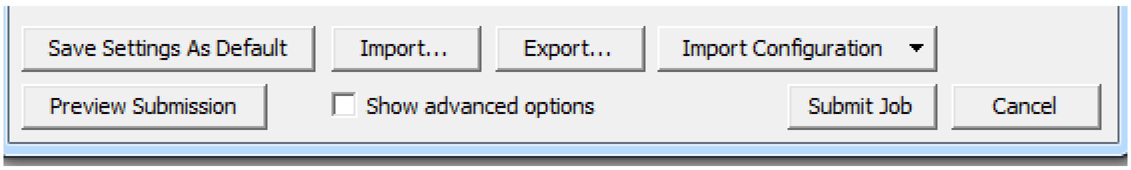


If desired, set the job name on the **Scheduler Options** tab. When the jobs are submitted, each will have a name given by job name (if any) appended with the name of the step for the job. For example, if the job name entered is “MultiStep”, then the individual jobs might be named “MultiStep-Adaptive” and “MultiStep-Sweeps”.



There is a new **Fix job name as necessary** check box for the job name. This applies only to certain schedulers where there are constraints on the job name. It is useful to ensure proper submission in the case where appending the step name results in an invalid job name.

When you click **Preview Submission**, you see a preview for each step, all in the same output window. Any errors or warnings for a step, generated during preview validation, are displayed with the text for the preview of that step.



When you click **Submit Job**, each job is submitted individually, and dependencies are set with the scheduler so that subsequent jobs wait for the prior step's job to complete before starting. Any errors in job submission for any step prevent further steps from being submitted.

The dialog reporting successful submission has been modified slightly for multi-step jobs. It will show the jobs IDs of all jobs that are part of the multi-step job sequence.

Multi-Step Job Monitoring

You can monitor the job step in progress. When one job completes, the status shows "Completed", but monitoring restarts once the next job step is running.

Aborting a Job Step

You can also abort the job step in progress. With multi-step jobs submitted from the Electronics Desktop GUI, this will also cause remaining jobs (which would otherwise remain queued in the scheduler) to be canceled.

Archive Projects for Multi-Step Job Submission

Note that you can submit archive projects. Monitoring is based on the archive for the first step, then on the extracted project for subsequent steps.

Submitting a Job without Opening the Project

You can also submit a job without opening the project. This can be done by choosing **Tools** > **Job Management** > **Submit Job…** and then manually entering the project path (You can also use **Browse** can to select the project.) Note that you must also select a single nominal setup before the **Use multi-step submission** check box is enabled on the **Compute Resources** tab.

Windows to Linux Job Submission

Given a set of prerequisites, Ansys Electronics Desktop can permit Windows to Linux job submission as part of HPC.

Prerequisites for Job Submission

Directory Shared between Windows and Linux

For all jobs submitted to a Linux cluster, the project file is required to be in a directory that is accessible from all execution hosts used by the job. For submission of jobs from a Windows host to a Linux cluster, the project file must also be accessible from the Windows host where the GUI runs. There must be a directory shared with both Windows and Linux hosts, and the project file may be in a subdirectory (at any level) of the shared directory.

Network Access from Windows Host to Linux Job Management Host

The job is submitted to the cluster from a Linux host configured for submission of jobs to the Linux cluster. We call this Linux host the “Job Management” host. The information about the job to be submitted is transmitted to the Job Management host over the network. As a result, the Windows host where the GUI runs must have network access to the Job Management host. If this communication is blocked, then job submission from a Windows host to the Linux cluster will not be possible. Communication could be blocked if there is a firewall or if the Linux cluster is only on a private network, for example.

Ansoft 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.

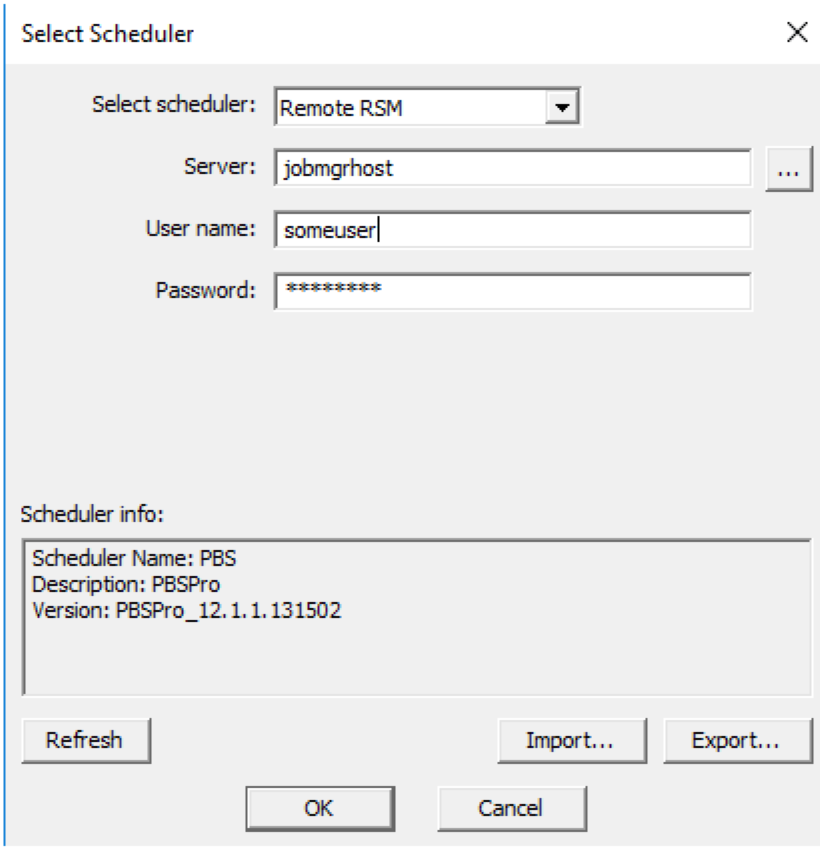
Supported Schedulers

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

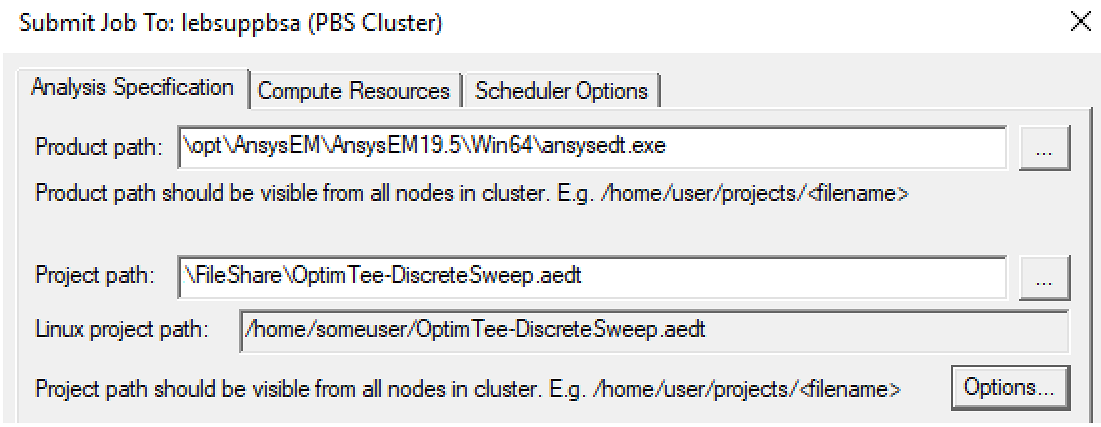
If you select the **Use a computer on the network** option, you can enter a username and password. This username and password are used when the job is submitted to the Linux scheduler.



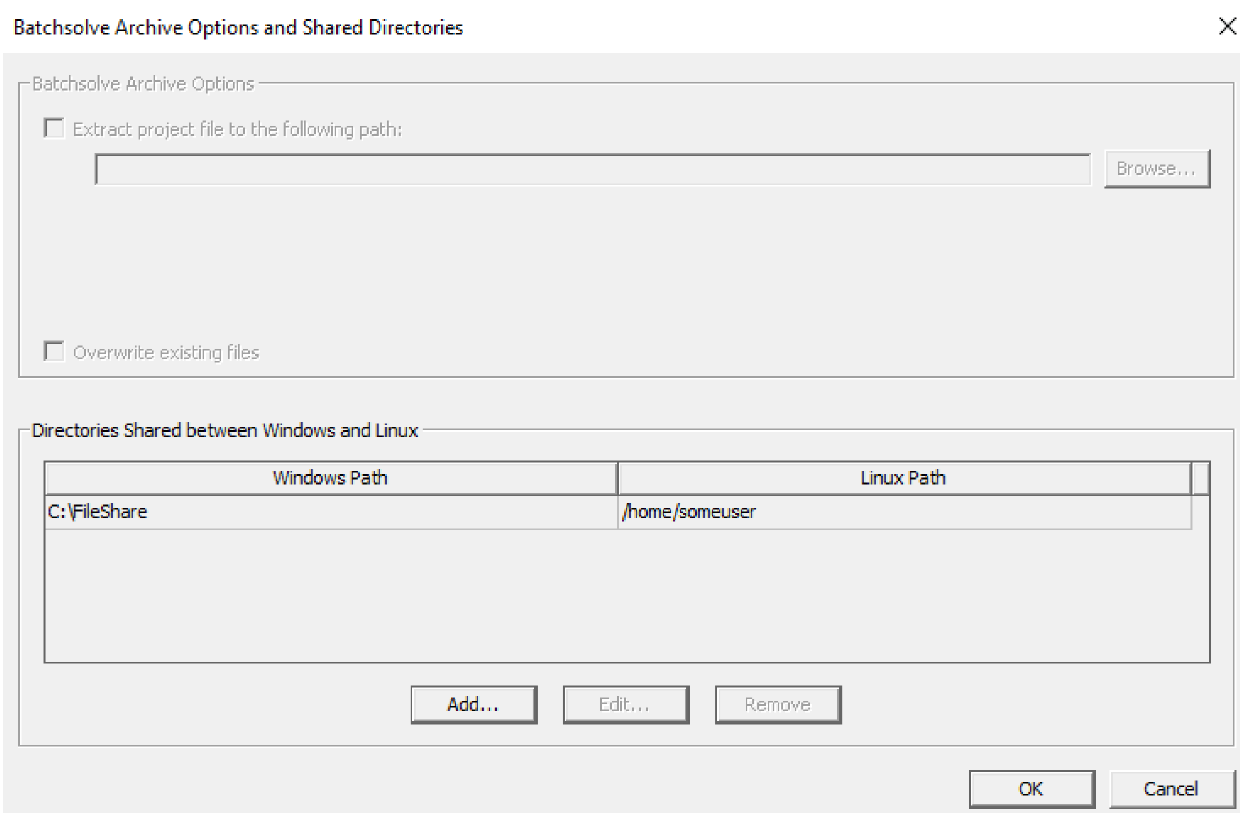
Submit Job

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.

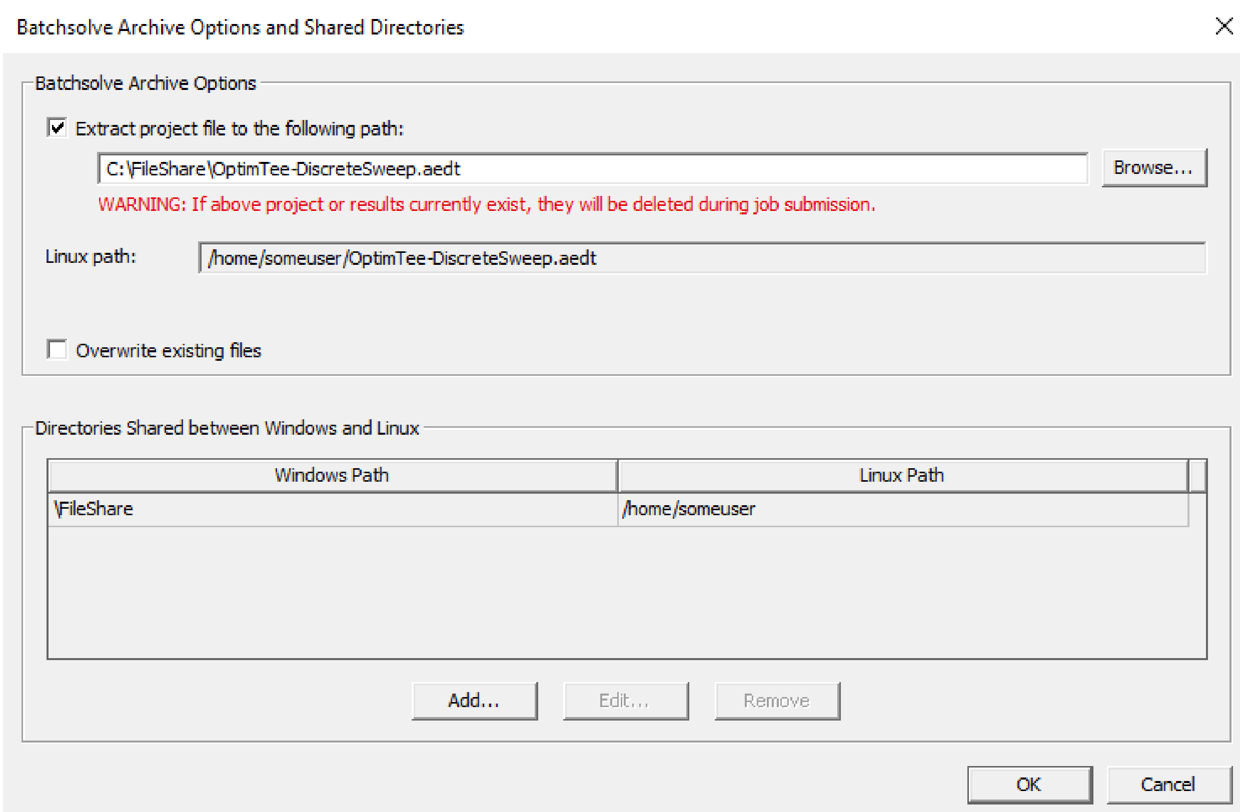
You must enter the Windows path of the project file in the Product path edit control or use the browse button (labeled "...") to select the Windows path of the project file. If the Linux path of the project file can be determined from the specified Windows path and the directories shared between Windows 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.



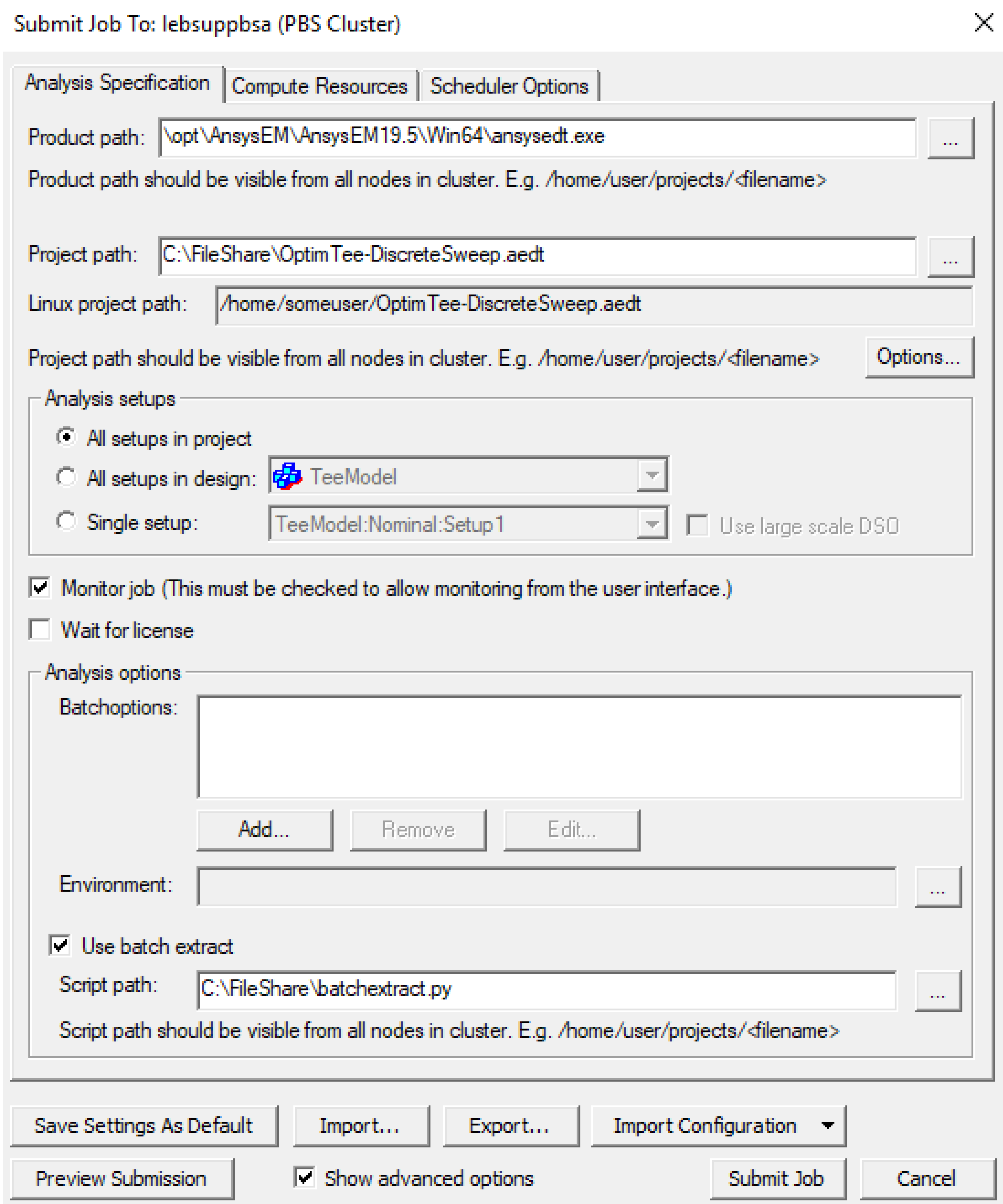
The **Options** button opens a window that you can use to specify archive options for a job. The lower portion of this window allows you to specify one or more directories shared between Windows and Linux. The mapping of directories between Windows and Linux is shown in a grid which displays the Windows 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 dialog.



If you specify a batchextract script, the Linux path of the batchextract script is determined from the Windows path of the batchextract script and the directories shared between Windows and Linux.



User Passwords are Encrypted

Jobs are submitted to the Linux cluster using the user name and password entered in the **Select Scheduler** 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.

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* dialog box appears. This is the only part of job submission scripting that may required user intervention. This same mechanism is used if, from within the SubmitJob command, there is failure to connect to the scheduler. Even though there are allowances for graphical user intervention if something goes wrong, if the password (if required) is cached and all settings are correct, the entire submission process can run non-graphically and fully automated.

Limitations

All settings besides the arguments passed to the **SubmitJob** command must be stored in the .areg file containing settings exported from the 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 re-submitted from within a single script, the job should be monitored (waiting for completion) before trying to submit it again. This monitoring/waiting can be done with a combination of a single **LaunchJobMonitor** command followed by a loop that checks the result of a **RefreshJobMonitor** command.

How to 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, since the backslash is an escape character. When the script is run, and all is successful, there should be a message in the message windows stating that the job was submitted, including the job ID(s). There could be multiple job IDs if multi-step submission is used.

See the Scripting help (click **Help** > **CircuitScripting 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 Jobs
- Monitoring Large Scale DSO Jobs
- Web Client for Batch solve Monitoring and Reporting

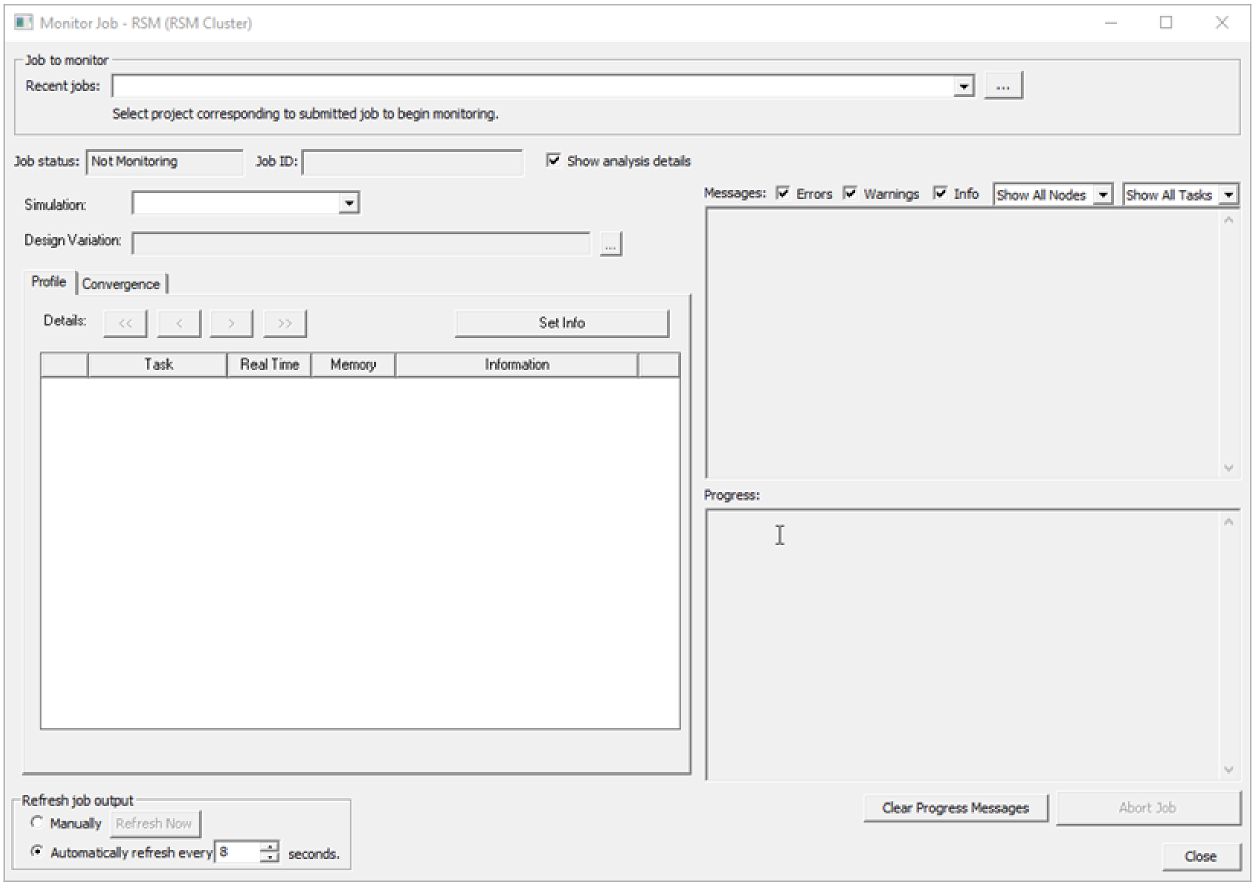
Monitor Job Window

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:

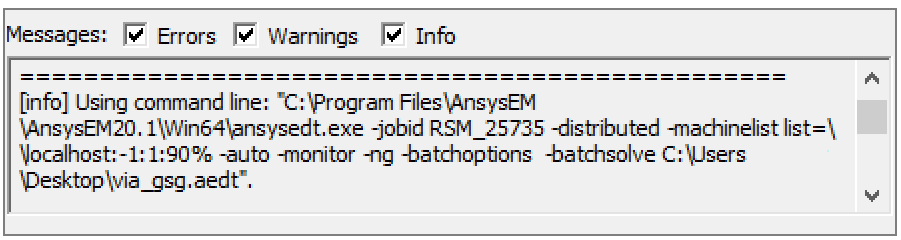
The window may look slightly different, depending on your selected scheduler and design type.

The **Monitor Job** window contains the following areas:

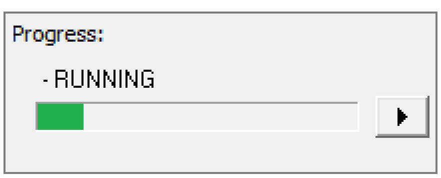
- **Job to Monitor** – Allows you to select either a recent job, or a project containing the job you want to monitor. You can select an archive.
- **Job Status** – The current job status. The normal progression is: Starting Monitoring, Queued, Running, Shutting Down, Completed. A status of Unknown may 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.

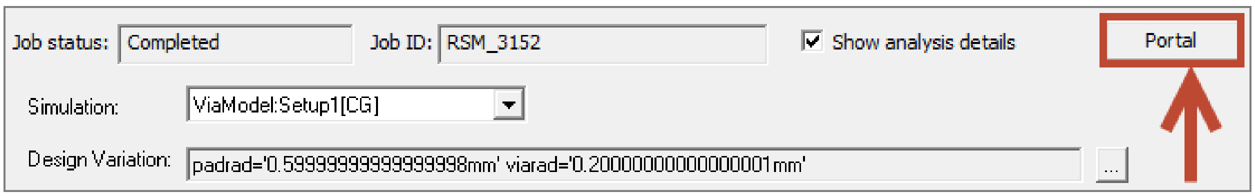
To display less information, deselect **Show analysis details**. This removes the **Simulation**, **Design Variation**, **Profile**, and **Convergence** information, instead displaying only **Messages** and **Progress**.

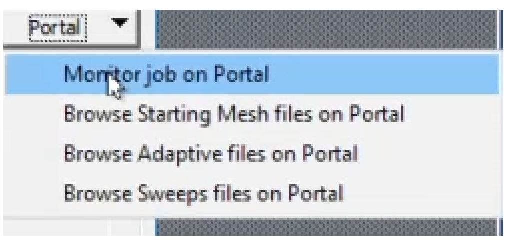
1. Click **Close** when you are done monitoring.

Monitoring Ansys Cloud Direct Jobs

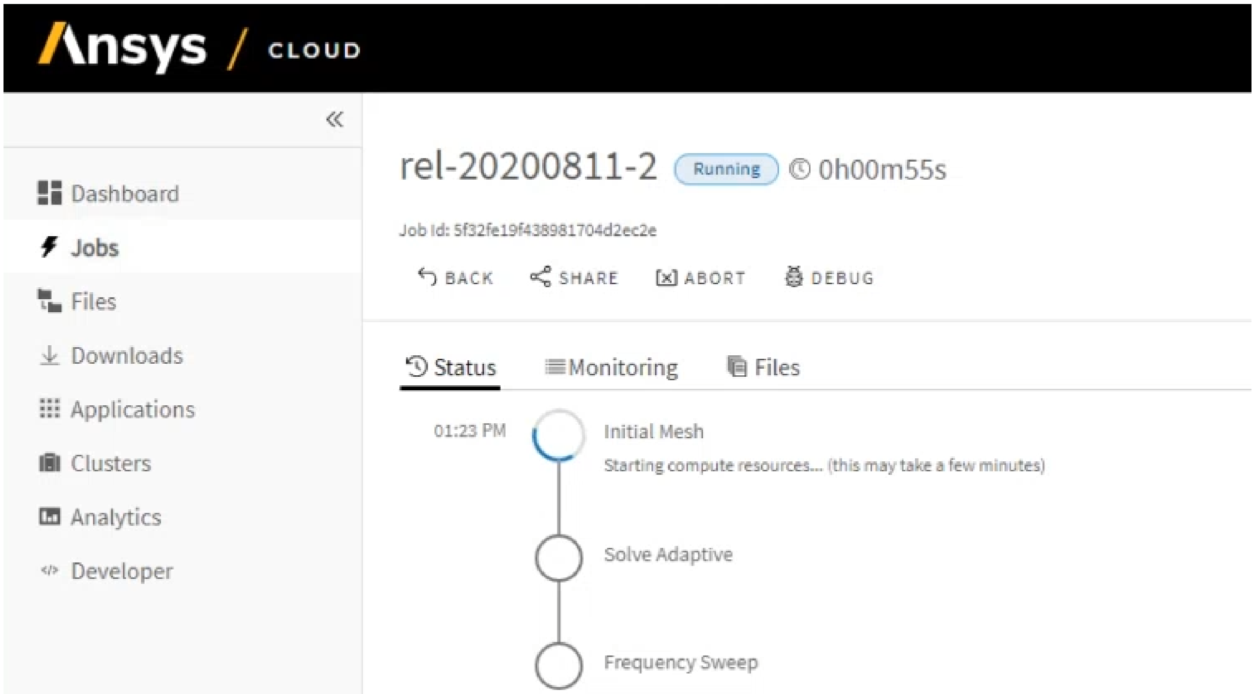
For Ansys Cloud Direct submissions, the **Monitor Job** window includes two additional items:

- **Portal** – link to the Ansys Cloud Direct Portal, where job details and additional monitoring information are available.

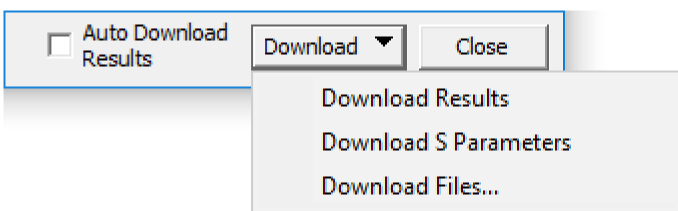




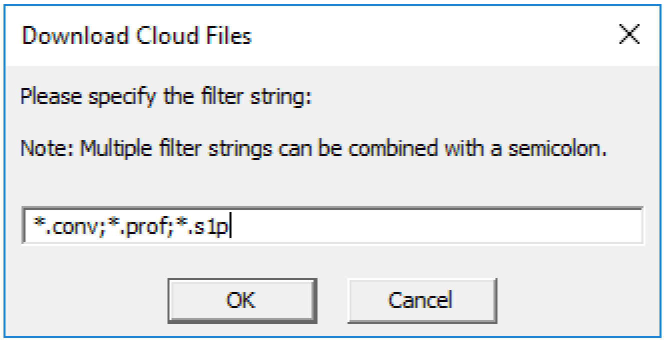
Monitor job on Portal Opens a Portal Web interface that lets you monitor an Ansys Cloud Direct job. The additional menu options for **Portal** are for multi-step submissions, opening the Portal Web interface directly to folders with Starting Mesh Files, Adaptive Files, or Sweeps files.



- **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:



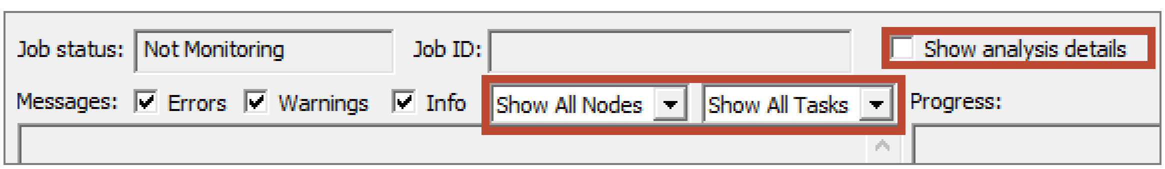
Specify the types of files you want to download (separated by semicolons) and click **OK**.

Monitoring 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.

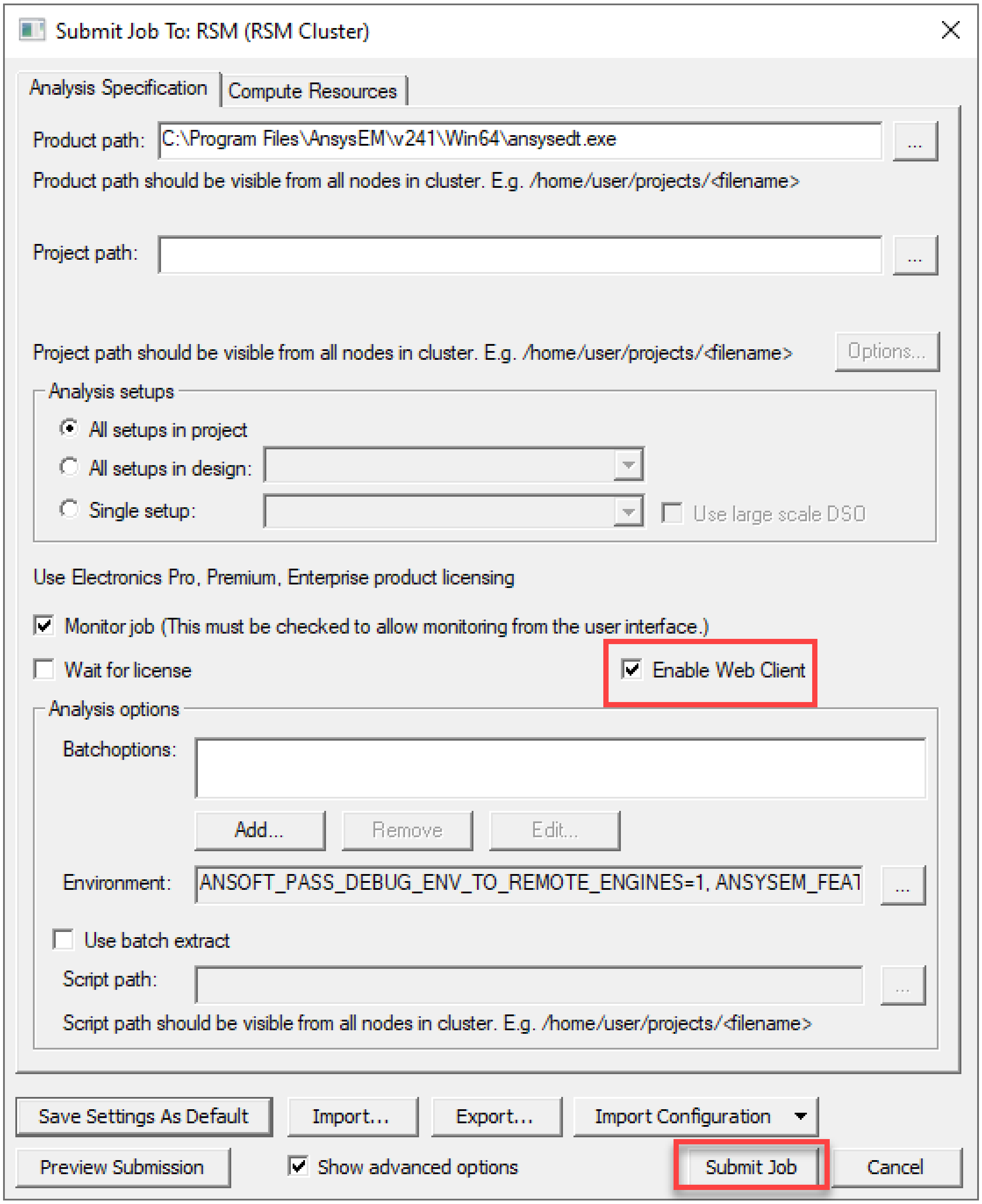


To leave this mode, select **Show analysis details**.

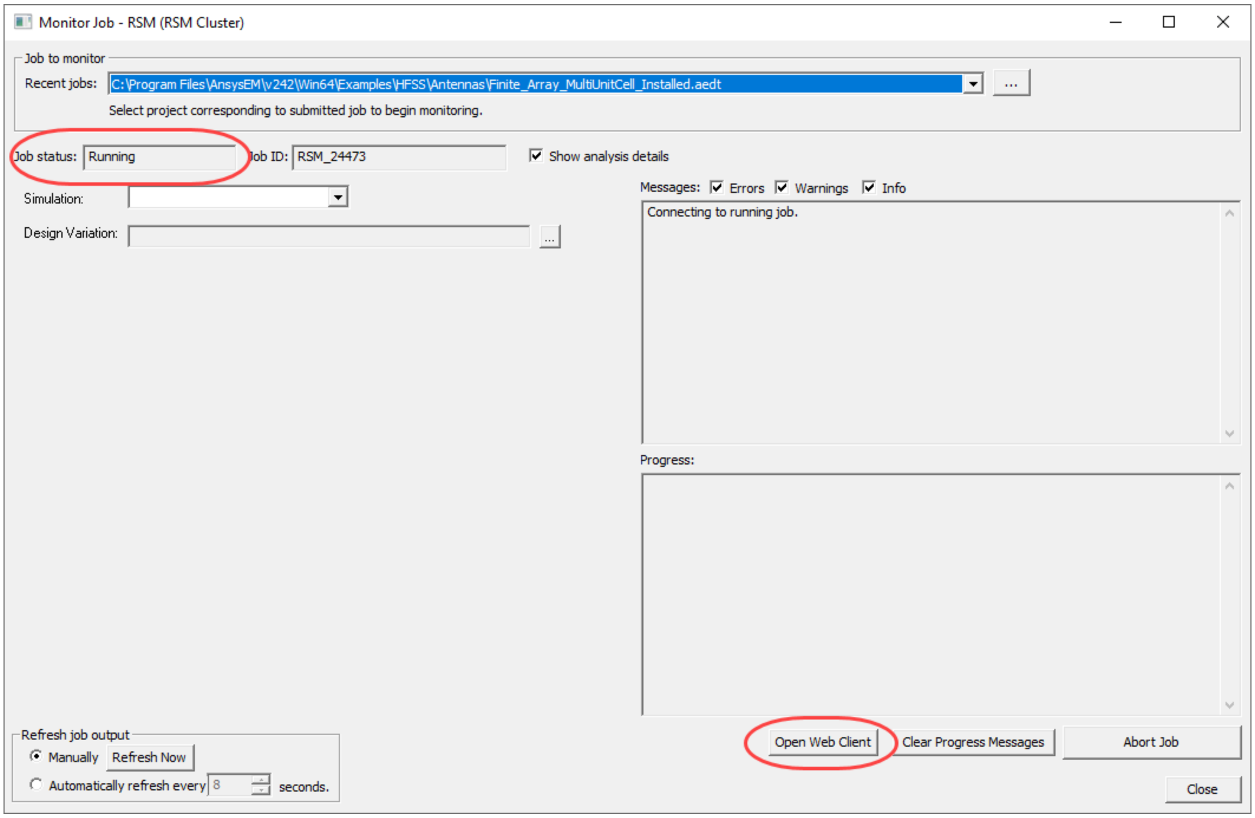
Web Client for Batch solve Monitoring and Reporting

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.

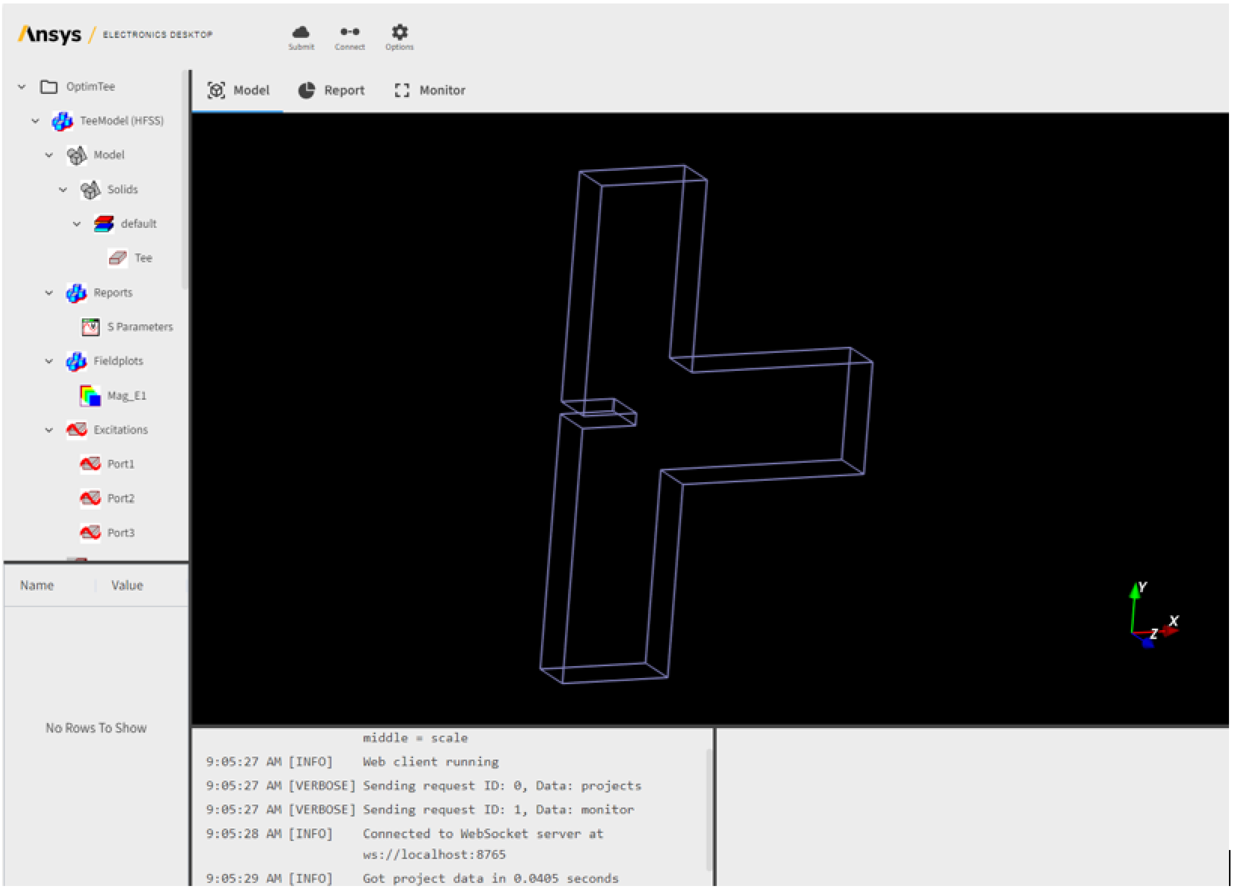
After you have selected the scheduler to use, when you access the **Submit Job To:** dialog, you well see Enable Web Client checked.



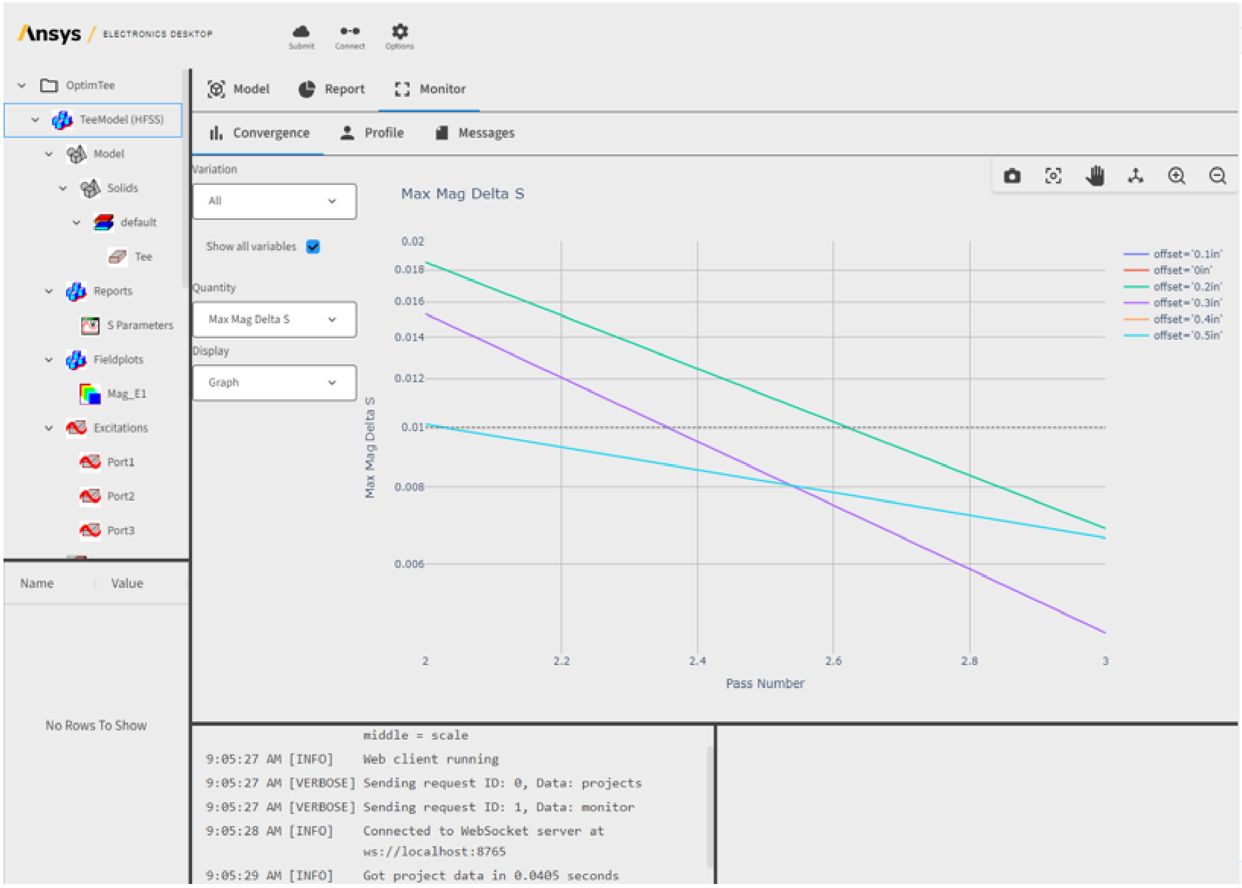
After specifying the project path and setting other parameters, you can then select **Submit Job**. This opens the Monitor Job dialog for your scheduler. Once the **Job Status** changes to Running, press **Open Web Client**.



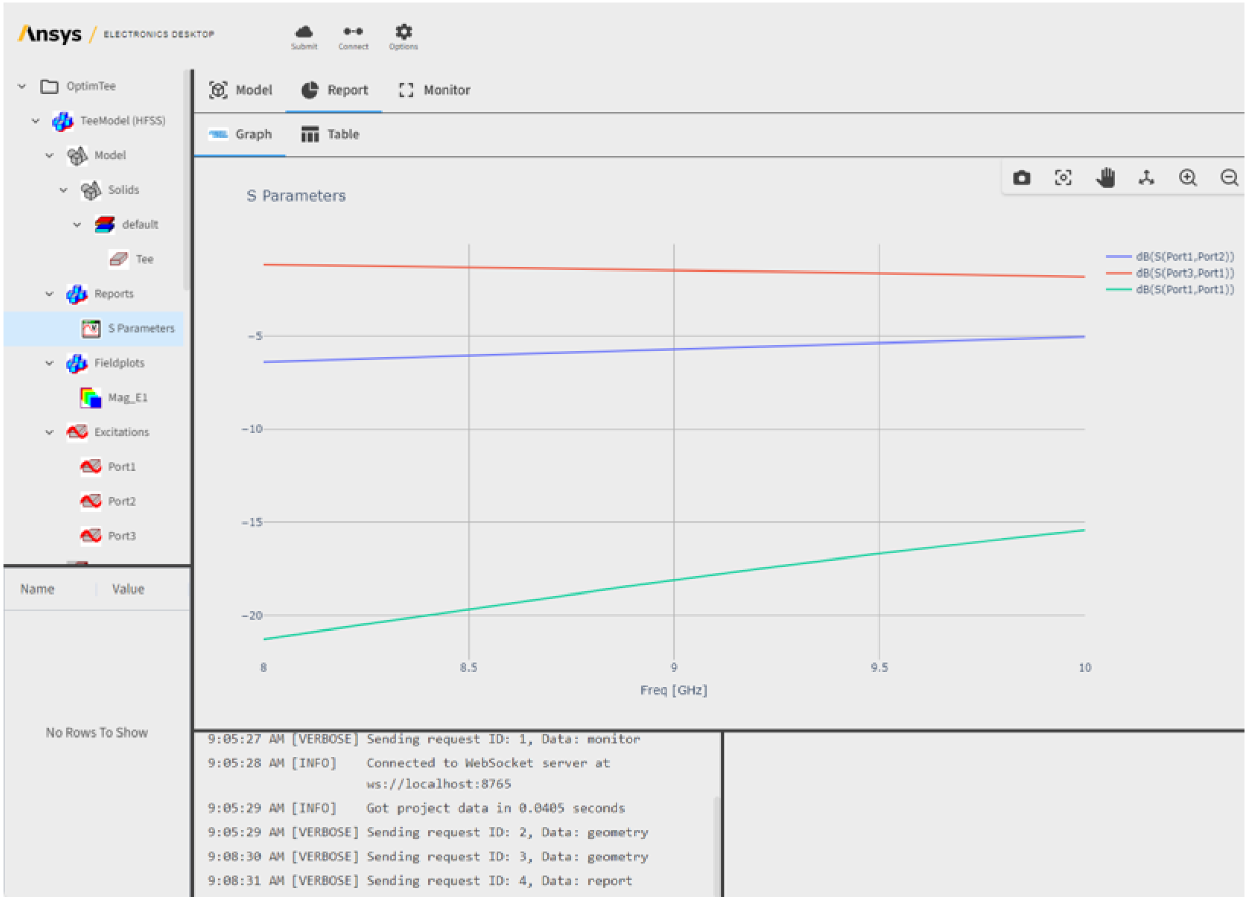
Click **Open Web Client** to open the Web Client in your default browser.



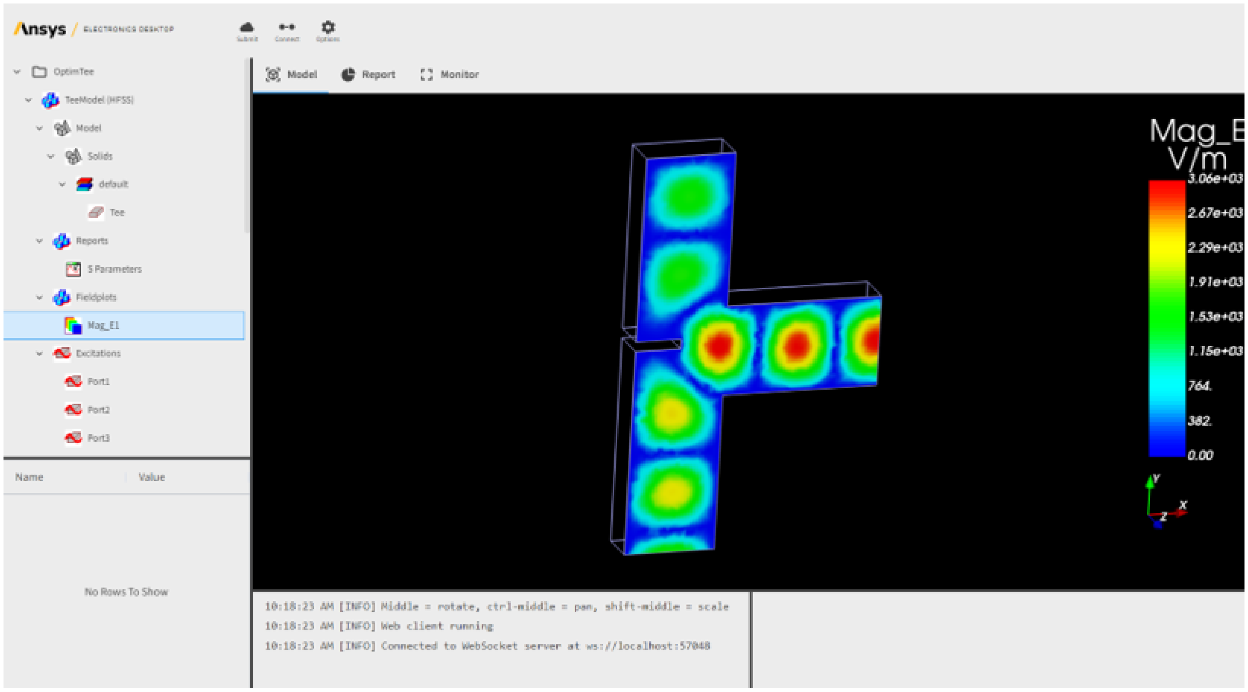
Switch to the **Monitor** tab to gain access to the subtabs for Convergence, Profile and Messages.



Select a Report in the **Project Tree** to view it.



Select a Field Plot in the **Project Tree** to view it.



Integration with Ansys Remote Simulation Management (RSM)

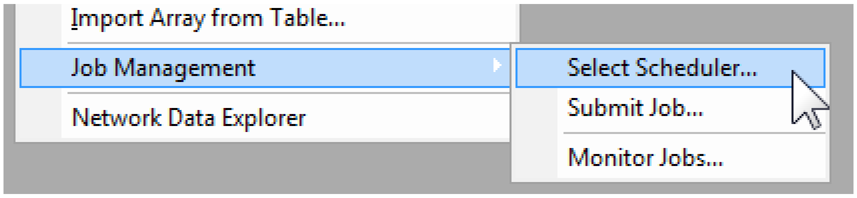
Ansys Electromagnetics supports its own Remote Simulation Management (RSM) software along with other High Performance Computing (HPC) software management programs (see High Performance Computing (HPC) Integration).

When do you need RSM?

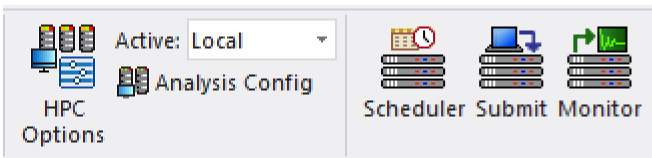
RSM is generally required if you want to run remote or distributed simulations. However, if you have a separate scheduling system that Ansys Electromagnetics supports, and you plan to run batchsolve simulations only, you may not need to install RSM. For details of installation and configuration of RSM, see the *Ansys Electromagnetics Installation Guides*.

Job Management UI for RSM

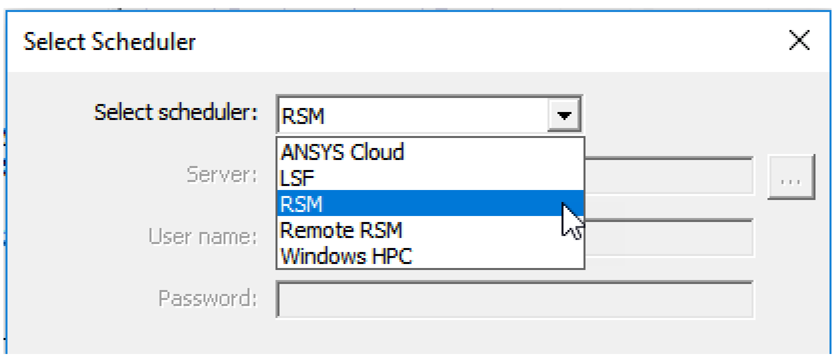
You can use the Job Management UI to submit batch jobs to RSM. The Job Management UI is accessed by running Ansys Electronics Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access 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.

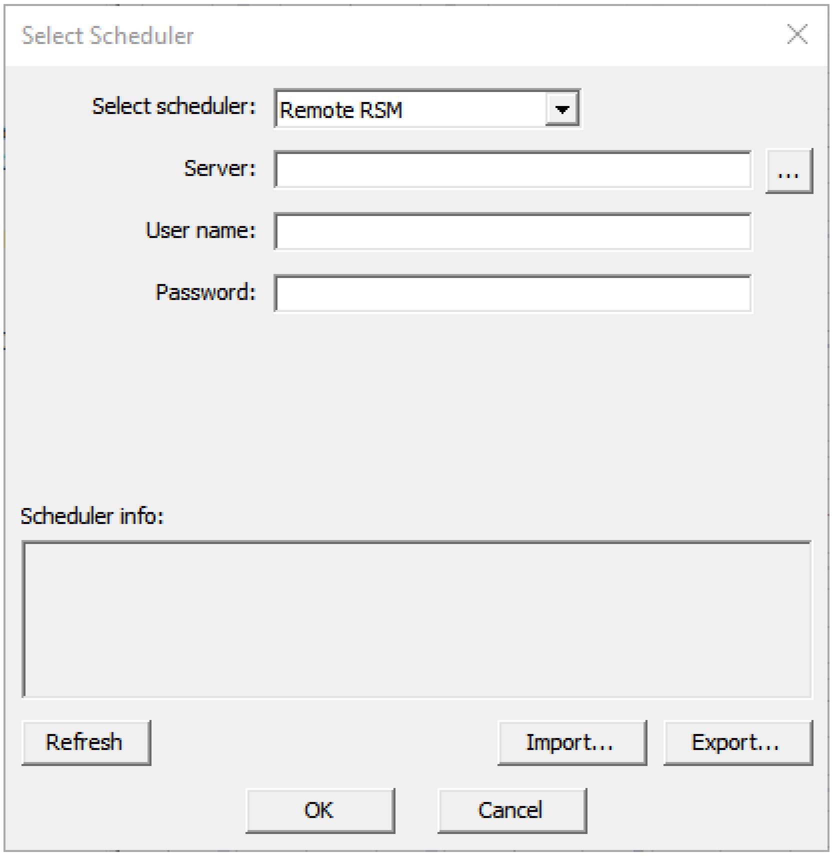


This displays the selection dialog box. The **Select scheduler** drop-down menu lists potential schedulers (which can include RSM, LSF, Windows HPC, or SGE, depending on the environment).

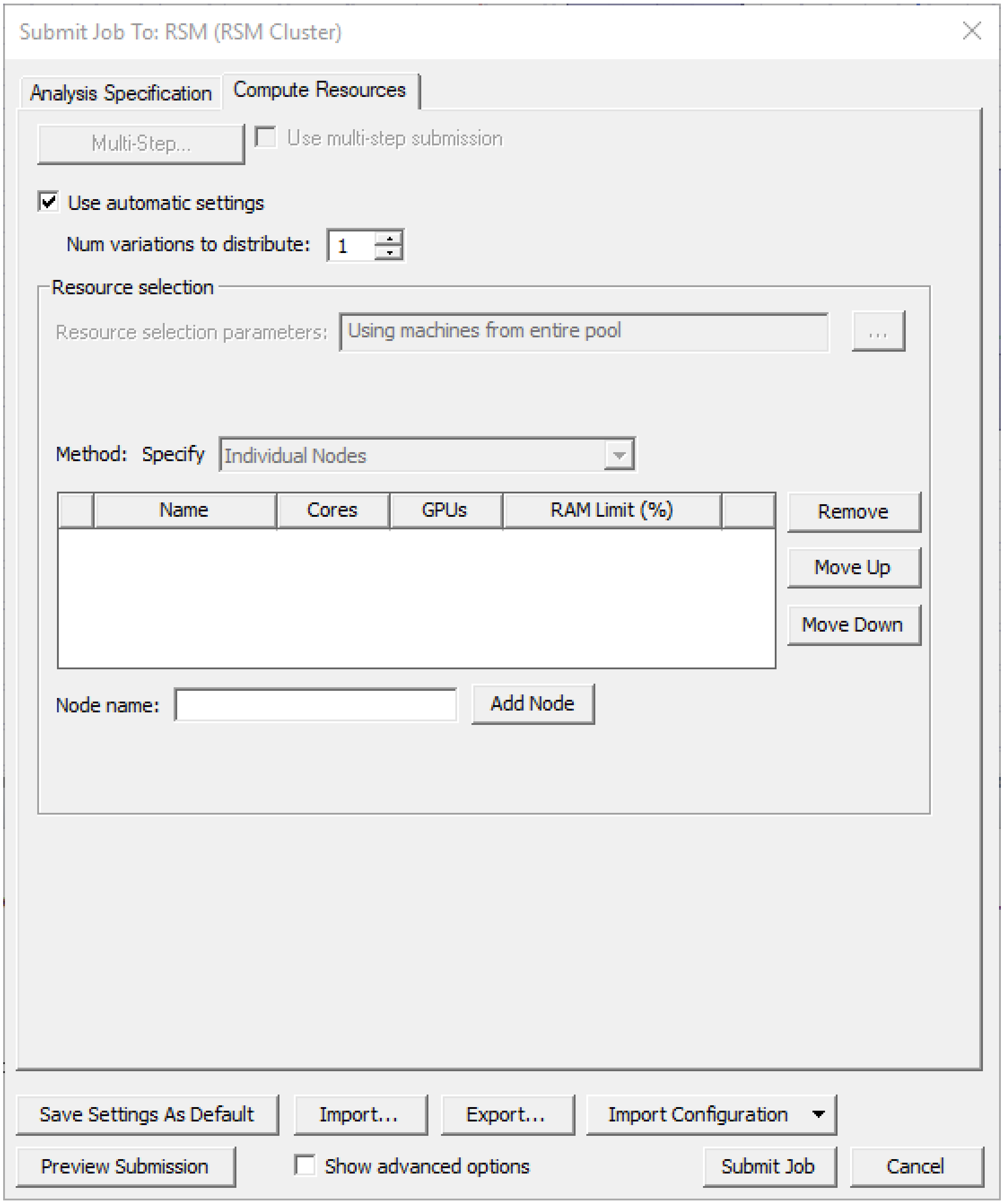


If you select a scheduler that is not supported in your environment, you receive a warning.

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).

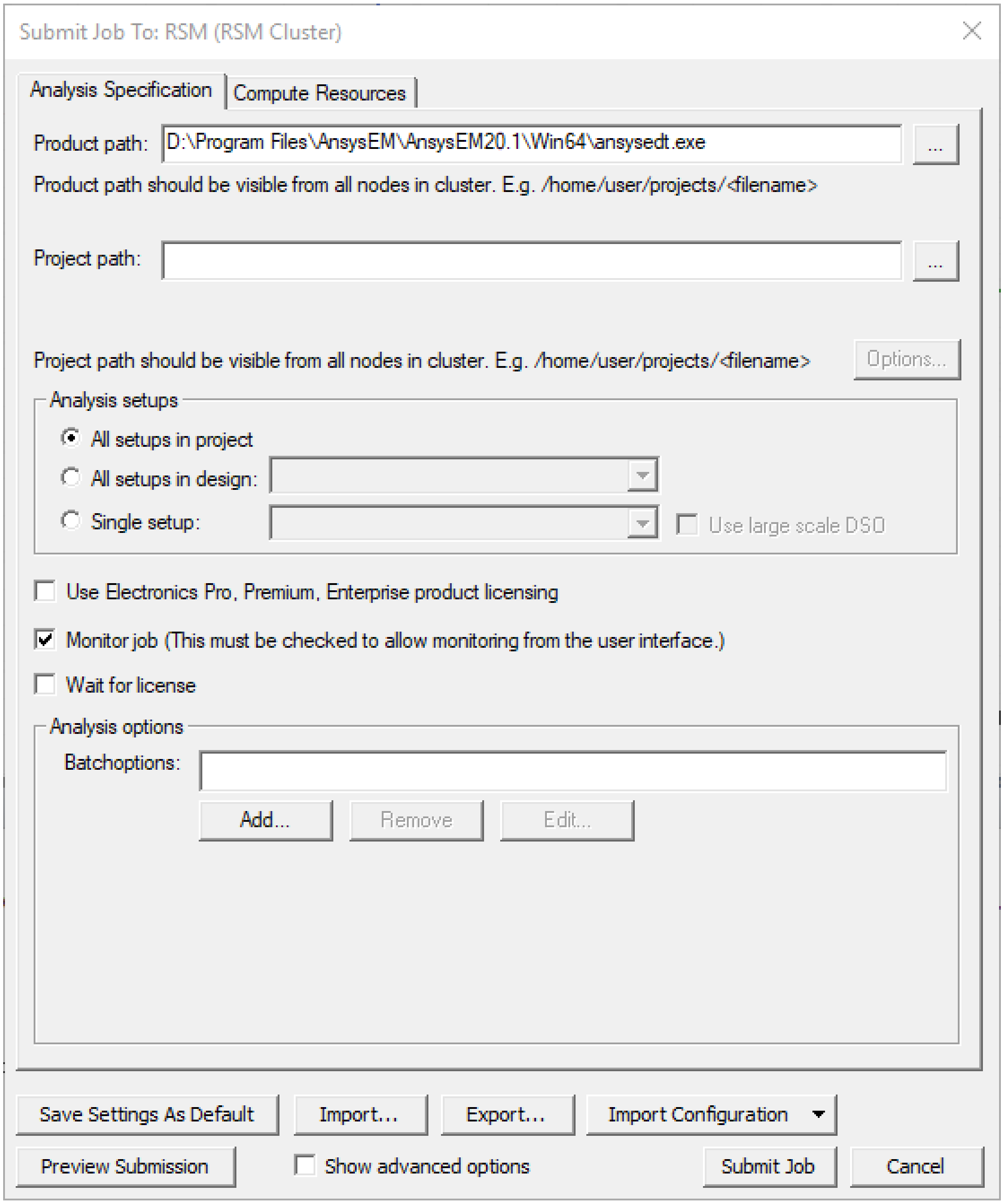
- In the RSM environment, you are required to specify a machine-list. See: HPC and Analysis Options. For example, say the machine-list is: 3 cores from 'm1' and 3 cores from 'm2', for a total of 6 engines. You select the list on the **Compute Resources** tab of the **Submit Job to RSM** window, as described 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:

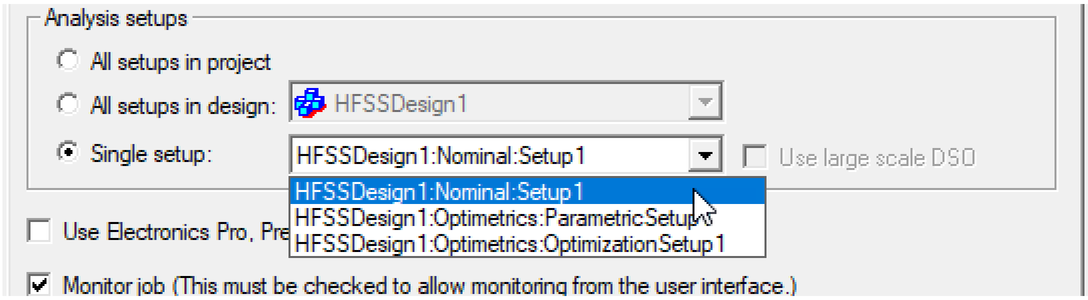
   - **Analysis Specification**– specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables. If you select the Analysis or Optimetrics setup, the Analysis Specification is pre-populated.
   - **Compute Resources** – specify 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

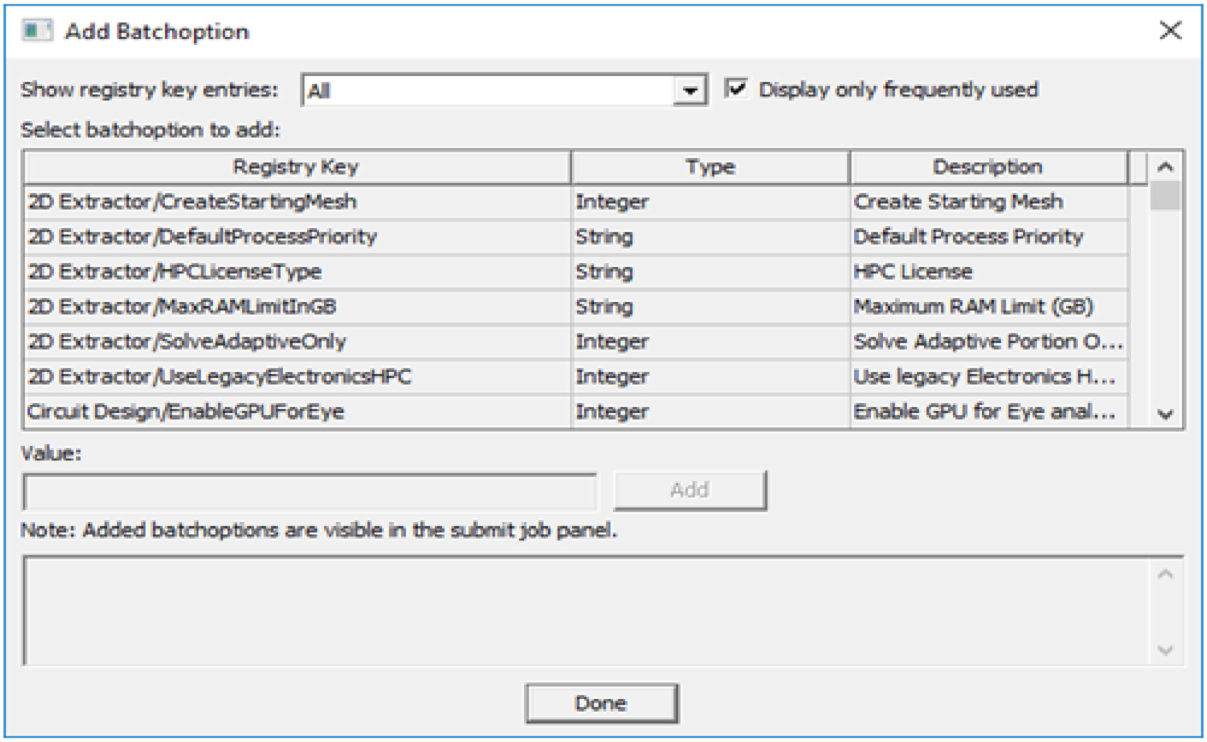
pre-populated.



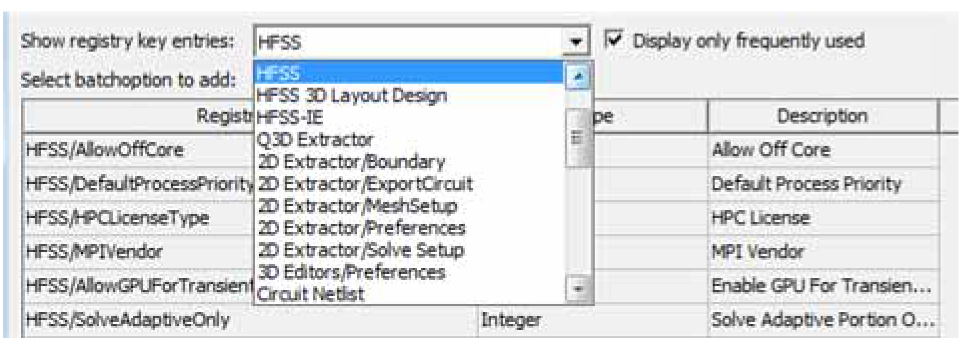
For Parametric setups, you have the option to select **Use Large Scale DSO**. For details on how and when you use this feature, see Job Management Interface for Large Scale DSO.

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.

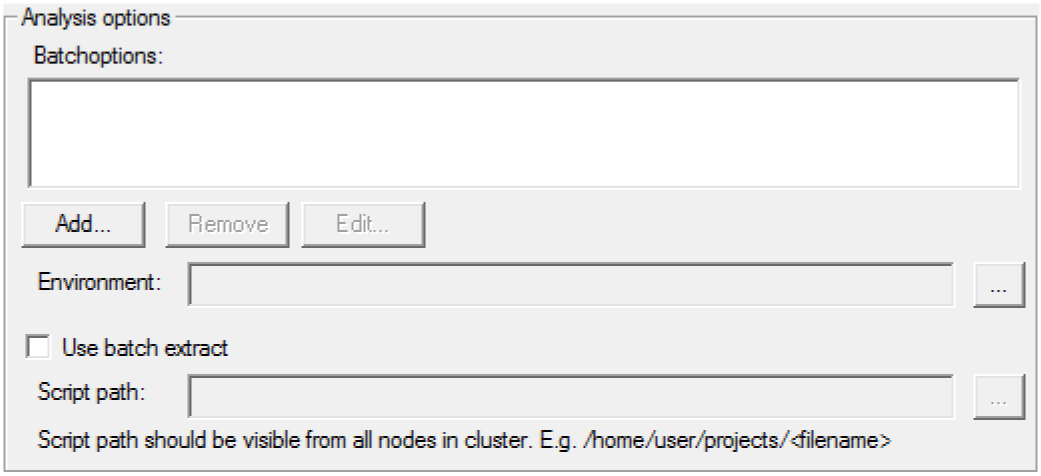


The **Show registry key entries** field lets you filter the entries displayed, by means of drop-down menu selection, and a check box to **Display only frequently used entries**.

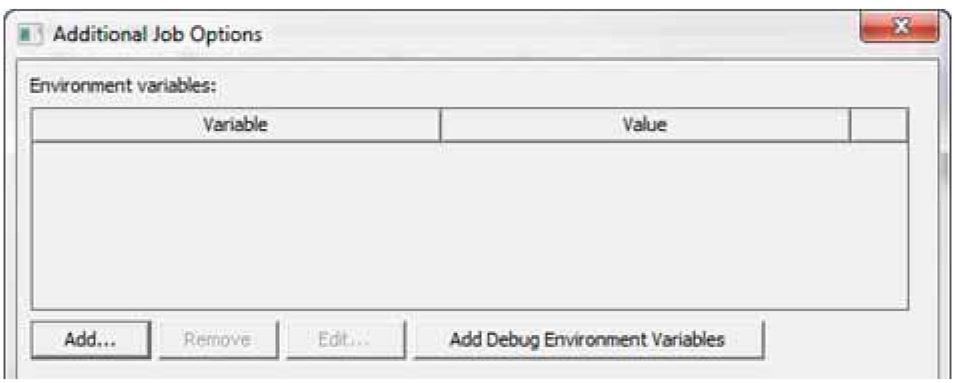


- When you have selected a batchoption, you can type the value in the field, and click the **Add** button to add the option to the batchcommand.
- In the **Submit Job To:** window, you can enable **Show advanced options** to display

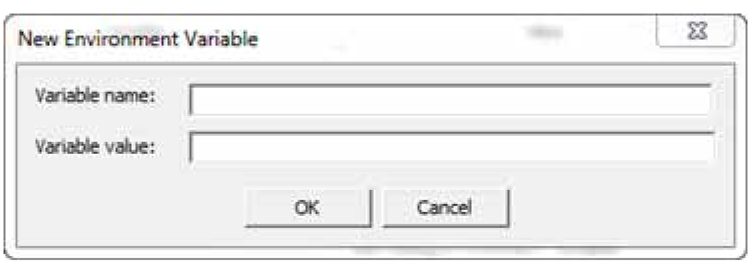
additional fields for environment variables, and whether to **Use batch extract**.



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.

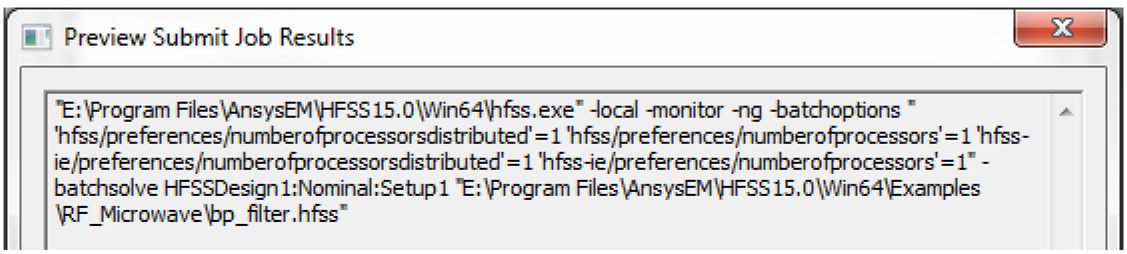


Clicking the **Add Debug Environment Variables** button automatically adds a set of debug variables. This can be useful in working with Ansys Application Engineering support.

Variable	Value
ANSOFT_DEBUG_MODE	2
ANSOFT_DEBUG_LOG_SEPARATE	1
ANSOFT_DEBUG_LOG	$PROJECTFILEDIR\debug_log\log
ANSOFT_PASS_DEBUG_ENV_TO_REMOTE_EN...	1

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.

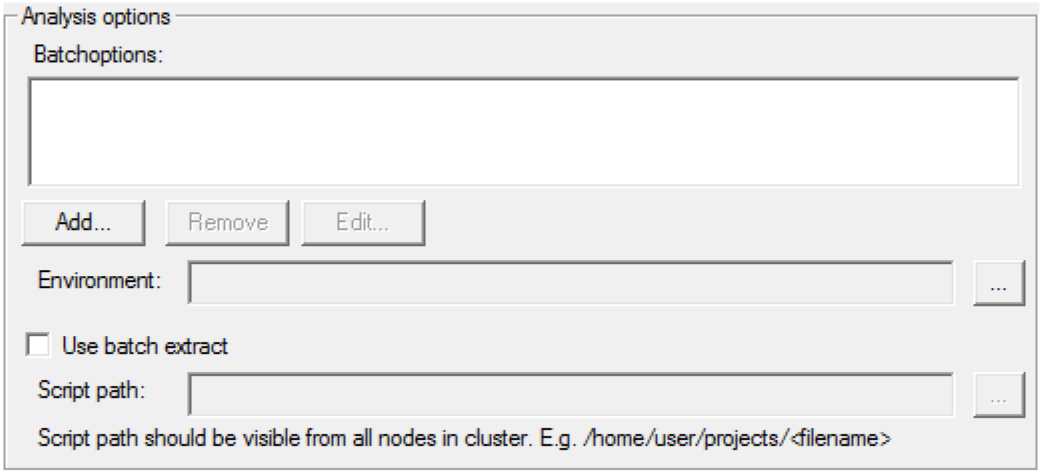
8. To see the command-line to be submitted to the scheduler, click **Preview Submission.** This opens a dialog box showing the command to be sent to the scheduler.



The text can be copied to the clipboard, if desired.

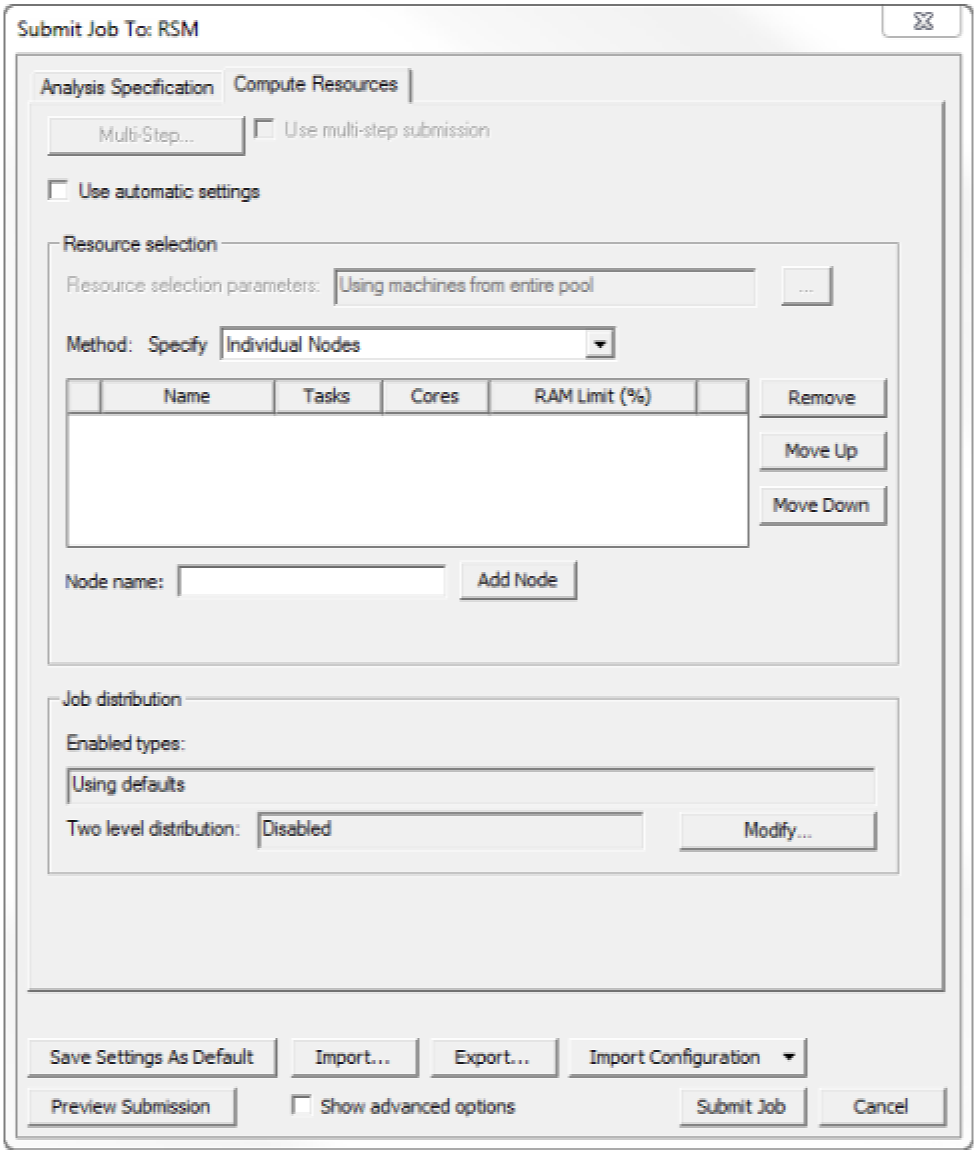
Use Batch Extract for RSM

Selecting **Show advanced options for RSM** also shows the Use batch extract fields.

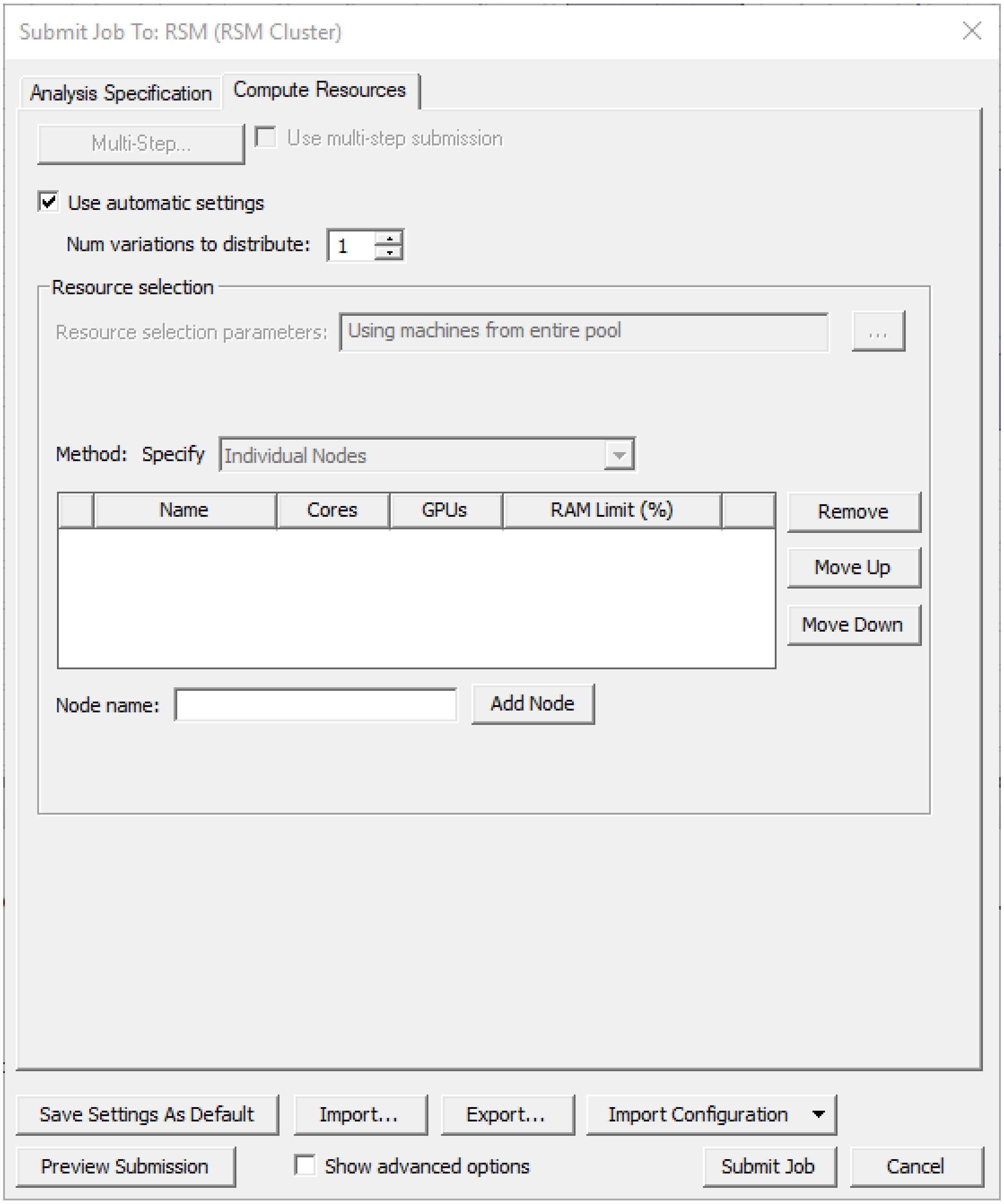


See the discussion on Running Ansys Electronics Desktop from a Command line for a discussion of the solve information available through batch extract.

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.

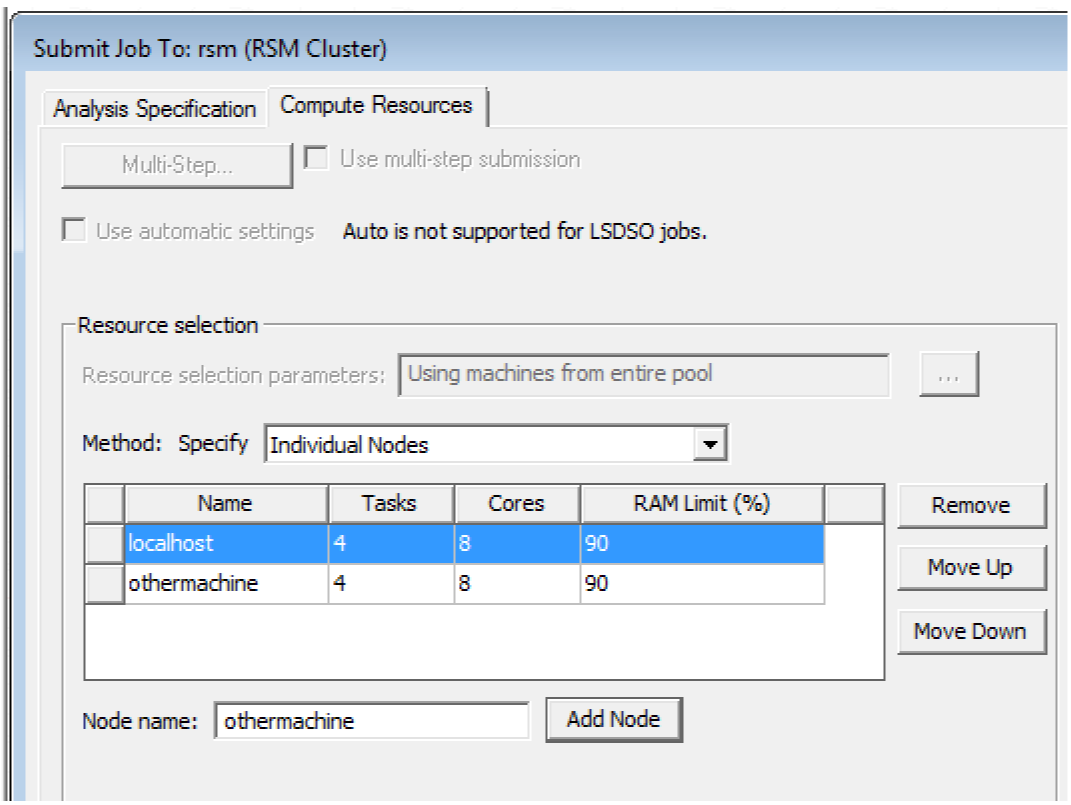

Submit Job To: RSM (RSM Cluster)
Analysis Specification
Compute Resources
Multi-Step...
Use multi-step submission
Use automatic settings
Num variations to distribute:
1
Resource selection
Resource selection parameters:
Using machines from entire pool
...
Method:
Specify
Individual Nodes
Name
Cores
GPUs
RAM Limit (%)
Remove
Move Up
Move Down
Node name:
Add Node
Save Settings As Default
Import...
Export...
Import Configuration
Preview Submission
Show advanced options
Submit Job
Cancel

> **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

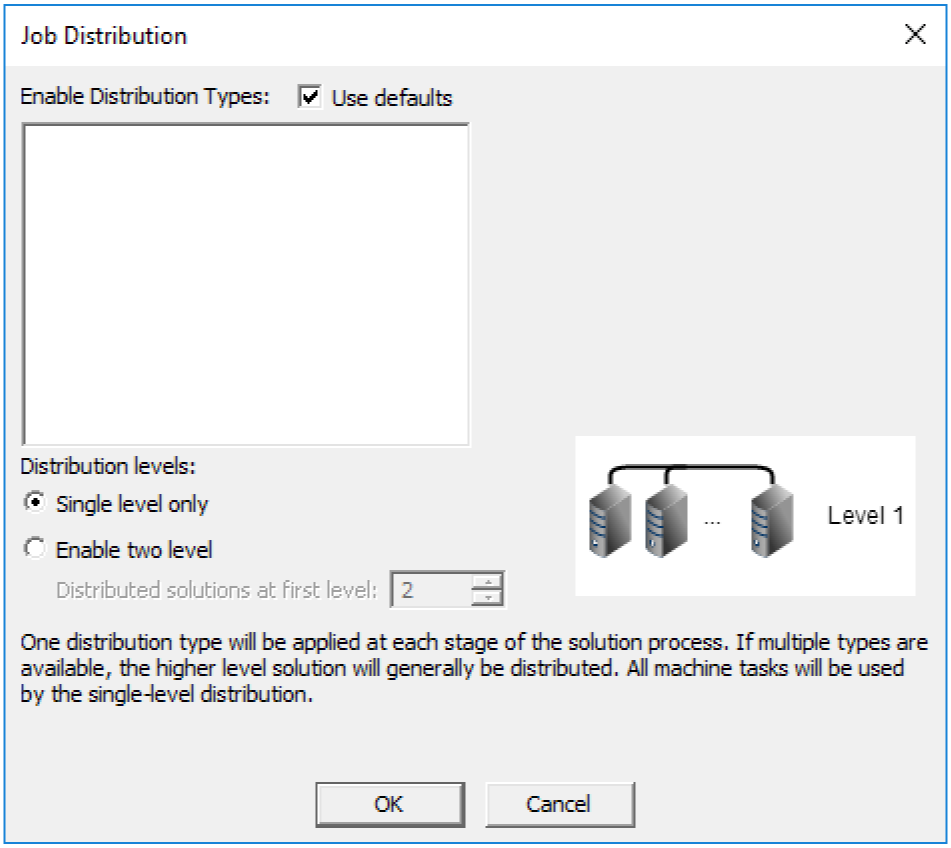
  Here you can specify a node list. In a computing environment where the available cores are not uniform, you can use this to control which resources your job will use. For use with Large Scale DSO for RSM, for jobs that are submitted from job submission panel, localhost must be the first node in the resource selection node list, other wise LSDSO solve with RSM will fail.



Job Distribution

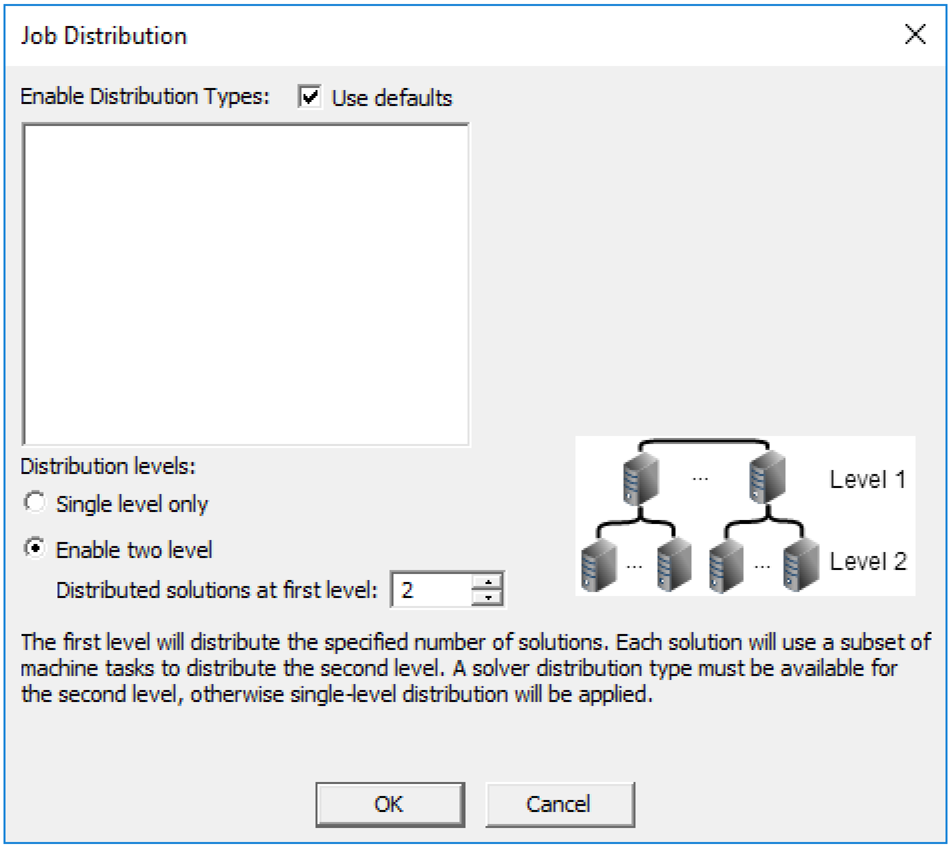
If you disable User automatic selection, you can modify the Job distribution settings.

- 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.

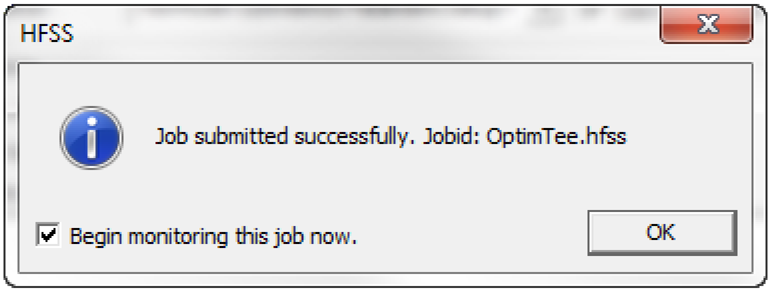


In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, 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\AnsysEM\RSM\Win64 on Windows.

The default location of this directory is /opt/AnsysEM/rsm/Linux64 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'

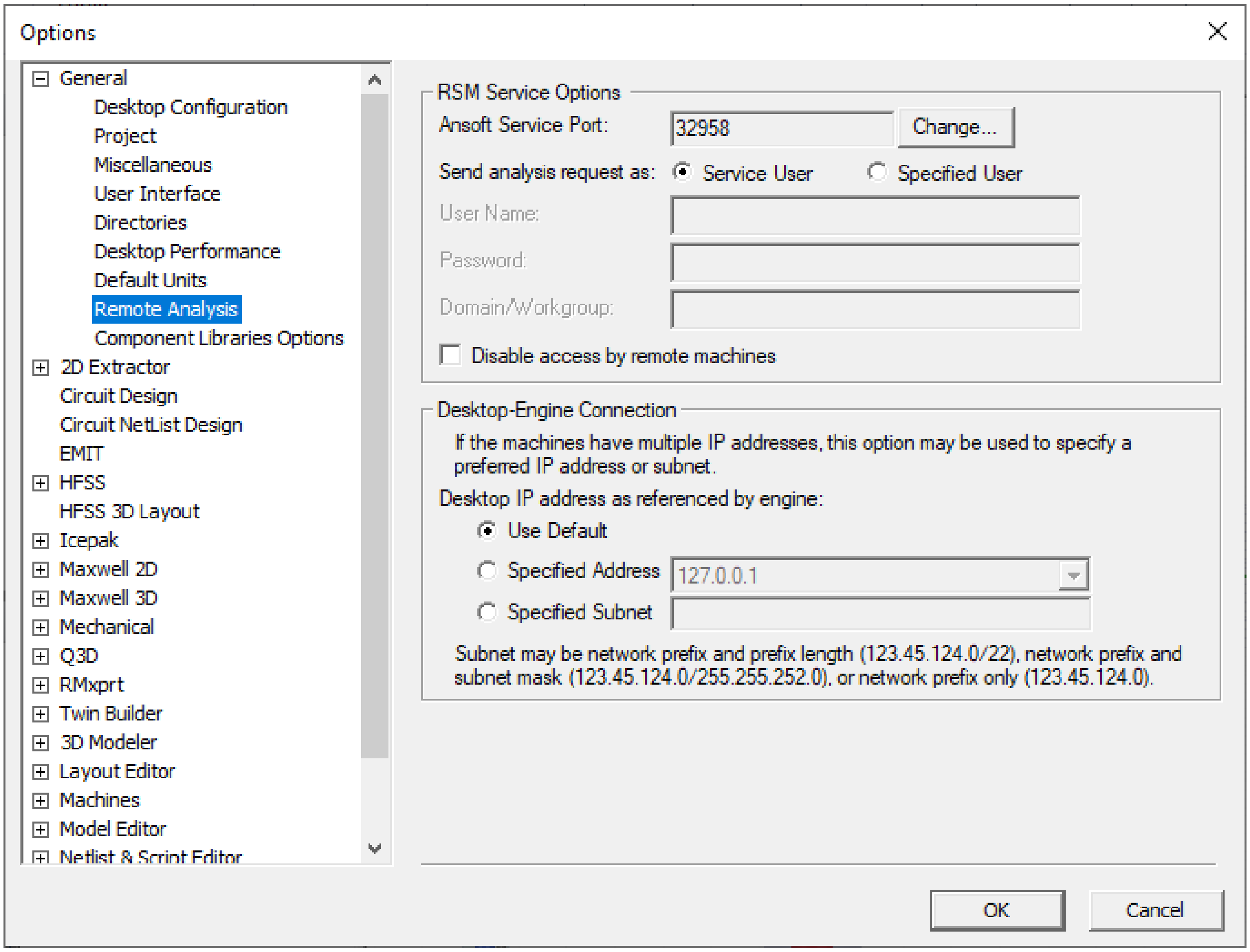
$end 'Default:CommDetails'

Ansoft Electromagnetics Desktop Configuration

The port number for connecting to the AnsoftRSMService is configured from the **General Options** window.

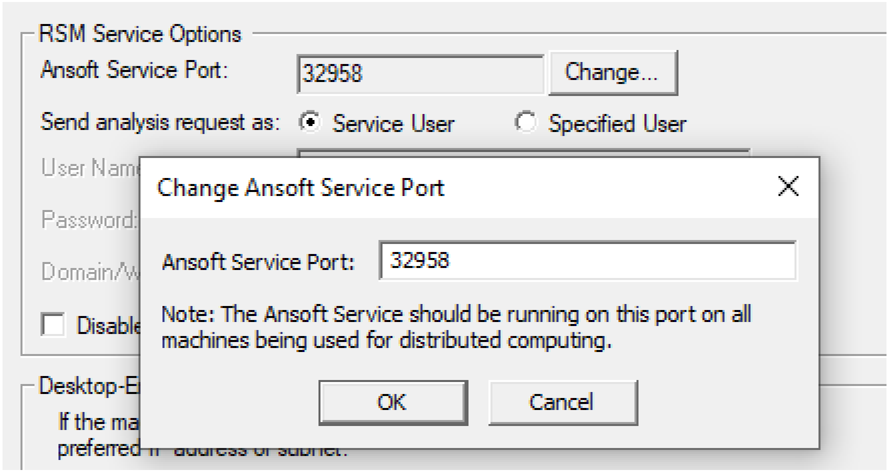
To access these options:

- 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.

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
- Running the diagnostics job
- Standard diagnostic job
- Using diagnostics scripts on Linux clusters
- Using Windows HPC job file
- Diagnostic report
- Site-specific diagnostics job
- Environment variables
- ANSYSEM_DIAG_PROD_DIR contents
- ANSYSEM_DIAG_RESULTS_DIR contents
- How does the diagnostic tool work

Supported schedulers

The tool supports diagnosis of issues on Linux and windows clusters managed by the following schedulers:

- LSF
- SGE
- PBS/Torque
- Windows HPC

For the above schedulers (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.

Basicdiagnostic 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:

```
.../Linux64/schedulers/diagnostics
```

Windows:

```
...\Win64\schedulers\diagnostics
```

Using diagnostics scripts on Linux clusters

The following basic scripts are provided in the diagnostics directory (.../Linux64/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:

```
...\Win64\schedulers\diagnostics.
```

To submit this diagnostic job, you must change the job description to suite your environment as following:

1. Select a directory for saving the diagnostic results. This directory must be accessible at the same path from all the hosts of the cluster.
2. Locate the directory for Ansys EM installation. This directory also must be accessible at the same path from all the hosts of the cluster.
3. Locate the winhpctest.xml in the diagnostics subdirectory of schedulers directory in 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 variable with the directories located in the first two steps:

   ANSYSEM_DIAG_PROD_DIR
   .

   ANSYSEM_DIAG_RESULTS_DIR

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 which (along with other related diagnostics results) is placed in the following directory

Linux:

`${HOME}/Ansoft/HPCDiag/Results/JOBID`

Windows:

`%ANSYSEM_DIAG_DIR%\Results\JOBID`

Report file:

`.../HTML/report.html`

where JOBID is the job ID assigned by the scheduler. On Windows, the user must specify ANSSEM_DIAG_DIR directory.

Site-specific diagnostics job

To run a diagnostic job with job submission parameters of your choice, you need to create your own job submission script. For example, you may want to specify a different LSF queue, or select a different SGE parallel environment. To run such a job, you need to create your own job submission script starting from the 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\v242\Win64
Linux example	/shared/AnsysEM/v242/Linux64
Comments	Windows: Required. Linux: Optional. Export this environment variable if you make a copy of the diagnostic script.

ANSYSEM_DIAG_RESULTS_DIR

Environment variable	**ANSYSEM_DIAG_RESULTS_DIR**
Description	Location of the diagnostic report and other results on a shared drive. This must be available at the same path from all the hosts of the cluster
Example	\\filer\Home\User\Ansoft\HPCDiag
Linux example	/shared/home/user/Ansoft/HPCDiag
Comments	Windows: Required. Linux: Optional. Export this environment variable if the home directory for the user is not accessible from the cluster.

ANSYSEM_DIAG_CUSTOM_DIR

Environment variable	**ANSYSEM_DIAG_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

Environment variable	**ANSYSEM_DIAG_CUSTOM_DIR**
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**.

Aborting an Analysis

To end the solution process before it is complete:

- Right-click In the **Progress** window and click **Abort**.

  The solver ends the analysis immediately.

The data for the currently solving pass or frequency point is deleted. All previously solved solutions are retained. For example, if you abort between the third and fourth adaptive pass, the solutions for the third pass will be available, and any solutions for the fourth pass are discarded. You must manually remove any solved results you do not want to keep.

To abort the solution process after the current adaptive pass or solved frequency point is complete:

- Right-click the **Progress** window, and click **Clean Stop** on the shortcut menu.

  The solver ends the analysis after the next solved pass or frequency point.

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

9 - Schematic Editor

The **Schematic Editor** is the tool used to create Circuit schematic designs. You create a design by opening the **Schematic Editor** and placing components, ports, connectors, and wires into a new schematic.

Starting the Schematic Editor

The **Schematic Editor** is the tool for creating schematic circuit designs. A design is a graphical representation of the electrical structure and characteristics of a circuit. A design may be *hierarchical*; that is, it may contain symbols that represent other designs. A design may also consist of one or more *pages* — multiple, associated graphical workspaces — that share a local namespace at the same hierarchical level.

To start the **Schematic Editor** and begin editing a new design, do either of the following:

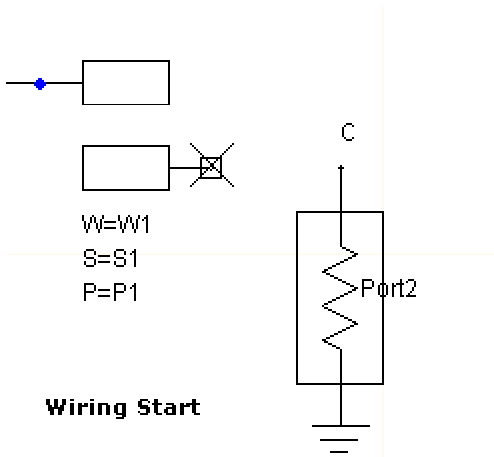
- From the **Project Manager** window, expand the **Project Tree**. Then right-click the [*active design folder*] and select **Insert** > **Insert Circuit Design**.
- From the **Project Manager** window **Project Tree**, select the project that contains the new design by clicking its icon. Then on the **Project** menu click **Insert Circuit Design**.

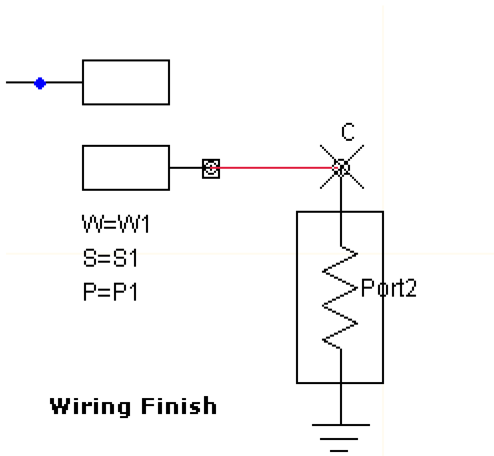
To edit an existing design, double-click its icon in the **Project Manager** window. For more information see Working with Projects.

The Schematic Editor Window

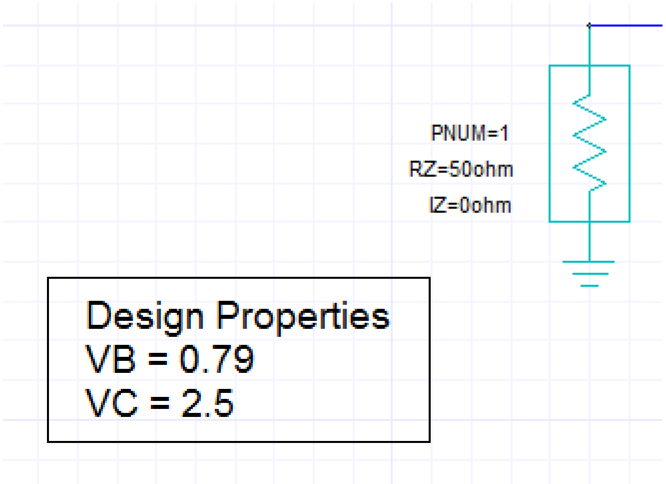
The **Schematic Editor** window allows you to place components and wire them together. You can move components by dragging and dropping them. Copy and paste can be used on components and their wires in the schematic editor. You can also copy and paste to other schematics.

When you place the cursor near a pin of a component, the cursor changes from an arrow to an **X**. This indicates that the **Schematic Editor** is in wiring mode. In wiring mode, click to start drawing a wire. Click again to end the wire.





As shown in the following figure, a Variable Text Block displays Design Properties. For a top-level circuit, the block displays Parameter Defaults and Variables. For a sub-design, the block displays Passed Parameter Values and Variables. The block is dynamic and updates automatically as changes are made. Double-clicking the block displays the Property window showing Design Variables to then modify. The block can be repositioned, and text fonts/colors are configurable using the Property Window. To control the visibility of the block, click **Settings > Design Variables** in **Electronics Desktop** Schematic Ribbon. There is script support to control visibility; for more information, see *Circuit Scripting: ShowVariableBlock*.



Commonly used items such as ports, *n*-port black boxes, grounds, and page connectors can be placed in the schematic by clicking their toolbar icons or by using the **Draw** menu.

View controls to zoom in, zoom out, and fit the drawing to the editor window are available on the **View** menu, and on the shortcut menu that opens when you right-click in a schematic.

The **arrow keys** scroll the view up, down, left, or right in small increments. **Page Up** and **Page Down** scroll the view up or down in larger increments. If you scroll so far that no objects are in the view, select **Fit Drawing** on the **View** drop-down in the menu (or **Ctrl+D**) to recenter the entire design, resized to fill the window.

Setting Schematic Editor Options

To set **Schematic Editor** options in Ansys Electronics Desktop:

1. Click **Tools>Options>General Options**, then in the left group box click to expand the plus sign (+) button directly next to **Schematic Editor**. This displays the following available selections:
   - General
   - Fonts
   - Colors
   - Wiring
   - Multiple Placement
   - Symbol Editor
2. Click each selection above that is shown beneath **Schematic Editor** and make the appropriate option choices in the right pane.
3. Click **OK**.

Schematic Editor Options: General Panel

The following options are set in the **General** group box under **Schematic Editor** in the **Tools>Options>General Options** window.

- **SymbolGraphics** – specifies the graphic symbol style (**IEEE** or **Traditional**) to be used on schematics. Each style contains two or more active levels for the graphic objects that comprise a symbol.
- **SubCircuit Pin Spacing** – specifies the pin spacing in grid units for pins on a subcircuit component symbol. The default value is 1 grid unit. (See Adding a New Subcircuit to a Design and Copying an Existing Subcircuit within a Design for additional information.)
- **Show Pin Labels** – specifies that pin labels can be displayed in the **Schematic Editor**.
- **Property Display Angle Follows Symbol Angle** – specifies that displayed text properties rotate when the associated component is rotated. Turned off by default.
- **Update Property Display on Definition Update** – specifies that when component symbol definitions are updated, the displayed properties of instances of the updated component on the schematic are automatically updated. Enabled by default. If not enabled, only the symbol graphics are updated.
- **Auto Scroll when close to edges** – specifies that the **Schematic Editor** display scrolls automatically when the cursor is placed close to the edge of the editor window.
- **Show advanced property data** – turns on the display of less often used property tabs, such as Symbol and General for components.
- **Net name display** – sets the net name property display distance on the net.
- **Symbol Scaling Factor** – Third-party vendors often follow a different symbol dimensioning system than that used in Twin Builder. Symbol sizes that look appropriate in a third-party application may not be appropriate forTwin Builder. To mitigate such issues, the Symbol Scaling factor can be used during import to scale the incoming symbol graphics for symbol formats such as SVG by the specified amount. Similarly, when exporting, the symbol can be rescaled to the original dimensions.
- **Selection Colors** – sets the colors of the first and subsequent objects as they are selected on a schematic.

Schematic Editor Options: Fonts Panel

Set the **FontName** (from a drop-down selection list) and **Size** for text used on schematics on the **Fonts** group box under **Schematic Editor** in the **Tools>Options>General Options** window. A **Sample Text** display window shows the appearance of the specified font and size. **Apply this font to all property displays in the active schematic** specifies that all displays in the **Schematic Editor** are to use the font style and size you selected.

Schematic Editor Options: Colors Panel

The following options are set in the **Colors** group box under **Schematic Editor** in the **Tools>Options>General Options** window.

The **Schematic Objects** group box allows you to set the color for each **ObjectType**.

- **ObjectType** – a drop-down menu used to select the object type whose colors you want to modify. Types are: **Components**, **InterfacePorts**, **PagePorts**, **GlobalPorts**, **Grounds**, **GraphicItems**, **Wires**, **Buses**, **TitleBlocks**, **PageBorders**, and **PageBorderText**.
- **SetColor** – implements the color you have selected for the object type.
- **ClearColor** – resets the color of the selected object type to its default value.

The **Individual Definitions** group box lets you set the color for specified **Components** and **WireDomains**.

- **ComponentName** – is used to identify the component whose colors you want to modify. (Visible only when the **ComponentsObjectType** is selected.)
- **Add** – allows you to add a component whose colors you want to define. (Visible only when the Components **ObjectType** is selected.)
- **WireDomain** – a drop-down menu used to select the wire domain whose color you want to modify. Each **Wire Domain** is assigned a unique color by default. **Domains** are: **Conservative - Electrical, Conservative - Magnetic, Conservative - Fluidic, Conservative - Translational, Conservative - Translational_V, Conservative - Rotational, Conservative - Rotational_V, Conservative - Radiant, Conservative - Thermal, Signal**, and **Quantity**.
- Set **Color** – implements the color you have selected for the specified component. All future instances of the component added to the active schematic window display possesses the selected color.
- **Clear Color** – resets the color of the component to its default value.
- Apply these color settings to the current schematic – applies the color settings for the selected **Object Type** to objects of that type in the active schematic window.

Schematic Editor Options: Wiring Panel

The following options are set in the **Wiring** group box under **Schematic Editor** in the **Tools>Options>General Options** window.

Connectivity Options:

- **Show Merge Wire window before combining wires** – displays the **Merge Wire** window before combining wires to allow for changes to the merge configuration.

- **Show Split Wire window before separating wires** – displays the **Split Wire** window before separating wires to allow for changes to the split configuration
- **When renaming a physically separate piece of wire** – offers two choices:

  – Split into a different net

  – Keep connected (name all pieces the same)

- **Show GlobalPort Disconnect** window before separating – displays the **GlobalPort Disconnect** window before separating wires to allow for changes to the global port configuration.

Schematic Editor Options: Multiple Placement Panel

The following options are set in the **Multiple Placement** group box under **Schematic Editor** in the **Tools>Options>General Options** window.

Check the appropriate objects in the **Items for multiple placement** group box.

- **Components**
- **Interface Ports**
- **Grounds**
- **Page Connectors**
- **Global Ports**

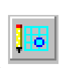
Schematic Editor Options: Symbol Editor Panel

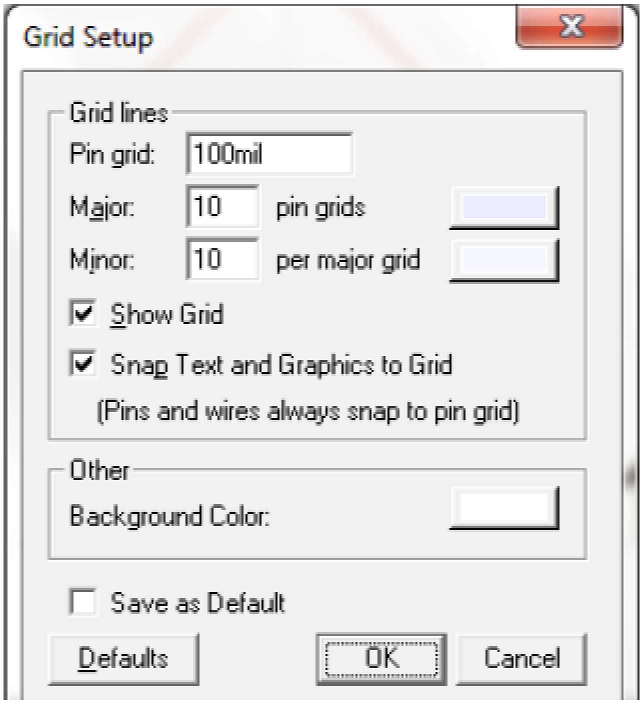
The following options are set in the **Symbol Editor** group box under **Schematic Editor** in the **Tools>Options>General Options** window.

Adjust Pin Orientation on Drag enables symbol pin orientation to be adjusted automatically as pins are dragged to the appropriate side of a symbol in the symbol editor.

Schematic Grid Setup

You can adjust the visibility, color, resolution, and other characteristics of the **Schematic Editor** grid. To access these settings, do either of the following to open the **Grid Setup** window:

- From the **Schematic** menu, select **Grid Setup**.
- From the **Schematic** toolbar, click **Grid Setup**  .



The **Grid Setup** window is used to set visibility, colors, resolutions, and snapping for the Schematic editor's alignment grid.

Grid Lines

Pin grid— Set the granularity of the grid lines in the pin grid expressed in units of millimeters.

Major —This check box specifies the spacing of major grid lines (default in mils). To specify a different value, click in the **Major** group box and type the new value. To change the color of the major grid lines, click the **Major** color button, specify a color in the **Color** window, then click **OK**

Minor —This check box specifies the spacing of minor grid lines (default in mils). To specify a different value, click in the **Minor** group box, then type the new value. To change the color of the minor grid lines, click the **Minor** color button, specify a color in the **Color** window, then click **OK**.

Show Grid —This check box toggles grid-line visibility. Check **Show Grid** to make the grid lines visible, or clear **Show Grid** to turn grid line display off. The default for **Show Grid** is *on/checked*.

Snap Text and Graphics to Grid —This check box controls whether graphics (arcs, circles, lines, polygons, rectangles, and text) placed near the grid, but not on, automatically *snaps* upon placement to the nearest grid intersection. The default for **Snap Text and Graphics to Grid** is *on/checked*.

Other

Background Color— This control sets the background color used for schematic editing. To modify the **Background Color**, click the color box to open the **Color** window. Then specify a color and click **OK**.

Save as Default —Check this box to save the current **Grid Setup** values for use across designs.

- Click **Defaults** to restore all **Grid Setup** settings to their installation defaults.
- Click **OK** to save changes made and close the window.
- Click **Cancel** to close this window without committing any changes.

> **Note:**
>
> - To ensure electrical connectivity among schematic elements, the pins and wires of placed components/ports always snap to a 100-mil (2.54-millimeter) connectivity grid regardless of the **Major** and **Minor** grid line settings. This snapping cannot be deactivated. Additionally, the default spacing of the connectivity grid cannot be adjusted.
> - Use of the **Minor** grid stetting varies depending upon the active Editor. In the Layout Editor, the minor grid-line setting specifies the number of units between each minor grid division. However, in the Schematic and Symbol Editors, the minor grid-line setting specifies the number of minor grid lines that appear between major grid lines.

Schematic Page Setup

You can adjust the page size, orientation, borders, title blocks, and other characteristics of the **Schematic Editor** pages. You can access these settings via the **Schematic** menu by clicking the **Page Borders and Title Blocks** menu item to open the **Page Border, Title Block, Page Properties** window, which contains the following three tabs:

- Page Borders
- Title Block
- Page Properties and Display

Page Borders Tab

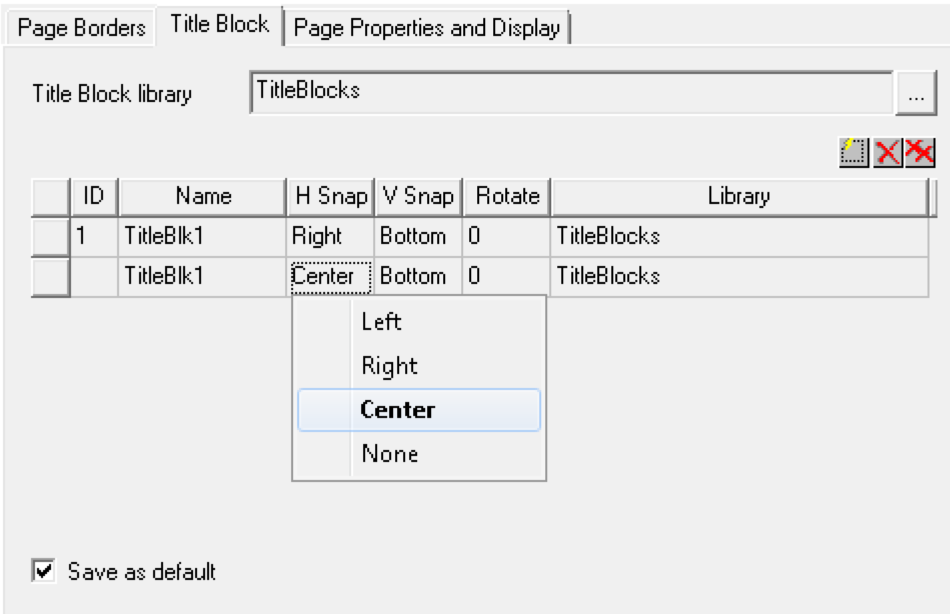
Click the **Page Borders** tab to set the border type, page size, page margins, and ANSI Zones for the **Schematic Editor** display. Check **Save as default** to save the current tab settings as defaults for new schematic pages.

- **Border Type**
  - **None** – no border is displayed. All other settings in the tab are unavailable if **None** is selected. If title blocks have been defined, they is hidden.

- ◦ **Outline only –** displays a simple outline of the specified **PageSize**. **Margins** can also be adjusted.
- ◦ **ANSI Zones –** displays fully ANSI-compliant borders of the specified **PageSize**. The specified **Number of Zones** are added to the border.
- ◦ **ISO Zones –** displays fully ISO-compliant borders of the specified **PageSize**. The specified **Number of Zones** are added to the border.
- ◦ **DIN Zones –** displays fully DIN-compliant borders of the specified **PageSize**. The specified **Number of Zones** are added to the border.

- **Page Size –** allows you to choose from a number of preset vertical and horizontal page sizes. Selecting **Custom** allows you to set your own page **Width** and **Height**.
- **Number of Zones –** allows you to set the number of **Vertical** and **Horizontal** zones to draw. The minimum and maximum number of zones allowed varies with page size. If you try to set a number outside of the allowable range, a window appears displaying the allowable range of values.
- **Margins –** allows you to set the **Vertical** and **Horizontal** margins in inches.

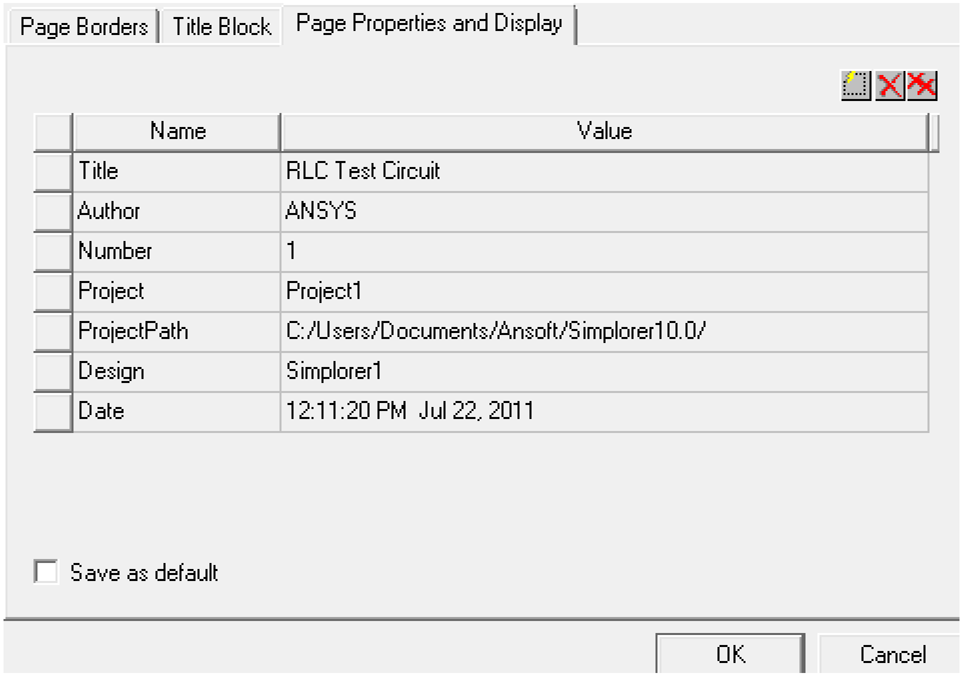
Title Block Tab



- From the **Title Block** tab, choose a **Title Block Library** by clicking the **[...]** browse button to locate a title block .aslb library; or accept the default library.
- To add a title block to the grid, click the yellow “lightning bolt” icon, and select a title block symbol on the list presented.

- Within the grid control, the title block can be set to snap along the bottom, top, or on either side, or can be set to not snap at all. You can also **Rotate** the title block 0, 90 or 270 degrees.
- You can move title blocks on the schematic much like any other element, though title blocks have optional snapping constraints. If a title block has been set to snap (any choice but **None**), it can be moved between snapping points: Left, Center and Right for horizontal; Top, Center and Bottom for vertical. The snap property on the title block is changed accordingly. The snap location can also be changed through the property window or the Page Borders and Title Blocks window.
- More than one title block can be added to a page. Unwanted title blocks can be deleted on the page using either the single red “X” button to delete a single entry, or the double red “X” button to delete all entries. An ID property on the title block corresponds to the number in the **ID** column of this tab, so a user can identify which title block to delete.
- Click **OK** to add all title blocks in the list to the schematic page, The blocks in the list snap to the border as specified, or as previously placed if the snap specification is **None**. If blocks are removed on the list, they are removed on the schematic. Any changes made to blocks in the list are applied to preexisting title blocks.
- Selecting **None** for Border Type after setting up border and title block(s) causes title blocks to be remembered but not drawn.
- Note that to create a symbol with company logo, *propdisplay* positioning, and as much additional information and graphics as needed and store it in a symbol library where it is available for any schematic page.
- Clicking on the **Save as default** check box causes the title block information to be saved to the registry and used as a default for the subsequent pages.

Page Properties and Display Tab



- Page properties can include fields such as: **ProjectPath**, **Project**, **Design**, **Title**, **Author**, and **Date**.
- Click a **Value** field and enter the appropriate information.
- Data entered in fields used by the title block(s) appear in the title block on the schematic.
- Page properties are saved if the **Save as default** check box is checked, then initialized when a new page is created.
- You can create additional page properties, by clicking the yellow “lightning bolt” button.

In the symbol editor, a *propdisplay* may be added to a title block symbol, and the title block *propdisplay* shows the value of the same-named page property.

Zooming and Panning the Schematic View

You can magnify (zoom in) or shrink (zoom out) the contents in the view window, or on a rectangular area in the view window. You can pan (scroll) the view in any direction.

To zoom in on the contents in the view window:

1. Click **Zoom In o** n the **View** menu, or right-click in the schematic window and select **Zoom In**.

   You can also use the keyboard shortcut **Ctrl ++**.

2. The view zooms in to a larger magnification. The absolute size of the model does not change.
3. Repeat the operation until the appropriate magnification is achieved.

To zoom out on the contents in the view window:

1. Click **Zoom Out o** n the **View** menu, or right-click in the schematic window and select **Zoom Out**.

   You can also use the keyboard shortcut **Ctrl+-**.

2. The view zooms out to a smaller magnification. The absolute size of the model does not change.
3. Repeat the operation until the appropriate magnification is achieved.

To magnify a specific rectangular area in the view window:

1. From the **View** menu, select **Zoom Area**, or right-click in the schematic window and select **Zoom Area**. The cursor changes to a magnifying glass.
2. Draw a rectangle (or square) by selecting two diagonally opposite corners. This is the area where magnification is increased.

The rectangular area is magnified in size and the cursor returns to normal. The absolute size of the model does not change.

Restoring a Previous Zoom View

After executing a zoom in or zoom out operation, you can return to the previous magnification by selecting **Zoom Previous** on the **View** drop-down.

Panning the View

To pan (scroll) the view in any direction, hold the left mouse button and tap **Shift**. The view is attached to the cursor. Drag the mouse to pan the view in any direction. Release the mouse button to end the panning operation.

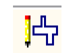
Displaying Symbols in Schematics

There are two display styles available for symbols in the **Schematic Editor**, **IEEE** and **Traditional**. The choice between **IEEE** and **Traditional** is configured in General tab of the **Tools** > **Options > Schematic Editor Options** window.

- Each object in a symbol is associated with a level.
- Graphics on levels 0 and 1 are always present and shown. Graphics on other levels are visible by default, but can be configured as visible/invisible programmatically.

- For **IEEE** and **Traditional** styles, all levels are independent — except for level 0 where pins are displayed.
- Both **IEEE** and **Traditional** symbols contain at least two active levels, and you may add any number of levels for each style.

Viewing Layout from Schematics

To view the layout corresponding to a schematic design, click the Edit Layout icon  on the menu. The Layout Editor window opens to open the layout corresponding to the current schematic design. See the Layout Editor User's Guide for further information.

Placing Components in Schematics

To facilitate the copying of design materials, drag and drop both designs and components on the **Project Manager** window to the **Schematic Editor**. In particular, drag and drop into a schematic:

- A *design* on the **Project Manager** window
- A *component* on the **Project Manager** window Definitions folder
- A *component* on the Components tab in the Component Libraries window

Components can also be placed by clicking on a component symbol in the Symbols tab of the Component Libraries window. Then click in the Schematic window to place to place the component.

> **Note:**
>
> You can drag and drop one or more copies of a component into a schematic. Alternately, right-click on a component and select **Copy**, then right-click in the schematic and select **Paste** (or use the **Ctrl** +**C** and **Ctrl** +**V** keyboard commands).
>
> If **Multiple Placement** is activated for components in the **Schematic Options** window, place multiple instances of a component in the schematic by clicking at multiple locations. To stop placing components during multiple placement, do either of the following:
>
> - Press **Enter**, the spacebar, **Backspace**, or **Esc** on your keyboard.
> - Right-click , then select **Place and Finish**, **Finish**, **Cancel**, or **Back**.

To place a component:

1. Select the schematic that contains the component that you places.

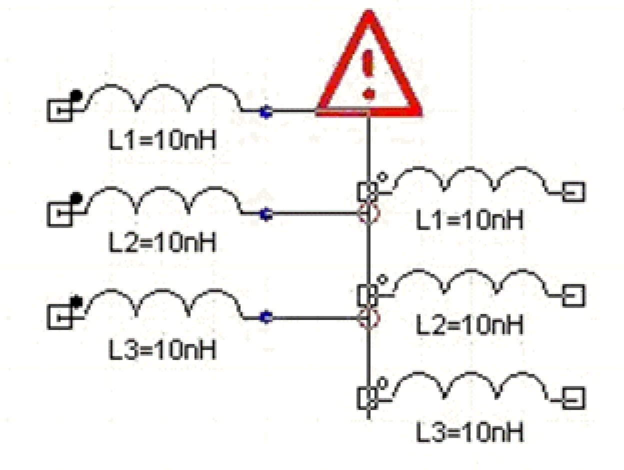
2. Next, select and place the component in the schematic using one of the methods described in the *Note* above. Inside the schematic window, the cursor is accompanied by the component symbol for placement.
   - You can rotate a component before placing it by repeatedly pressing **R** on your keyboard. Each press rotates the component 90º counterclockwise.
   - You can flip the component left-to-right by pressing **X**.
   - You can flip the component top-to-bottom by pressing **Y**.

   For components with physical layers, the **Merge Layers** window may appear. See the Merge Layers window topic for further information.

Note:

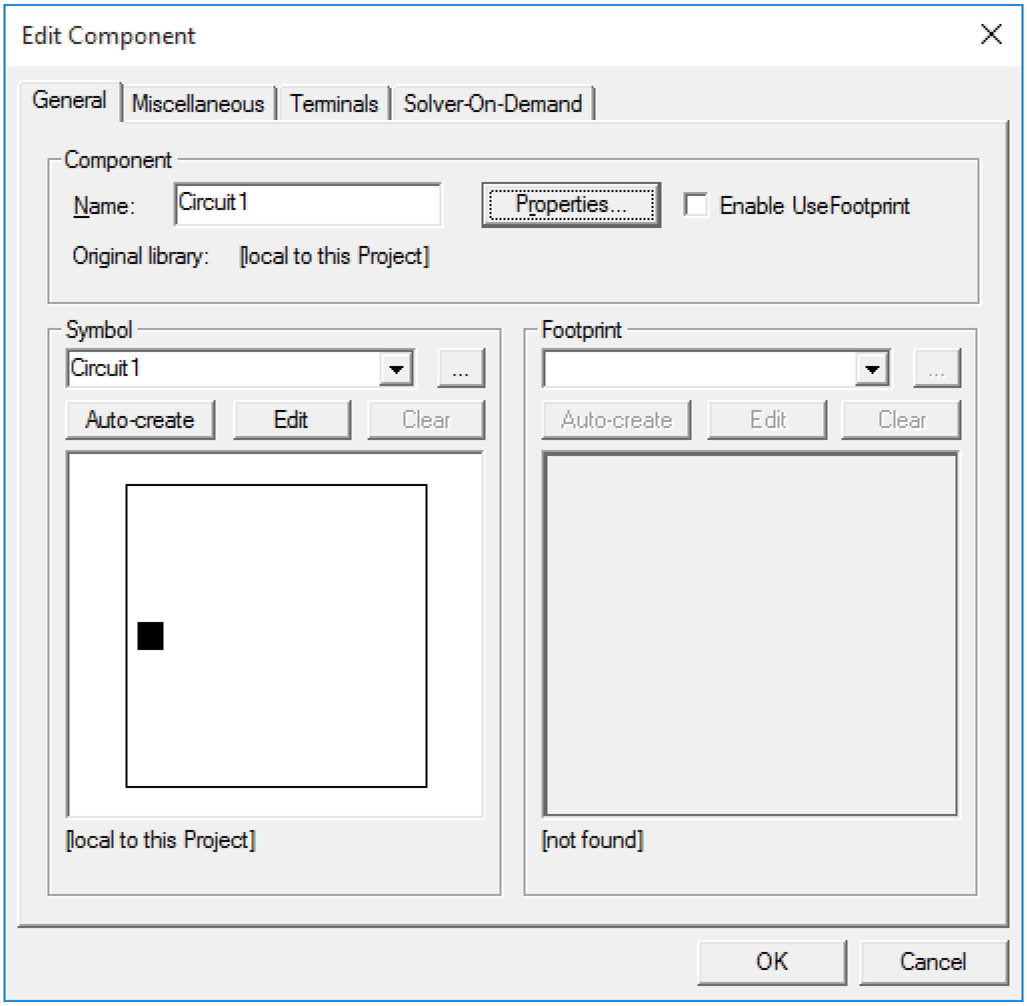
1. The first time a component project is placed, entries for it are added in the Component Libraries window's **Most Recently Used** and **Project Components** headings. To save time as your work progresses, place new instances of a component by double-clicking these icons in the Components tab or selecting the symbols in the Symbols tab. Please note that for security reasons, encrypted components are not saved in the Most Recently Used list or the Favorites list.
2. To ensure electrical connectivity among schematic elements, the pins of placed components snap to a 100-mil (2.54-millimeter) grid. This snapping cannot be deactivated, and the spacing of the connectivity grid cannot be adjusted.
3. To move an existing component, select and drag the component to a new position. In-place wiring to the component is automatically adjusted. To retain the in-place wiring, hold down **Alt** as you drag the component to a new position.

   Please note that when dragging a component to a new position, if the wires attached to it make an unintended connection, a red exclamation mark is displayed as a warning. The following figure illustrates that red circles are used to indicate where connections is made and the exclamation point warns that there may be a connection that is not what you intended.



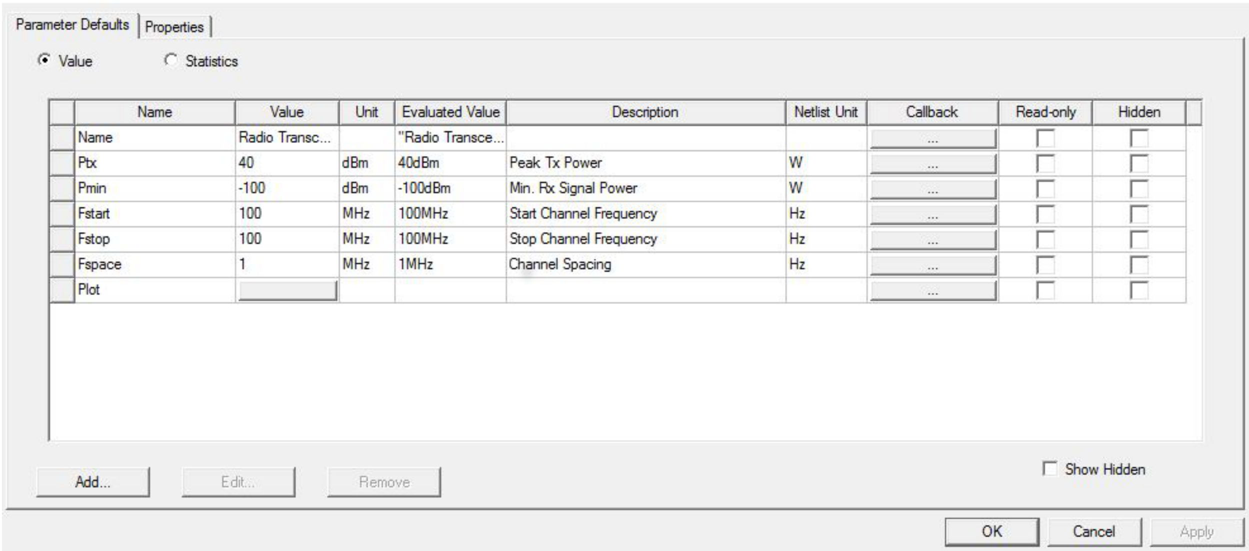
Displaying and Editing Component Properties

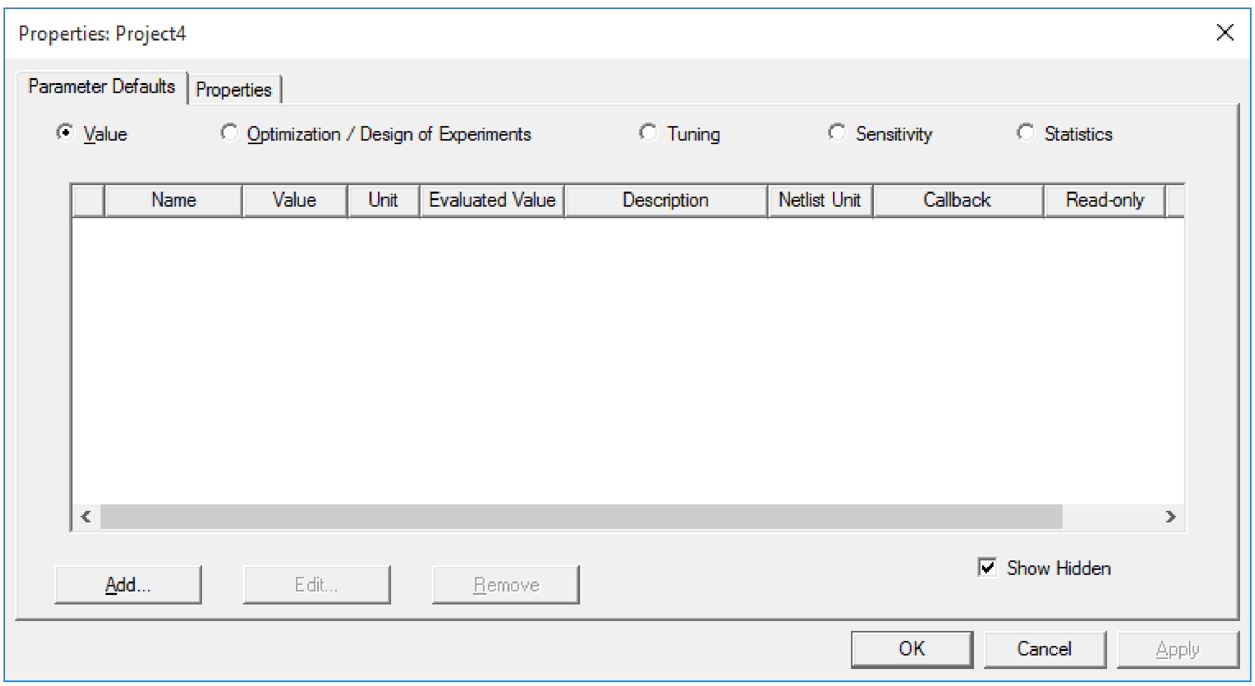
To open the **Properties** window for a selected component, double-click the component in the **Schematic Editor**, or double-click the component definition in the **Project Manager** window to open. This opens the **Edit Component** window.



- To display and edit component properties, in the **Edit Components** window, click **Properties** to open the **Properties** window for the component.

The **Properties** window is context sensitive and presents different options depending upon the type of component whose properties are being viewed.



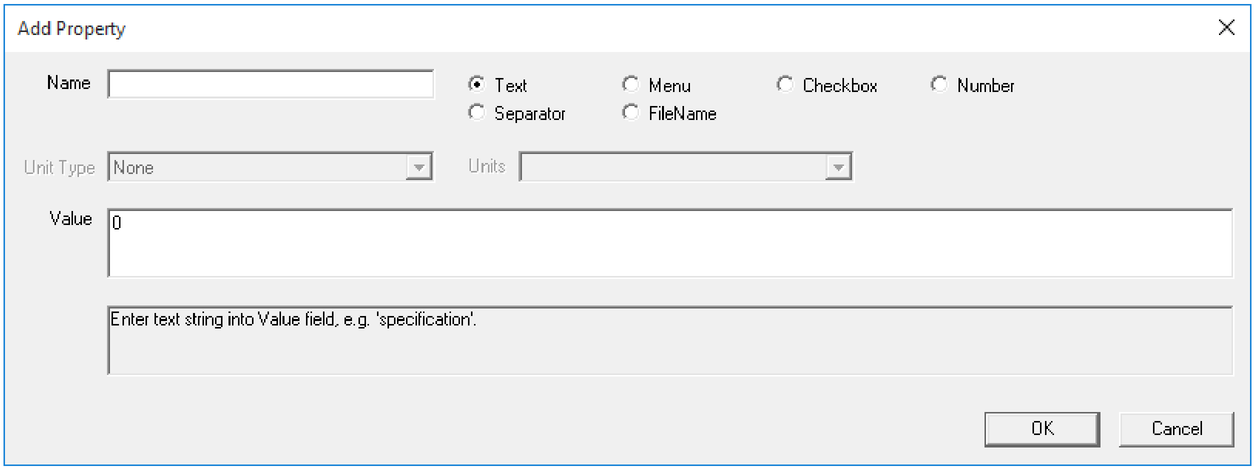


Parameter Defaults Tab

The **Parameter Defaults** tab lists, displays, and sets the default values for component parameters during general analysis.

- Select **Value** (general analysis) or **Statistics** (statistical analysis) to set the default values for component parameters. Alternately, you may also be given the option to select on the following analyses: **Optimization / Design of Experiments**, **Tuning**, **Sensitivity**.

- Use **Edit** to modify the values of a definition.
- Use **Add** to add a parameter to the definition. You are prompted to define the parameter name, value, and value type (text input, menu, check box, etc.). You may **not** add reserved system parameters to the definition.



- To remove a property on a component, select the property and click **Remove**. The property is deleted on the component definition (model) and from all instances in the design.
- You can use an expression for a numeric property value. The **Electronics Desktop** internal expression syntax is the same as the one supported in the Reports window. See "2D and Circuit Selecting a Function" on page 18-323 for more information on this expression handler.
- Clicking the **Show Hidden** check box allows you to view hidden properties of the component. Hidden properties contain system-defined values and rules for interpreting predefined component parameters. Modifying hidden properties requires specialized knowledge of the component, and is not needed for normal operation.

> **Note:** Removing a property from a component from an Electronics Desktop component library may produce undesirable results when the component is simulated.

Properties Tab

Depending on the attribute type, 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, 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 window.

- Click the button to open a settings window. Then make any necessary changes.
- Click the attribute to open a setting. The make any necessary changes.

Related Topics

Parameter Statistics Display

Reserved Component Parameters

Copying and Pasting Properties

You can copy and paste properties for primitive drawing elements (graphical objects) and components.

Primitive drawing elements (graphical objects)

You can copy and paste the common properties of the following primitive drawing elements (graphical objects):

- Arcs
- Circles
- Lines
- Rectangles
- Polygons
- Text
- Images

Components

For components, copy either the value of properties or the value and attribute of properties. Properties are copied only when property names match.

Parameters, quantities, signals, and property displays are copied along with properties and their attributes.

> **Note:**
>
> Property displays are copied to same location as the source if they are on the left, bottom, right, top or center. Custom location property displays are copied to the default bottom location.

There are two types of components: components that have user-defined parameters, and those that do not have user-defined parameters. user-defined properties are created in two ways:

- Properties created by users in equations such as FML and FML_INIT
- Properties that are dynamically created depending on certain parameters. An example is the GS component, where coefficient parameters are created dynamically based on the values of numerator (n) and denominator (d).

The options available in the **User Defined Properties of Component** pane differ according to the type of component, as shown in the following table.

Component type	User-Defined Properties Information	Examples
Component has user-defined parameters	Components for which copy properties allow replacing only existing parameters.	GS, NXMY, NDTAB, and DES
	Components for which copy properties allow either replacing existing parameters or adding parameters to the existing list. Note: Only components that have equations allow adding more to their existing properties.	FML, FML_INIT, EQUBL, and State components, such as STATE_01, STATE_10, STATE_33, and STATE_Flexible
Component does not have user-defined parameters	The **User Defined Properties of Component** pane is unavailable when there are no user-defined properties for the component.	R, L, G, C, E, D, 2, DELAY, DIFF, and GAIN

Related Topics

Copy and Paste Selected Properties for Components

Copy and Paste Common Properties for Primitive Drawing Elements

Copy and Paste Selected Properties for Components

To copy and paste selected properties from one component to one or more components:

1. Select the component you want to copy from.
2. Use one of the following methods to access Copy Data:
   - Click **Edit** > **Copy Data**.
   - Right-click the object and select **Copy Data**.
   - Press **Ctrl** +**Shift** +**C**.

3. Select the components you want to paste to.
4. Paste the properties by using one of the following methods:
   - Select **Edit** > **Paste Data**.
   - Right-click the component and select **Paste Data**.
   - Press **Ctrl** +**Shift** +**V**.

The **Copy Properties Data** window is displayed.

1. In the **Copy data** pane, select **Value and attributes** or **Only value**. When you select **Only value**, the attributes columns are hidden.
2. In the **User Defined Properties of Component** pane, select an option. The pane is unavailable when there are no user-defined properties for the component.
   - **Do not add or replace any properties**. Use this option to copy fixed properties without copying the user-defined properties.
   - **Add new properties**. This option is active if you are copying an equation component. If there are no new properties to add, this option is unavailable.
   - **Replace existing properties**.Use this option to copy data from a component that has user-defined properties but has no equations.
3. Do one of the following to select the properties to copy.
   - Select one or more of the following check boxes: Parameters, Quantities, Property Displays, or Signals. If the component has no signals, the Signals check box is not available.
   - Click **Select** in the first column to toggle between selecting all and clearing all.
   - Click the **Select** check box for the property. If you do not want to include the property, clear the **Select** check box.

In the **Filter by** options, filter the view by selecting **All**, **Parameters**, **Quantities**, or **Signals**.

Click the **Show Description** check box to open the descriptions for each property.

1. Click **OK**. The properties are pasted into the component.

Copy and Paste Common Properties for Primitive Drawing Elements

To copy and paste the common properties from a primitive drawing element (graphical object) to another:

1. Select the object you want to copy from.
2. Use one of the following methods to access Copy Data:
   - Click **Edit** > **Copy Data**.
   - Right-click the object and select **Copy Data**.
   - Press **Ctrl** +**Shift** +**C**.

3. Select the object you want to paste to.
4. Paste the properties by using one of the following methods:
   - Select **Edit** > **Paste Data**.
   - Right-click the component and select **Paste Data**.
   - Press **Ctrl** +**Shift** +**V**.

The **Copy Properties Data** window is displayed.

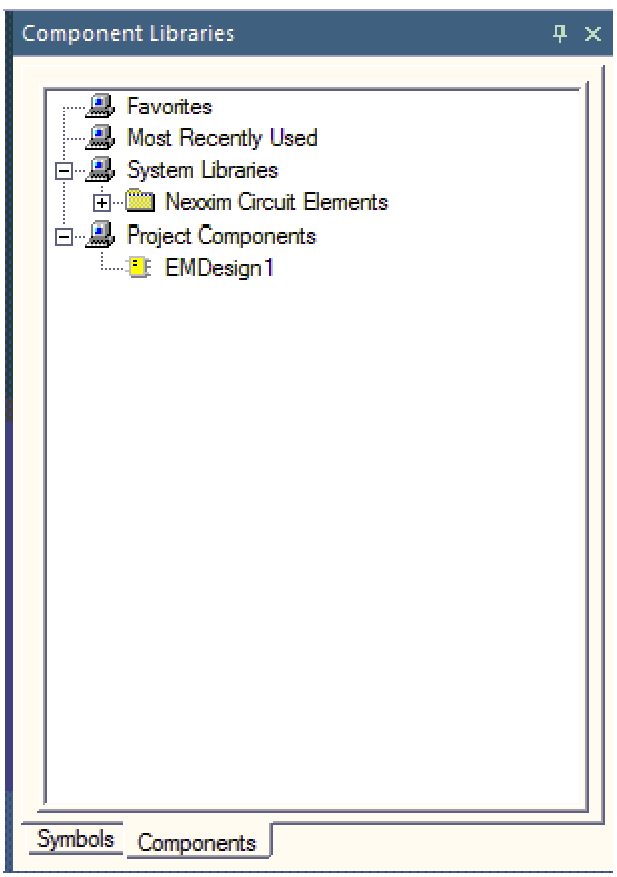
5. In the **Properties** pane, keep the default selections or clear the check boxes for the properties you do not want to copy.

Click the **Show Description** check box to open the description for each property.

6. Click **OK**. The properties are pasted into the object.

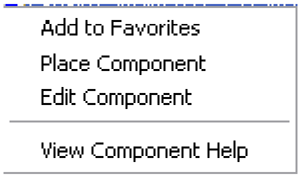
Favorites and Most Recently Used Components

When the **Schematic Editor** is the active window, the Components tab in the Projects window keeps track of the components that have been placed most recently. You can also create a Favorites list of components that you use most frequently.

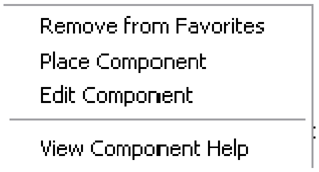


> **Note:**
>
> For security reasons, encrypted components are not saved in the Most Recently Used list or in the Favorites list.

To add a component to the Favorites list, right-click on its icon in the Components Libraries window to open the following menu:



To remove a component on the Favorites list, right-click the component in the Favorites list, then select **Remove from Favorites**:



Finding Elements

To find one or more elements in the **Schematic Editor**:

1. From the **Edit** menu, select **Find Elements**. This opens the **Find Elements** window.

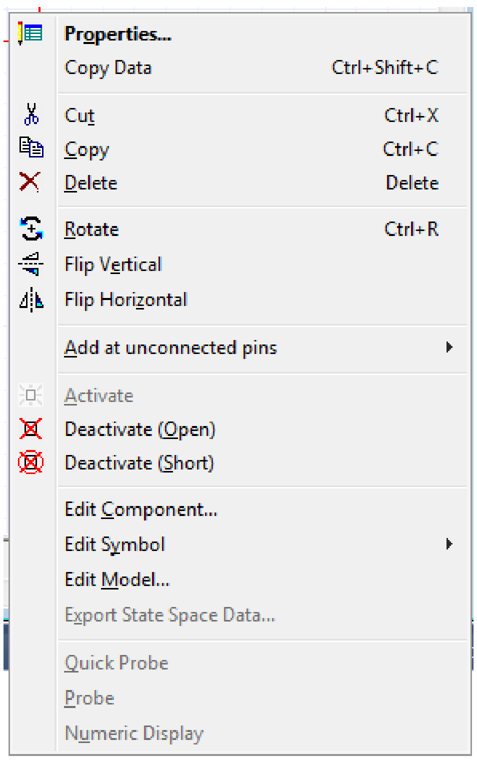
2. Click **Advanced**.

3. Click **Instancename** to open a drop-down menu that displays the elements to search for.
3. Click **Contains** to open a drop-down menu that displays the criteria to search with.
4. You can also enter an amount in the **Value** column that the element **Contains** or **Does not Contain**.
5. You can find elements using the following controls:
   - Check a **Filter** group box to include an element in the search
   - Check an **Options** group box to vary the scope of the search.

Elements found in the search are displayed in the **Results** window of the **Find Element** window and are highlighted in the **Schematic Editor**.

Operations on Components

Right-clicking on a component or other object in the schematic selects the object and open a context-sensitive menu similar to the following:



This menu contains **Schematic Editor** commands.

- **Properties** opens the Properties window.
- **Copy Data** copies the selected data into the clipboard.
- **Cut** deletes the component, and retains a copy for pasting into a schematic in the same application.

- **Copy** creates a local copy for pasting into a schematic in the same application.
- **Delete** deletes the component.
- **Rotate** rotates the component 90 degrees counterclockwise.
- **Flip Vertical** flips the component about the X-axis.
- **Flip Horizontal** flips the component about the Y-axis.
- **Add at unconnected pins** opens a menu to add specific terminations to each unconnected pin of the component.
- **Activate** causes the activation of one or more selected components.
- **Deactivate (Open)** temporarily converts the component into an open circuit. **Activate** restores a deactivated component to the circuit.
- **Deactivate (Short)** temporarily converts the component into a short circuit. **Activate** restores a deactivated component to the circuit.
- **Push Down** moves down one level in the project design.
- **Pop Up** moves up one level in the project design.
- **Edit Component** brings up the Edit Component window for the component.
- **Edit Symbol** is a menu with the following commands:
  - **Geometry** brings up the Symbol Editor for the default Symbol.
  - **Bus Pins** brings up the Bus Pin Editor window.
  - **Pin Locations** brings up the Edit Symbol Pin Location Editor.
  - **Pin Properties** brings up Editing Pin Properties window.
  - **Create Instance Symbol** lets you move unconnected pins around the circumference of the bounding rectangle of a selected component (such as model components and components that represent subcircuits). Also lets you resize the bounding rectangle. See Adjusting Symbols and Pins for additional information.
- **Edit Instance Symbol** has the following commands and replaces the **Edit Symbol** submenu when the component instance has an instance symbol (See **Create Instance Symbol** above).
  - **Geometry** brings up the Symbol Editor for the default Symbol.
  - **Pin Locations** brings up the Edit Symbol Pin Location Editor.
  - **Move Pins** lets you move unconnected pins around the circumference of the bounding rectangle of a selected component (such as model components and components that represent subcircuits). Also lets you resize the bounding rectangle. See Adjusting Symbols and Pins for additional information.
  - **Revert to default symbol** removes the instance symbol and allows the component instance to use the default symbol in its component.
- **Edit Model** brings up an appropriate window to edit the associated model; this command is not present if there is no model.

- **Export State-Space Data** is enabled only if the model has state-space data available. The command opens a **Save File As** window. When you provide a filename and location, the .sss file is generated and updated with the port names in the model.
- **Quick Probe** creates a probe port connector.
- **Probe** adds an XY Plot into the schematic. The user choose what to display on the menu.
- **Numeric Display** adds a data table into the schematic. The user choose what to display on the menu.

Right-clicking a component in the **Components** tab of the Project window opens a menu similar to the following:

This menu contains the following commands:

- **Add to Favorites** adds the component to your list of frequently used components (See Favorites and Most Recently Used Components).
- **Place Component** attaches the component symbol to the cursor for placement in the schematic.
- **Edit Component** opens the Component Editor window. (See The Edit Component window in the Component Libraries topic for details).
- **View Component Help** opens the help topic for the component.

Related Topics

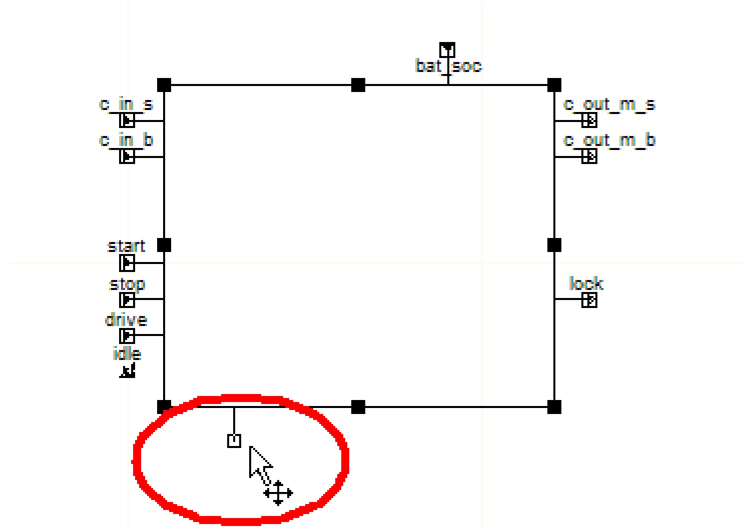
Adjusting Symbols and Pins

Adjusting Symbols and Pins

The **Adjust Symbol and Pins ...** menu item appears in the shortcut menu when a single component is selected and the component's symbol has a rectangle that matches the graphic extent such as those for auto-generated symbols, including those for model components and components representing subcircuits.

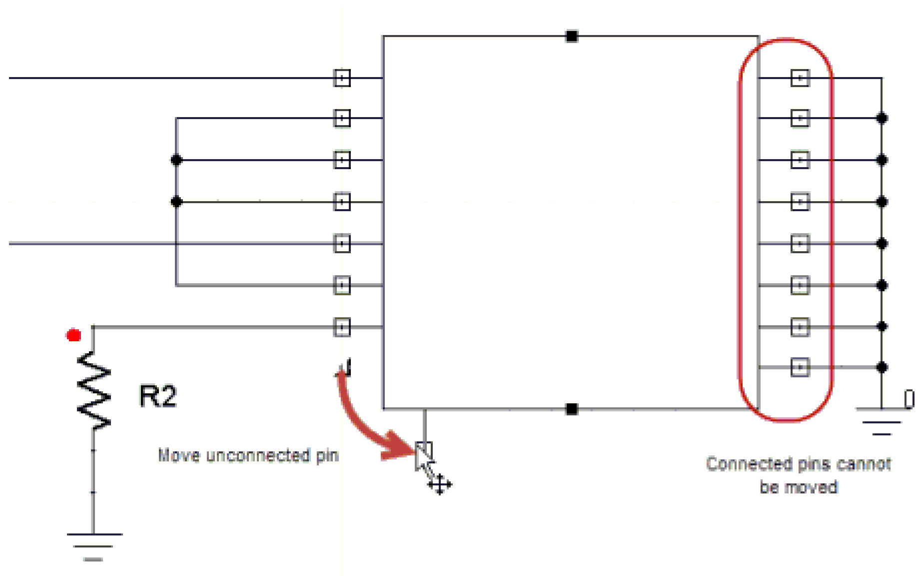
This command allows you to move unconnected pins around the circumference of the bounding rectangle. You can also resize the rectangle.

- A pin may be moved to any vacant grid location (a grid location not occupied by any other of the selected component's pins). Its pin stem automatically rotates depending on the side to which the pin is moved.



> **Note:** Pins with connections cannot be moved. The Message Manager displays a reminder if you attempt to move a connected pin.

- Moving an unconnected pin onto a wire or pin of another component and accepting the changes causes a connection to be made.
- With no pins connected, there are eight handles that may be grabbed to resize the rectangle. When the rectangle is resized, any unconnected pins moves proportionately, snapping to the nearest grid point.
- Connected pins constrain to how the rectangle may be resized. A connected pin on one side of the component prevents that side from moving (there is no handles on that side or its corners), though that side of the rectangle may be lengthened or shortened by moving a corner on the opposite side.



- If the symbol has a picture whose size matches the rectangle (such as for coupling components), the picture is resized as the symbol is adjusted.
- To accept the changes, press **Enter**, or right-click and select **Finish** or **Place and Finish**. To abort the command and discard changes, press **Backspace** or **Esc**, or right-click and select either **Back** or **Cancel** on the shortcut menu.

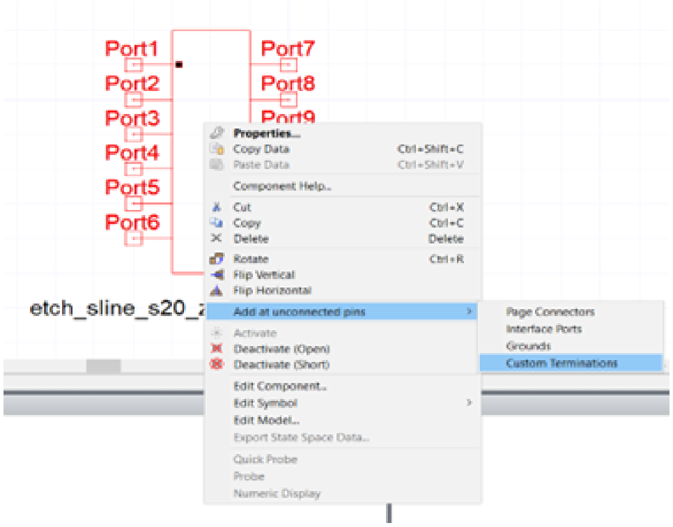
The changes you make using this command are applied to a *copy* of the component's original symbol, which is used uniquely by the selected schematic component instance. To revert to the original symbol, right-click the component and select **Revert to default symbol** in the **Edit Symbol** shortcut menu.

Custom Terminations

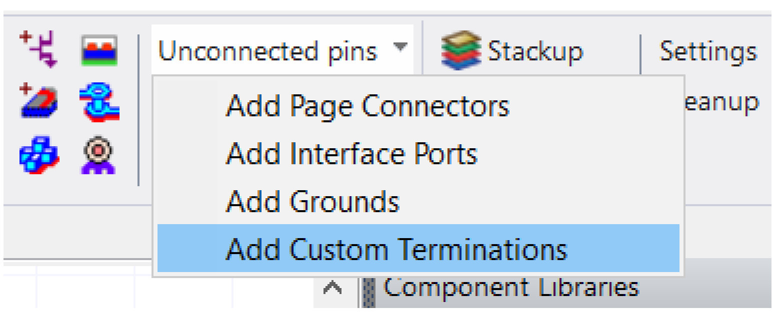
The custom terminations dialog can be used to create different types of terminations for unconnected pins in one place.

To launch the Custom Terminations dialog, use one of 3 methods:

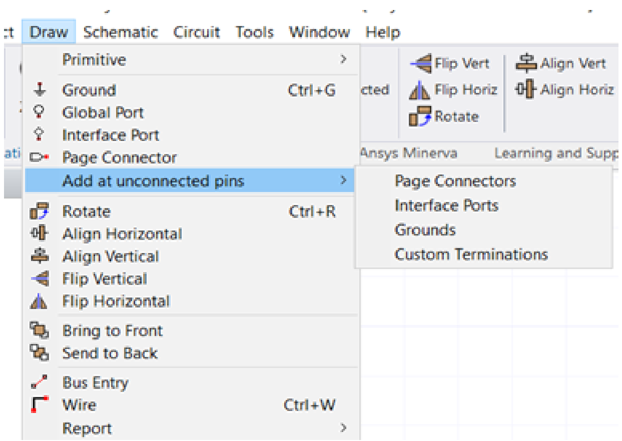
1. Right-click on a component in schematic.



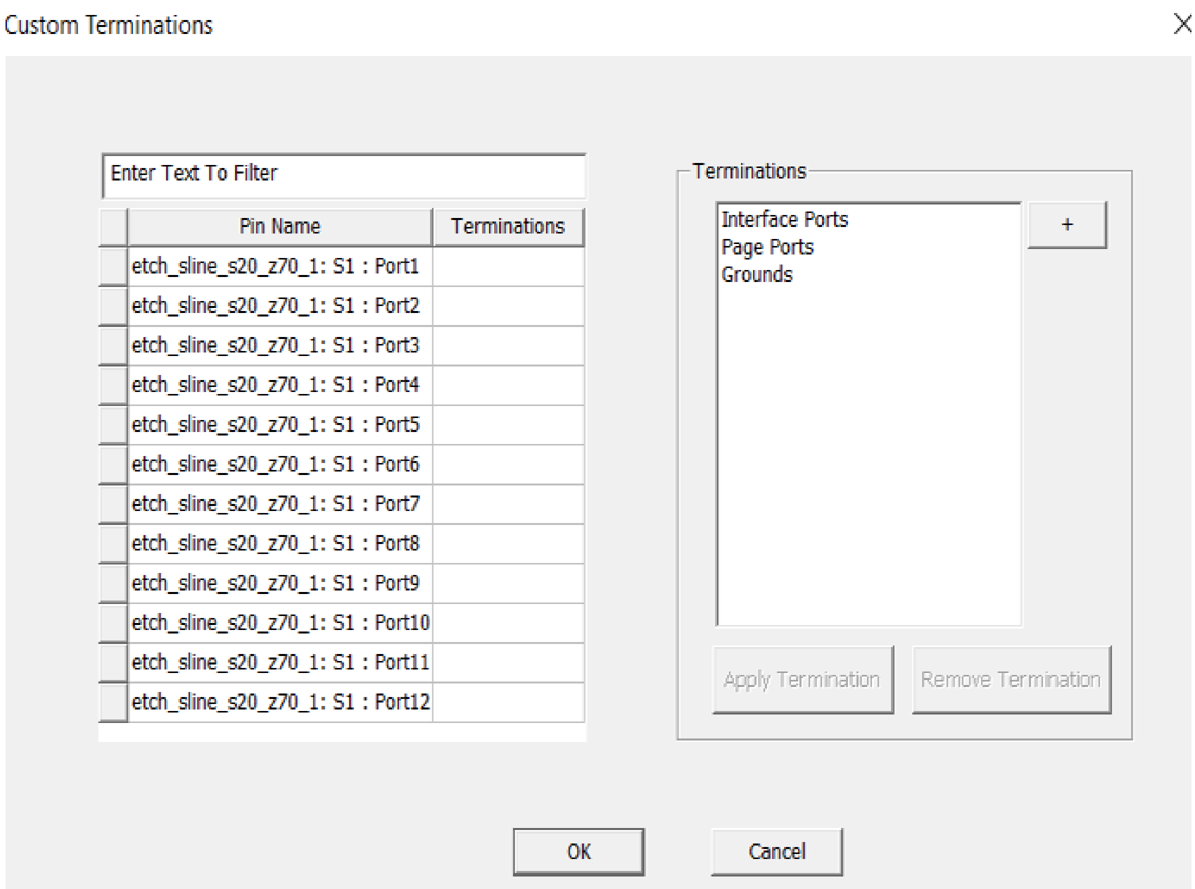
2. Choose the **Add Custom Terminations** option from the **Unconnected pins** drop-down menu in the ribbon.



3. Use the **Draw** menu: Select **Draw** > **Add at unconnected pins**> **Custom Terminations**.



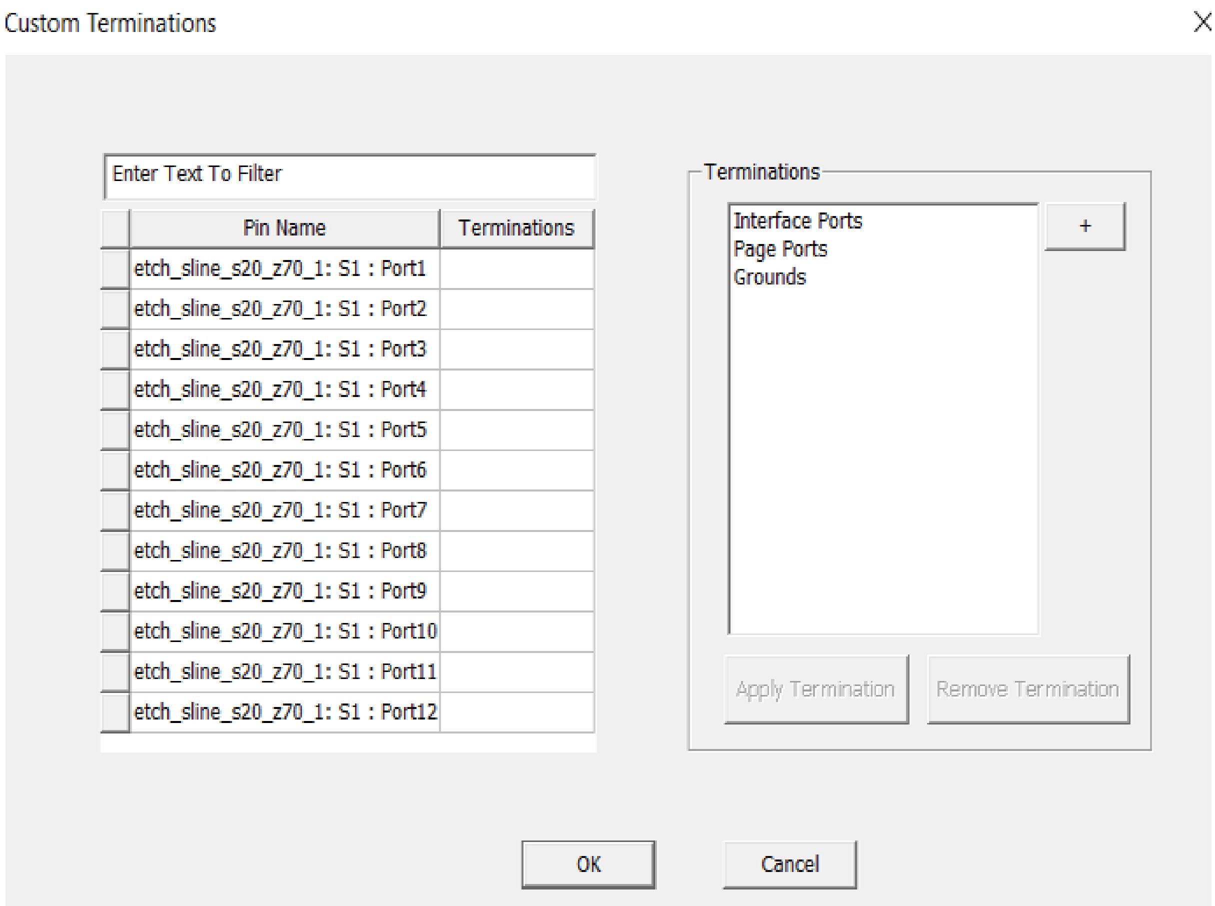
About the Custom Terminations Dialog:



- The custom terminations dialog supports different types of terminations in a single dialog.
- 1 port and 2 port terminations are supported.
- The components in the project are added to the dialog by default for the user to add them as terminations.
- The user can import any other components using the component browser to add them as terminations.

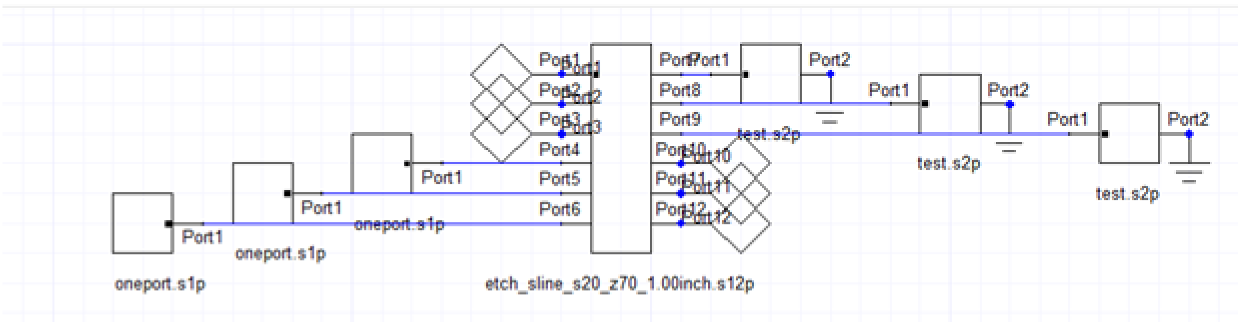
  Note: the components that the user can add are the same that show up in the Component Libraries > Components list. This includes both built-in components as well as user-defined components and Project Components.

Custom Terminations in Schematic:



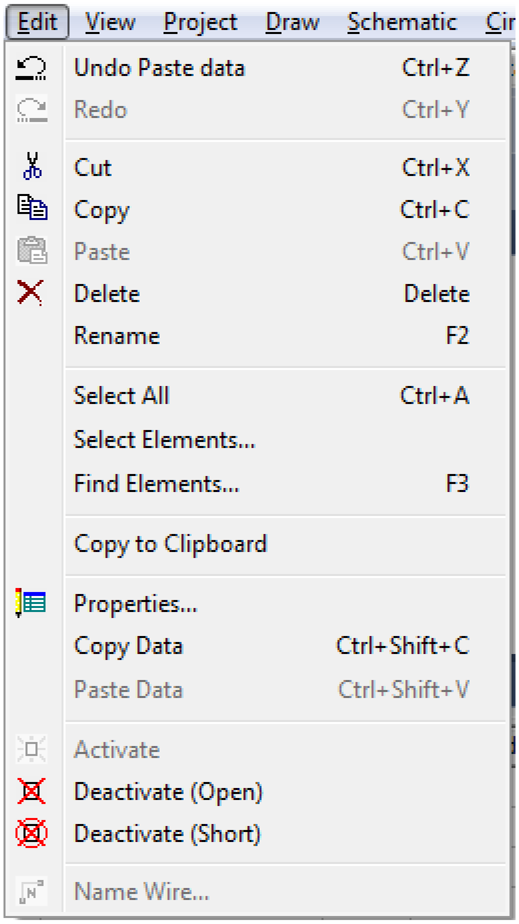
- Users can click on the + button in the dialog to bring up the component chooser dialog .
- Users can select multiple components from this dialog and select ok.

- The selected components will be added to the Terminations list in the custom terminations dialog.
- Users can apply terminations to pins by selecting one or more pins in the grid, selecting terminations they want to add, and clicking on the **Apply Termination** button.
- Users can remove or reset the applied termination in the grid by selecting the pin and clicking on **Remove Termination**.
- Once the terminations are selected, users can click **ok** to apply the selected terminations in the schematic.
- Adding pin connections is scriptable and supports undo/redo.
- Terminations chosen in the dialog will be connected to the corresponding unconnected pins by auto wiring in the schematic as shown below.



Editing Operations

When the **Schematic Editor** is the active window, clicking **Edit** on the menu accesses the following menu:



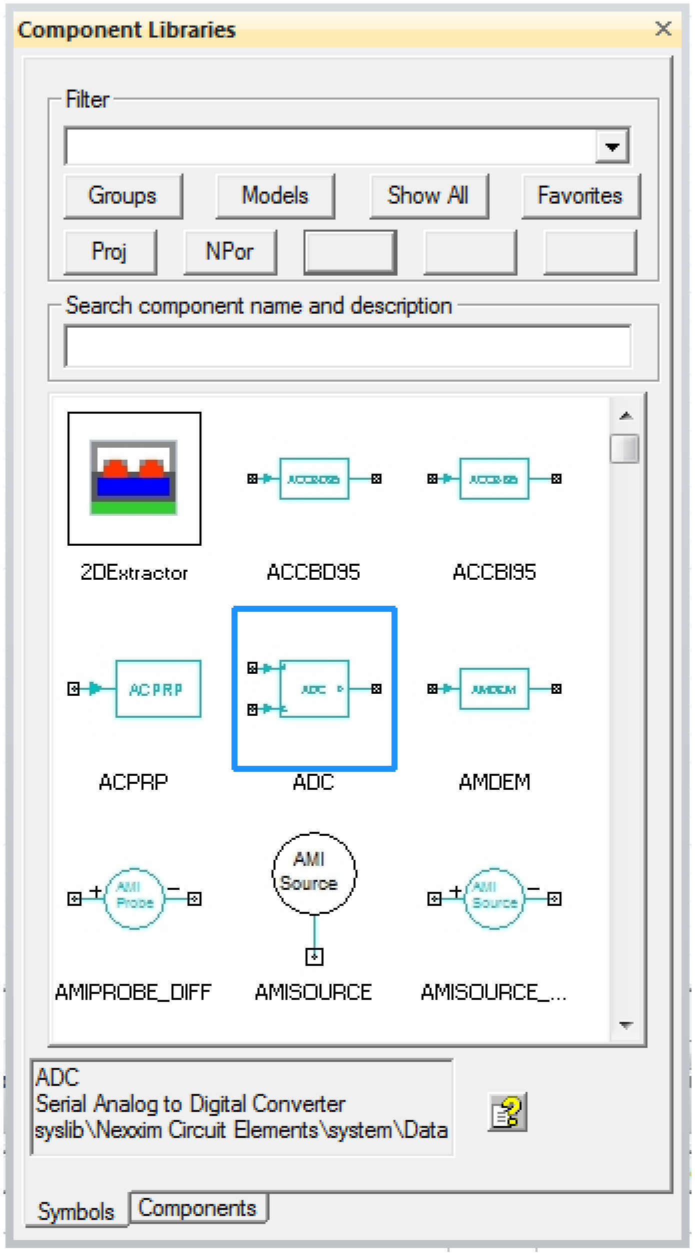
This menu contains the following Schematic Editor commands. The commands that are active depend on what is selected and on the previous command.

- **Undo** undoes the last operation (the last operation was changing the value of a property in the above picture, but the menu shows the last operation actually performed).
- **Redo** re-executes the last operation that was undone.
- **Cut** deletes the selected component or wire segment, and retains a copy for pasting into a schematic in the same application.
- **Copy** creates a local copy for pasting into a schematic in the same application.
- **Paste** puts the local object on the previous copy or cut into the schematic.
- **Delete** deletes the component.
- **Select All** selects all the components and nets in the schematic.
- **Select Elements** opens a window to select a component by name. See Selecting Components.
- **Find Elements** opens a window to find all elements that match the criteria you specify. See Finding Elements.

- **Copy to Clipboard** creates a global copy on the clipboard for pasting into a different application.
- **Properties** opens the **Properties** window for the selected element. See Displaying and Editing Component Properties.
- **Activate** restores a deactivated component to the circuit.
- **Deactivate (Open)** temporarily converts the component into an open circuit.
- **Deactivate (Short)** temporarily converts the component into a short circuit.
- **Name Wire** opens a window to assign a new name to the selected wire node.

Using the Component Libraries Window

Select the Symbols Tab of the Component Libraries Window to place components in the **Schematic Editor**. Components can be placed by selecting a component symbol , then clicking in the Schematic Editor. The Component Window can also be used to filter the components to be placed.

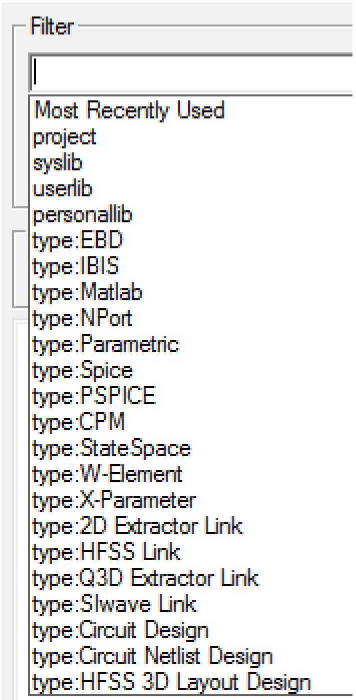


Filter Pane

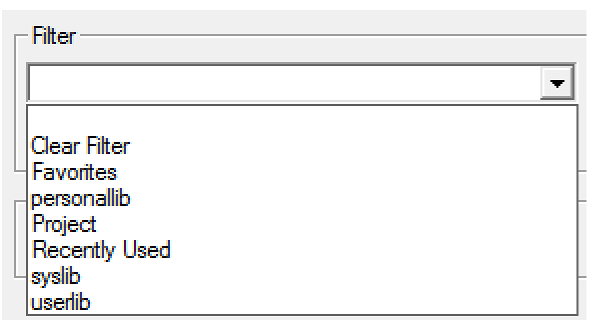
The Filter pane consists of two elements: a Filter Combo Box that contains a drop-down arrow, and the Filter Preset buttons beneath. Clicking the drop-down arrow of the Combo Box shows available filter choices.

The Filter pane allows you to filter and choose from components to be placed, either by using filters listed in the drop-down menu presets, or by typing text into the edit area of the Combo Box.

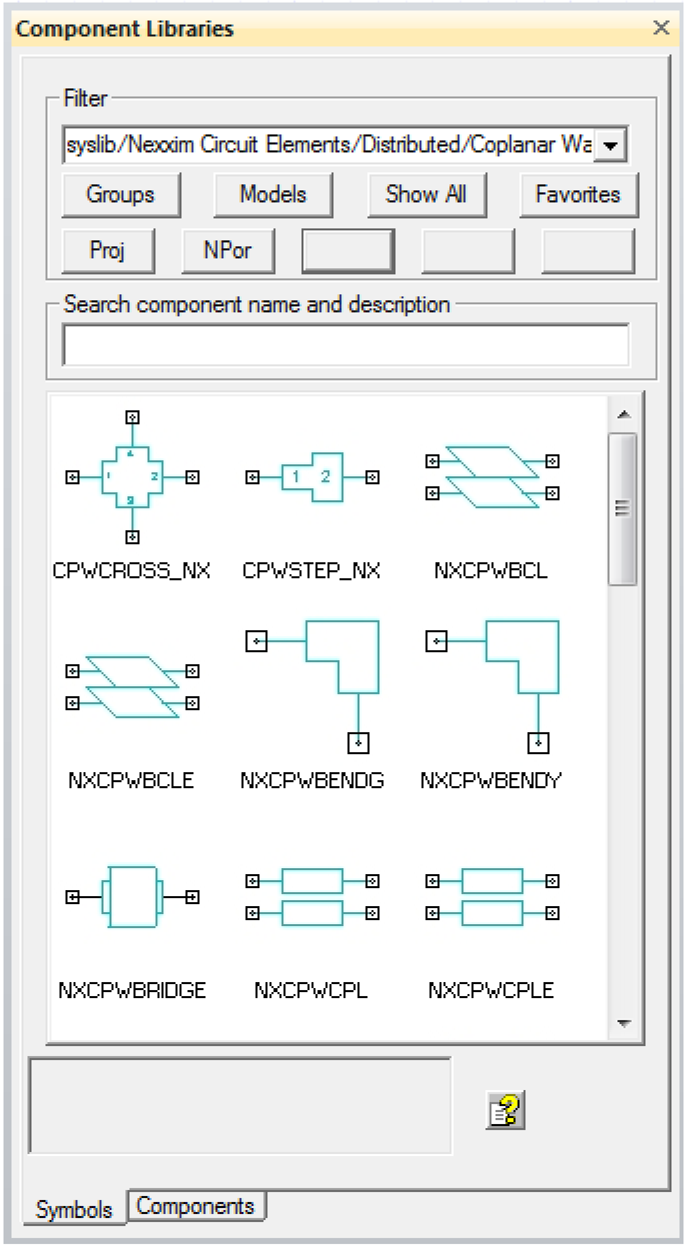
These filtering tools then show the names of components that match. For example, typing "Mi" into the edit area shows the following:



You can also click the drop-down selection list in the Filter pane and select one of the filter choices that are available:



- Choosing a filter on the list applies it to the symbols shown in the Listbox, and only components that satisfy the filter is shown. For example, choosing "Favorites" shows only the components that you explicitly marked as Favorite, and choosing "userlib" shows only those components that are in the userlib location on disk. For example, here is the Symbols tab display when the "syslib/Nexxim Circuit Elements/Distributed/Coplanar Waveguide" filter is chosen:



Types of Filters

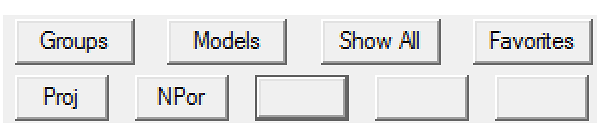
The following types of filters are available:

- A fully qualified library file name, e.g., syslib/Nexxim Circuit Elements/Distributed/Coplanar Waveguide/Transmission Lines. This shows only the components in that component library

- Any folder in a library (which in turn can contain other folders and component library files, e.g., syslib/Vendor Elements/vendors. This shows all components in all libraries in that folder.
- A component type. Supported types are:
  - — Spice
  - — IBIS
  - — EBD
  - — W-Element
  - — X-Parameter
  - — 2D Extractor Link
  - — HFSS Link
  - — Matlab
  - — NPort
  - — Parametric
  - — Q3D Extractor LInk
  - — SIwave Link
  - — StateSpace
  - — Circuit Design
  - — HFSS 3D Layout Design
  - — Circuit netlist Design
- A special category. Supported categories are:
  - — Project
  - — Favorites
  - — Recently Used

Filter Preset Buttons

The Filter Preset buttons are listed under the Combo Box:

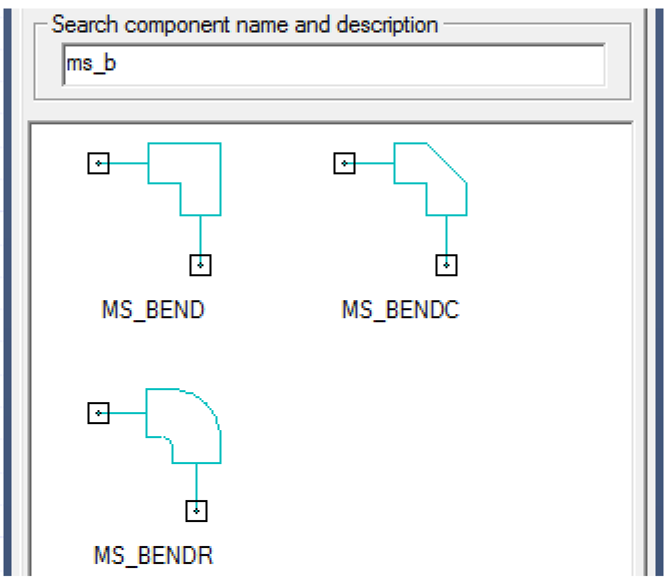


The top row of preset buttons are not user-configurable. They allow quick access to important filters. The Models button is particularly useful (See Selecting a Model to Import). The Groups filter button is described in Using Component Groups.

- The second line of filters are user-configurable and show the first four letters of the filter string. You can see the entire filter string by hovering the mouse pointer over a button.
- To configure a button, choose a filter string so it shows in the Combo Box, then click and hold for three seconds on a Preset button. When the text on the button changes, release the mouse button and the Preset is programmed with the filter string.
- The first two buttons come pre-programmed to show the Project and NPort filters, but change them to whatever you want.

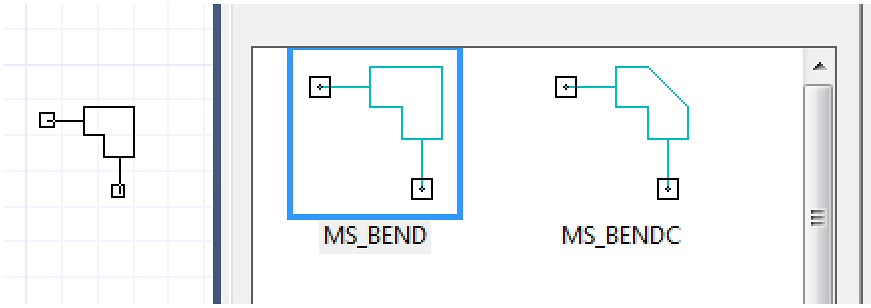
Search Pane

The Search box allows you to search for components within the filtered Components. Text entered into the search box matches component words in the component name or words in the component description. For example, filtering on "microstrip" and searching on "ms_b" shows the following:

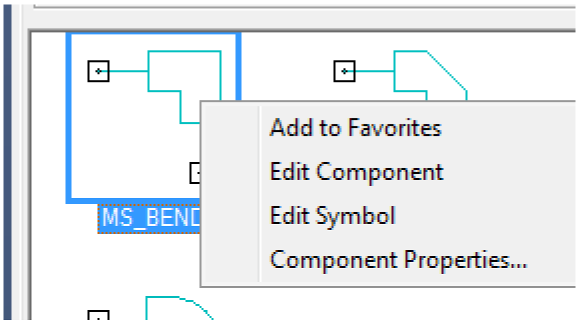


Listbox

The Listbox is located beneath the Search Pane and shows the symbols that correspond with each matching component, along with the component ID string. A blue selection-rectangle appears when you hover over a component in the Listbox. Clicking on a component in the listbox selects it and sets up for placing the component in the active editor (schematic or layout). Click to select a component, then click to place it in the active editor. The blue selection-rectangle disappears when the component placement is completed.



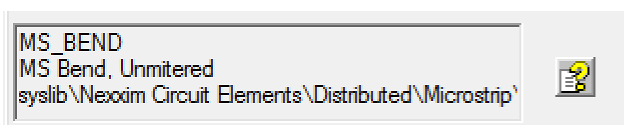
To add or remove a component on the Favorites list, right-click the component in the listbox.



Right-click the component in the listbox to show its drop-down menu. These commands allow you to add (or remove) a component from Favorites, edit the component or its symbol, and quick access to the component's property window.

Information Area

Under the Listbox is the Information Area. Hovering over a component in the Listbox displays its information in this area.

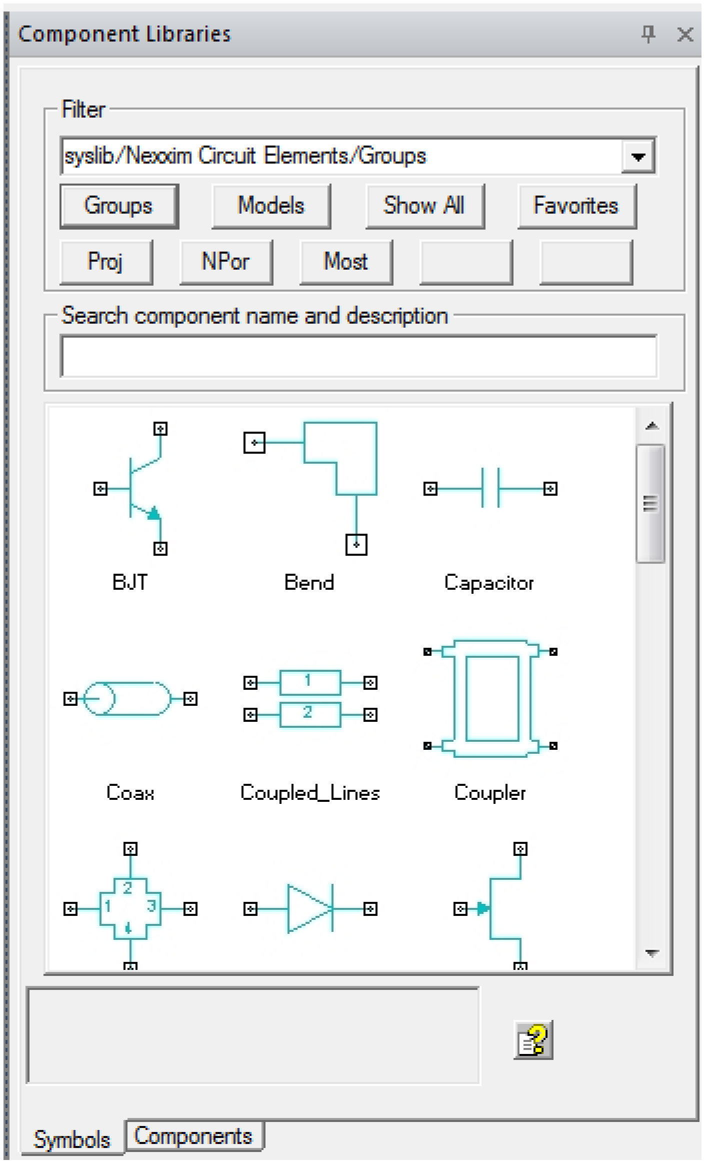


The Information Area shows the ID name of the component, the path to its library, and any text that was entered into the Description field for the component — all searchable using the Search Box as described above. Clicking **Info** opens **Electronics Desktop** Help and show the information page for the selected component.

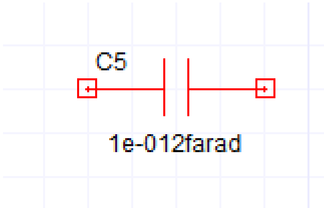
For more information, see Selecting a Model to Import.

Using Component Groups

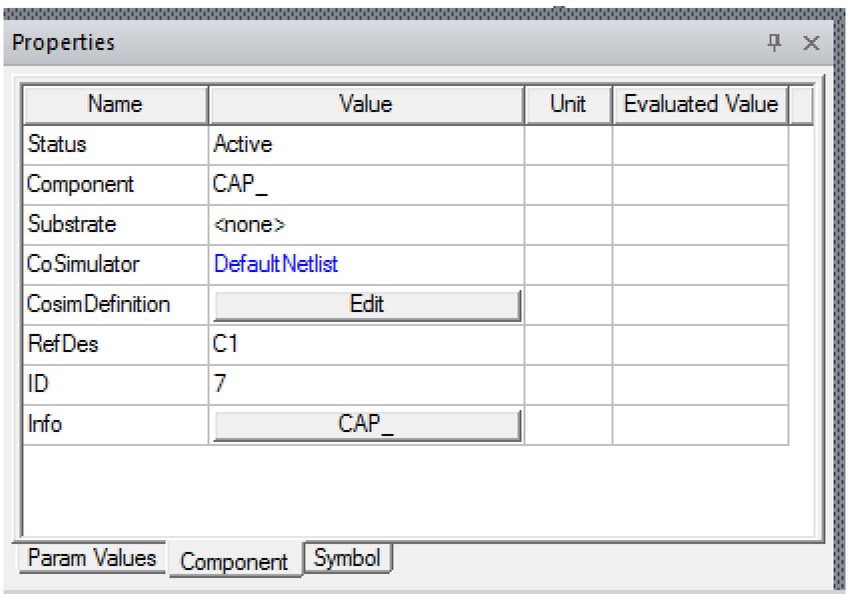
Click **Groups** in the **Component Libraries** window shows one component choice for each category of component (e.g., resistor, TRL, source).



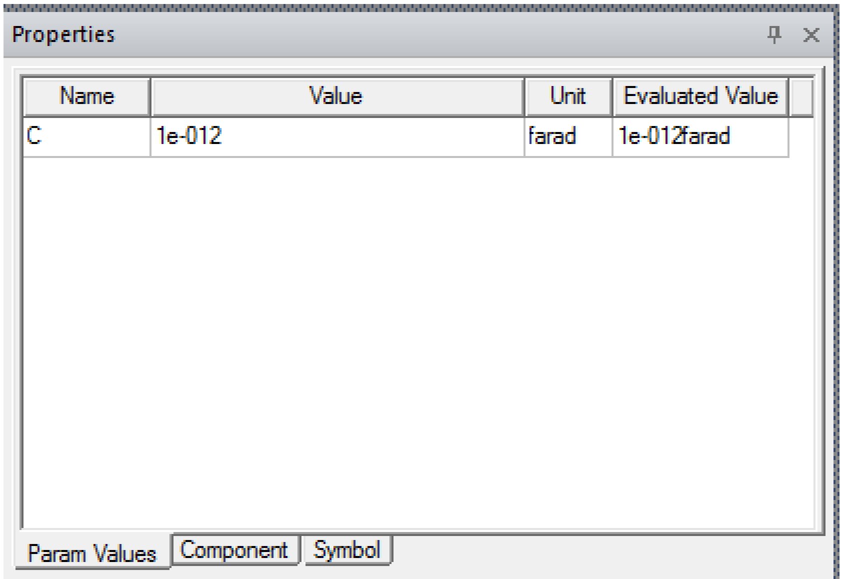
Choosing a group component and clicking in the schematic places the top choice in that group. For example, here is a capacitor placed in the schematic on the **Groups** filter.



Here is its Properties window showing the Param Values tab. Parameter values are netlisted and tell the solver what to do.

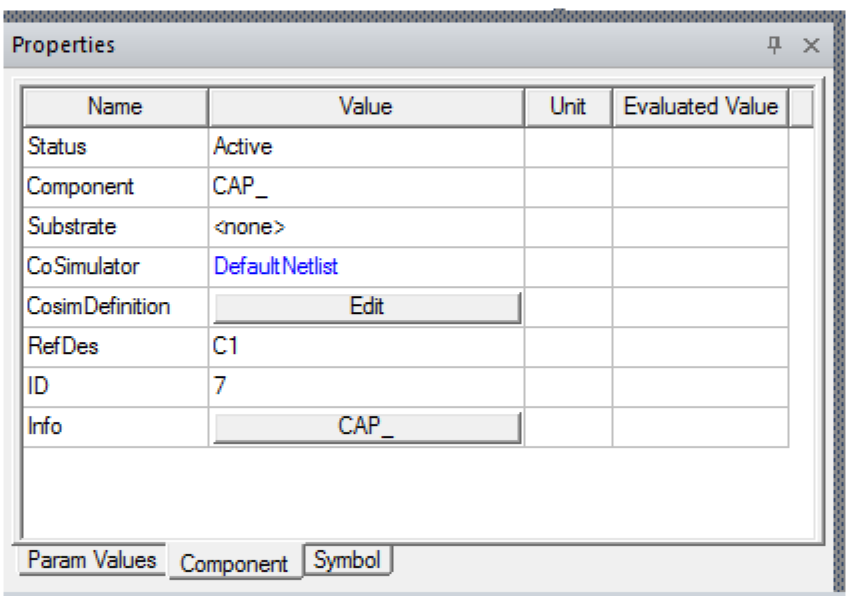


Name	Value	Unit	Evaluated Value
Status	Active		
Component	CAP_		
Substrate	<none>		
CoSimulator	Default Netlist		
Cosim Definition	Edit		
RefDes	C1		
ID	7		
Info	CAP_		



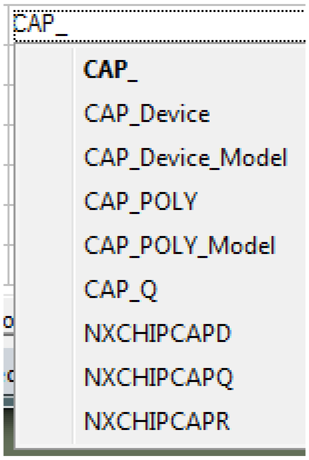
Component Tab

Switching to the Component tab, the following Properties that are not parameters appears. But the Properties control many features of this Component.



- **InstanceName** — The customizable identifier for this Component Instance; it must be unique in the Circuit Design.
- **Status** — A menu with three choices: Active, Inactive Open, and Inactive Short (See Editing Operations).

- **Component** — A menu that has other choices for the Capacitor Group and the <none> substrate; in this case it shows nine choices.



Substrate — A menu showing the substrates defined for this Circuit Design plus the New command. Choosing the New command opens the Substrate Stackup window to create a new substrate for this CompInstance.

Cosimulator — Chooses which Cosimulator to use (See Selecting a Model for an Individual Component).

CosimDefinition – Shows the information needed by the current Cosimulator (See Component CosimDefinition Property).

RefDes — This read-only identifier is unique in any hierarchy.

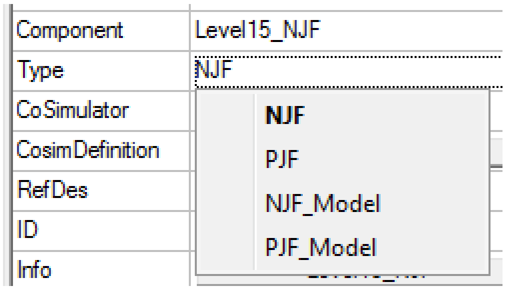
ID — This read-only identifier is unique in this Circuit Design.

Info — click to see the Help for the current Component.

The choices shown in the Component drop-down menu are controlled by the Group (the Resistor, Substrate property, and an optional third property that depends on the Group). For example, here are the CompInstance tab properties for a Coupled Line.

Component	MS_MCPL04		
Substrate	MS:MS_TOP		
Numlines	4		

The Component, MS_MCPL04, is for microstrip substrates only and also shows 4 coupled lines because of the setting of the Numlines property. Editing the Numlines property and entering “8” specifies 8 lines and therefore change the Component to MS_MCPL08. Other Component Groups, like FET and source have an additional property named “Type”. Here is the Type property choices for FET.

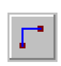


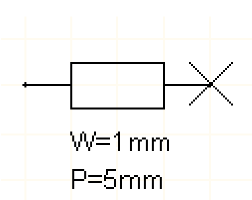
Note:

1. The order of entries in the Component menu changes as you place more components of that type, such that the most-placed components are always shown at the top of the list.
2. If there are too many Component choices to list, the last menu command is "**more…** ". Clicking "**more…** " opens a list box that shows all the choices. Clicking an entry in the listbox chooses that component.
3. When switching between components, the Param Values tab can show different parameters for each Component. If you override a parameter and switch to another component, the override is applied to the new Component choice. Switching back to a previous Component, you finds that overrides have not changed.

Wiring Components

To start drawing a wire, do one of the following:

- From the **Electronics Desktop** (**Electronics Desktop**) **Draw** menu, select **Wire**.
- Press **Ctrl** +**W** on your keyboard.
- Click the **Electronics Desktop Wire** icon .
- Move the mouse cursor over a component pin to display the cross-hairs wiring cursor:



To draw connections:

- Click at the point where you want the wire to start, and move the cursor to extend the wire.
- As you move the cursor, click wherever you want to turn a corner.
- All connection points are highlighted in red.

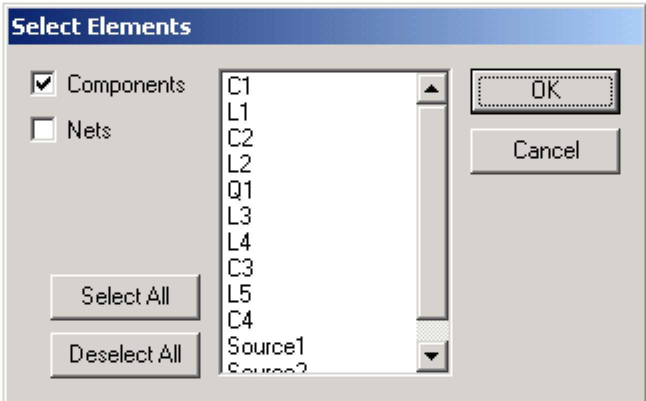
To stop drawing a wire, do one of the following:

- Click, if the wire ends at a port or component pin.
- Double-click, if the wire does not end at a port or component pin.
- Press the **Spacebar** on your keyboard to terminate the wire operation and leave the net unselected.
- Press **Enter** on your keyboard to terminate the wire operation and leave the net selected.
- Right-click, then select **Place and Finish**.

Selecting Components

- To select a single element (component or wire segment) in the **Schematic Editor**, place the cursor over the element and click to select it. Click again anywhere in the schematic window to deselect the object.
- To select multiple elements, hold down **Ctrl** while clicking the elements. To deselect one of the multiple elements, hold down **Ctrl** and click the element. To deselect all selected elements, click anywhere in the schematic window.
- Also select multiple elements by holding down **Ctrl**, then either click an object to add it to the selection or click-drag a rectangle to enclose objects. To remove items on the selection, hold down **Ctrl** and click an object that is selected.
- When you move a selected component, any wiring normally stays attached to the component. To move a component and break its connections, hold down **Shift** +**Ctrl**, then click the component and drag it to its new location.
- To select all connected segments of a wire, hold down **Shift** +**Ctrl**, then click any segment of the wire. Click an drag to move the wire segments to a new location, breaking any connections.
- If you have difficulty selecting an object, you can select it on the **Select Elements** window:

1. From the **Edit** menu, select **Select Elements**. This opens the **Select Elements** window.



2. In the **Select Elements** window, click the components you want to select in the **Schematic Editor**. You can select/deselect the **Components** or **Nets** group box to vary the list of elements to choose from. You may also click **Select All** or **Deselect All**.

Selecting a Wire

Although you may draw a wire that includes bends in a single operation, the **Schematic Editor** treats each portion of a wire between bends as a separate segment during selection operations.

- To select a single wire segment, click it.
- To select all segments of a wire, hold **Shift** and **Ctrl** while clicking any of its segments.
- To select multiple segments of a wire, contiguously or otherwise, hold **Ctrl** while clicking each segment in turn.
- To cancel a selection, click outside it in the schematic window.

Displaying Wire Properties

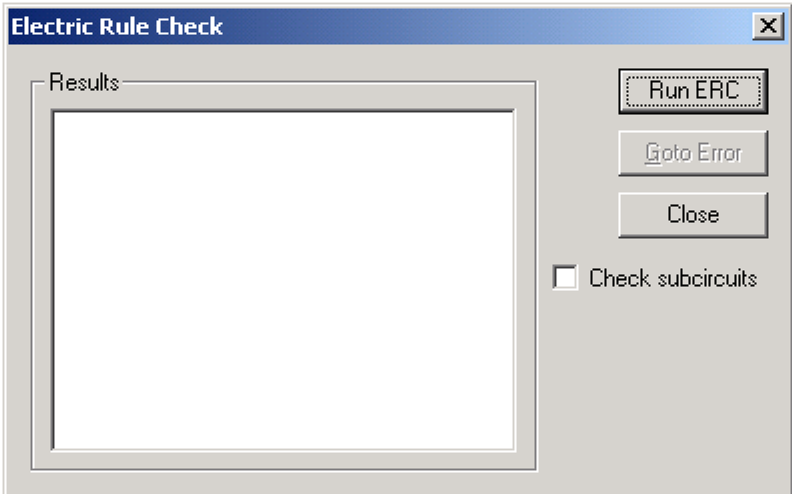
Double-click a wire to open its **Properties** window.

- **NetName** and **PinCount** are two wire properties of particular interest. **NetName** is the name of the unique interconnecting node, or *net*, a wire represents. NetNames correspond to the node names shown in the schematic's corresponding circuit file, or *netlist*.
- **PinCount** reflects how many component pins a wire interconnects, with one exception: Port connections are not included in PinCount values. A wire connecting a port and three component pins therefore has a PinCount of 3.

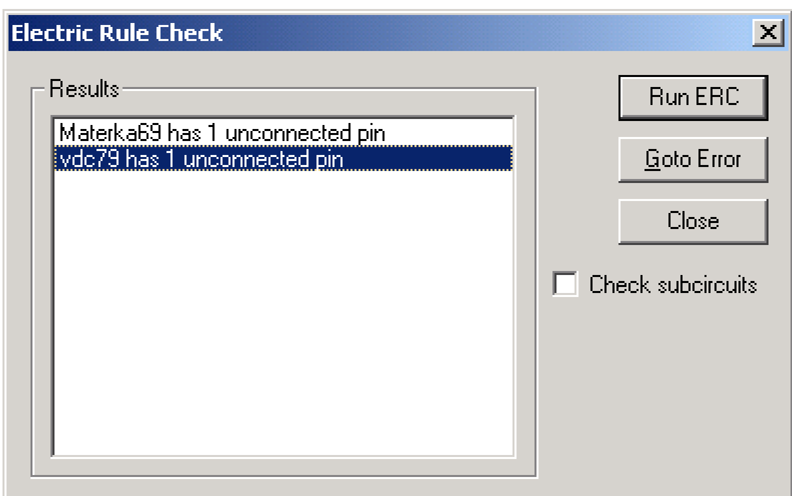
Checking Connectivity

The Electric Rule Check (ERC) feature checks the circuit for valid connectivity. ERC automatically conducts rule checking for ports, connections, and components of the active schematic.

1. To test for connectivity, select **Schematic > Electric Rule Check**, which opens the following window:



2. Select **Check subcircuits** to run the electric rule check on subcircuits of the active schematic display
3. Click **Run ERC** to begin the error check
4. If an error is displayed in the **Results** window double-click the error message or select the message and click **Goto Error** to go directly to the object in the **Schematic Editor** that caused the error.



Possible causes of an **Electric Rule Check** error include:

- **Unconnected Pins** – A component, or port, with a pin that is not connected to anything else.

- **Overlapping Components** – A component that completely overlaps another such that the two components appear as one component. (Often caused by accidentally clicking twice when placing a component.)
- **Nets with Multiple Output Pins** – A net which has more than one output pin connected to it. (This is rare since most component pins are labeled as input and output.)

Using Buses

The following topics describe various operations using buses.

Nets, Buses, and Bundles

A net or node is a single wire segment or a connected set of wire segments. A node name can contain any alphanumeric characters except the space ( ), ampersand (&), and asterisk (*). For more information, see Wiring Components.

Buses and bundles are ways to name multiple related wires, pins and ports conveniently. This topic presents the rules for using bus and bundle wire names, and for the separation of sets of connected wire segments into different wires based on different names.

Bus Format

A bus is a collection of schematic wires that are indexes of a base name (e.g., Data[0-31]). A bus is a schematic concept only. In the circuit and layout, all signals are individual.

A bus name consists of :

- A base name. A base name can contain any alphanumeric characters except the space ( ), ampersand (&), and asterisk (*).
- A square or angle open bracket (‘[‘, ‘<’]
- A number, a range of numbers specified with a hyphen or colon (*n1***-***n2* or *n1***:** *n2*), or a comma-separated list of numbers or ranges.
- A square or angle close bracket

Following are two examples:

```
DATA[1:5] \\ A bus with five signals
inputbus[1, 3-5, 11-22] \\ A bus with 15 signals
```

Bundle Format

A bundle is a collection of schematic wires including individual wires and buses (e.g., A,B,C[7-0]). A bundle is a schematic concept only. In the circuit and layout, all signals are individual.

A bundle name consists of a comma-separated list of single wire names and/or bus names. For example:

```
DATA[1-5],node5
```

Creating a Bus or Bundle

To create a bus or a bundle, select **Draw > Wire** to first create a wire, then change the wire's name to specify more than one signal, in either bus or bundle format. The width of the wire changes to a wide appearance when the number of signals is more than one.

- When a bus or bundle is copied and pasted, its name is lost and it becomes a single-signal wire.
- Drawing a wire by starting at a bus or bundle vertex extends/adds segments to the bus or bundle.
- Drawing a wire by ending on a bus or bundle extends/adds segments to the bus/bundle, unless the wire started at a named wire, bus or bundle. If it started at a named wire, that named wire is extended, and it connects to the bus or bundle if the bus or bundle contains the signal(s) in the named wire.

Deleting Wire Segments from a Bus or Bundle

Deleting one or more wire segments results in the remaining segments of the wire being assembled into connected sets (based on physical and port connections) with each unique set being an independent wire. Only one of the sets retains the original name. The others become single-signal wires with automatically assigned names.

Naming a Wire in a Bus or Bundle

Selecting one or more connected segments of an existing wire and assigning a name results in the following:

- The wire is broken into sets as if the newly named segment(s) had been deleted. One set retains the original name. The selected segments become a new wire and are assigned the new name. The new name spreads to connected segments that are automatically named, or that have fewer signals than the new name, and those segments is added to the new wire. Several different wires may result from this operation, each having a different name.
- Thus, renaming a segment of A[0-3] to A[7-0] renames the entire wire. Renaming a segment of bus B[31-0] to B[0-63] with a sub-bus connected to it, B[16-31], renames the connected segments of B[0-31] but does not rename the sub-bus, since it is a separate wire. Renaming the end segment of wire A[0-31] to A[0-15] makes that end segment into a sub-bus that is a different wire than A[0-31].
- If a wire with one or more pageports attached is renamed to a different width, the wire follows naming rules, and any pageports physically attached is also renamed.

Naming a Pin or Port on a Bus or Bundle

- The name of a pin or port can be a simple signal name, or can specify multiple signals through bus or bundle format.
- Changing the name of a port changes the name of attached wires.

Connections using Buses and Bundles

- Connections between wires, between pins and ports, and between wires and pins/ports depend on widths and names.
- Wire to wire connections are made if one wire contains the signals of the other. Thus, a wire A[0-7] has 4 signals electrically connected to physically connected wire A[7-4], namely A[4], A[5], A[6] and A[7]. A bundle A,B,C has one signal electrically connected to physically connected wire A. A wire A,B,C,D has two signals connected to physically connected wire C,D,E,F.
- Wire to pin or port connections are determined by the number of signals in the wire and pin/port. If the numbers of signals are the same, electrical connections are made in order. If the widths are different, there is no connection.
- Pin/port to pin/port connections are identical to wire to pin/port connections. If the widths are the same, all signals are connected in order. If the widths are not the same, none are connected.
- Changing the name of a wire that is attached to a pin, a port, or another wire may cause connections to be made or broken, based on the considerations above.

Rippers

A ripper is a two-pin schematic connectivity object, primarily for display and compatibility with third party schematic concepts (e.g., Mentor). A ripper can connect to any bus or bundle on one side, and a sub-bus or bundle on the other side. It functions as an extension of the bus or bundle, generally at a 45 degree angle.

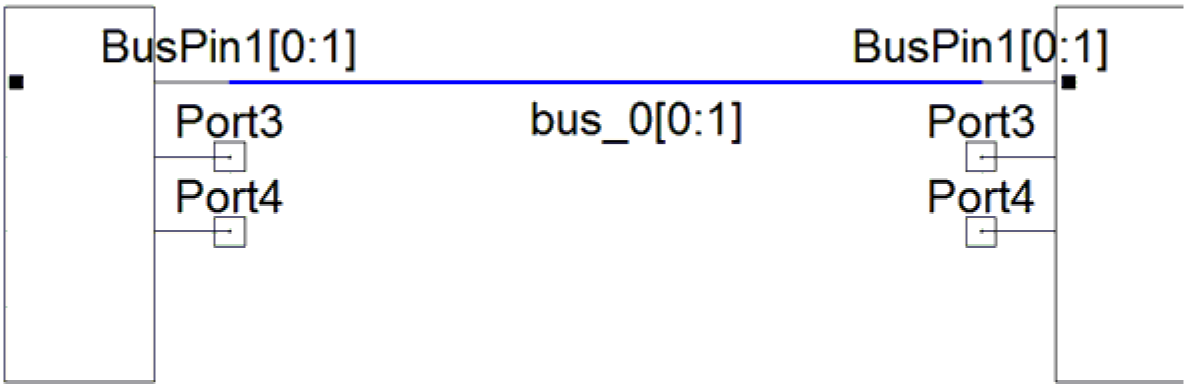
Net and Bus Wiring

This topic describes the net and bus wiring that results from connecting bus wires, including auto naming, pins of different width, and angled wires.

Connecting Bus Wires

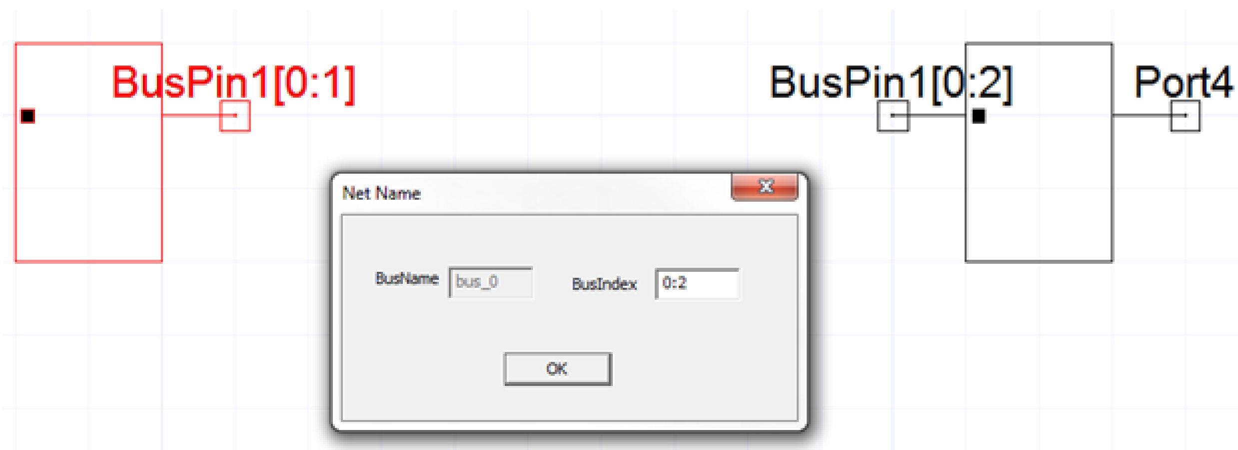
A wire can be drawn from a Bus pin to a pin of any width.

- When a wire is drawn between two bus pins of equal width, the wire is auto-named as shown in the following figure.

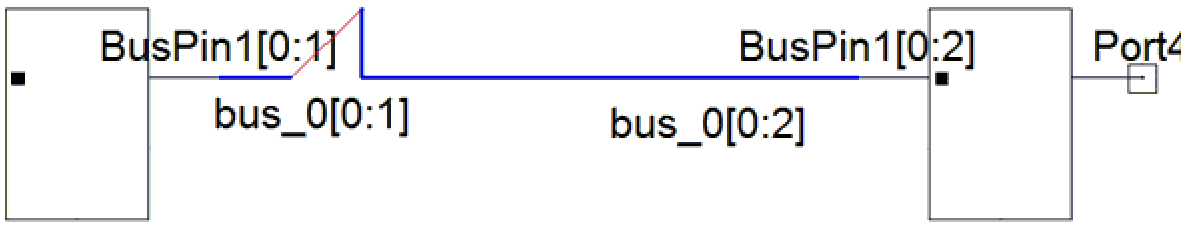


- When a wire is drawn between pins of different width, a bus entry is automatically added between the pins. A message box is popped up for the user to rename the name and index of the new net. The busIndex can be a singleIndex if it is a single width pin or startIndex:stopIndex for a Buspin. The index should be between 0 and one less than bus width.

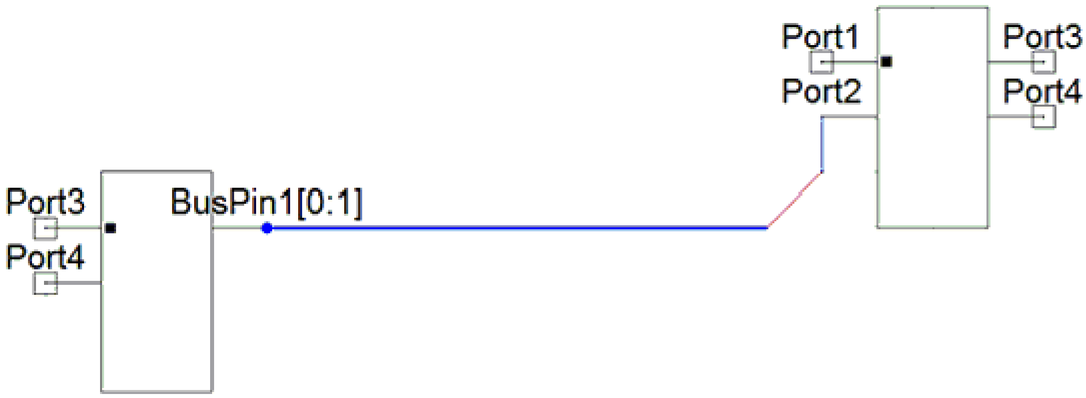
In the following figure, a wire is drawn between two pins of unequal width. A message box is opened for the user to rename the index of the new bus net.



Once the name and index of the bus net is set, the wire is drawn as shown in the following diagram.



- When a wire is drawn with an angle in it, the bus entry is placed at that angle.The bus entry is drawn at the corresponding angle as shown in the following diagram. In the figure, wire is drawn from BusPin1[0:1] to Port2.

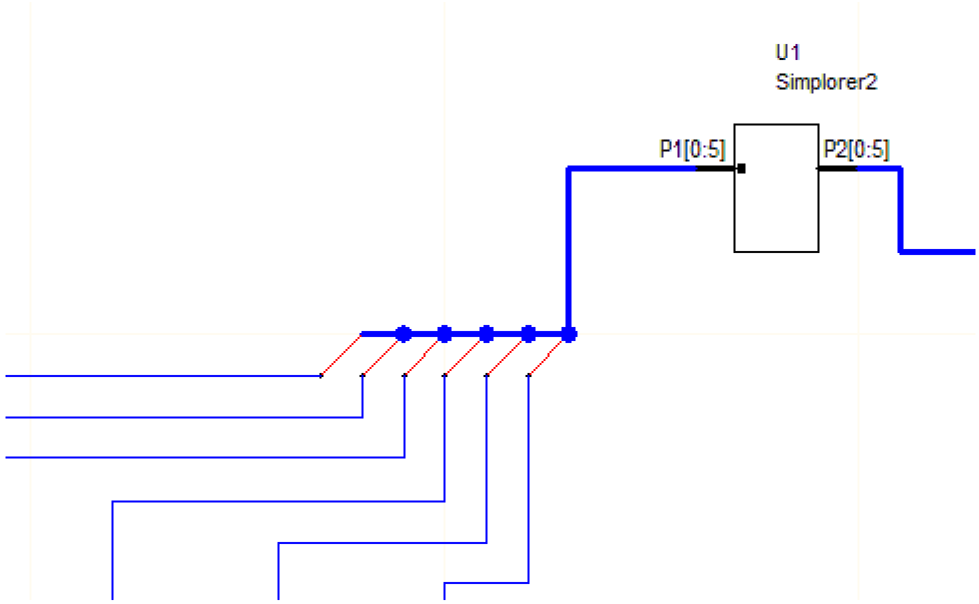


- Under some circumstances, the bus entry cannot be placed because of overlap due to space constraints. In this case, you should either manually add the bus entry and connect it or move components around to create more space.

Drawing Bus Entry Objects

The **Draw>Bus Entry** object menu command provides the user a way to add visual representations of connections from a bus to an individual net or a sub-bus.

On a schematic, a bus may be shown with individual nets (wire) or with wires where the bus width is greater than 1 (drawn with a thick blue line). The **Bus Entry** object provides a mechanism for visually representing individual wires connected to a bus consisting of multiple nets as shown in the following example. In this example, the P1[0:5] bus is visually “connected” to individual wires via the bus entry objects (the diagonal red lines).



Note:

- **Bus Entry** objects provide only a visual representation of connections on the schematic - not actual connections. The user still must ensure that connectivity is maintained by properly naming the wire segments drawn out of a bus.
- Because they are merely visual representations of connections, **Bus Entry** objects have no significance or effect on the actual circuit, and thus do not appear in the net list or in any results.
- **Bus Entry** objects can be used to show connections to nets of any width and any domain.
- Both ends can only be attached to nets and cannot be directly attached to other pins or ports.
- If the wires on each end of the Bus Entry are not correctly named to show that they are connected, then Disconnects are drawn.

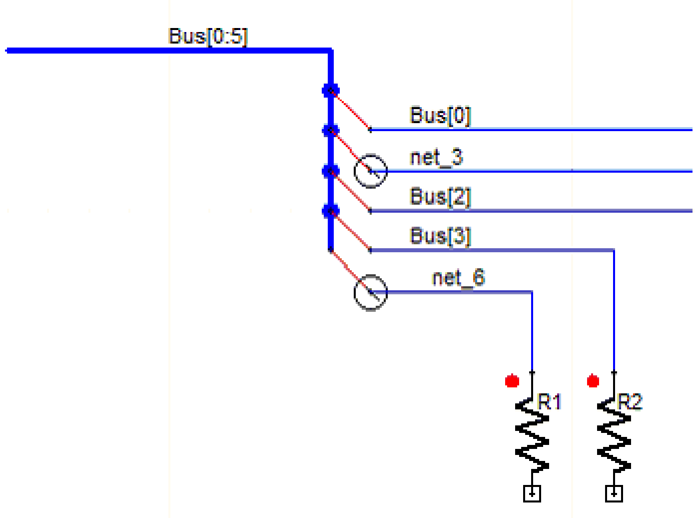
To add bus entry objects to an existing schematic:

1. Add a bus to the schematic by drawing a wire and naming it as a bus. For example: **DataBus[0:2]**.
2. Place a **Bus Entry** on the Draw menu such that one end intersects the bus.
3. Draw another wire on the other end of the Bus Entry object.
4. Rename this new wire as **DataBus[0]** to use that element of the bus as the individual net.
5. Repeat as needed for the remaining bus elements.

Disconnects

A disconnect is a circular visual indicator at the pin of a bus entry object. Its presence indicates that two nets – *though visually connected through the bus entry object* – are actually disconnected as indicated by the differing names.

If a bus entry has both pins connected to two different nets (which can be of different sizes), and if neither of the named nets are a subset of the other named net, a disconnect circle is drawn.



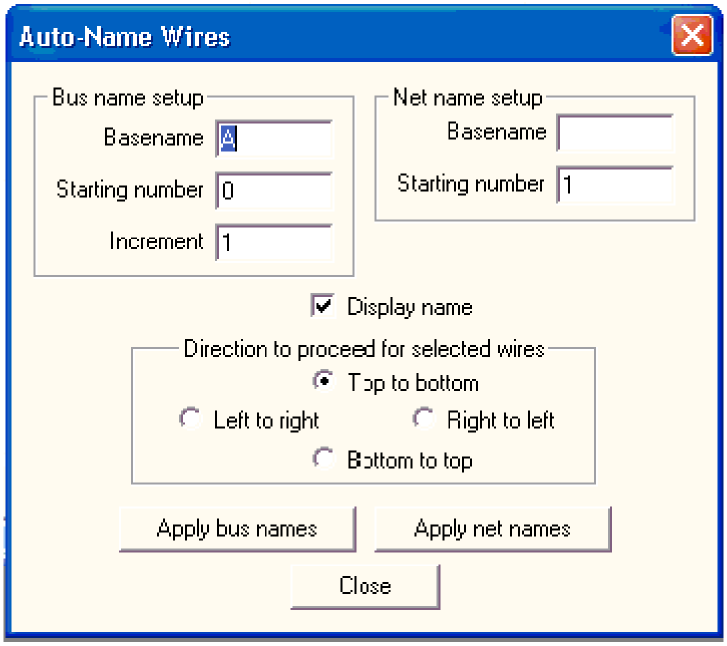
The preceding figure,shows a bus named Bus[0:5], which has five nets that are branched out through bus entry objects. Three nets are named properly as: Bus[0], Bus[2] and Bus [3]. However two nets: net_3 and net_6 are not connected to the bus by name – and so are shown to be drawn with circular disconnect-objects.

Net and Bus Auto-Naming

Selected wires can be named as a group with either bus-style names or net-style names (A[0], W1) using the **Auto-Name Wires** window.

To name wires as a group:

1. In the **Schematic Editor**, select the wires to be auto-named.
2. Click **Schematic** in the menu and select **Auto-Name Wires** to open the **Auto-Name Wires** window.



- Adjust the **Bus name setup** or **Net name setup**, depending on which naming style you want to use.
- Toggle **Display name** off if you do not want the wire names to display on the schematic.
- Select the direction for applying the names to the selected wires.
- Click **Apply bus names** and **Apply net names** to set the names of the selected wires.

Click **Close** to close the window.

Cross-Probing Elements

When both the Layout Editor and the **Schematic Editor** are displaying the same design, select one or more component instances and interface ports in both editors. Select the schematic elements or layout elements to be cross-probed. Then use one of the following methods to effect the cross-selection:

- In either editor, press **Ctrl** +**K**.
- In the **Schematic Editor**, select **Schematic > Cross-Probe Layout**.
- In the Layout editor, select **Layout > Cross-Probe Schematic**.

The corresponding elements in the related editor is selected and that editor window is brought to the front. If you select only component instances or only interface ports, then after cross-probing the primary selection in the first editor is also the primary selection in the related editor. Note that all previous selections in the related editor is cleared and only the cross-probed elements is selected when the command is invoked.

Note:

Changes to a design should be made on the editor that was used to create the design. For example, if a design was created using the **Schematic Editor**, it should not be modified on the layout editor. In some cases, attempts to change a schematic design on the **Layout Editor** results in a warning such as:

```
[warning] Port Port2, ID %2, was disconnected from net 98, but
it was not possible to unwire it. This must be done manually.
```

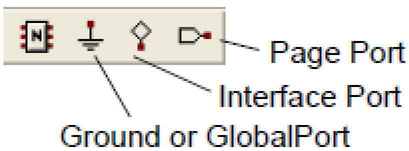
Similarly, a **Layout Editor** design can be viewed as a schematic using the cross-probing feature, but the schematic view should not be used to make changes to the design. Although an operation such as deleting a port in the schematic may successfully delete the cross-probed layout port, the operation cannot be undone in the **Layout Editor** after the schematic window has been closed (the undo attempt may cause an error).

Ports in Schematics

The following types of ports are available in the schematic editor: *interface*, *global*, *ground* and *page*.

- An **interface port** serves as a connector into or out of a given design, and may contain termination and signal source definitions. The names of interface ports must be unique within a given design but may be duplicated from design to design.
- A **global port** serves as a means of common electrical connection—in effect, a connection to a bus—within a given design, across hierarchical levels. You can define one or more circuit nodes to connect to global ports that cross hierarchy and schematic page boundaries. Within a design, all global ports with the same name are treated as if they are connected together electrically.
- A **ground port** is a special case of a global port. All ground ports connect to the reference or ground node (node 0 in Circuit solver designs).
- A **page port** serves as a named connection to a signal that is common to two or more pages of the schematic.

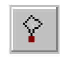
Commands to create ports are available from both the **Schematic** drop-down menu and from toolbar icons:



Interface Ports

An interface port serves as a named connector, whether the port is currently within a design or not. The interface port parameters specify the port name, its impedance, a reference node, and one or more power/current/voltage source specifications.

To place an interface port, select the schematic that contains it, then do one of the following:

- From the **Draw** menu, select **Interface Port**.
- From the **Schematic Draw** toolbar, click the **Interface Port** icon  .

The cursor, which is now associated with an interface port symbol for placement, moves to the center of the schematic window. To place the port, click at the appropriate location. (If **Multiple Placement** is activated for interface ports in the Schematic Options window, place additional ports by clicking at additional locations.)

To stop placing interface ports, do one of the following:

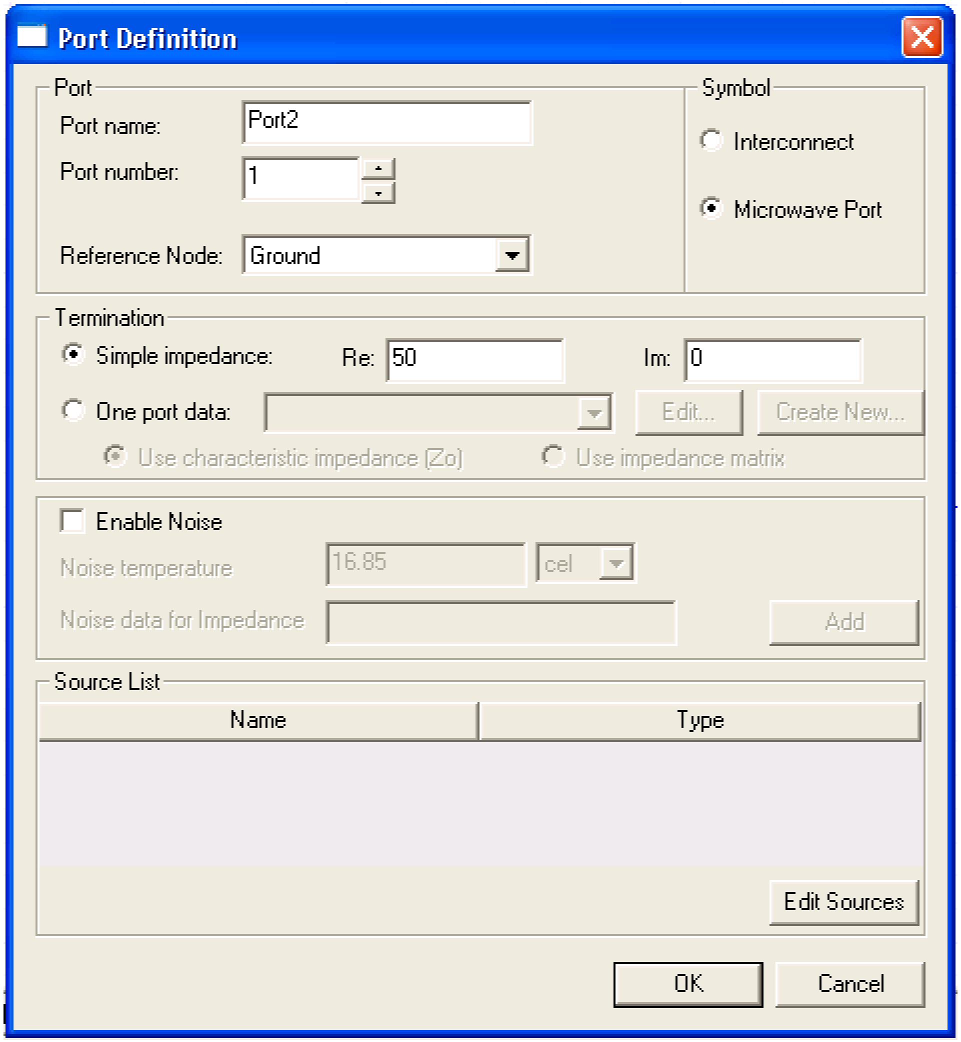
- Press **Enter**, the **Spacebar**, or **Esc** on your keyboard.
- Right-click, then select **Place and Finish**, **Finish**, or **Cancel**.

Hint You can rotate an interface port before placing it by repeatedly pressing **R** on your keyboard. Each press rotates the port 90º counterclockwise.

> **Note:**
>
> To ensure electrical connectivity among schematic elements, the pins of placed interface ports snap to a 100-mil (2.54-millimeter) grid. This snapping cannot be turned off, and the spacing of the connectivity grid cannot be adjusted.

Editing an Interface Port Definition

To edit the definition of an interface port, right-click the port symbol and select Edit Port on the menu. The Port Definition window opens:



Port Name and Number

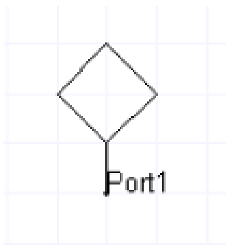
Click the **Port name** and **Port number** fields to rename and renumber the port as appropriate. The node to which the port is attached changes to match the specified port name. Click the **Reference Node** drop-down menu to select on the existing ports the one to be used as a common reference node for all sources attached to this port; the default is **Ground**.

Interface Port Symbol

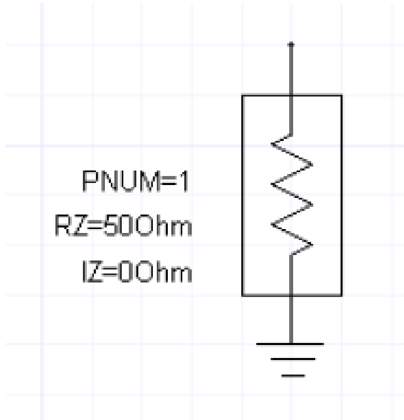
The default symbol for an interface port is a diamond. Optionally, choose a more complex symbol to represent a microwave port:

- To choose a new **Microwave** port symbol, select **Microwave** on the **Symbols** group box of the **Port Definition** window. The symbol indicates when a source is enabled for that port.
- To restore the default symbol, click **Interconnect**.

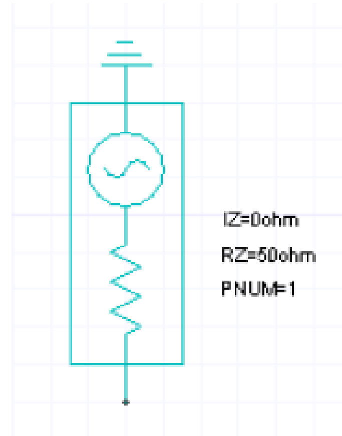
The definition of the port remains the same for both symbols.



Interconnect Port Symbol



Microwave Port Symbol



Microwave Port with Source

Interface Port Termination

Click the **Termination** group box to control the termination to the port.

- With **Simple termination** selected, set the real (**Re**) and imaginary (**Im**) parts of the port impedance. The default impedance is 50 ohms for the real part and 0 ohms for the imaginary part.

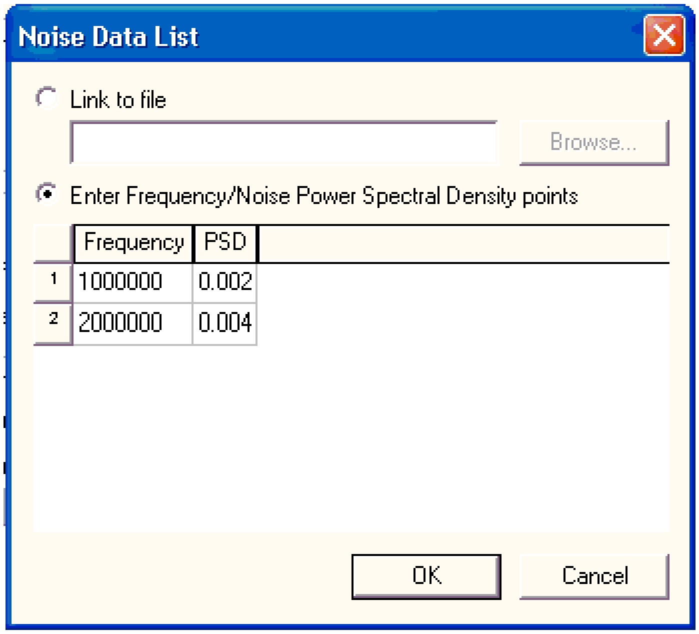
  Port impedance values can also be set using expressions referencing variables or ‘F’. Note: ‘F’ can be used only for Linear Network Analysis.
- With **One port data** selected, use **Edit** or **Create New** to edit a one-port device or create a new one. You can then choose between **Use characteristic impedance (Zo)** or **Use impedance matrix**.

Interface Port Noise Data

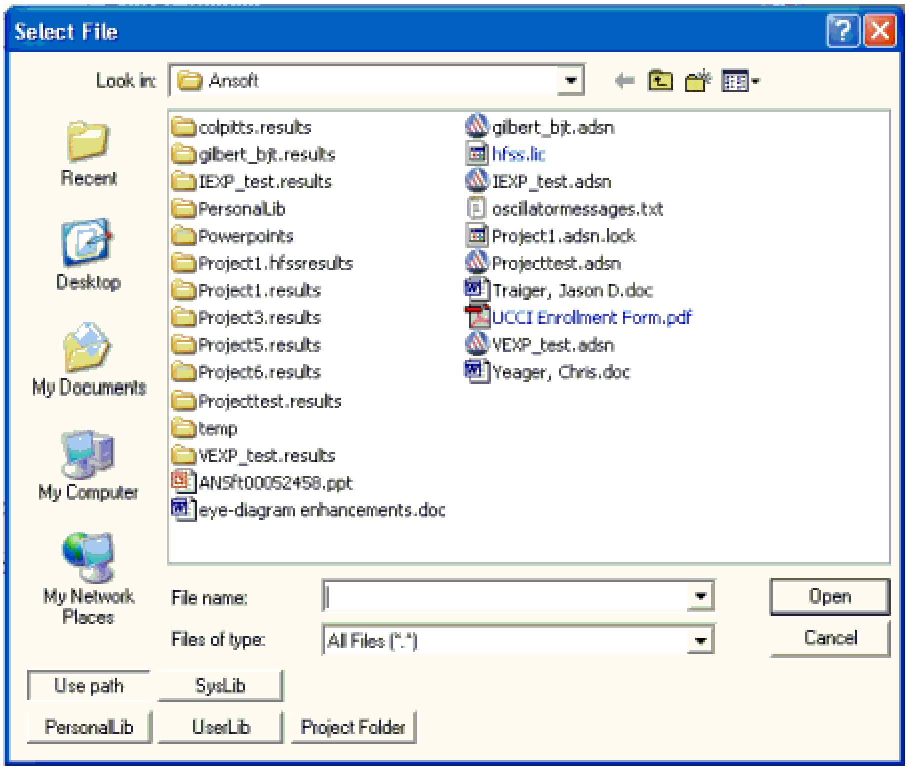
You can specify noise data for the interface port. Click the check box **Enable Noise**. The **Noise Temperature** and **Noise Data for Impedance** fields are activated.

- The default for **Noise Temperature** is 16.85× Celsius. Enter a new temperature as appropriate.

- To add **Noise Data for Impedance**, click **Add**. The **Noise Data List** window opens:



The default selection is **Link to File**. To use data from a file of noise data, click **Browse ...** to open the **Select File** window.



Noise data files have the format. The frequencies are in Hz and the corresponding noise power spectral densities are in Amp^2/Hz.

```
#Hz
freq1 psd1
freq2 psd2
freq3 psd3
.
.
.
freqn psdn
```

- Click the **Look in** field to specify a directory, or locate the file by clicking **Use Path**, **PersonalLib**, **UserLib**, or **SysLib**.
- Within the folder, select the file from those listed, or type the name of the file in the **File Name** group box.
- If you select **Use path**, type the name of the file in the **File Name** group box, or Click the **Look in** field to navigate to the file and record its name. Note that components or libraries imported with Use Path may not be portable when the project is moved to another machine.

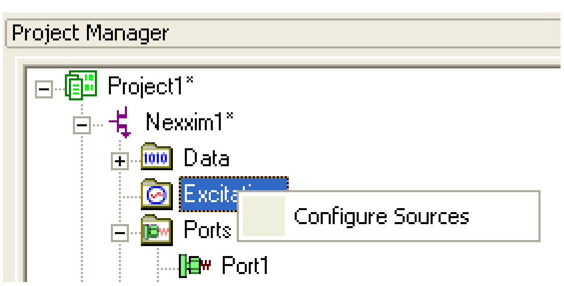
- When **In project folder** is selected, references to the file in the design are relative to the directory where the project resides. In this case, the path is saved in the project **.adsn** file as a variable such as: .lib '$PROJECTDIR/x_113854.lib'
- The variable $PROJECTDIR is expanded to the current location of the project when the design is converted to a netlist and run by an analysis tool. Otherwise, an absolute path is saved. If you move a project and its library files together to a new directory, preserve the file references by selecting the **Project Folder** option.
- To enter the noise data directly, click **Enter Frequency/Noise Power Spectral Density points**. Enter the frequencies (in Hz) and the corresponding PSDs (in Amp^2/Hz). Press **Tab** to advance to the next field.

When all the noise data is as appropriate, click **OK** to close the **Noise Data List** window and return to the **Port Definition** window. Click **OK** to complete the port definition and close the window.

> **Note:**
>
> The **PersonalLib** (personal library) folder is located at **<Project Directory>/PersonalLib** ; the **userlib** (user library) folder is located at **<Installation Directory>/userlib** ; and the **syslib** (system library) folder is located at **<Installation Directory>/syslib**, where **<Project Directory>** is the location where the projects are saved and stored, and **<Installation Directory>** is the directory into which **Electronics Desktop** was installed during setup.
>
> **SysLib** is reserved for libraries supplied with **Electronics Desktop**.

Port Sources

To create a power, voltage, or current source and associate it with one or more ports, right-click the **Excitations** icon in the **Project Manager** window and select **Configure Sources**:



For more information, see Sources in Schematics.

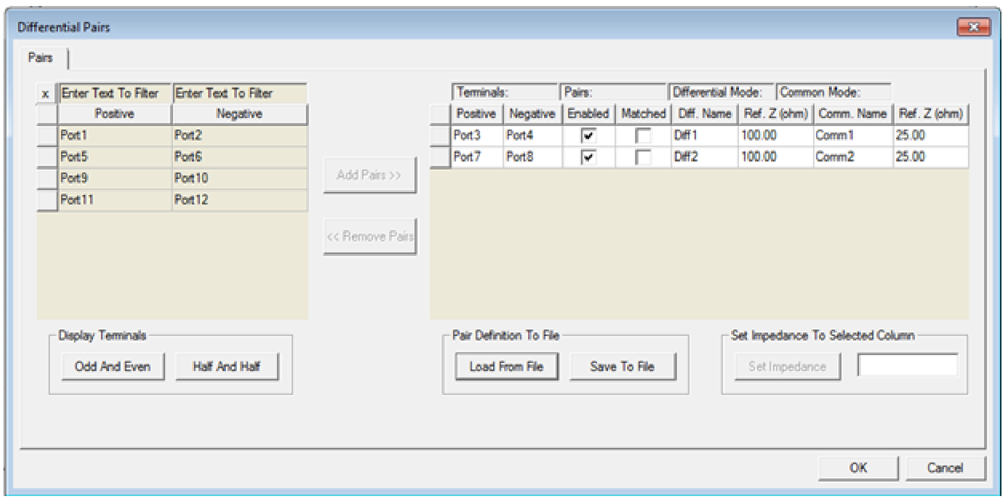
Setting Up Ports as Differential Pairs

A differential pair represents two circuits, one positive and one negative, which are routed close together so they pick up nearly the same amount of noise. The two signals are then subtracted from each other by a receiver, yielding a much more noise-free version of the signal.

You can define existing ports to be differential pairs by first selecting the ports , then accessing the **Differential Pairs** window.

Differential Pairs window can be opened from multiple places:

- Network Data Explorer: Edit > Define Differential Pairs
- Ports Folder in a Nexxim Design: right-click Ports > Differential Pairs
- Excitations Folder in a 3D Layout Design: right-click Excitations > Differential Pairs



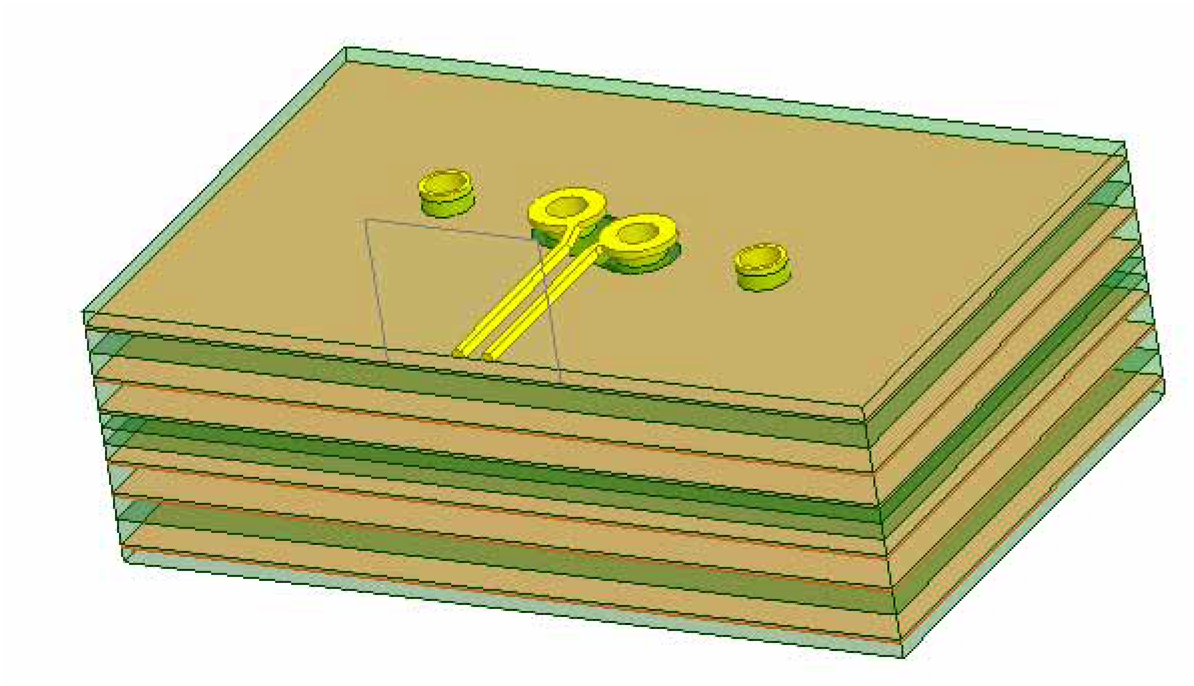
When differential pairs have been set up, the Diff.Name and Comm.Name entries are available in reports. For more information see Set Differential Pairs.

Set Differential Pairs

A differential pair represents two circuits, one positive and one negative routed close together so they will pick up nearly the same amount of noise. The two signals are subtracted from each other by a receiver, yielding a near "noise-free" version of the signal.

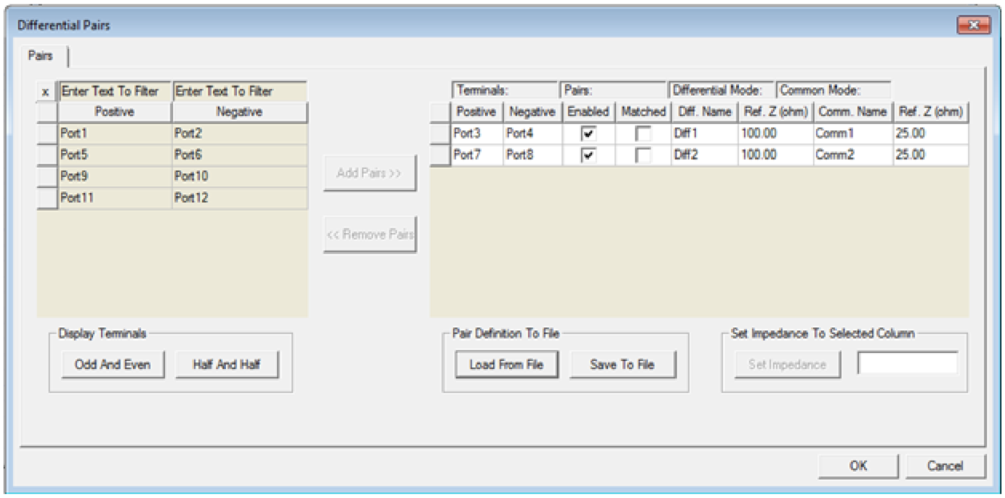
You can define one or more differential pairs from terminal excitations assigned on existing wave ports. Differential pairs can span ports, use lumped ports, and be enabled or disabled. To allow automated calculation of differential S-parameters from lumped ports, you can select terminals from two arbitrary ports, whether wave ports or lumped ports, for use in a differential pair.

Because differential pairs can span ports or occur within a port, the **Differential Pairs** command is accessible at corresponding levels in the Project tree via the right-click menu both at the **Excitations** level, and at the port name level. If a differential pair involves terminals from two different ports, the **Differential Pairs** command for those ports can only be accessed at the **Excitations** level. If an individual wave port has multiple terminals defined, the **Differential Pairs** command is enabled when you select that port and right-click to display the shortcut menu. In order to combine differential pairs across ports, both ports must have the same renormalization setting; that is, either ports have **Do not Renormalize** on, or both have it off. For Transient Network solutions, differential pairs cannot include passive terminals. We will use a differential pair via model to assign the differential pairs. See Figure below.

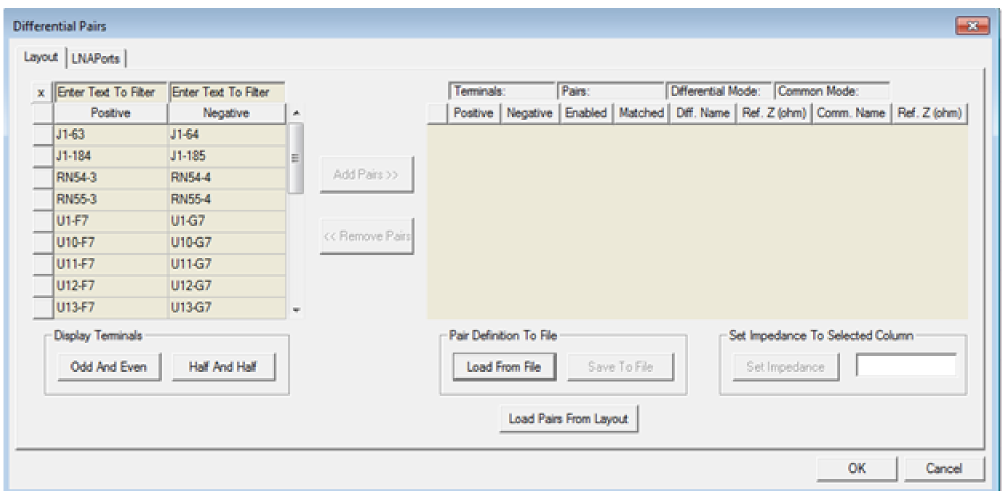


Differential Pairs Dialog can be opened from multiple places:

- Network Data Explorer: Edit > Define Differential Pairs
- Ports Folder in a Nexxim Design: right-click Ports > Differential Pairs
- Excitations Folder in a 3D Layout Design: right-click Excitations > Differential Pairs



Differential Pairs can be defined for both design layout and LNA setups if both are available. Both pairs will be available in two tabs of the same dialog as shown below.



To set up a differential pair:

1. All terminals that are not already in pairs are shown in columns Positive and Negative of the Terminals list (on left):

- Clicking on a cell, selects that cell; Multiple selection is allowed using Ctrl+click and Shift+click; if one cell in each column is selected, Add Pairs button is enabled; clicking that button will add the pair to the bottom of the Pairs list (on right)
- Clicking on a row button in Terminals list selects that row; user can select multiple rows using Ctrl+click and Shift+click; when one or more rows are selected, Add Pairs button is enabled; clicking that button will add the pairs to the bottom of the Pairs list
- If anything is selected in Terminals list, accelerator Ctrl+A will select all rows

- Click-dragging a cell in Terminals list to different position in same column or to any position in the other column will move that terminal to the drop position
- Entering text into the first cell in each column will be used to filter which pins are shown in the column; * wildcard character is supported. See the Layout List dialog for an example of this.

2. Display Terminals group:

- Clicking Odd and Even will move odd position terminals to Positive column and even position terminals to Negative column; the filters will still be applied
- Clicking Half and Half will move terminals in the first half of the original list to the Positive column and terminals in the second half to the Negative column; filters will still be applied

3. Pairs list shows all terminal pairs that have been defined:

- Clicking Enabled or Matched header will toggle the check boxes in those columns; if Matched check box is checked, remove text from both Z Ref cells and disable those cells; if check box is subsequently unchecked restore Z Ref cells to previous values
- Clicking Positive, Negative, or Name column headers will toggle sorting the pairs alphanumerically on that column
- Clicking the Ref Z column headers will select the column
- Clicking on a row button in Pairs list selects that row; user can select multiple rows using Ctrl+click and Shift+click; when one or more rows are selected, Remove Pairs button is enabled; clicking that button will remove the pairs and add the corresponding Terminals back into the Terminals list
- If anything is selected in Pairs list, accelerator Ctrl+A will select all rows; if Del key is clicked, it will remove the selected pairs
- Clicking in a cell for Name or Ref Z will allow user to edit that value directly; names must satisfy pin name restrictions; Ref Z must be positive number with optional resistance units (typing “k” is acceptable for kohm)

4. Set impedance group:

- Entering a positive number with option units into the edit box will enable the Set impedance button
- If a Ref Z column is selected in the Pairs dialog box, clicking the Set impedance button will change all values in that column to the value in the edit box

5. Pair Definition file group:

- Clicking Save To File button will open Save File dialog with library location buttons; entering a name and clicking Save will save the Pair configuration to that file in csv format; this button is not enabled if there are no pairs defined
- Clicking Load from File button will open Open File dialog with library location buttons; choosing a .txt file will load the previously saved Pair configuration; this will show an error and NOT load if the file syntax is incorrect. Note that the file may contain pair definitions that do not reference pins that exist in the circuit; those definitions won’t be applied. Only definitions where both terminals actually exist will be applied.

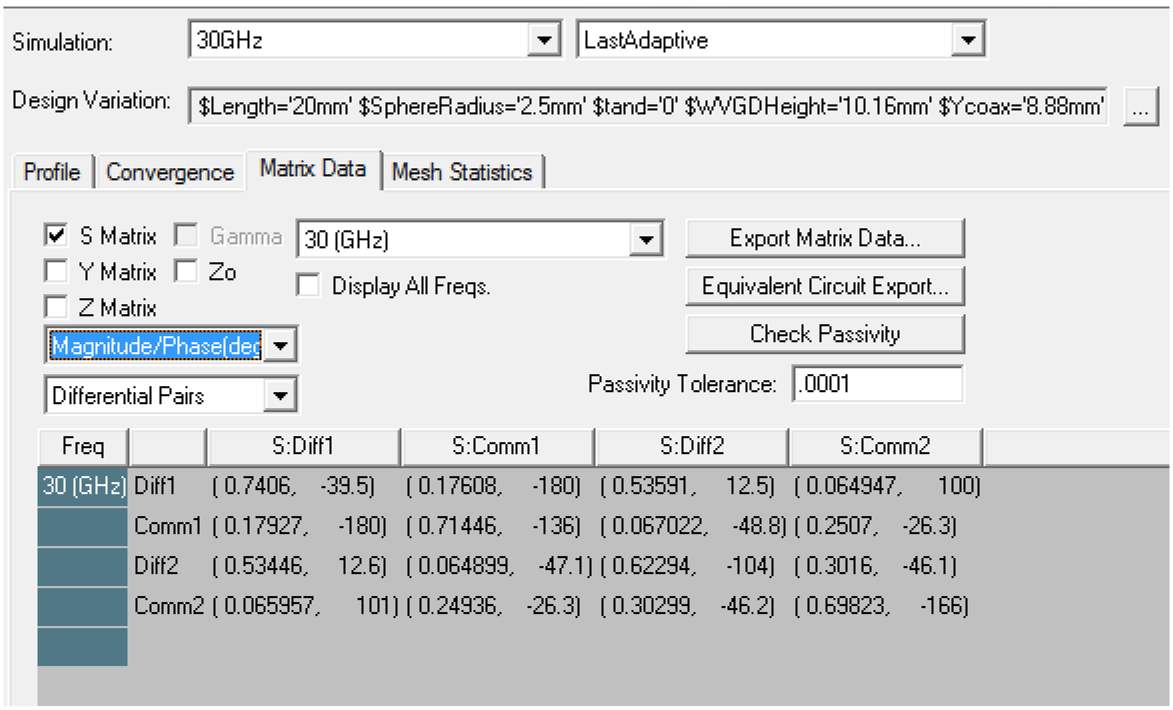
- Saving a file and then subsequently loading a file should open a window already showing the last file saved or loaded.

6. Load Pairs from Layout button:

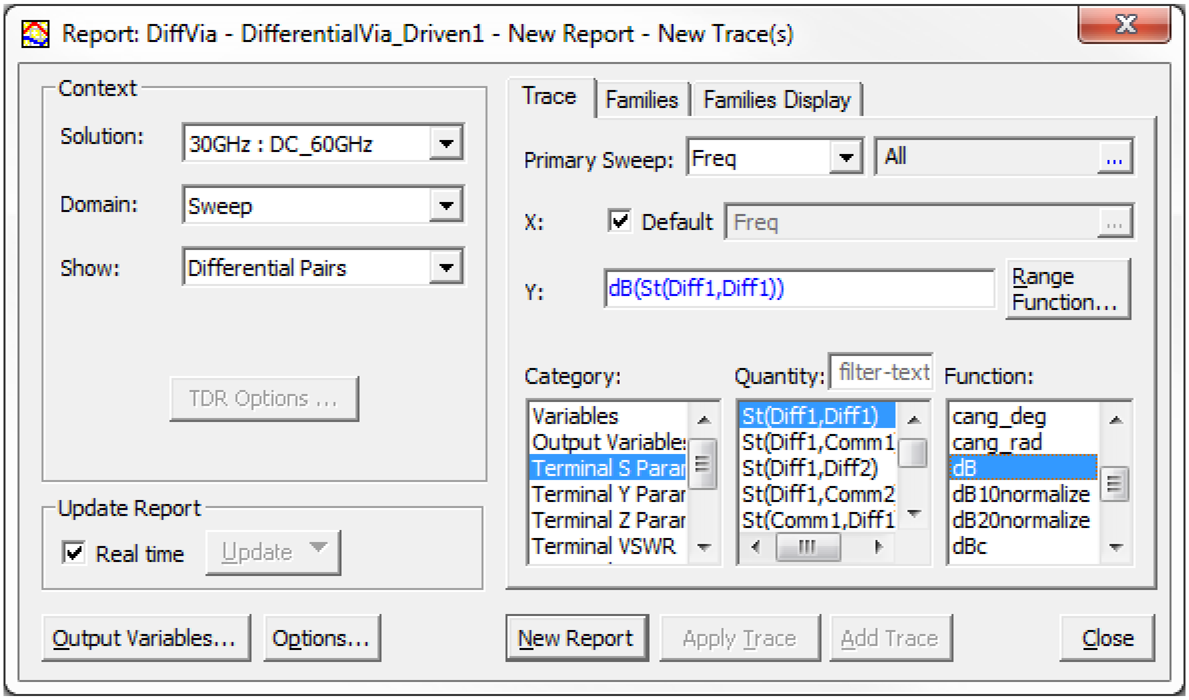
- This button will only be shown when the pairs dialog is launched from HFSS 3D Layout; clicking it will load pairs corresponding to Differential Pair net assignments

> **Note:**
>
> Scripting for loading/saving differential pairs from/to a file will not be recorded as part of the scripting in the dialog box. The user can hardcode the scripting as follows:
>
> - LoadDiffPairsFromFile(filename)
> - SaveDiffPairsToFile(filename)

After a solution has been generated, view the common and differential quantities of the differential pair under the **Matrix** tab of the **Solution Data** window as shown below.



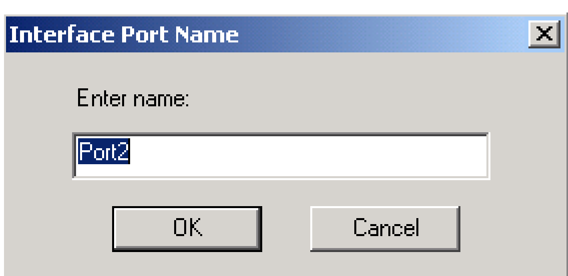
When the design has differential pairs (link), the reporter can display quantities for the defined pairs or for the single-ended terminals upon which they are based. A drop-down menu will appear in the **Context** area of the **Report** creation dialog which allows the user to select which quantities will be displayed.



You can freely mix differential and single-ended terminal quantities. However, single ended quantities are computed as if no differential pairs existed. So, in the unlikely case of several terminals where only a subset are combined into pairs, the results may not be as expected.

Renaming an Interface Port

Each interface port you place appears as a **Port** object under the selected design's icon in the **Project Manager** window. The name of the port is the name of the node or net to which it is connected. Node names of the form "net_*n*" are automatically generated during the wiring operation, but can be renamed. To rename an interface port, select it by clicking its schematic symbol or its icon in the project tree. Then click the port name button in the **Properties** window to open the **Interface Port Name** window.



Enter the appropriate name, then click **OK**. The property window reflects the new name. Note that the port number (pnum) does not change:

- If the port is placed so that it connects to an existing node, the node is renamed to the name of the port , even if the node was manually renamed. (You can also change a port name attached to a node by selecting the node and clicking on the node name in its property window. A **Net Name** window similar to the one above opens to specify a new name. Changing the net name changes the names of any ports attached to that node.)
- If you name a port the same as an existing net, the two nodes are merged into a single node with that name. A window box appears to confirm the merge operation.
- If you delete an interface port, the node to which it was attached is renamed to a system-defined name (“net_*n*”).

Page Ports

A page port serves as a named connection to a signal that is common to two or more pages of the schematic.

- To place a page port in the schematic, click on the Page Port icon in the **Schematic Editor** menu.
- Click again in the **Schematic Editor** window to place the port symbol. By default, the port is named “Pageport_*n*”, where *n* is an arbitrary integer.

The name of the port is the same as the name of the node or net to which it is connected. Node names of the form “net_*n*” are automatically generated during the wiring operation, but they can be renamed by clicking on the wire , then clicking on the node name in the **Parameters** window. If the port is placed so that it connects to an existing node with a system-assigned name (“net_ *n*”), the node is renamed to the name of the port. If the node was manually named, the node retains its user-defined name and the page port gets that name.

To rename a port (and the node to which it connects):

- Click the port to select it and view its properties in the property window.
- Click the port name in the **Port Name** field to open the **Page Port Name** window: Enter the appropriate name, then click **OK**. The property window reflects the new name.
- You can also change a port name attached to a node by selecting the node and clicking on the node name in its property window. A **Net Name** window similar to the one above opens for you to specify a new name. Changing the net name changes the names of any ports attached to that node.
- If you name a port the same as an existing net, the two nodes are merged into a single node with that name. A window box appears to confirm the merge operation.
- If you delete a page port, the node to which it was attached retains the name it had before the delete, either system-defined or user-defined.

Page ports serve as a graphic reminder that a signal may be present on another page. Any node with a given name (system-generated or user-defined) is automatically connected across all pages that reference it.

Ground Ports

A ground port serves as a connection to the reference or ground node (node 0 in schematic designs).

1. To place a ground, select the schematic that contains it.
2. From the **Schematic Editor** menu bar, menu, click the **Ground** icon or click **Ground** on the **Draw** dropdown menu.
3. The cursor, now carrying a ground symbol for placement, moves to the center of the schematic window. Click at the appropriate location to place the ground.

If Multiple Placement is activated for grounds in the Schematic Options window, place additional grounds by clicking at additional locations. To stop placing grounds during multiple placement, do either of the following:

- Press **Enter**, the **Spacebar**, **Backspace**, or **Esc** on your keyboard.
- Right-click, then select **Place and Finish**, **Finish**, **Cancel**, or **Back**.

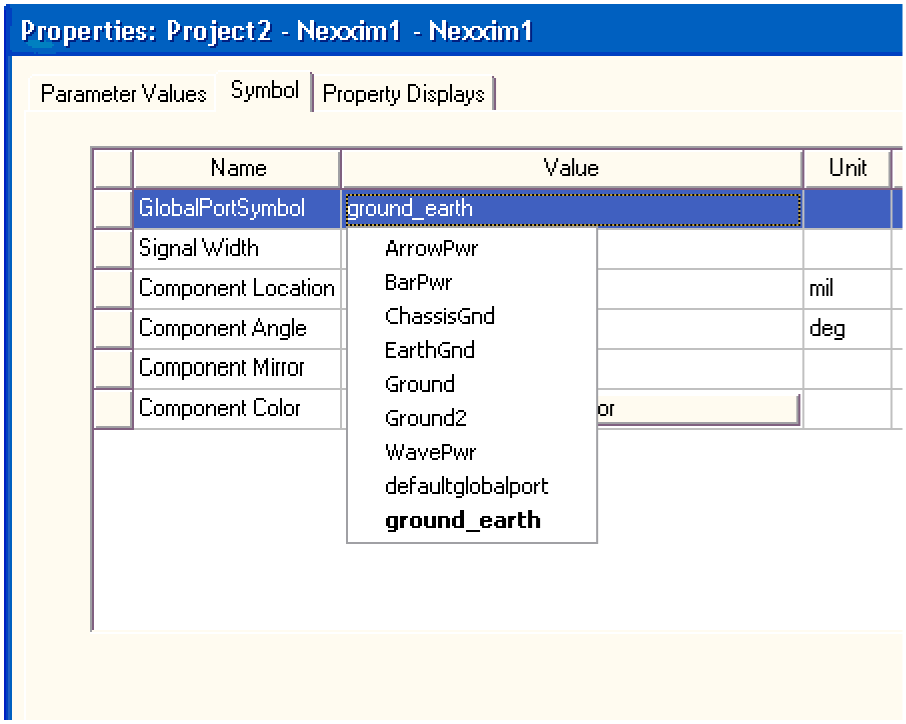
Hint You can rotate a ground before placing it by repeatedly pressing **R** on your keyboard. Each press rotates the ground 90º counterclockwise.

> **Note:**
>
> To ensure electrical connectivity among schematic elements, the pins of placed grounds snap to a 100-mil (2.54-millimeter) grid. This snapping cannot be deactivated, and the spacing of the connectivity grid cannot be adjusted.

Selecting a Ground Symbol

Several different symbols are available to represent a ground node.

1. To select a symbol other than the default, double-click the ground symbol to open the **Properties** window box.
2. Select the **Symbols** tab and click the **Value** field for the **GlobalPortSymbol** to display a drop-down with the choices for symbols.



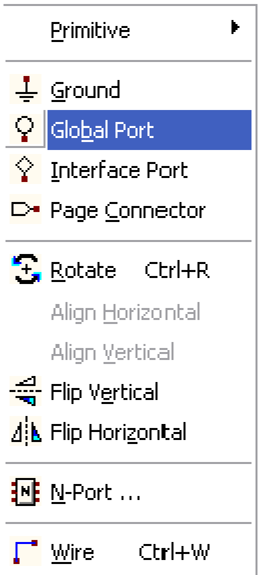
3. Select a symbol and click **OK** to close the **Properties** window. The new symbol is displayed in the schematic.

> **Note:**
>
> All of the global port symbols except the **defaultglobalport** symbol can be edited in the Symbol Editor (part of the Library Editor).

Global Ports in Schematics

In Circuit designs, a global port serves as a named connection to a signal that is common to all parts of the circuit. A global port can represent local power, a special ground node, or any other global signal.

- To place a global port, select the schematic that contains it.
- From the **Draw** drop-down menu, select **Global Port**.



The cursor, now carrying a global port symbol for placement, moves to the center of the schematic window. To place the global port, click at the appropriate location. To end the placement operation, do either of the following:

- Press **ENTER**, the **SPACEBAR**, **BACKSPACE**, or **Esc** on your keyboard.
- Right-click, then select **Place and Finish**, **Finish**, **Cancel**, or **Back**.

Hint You can rotate the symbol before placing it by repeatedly pressing **R** on your keyboard. Each press rotates the symbol 90º counterclockwise.

Note:

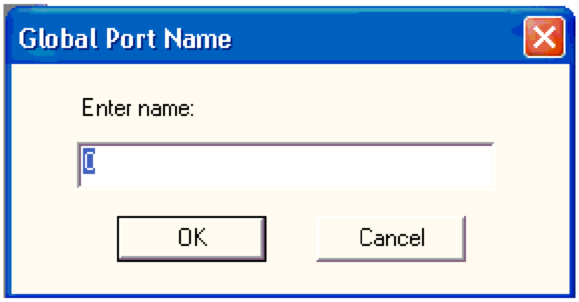
To ensure electrical connectivity among schematic elements, the pins of placed global ports snap to a 100-mil (2.54-millimeter) grid. This snapping cannot be turned off, and the spacing of the connectivity grid cannot be adjusted.

Selecting a Global Port Name

A global node created on the **Draw** drop-down is listed in a **.GLOBAL** statement in the generated netlist with a default node name **G_***n*, where *n* is an integer. A global port can also be defined by naming a ground port to a node name other than zero (0).

See Global Nodes in Subcircuits for further information.

1. From the **Schematic Editor** toolbar, click the **Ground** icon.
2. Select the ground symbol. The properties of the ground port appear in the **Properties** window. (Alternatively, double-click the symbol to open the **Properties** window for the ground port.)
3. To change a ground port into a global port, or to rename a global port that has a default node name, double-click on the **Value** field for the PortName parameter to open the **Global Port Name** window:

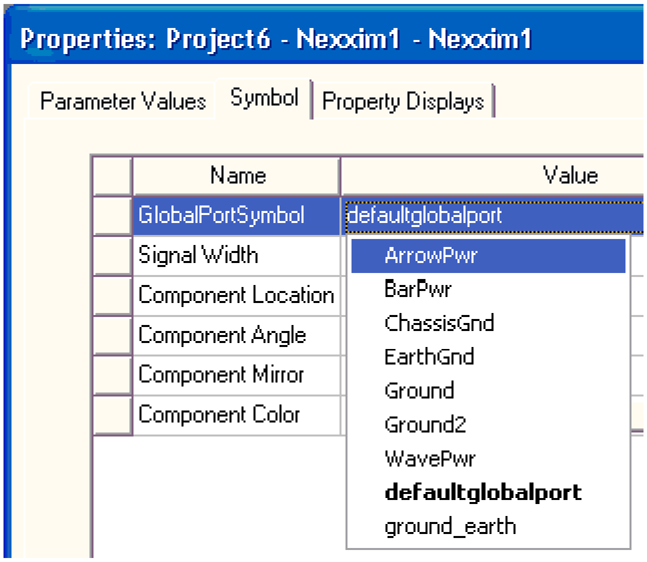


4. Enter the appropriate name for the global node and click **OK**.

> **Note:**
>
> When you wire two differently named global nodes together, a .CONNECT statement is added to the netlist.
>
> See CONNECT Statement for further information.

Selecting a Global Port Symbol

Eight different symbols are available to represent various kinds of global ports. To select a symbol other than the default, double-click the global port symbol to open the **Properties** window. Select the **Symbols** tab. Click the **Value** field for the **GlobalPortSymbol** to display a drop-down with the choices for symbols.



Select a symbol and click **OK** to close the **Properties** window. The new symbol is displayed in the schematic.

Note:

All of the global port symbols except the defaultglobalport symbol can be edited in the Symbol Editor (part of the Library Editor).

How Ports Affect Node Names

When you connect a port to a node or net, the net name changes to that of the port. The rules for net renaming depend on the kinds of ports attached to the node. The precedence for port types is **Global Ports > Interface Ports > Page Ports**.

Global Ports

- If a global port is attached to a net, the net name is the global port name, and all connected page ports also share that name.
- If more than one global port is attached to a net, the net name is the first global port name, though any name should work to access the net.

> **Note:**
>
> When you wire two differently named global nodes together, a .CONNECT statement is added to the netlist.
>
> See CONNECT Statement for further information.

Interface Ports

- If an interface port is attached to a net (but no global ports), the net name is the interface port name, and all connected page ports also share that name.
- If more than one interface port are attached to a net, the net name is the first interface port name.

Page Ports

- If a page port is attached to a net (but no global or interface ports), the net name is the page port name.
- If more than one page port is attached to a net, they all share the same name.

Adding Ports

- If a global port is added to a net, the name of the net and page ports on the net are changed to the global port name.
- If an interface port is added to a net (without a global port) the name of the net and page ports directly connected are changed to the interface port name.
- If a page port is added to a net (without any global or interface ports) the name of the net and pageports directly connected are changed to the page port name.

Changing Node Names

You can change the name of a node (net) in the schematic by double-clicking it to open its **Properties** window, then click the **Value** field for the **Name** property. Then click **OK** to apply the new name.

Following are some guidelines to keep in mind when changing the name of a node:

- Net names may not contain spaces, ampersands (&), or asterisks (*).
- When you try to change a global or interface port name to that of an existing net, the existing net of the same name as the new port name is connected to the new port's net. A window opens, to resolve the name conflict by splitting the net into two nets, or allowing the nets to remain connected.
- When you try to change a global or interface port name to that of an existing net, the existing net of the same name as the new port name is connected to the new port's net. A

window opens, to resolve the name conflict by splitting the net into two nets, or allowing the nets to remain connected.

- When you try to change a page port name to that of an existing net, the existing net of the same name as the new port name is connected to the new port's net. A window opens, to resolve the name conflict by splitting the net into two nets, or allowing the nets to remain connected.
- If a global port name is changed, the name of the net it is attached to and any connected page ports are changed.
- If an interface port name is changed, the name of the net it is attached to and any connected page ports are changed.
- If a page port name is changed, the net is examined to see what is physically connected. Wires and pins that are directly attached to the renamed page port are connected on the net with the page port. If there is another net with a page port of the new name, the page port and directly connected objects are connected to that net. If there is no such other net, the page port and directly connected objects become their own net.
- If a net name is changed, the ports on the net that had the same name as the old net name is changed to the new name.

The Effect of Deleting Ports

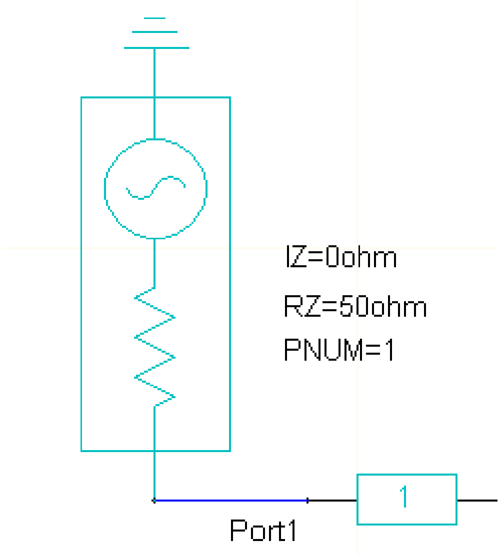
- If a global port is deleted from a net, the name of the net is determined by the remaining global or interface ports, or becomes a new unique name.
- If an interface port is deleted from a net, the name of the net is determined by the remaining global or interface ports, or becomes a new unique name.
- If a page port is deleted from a net, the name of the net is determined by the remaining global or interface ports, or becomes a new unique name.

Sources in Schematics

The Configure Sources window allows you to create power, voltage, or current sources, then associate each source with a particular port and with a particular analysis. The Configure Sources facility allows you to specify the ports and sources to be incorporated in a given analysis.

Power Calculation for Port Sources

For all sources associated with ports in the schematic, as in the following diagram:



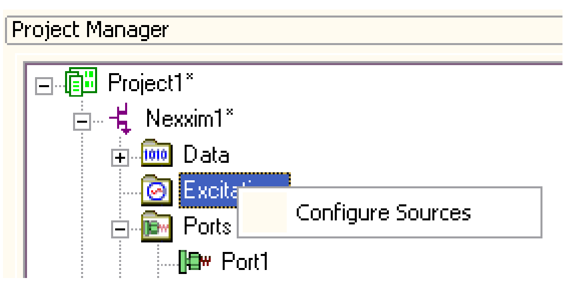
The source (if present) and the impedance are in series. The power in dBm computed at the node labeled Port 1 is the **Apparent Power**:

$$mag(0.5 \times conj(V(Port1) \cdot I(Port1)))$$

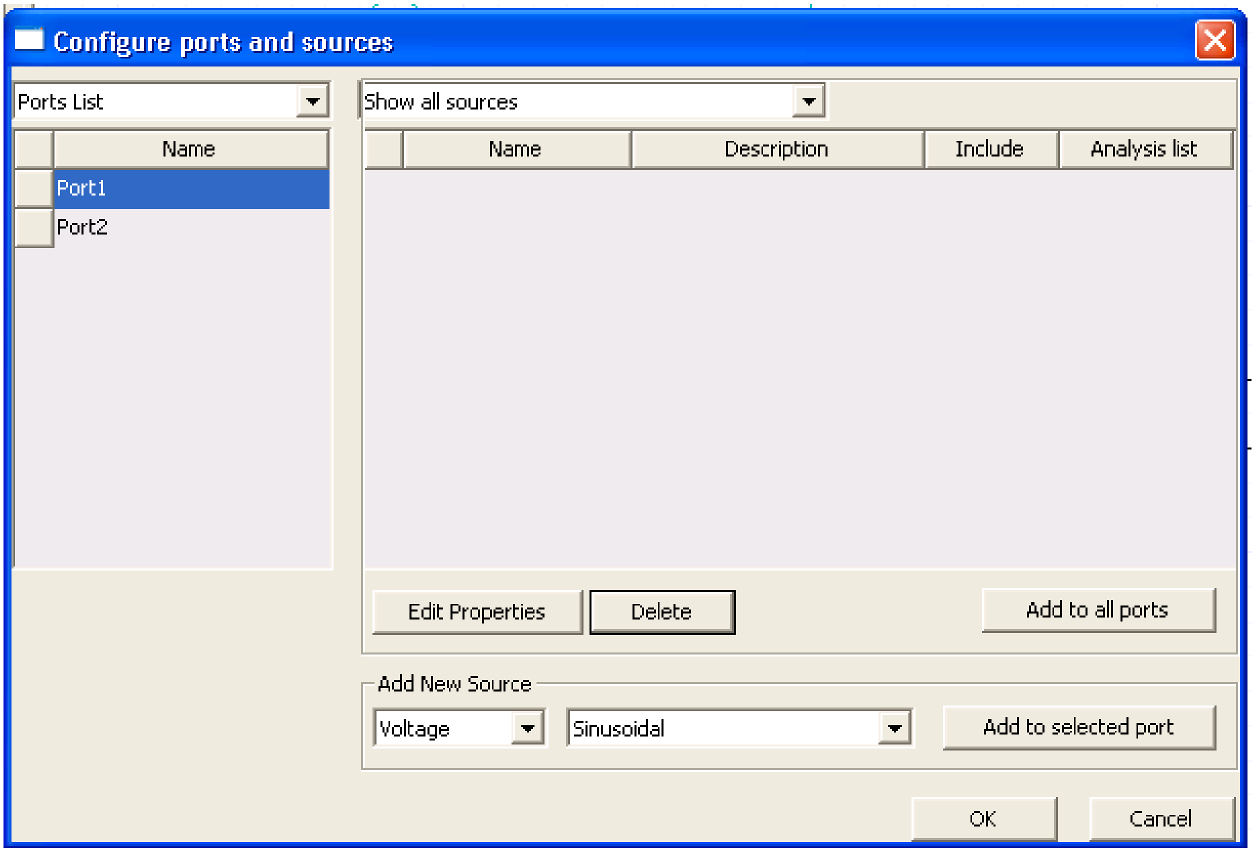
The power computed is the **Apparent Power** as seen by the circuit. It is not the Total Apparent Power as is seen at the intermediate node between the source (if present) and the impedance.

Creating Sources for Ports

To create a power, voltage, or current source, right-click the **Excitations** icon in the **Project Manager** window and select **Configure Sources**:.

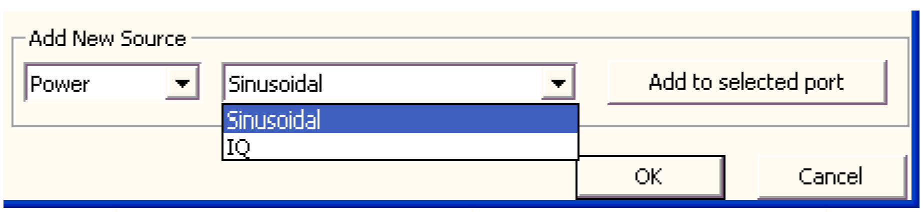


The **Configure ports and sources** window opens:



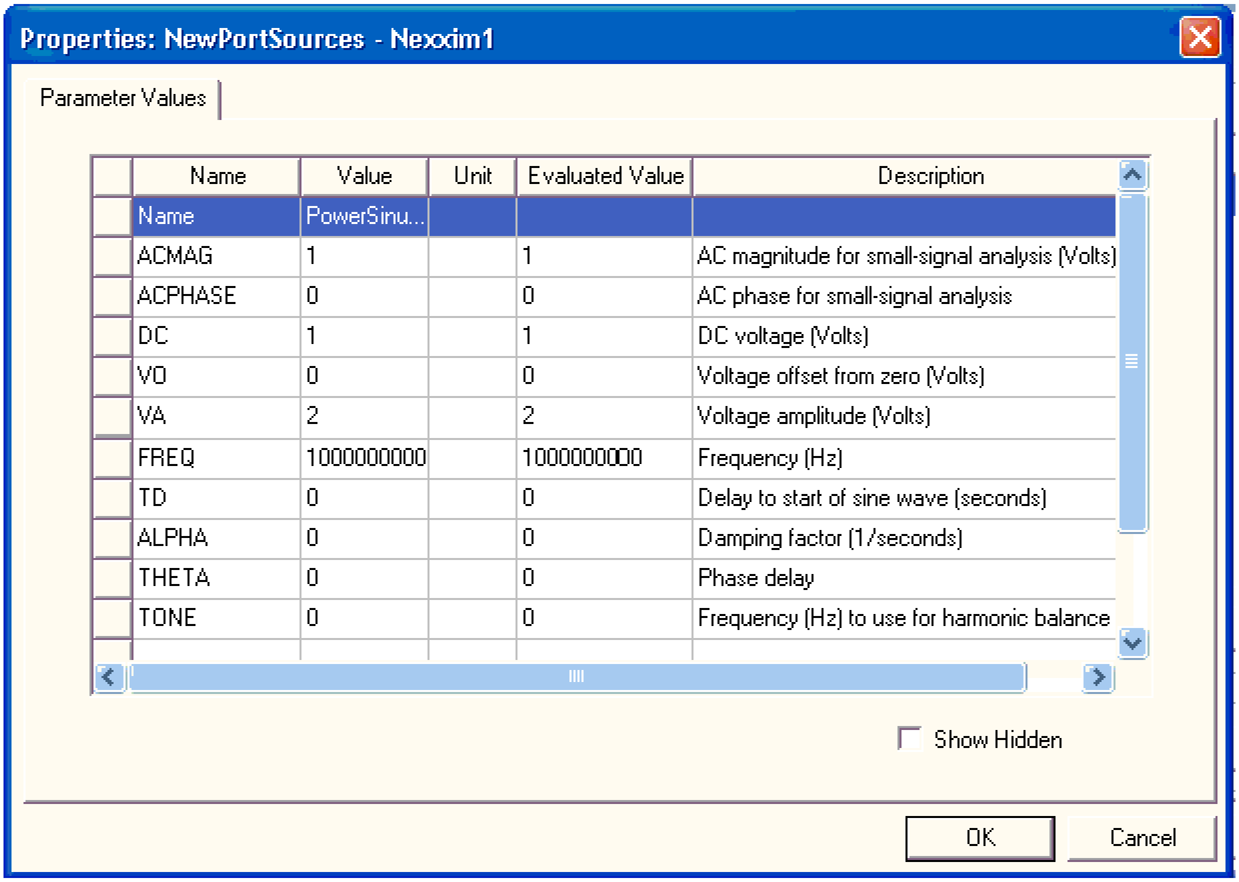
Port Power Source

You can set up an interface port to be a power source for the circuit. A power source can be sinusoidal or IQ-modulated. In the **Add New Source** group box, select **Power** and either **Sinusoidal** or **IQ**:



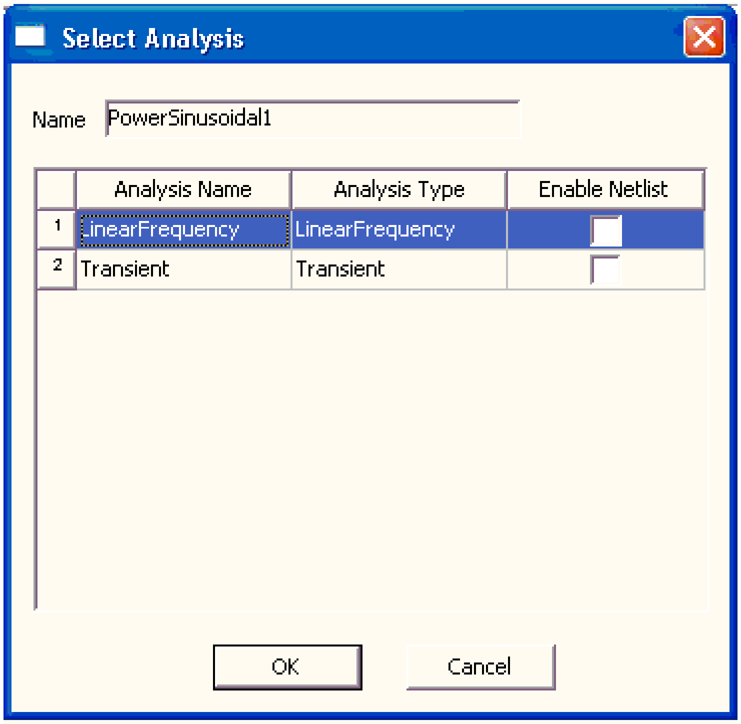
Sinusoidal Port Power Source

Select **Sinusoidal** as the **Power Source Type**. Click **Add to selected port**. The **Parameters** tab lists the sinusoidal parameters:

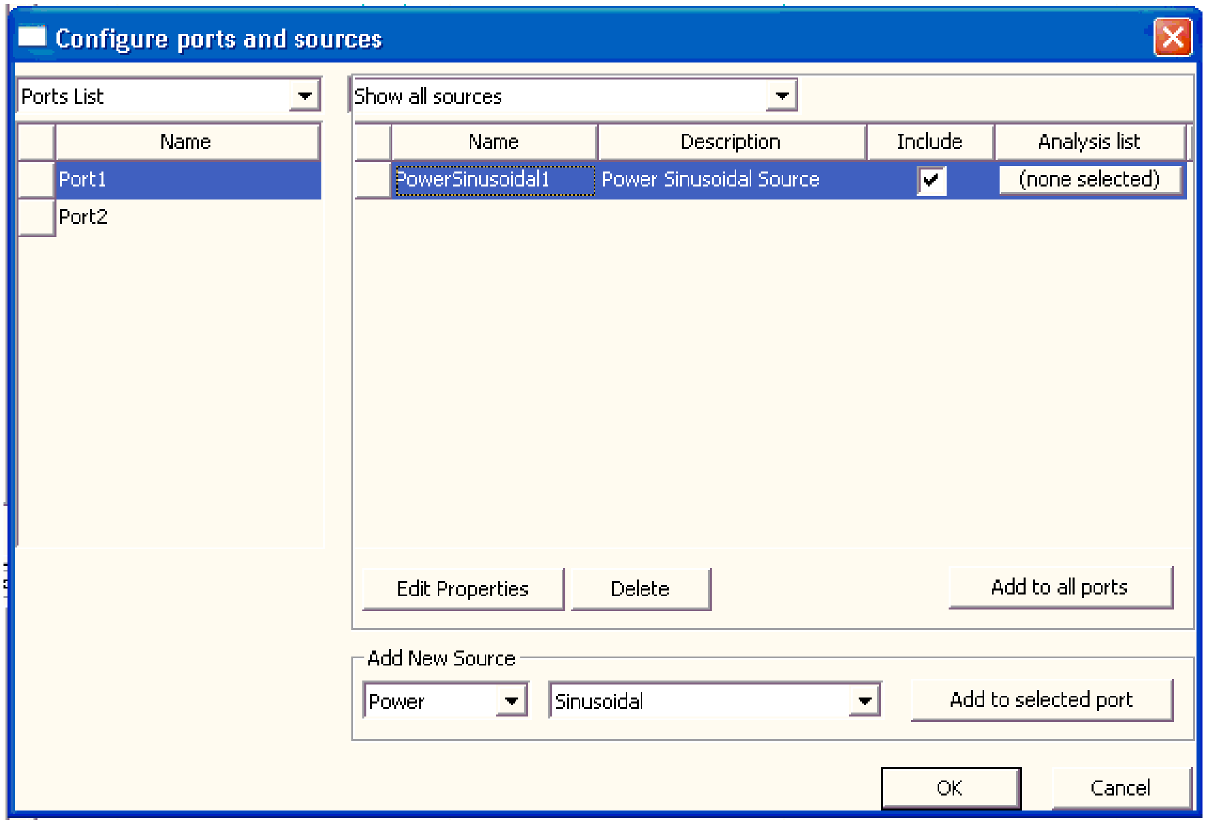


- Click in the **Name** field to assign a name other than the default (“PowerSinusoidal*n*”).
- Assign new values or units to the source parameters in the **Parameters** window by clicking in the **Value** or **Unit** field to be changed.

Click **OK** in the **Properties** window to open the **Select Analysis** window.



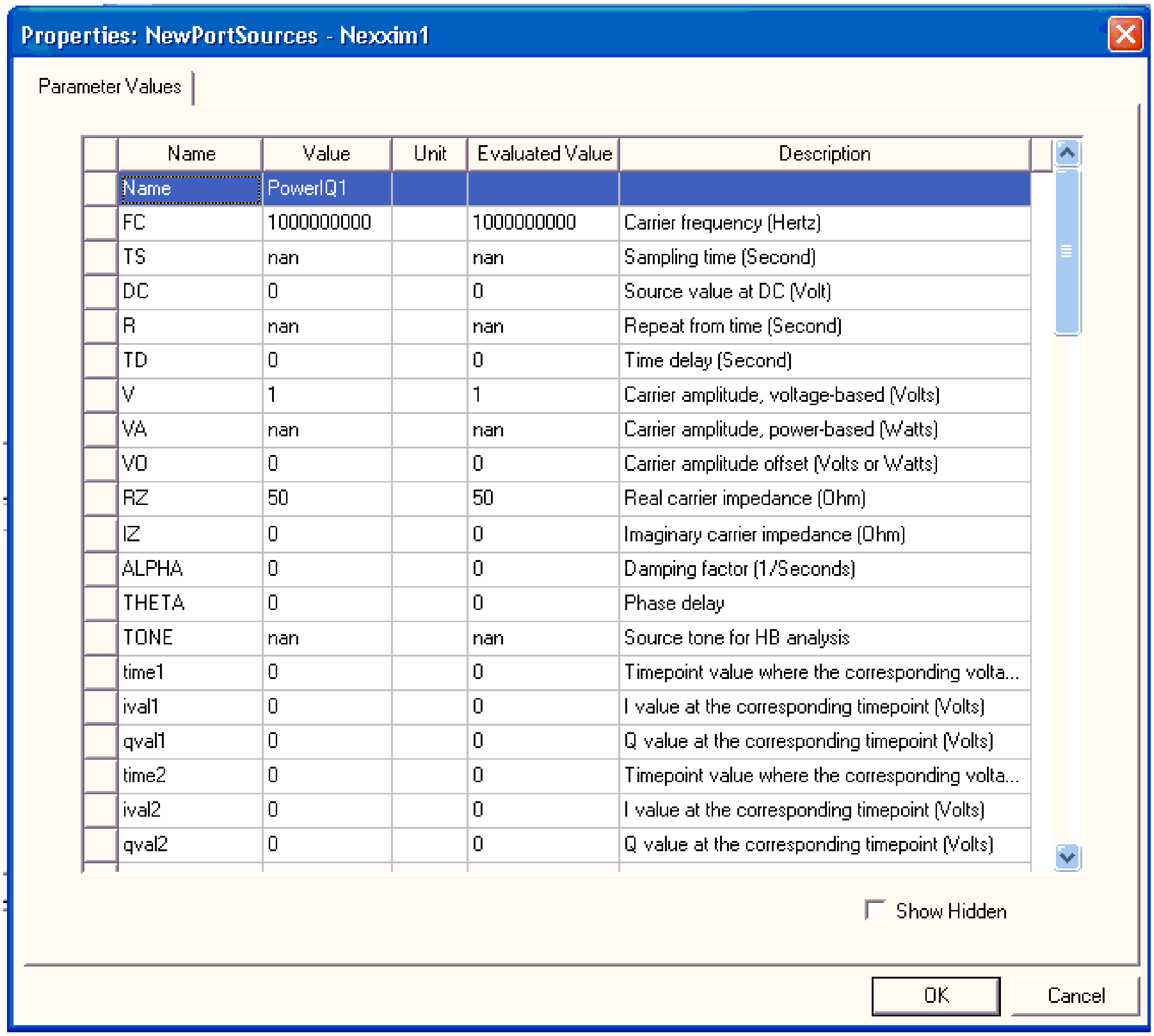
Use the check boxes to associate the source with one or more analyses. Click **OK** to return to the **Configure sources and ports** window. The **Configure ports and sources** window now shows the new source.



The **Include** check box is automatically checked to show that the source is associated with the selected port (**Port1**).

IQ-Modulated Port Power Source

In the **Add New Source** group box, select **Power** and **IQ**. Click **Add to selected port**. The **Properties** window lists the IQ source parameters, as in the following screenshot.


Properties: NewPortSources - Nexxim1

Parameter Values

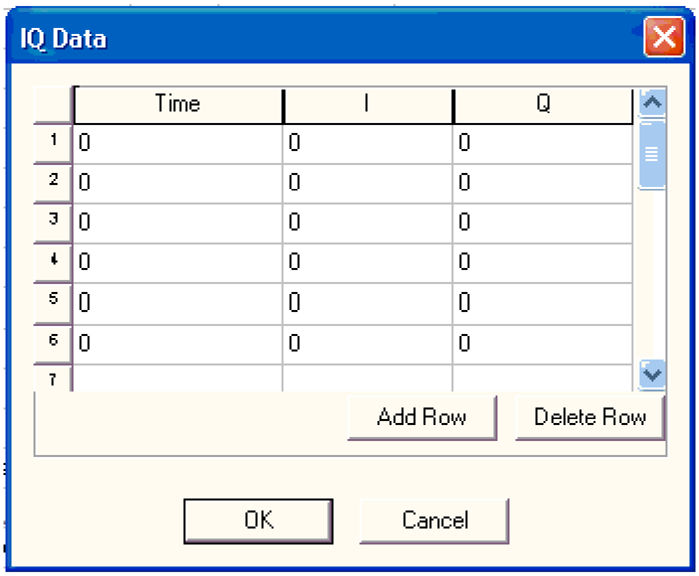
Name	Value	Unit	Evaluated Value	Description
Name	PowerIQ1			
FC	1000000000		1000000000	Carrier frequency (Hertz)
TS	nan		nan	Sampling time (Second)
DC	0		0	Source value at DC (Volt)
R	nan		nan	Repeat from time (Second)
TD	0		0	Time delay (Second)
V	1		1	Carrier amplitude, voltage-based (Volts)
VA	nan		nan	Carrier amplitude, power-based (Watts)
VO	0		0	Carrier amplitude offset (Volts or Watts)
RZ	50		50	Real carrier impedance (Ohm)
IZ	0		0	Imaginary carrier impedance (Ohm)
ALPHA	0		0	Damping factor (1/Seconds)
THETA	0		0	Phase delay
TONE	nan		nan	Source tone for HB analysis
time1	0		0	Timepoint value where the corresponding volta...
ival1	0		0	I value at the corresponding timepoint (Volts)
qval1	0		0	Q value at the corresponding timepoint (Volts)
time2	0		0	Timepoint value where the corresponding volta...
ival2	0		0	I value at the corresponding timepoint (Volts)
qval2	0		0	Q value at the corresponding timepoint (Volts)

Show Hidden

OK Cancel

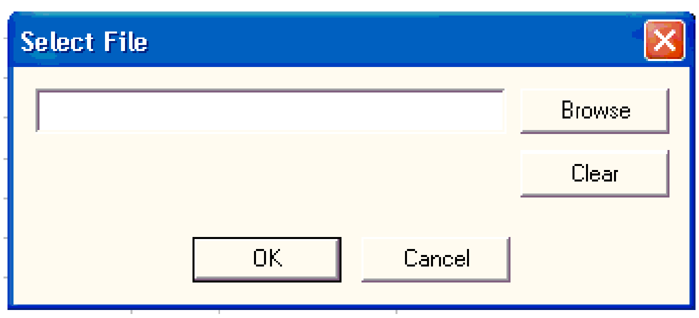
- Click in the **Name** field to assign a name other than the default (“PowerIQ*n*”).
- Assign new values or units to the source parameters in the **Parameters** window by clicking in the **Value** or **Unit** field to be changed.
- To assign the IQ time and data point manually, either enter the values in the *time*, *ival*, and *qval* parameter fields, or scroll down to the bottom of the property listing and click **IQ Data**.

The **IQ Data** window opens:

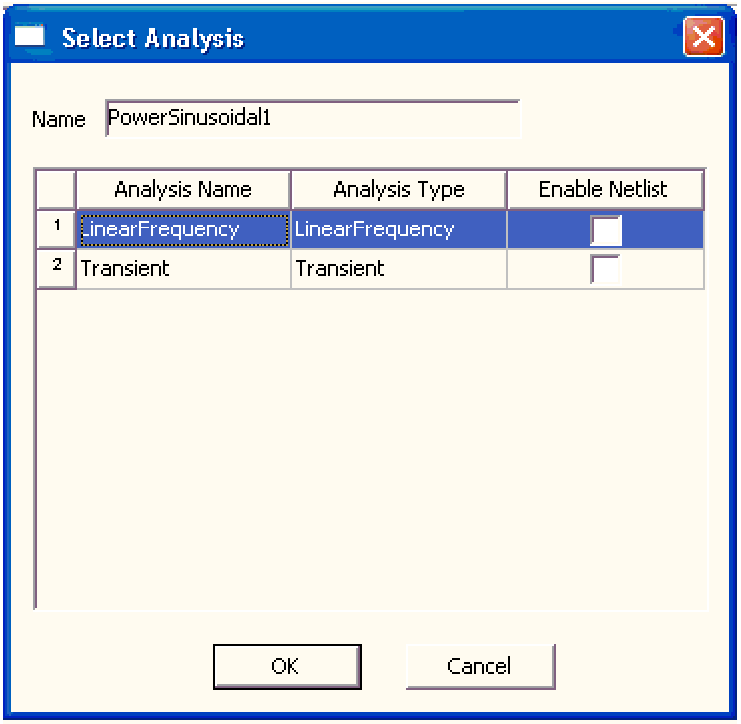


Enter the data in the fields. New rows are added automatically as you type. You can also add a row (to the end of the list) by clicking **Add Row**, or delete a selected row by clicking **Delete Row**. Click **OK** to return to the Properties window.

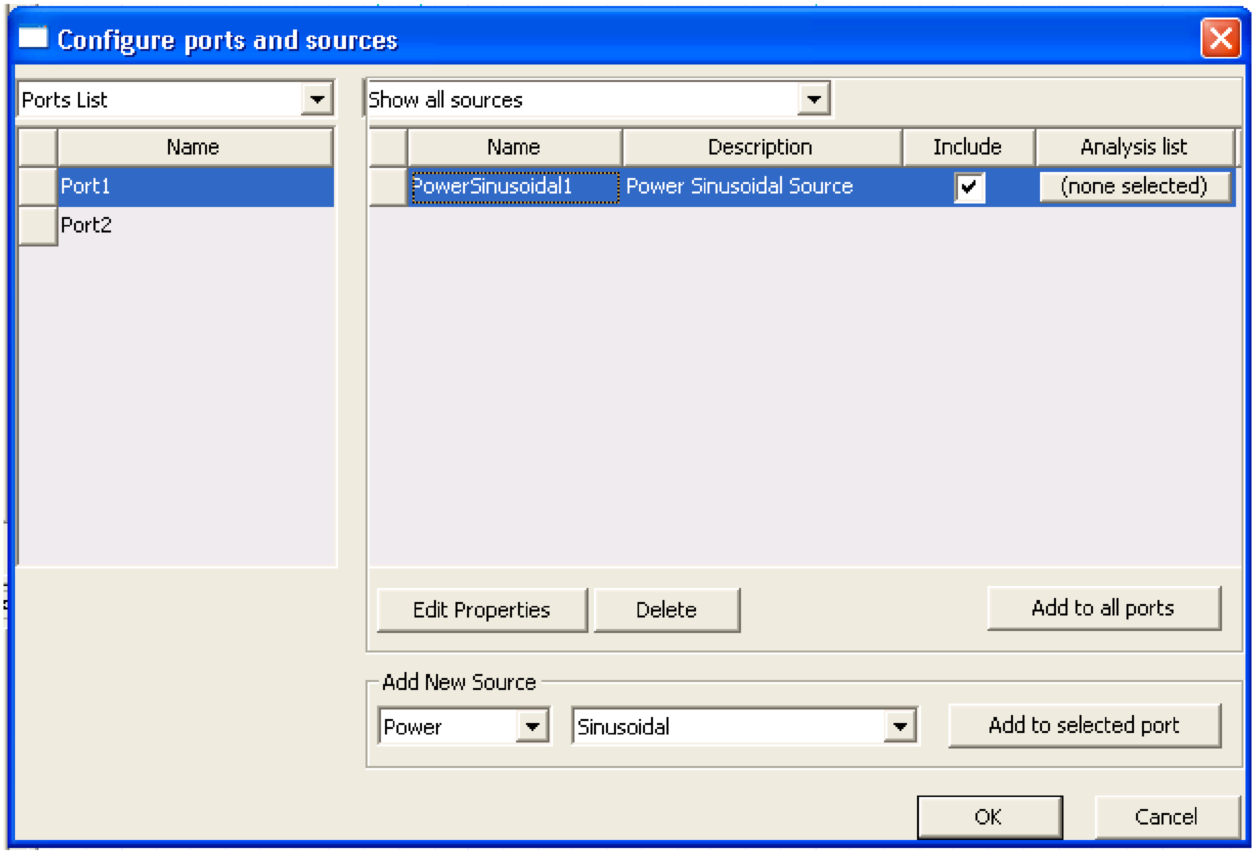
- If the IQ data is in a file, click **File** near the bottom of the property listing. The **Select File** window opens:



- Click **Browse** to locate the data file. Click **OK** to return to the Properties window.
- When the IQ data has been specified, click **OK** to close the **Properties** window to open the **Select Analysis** window.



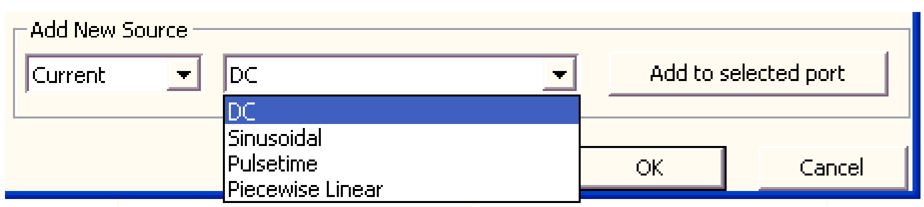
Use the check boxes to associate the source with one or more analyses. Click **OK**. The **Configure ports and sources** window box now shows the new source.



Port Current Source

You can set up an interface port to be a current source for the circuit.

In the **Add New Source** group box, select **Current** and one of **DC**, **Sinusoidal**, **Pulsetime**, or **Piecewise Linear**:

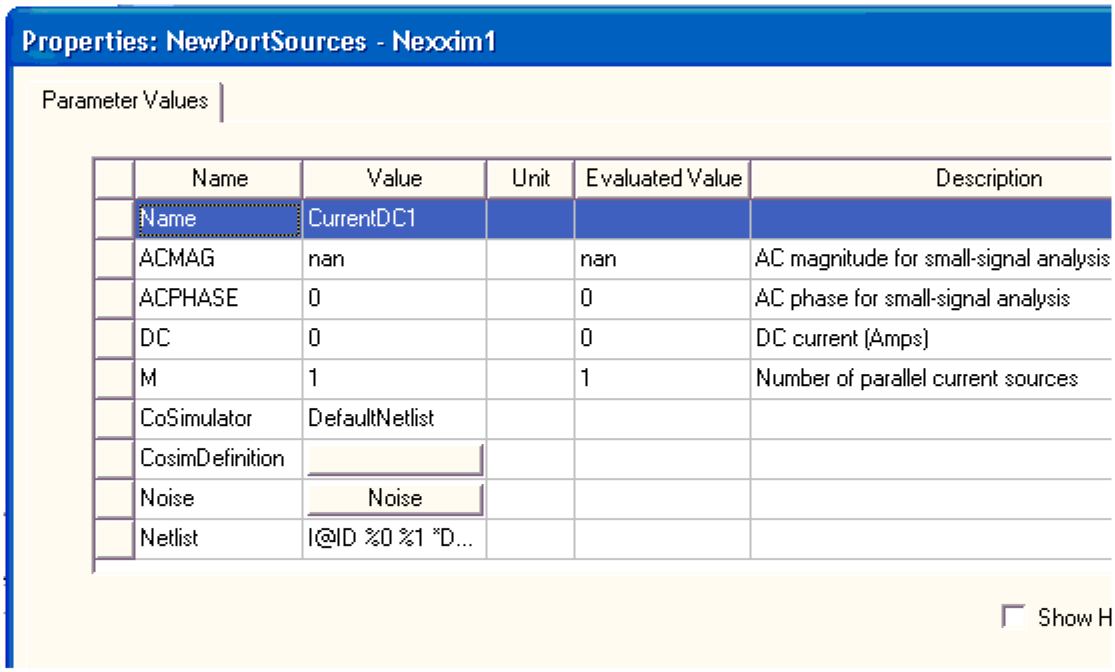


Port current sources in schematic designs can be DC, Pulse, Piecewise Linear, or Sinusoidal.

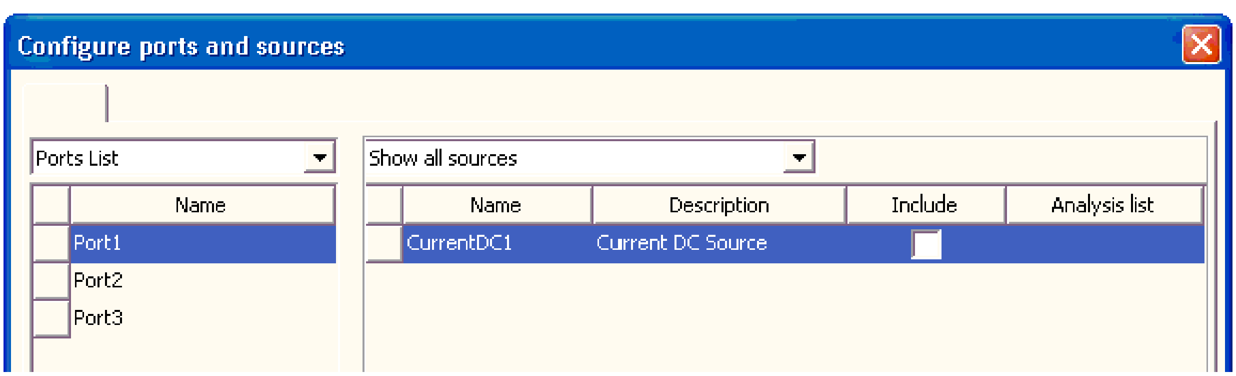
DC Port Current Source

In the **Add New Source** group box, select **Current** and **DC**. Click **Add to selected port**.

The Properties window displays the parameters for the DC current source:



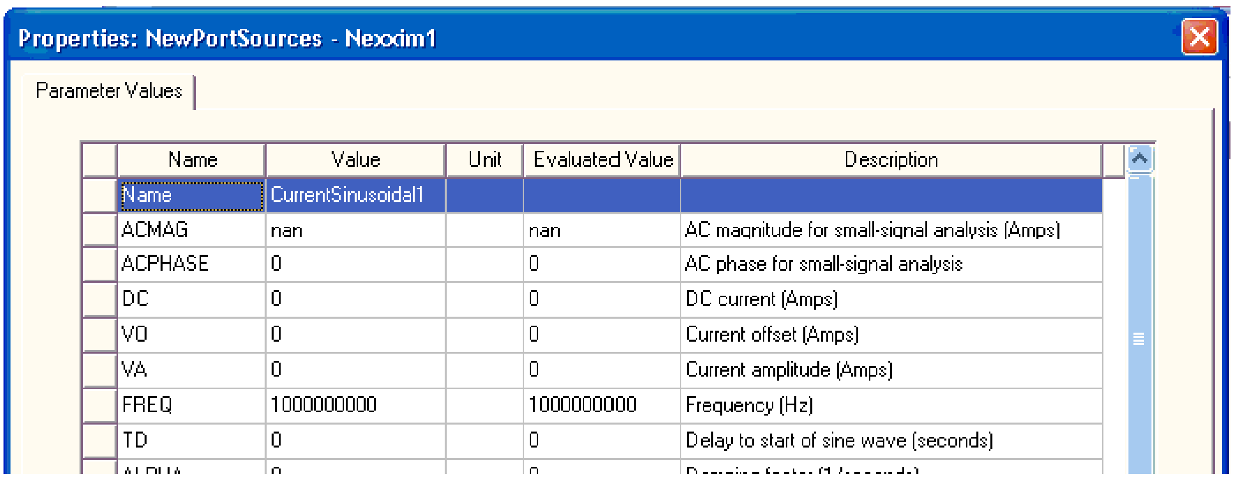
- Assign new values or units to the source parameters in the **Parameters** window by clicking in the field to be changed.
- To specify noise data, click **Noise** in the **Value** field. See Adding Noise Data to a Port Source for further information.
- When all parameters have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.
- The **Configure ports and sources** window box now shows the new source.



Sinusoidal Port Current Source

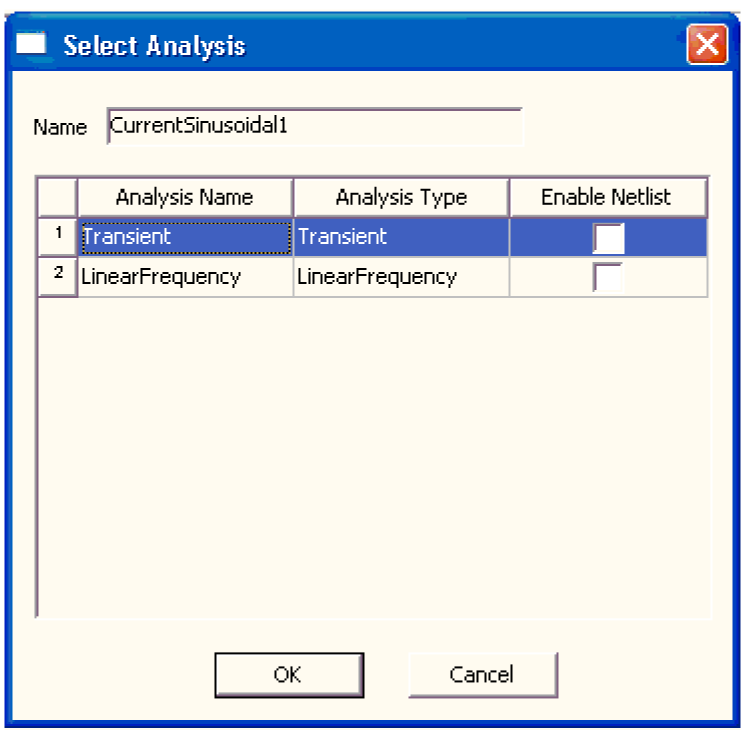
In the **Add New Source** group box, select **Current** and **Sinusoidal**. Click **Add to selected port**.

The Properties window shows the Sinusoidal Current source parameters.



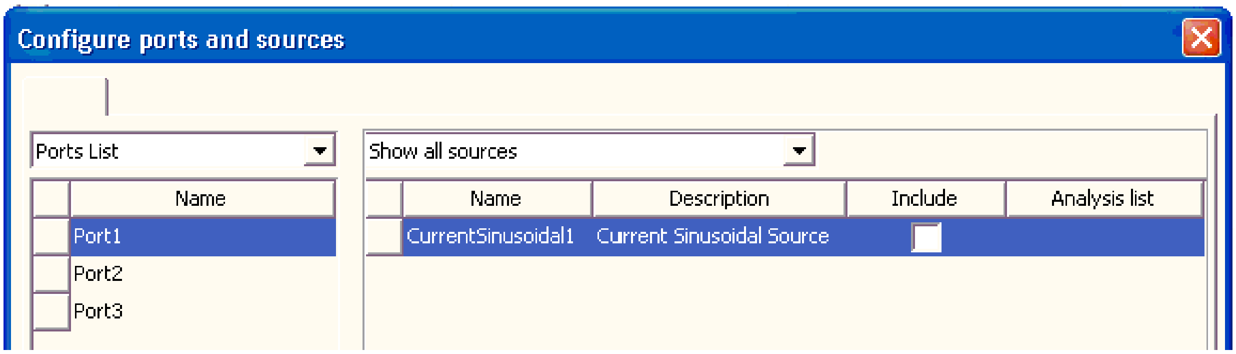
- Click in the **Name** field to assign a name other than the default ("CurrentSinusoidal*n*.").
- When all data for the port current source have been entered, click **OK**.

The Select Analysis window opens:



Use the check boxes to associate the source with one or more analyses. Click **OK**.

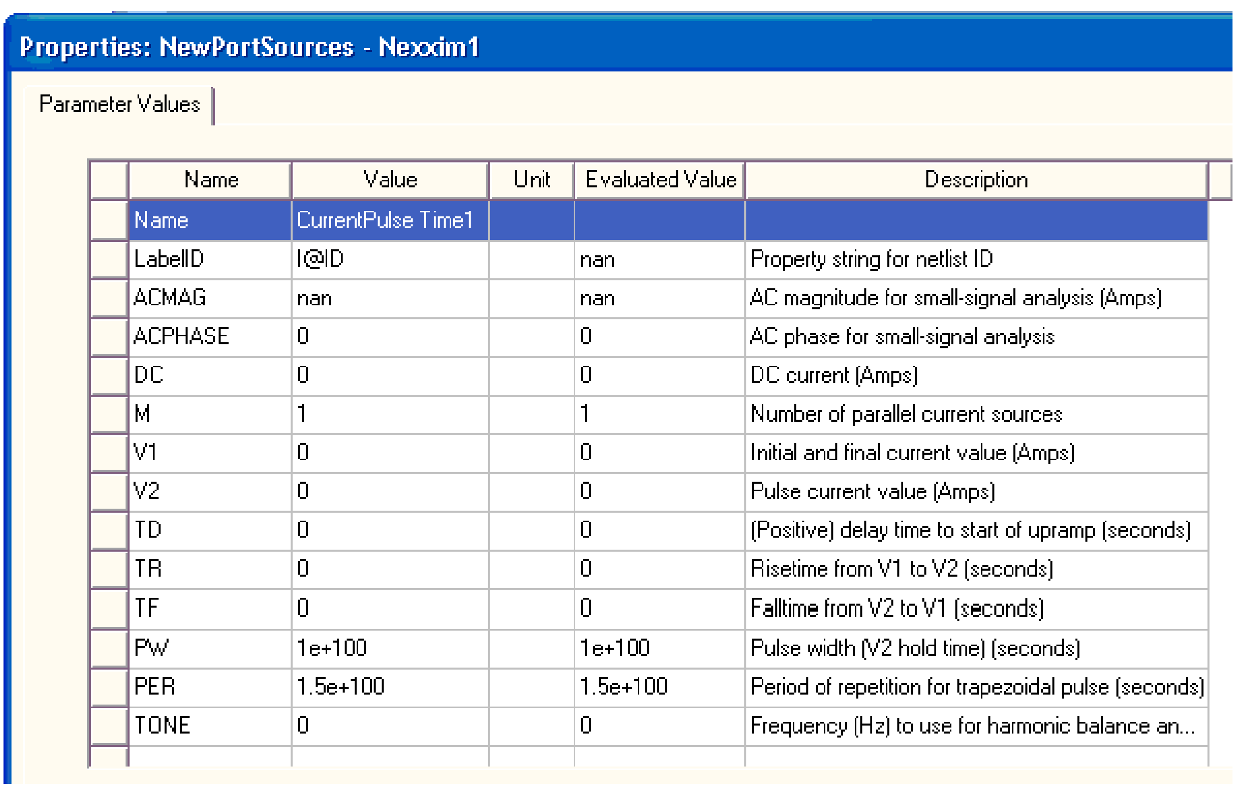
- The **Configure ports and sources** window box now shows the new source.



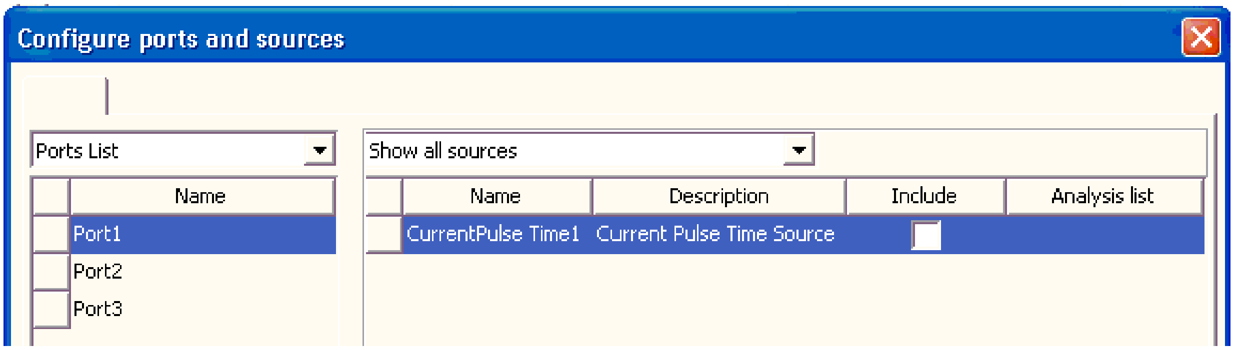
Pulse Port Current Source

In the **Add New Source** group box, select **Current** and **Pulsetime**. Click **Add to selected port**.

The Properties window displays the parameters for the Pulse Current Source:



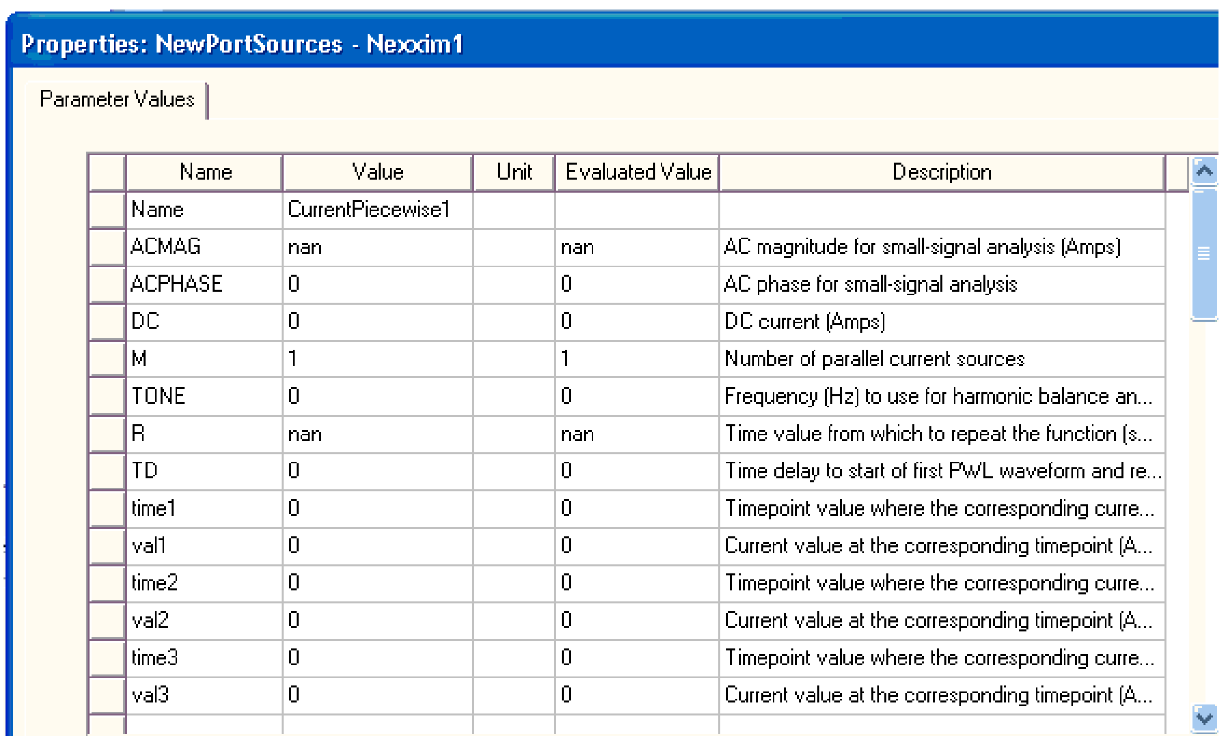
- When all data for the port current source have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.
- The **Configure ports and sources** window box now shows the new source.



Piecewise Linear Port Current Source

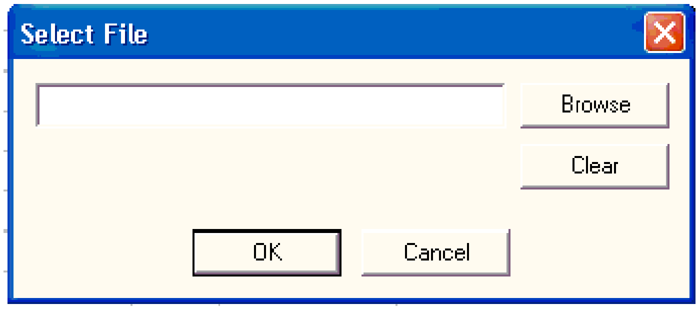
In the **Add New Source** group box, select **Current** and **Piecewise Linear**. Click **Add to selected port**.

The Properties window displays the parameters for the Piecewise Linear current source:



Name	Value	Unit	Evaluated Value	Description
Name	CurrentPiecewise1			
ACMAG	nan		nan	AC magnitude for small-signal analysis (Amps)
ACPHASE	0		0	AC phase for small-signal analysis
DC	0		0	DC current (Amps)
M	1		1	Number of parallel current sources
TONE	0		0	Frequency (Hz) to use for harmonic balance an...
R	nan		nan	Time value from which to repeat the function (s...
TD	0		0	Time delay to start of first PWL waveform and re...
time1	0		0	Timepoint value where the corresponding curre...
val1	0		0	Current value at the corresponding timepoint (A...
time2	0		0	Timepoint value where the corresponding curre...
val2	0		0	Current value at the corresponding timepoint (A...
time3	0		0	Timepoint value where the corresponding curre...
val3	0		0	Current value at the corresponding timepoint (A...

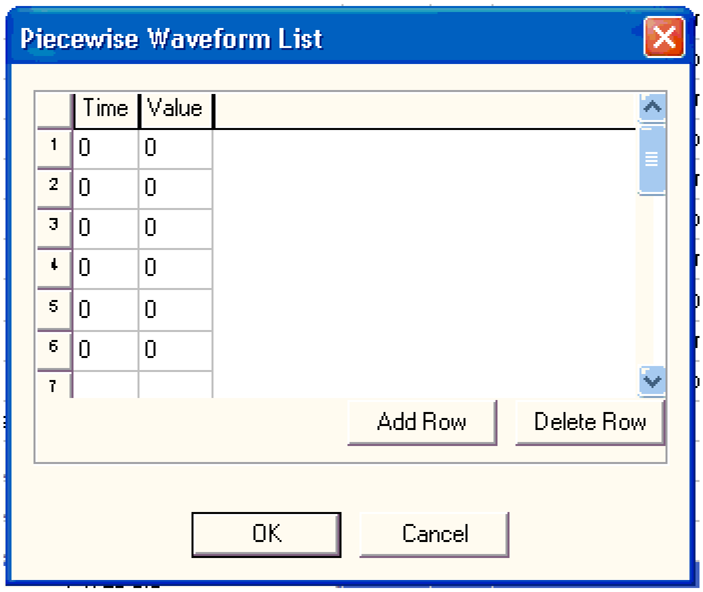
- Click in the **Name** field to assign a name other than the default (“CurrentSinusoidal*n.*”).
- If the PWL data is in a file, click **PWL_FILE** near the bottom of the property listing. The **Select File** window opens:



Use **Browse** to locate the file, then click OK to return to the Properties window.

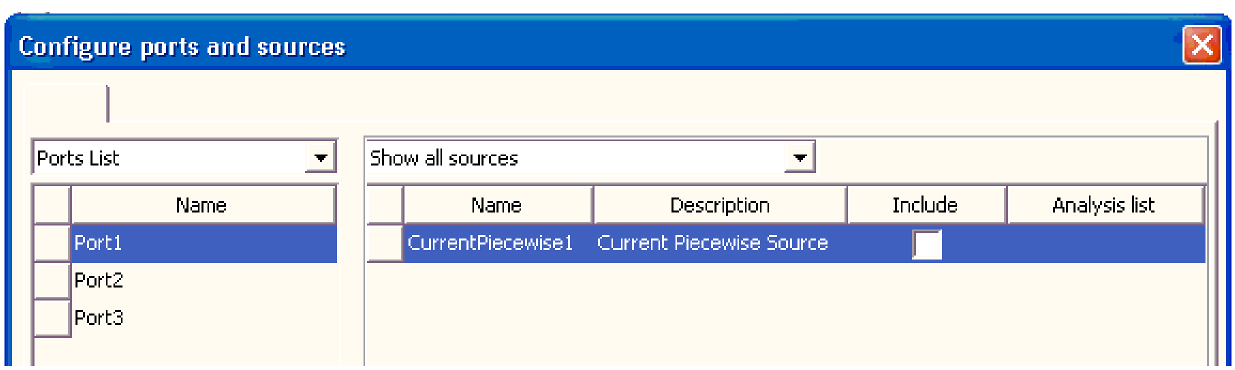
- The PWL data file should contain two columns of data: the first for time points and the second for corresponding data points.
- A pound sign (#) on the first line indicates a header which is used to set the units of measure for the time and data points. For example, the header “# ns mv” sets the time unit to nanoseconds and the data unit to millivolts. (The header is optional, however, and the default unit values are seconds and volts.)
- Comment lines are indicated by an exclamation point (!), asterisk (*), or semi-colon (;) and all data appearing on a comment line is ignored.
- Tabs or spaces are used to separate columns of data — commas are NOT allowed.

To enter the PWL data manually, click **PWL_DATA** to open a window for entering the PWL time points and corresponding current values:



When all time points and values have been entered, click **OK** to return to the **Properties** window.

- To specify noise data, click **Noise** in the **Value** field. See Adding Noise Data to a Port Source for further information.
- When all parameters have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.
- The **Configure ports and sources** window now shows the new source.



Port Voltage Source

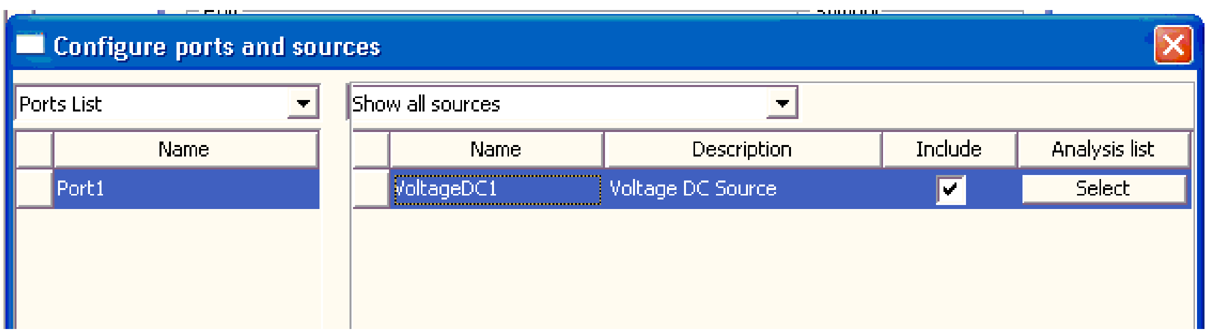
You can set up an interface port to be a voltage source for the circuit. In the **Add New Source** group box, select **Voltage** and **DC**, **Sinusoidal, Pulsetime**, **Piecewise Linear**, **PRBS**, or **IQ**.

DC Port Voltage Source

In the **Add New Source** group box, select **Voltage** and **DC**. Click **Add to selected port**. The Properties window displays the parameters for the DC voltage source:

Name	Value	Unit	Ev...	Descriptior
Name	VoltageDC1			
ACMAG	nan		nan	AC magnitude for small-signal
ACPHASE	0		0	AC phase for small-signal anal
DC	0		0	DC voltage (Volts)
CoSimulator	DefaultNetlist			
CosimDefinition				
Noise	Noise			
Netlist	V@ID %0 %1 *DC(DC=@DC) *ACMAG(AC @...			

  - Assign new values or units to the source parameters in the **Parameters** window by clicking in the field to be changed.
- Click the button in the **Noise Value** field. See Adding Noise Data to a Port Source for further information.
  - When all parameters have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.
- The **Configure ports and sources** window box now shows the new source.

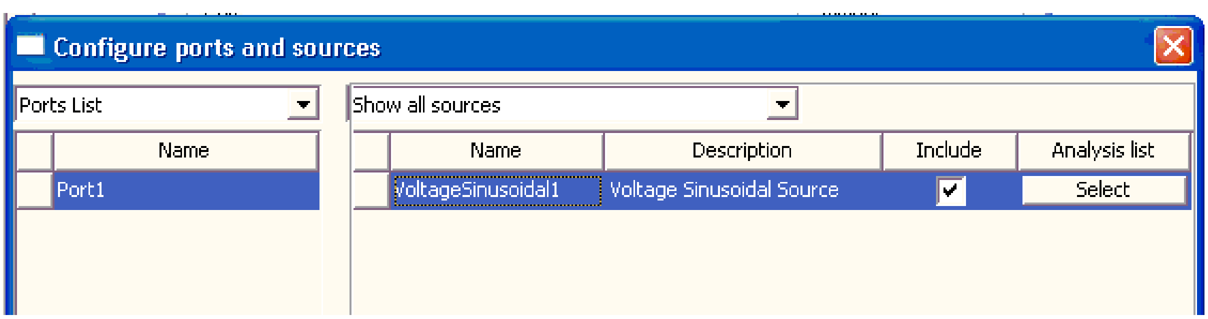


Sinusoidal Port Voltage Source

In the **Add New Source** group box, select **Voltage** and **Sinusoidal**. Click **Add to selected port**. The following display shows the Sinusoidal parameters.

Name	Value	Unit	Ev...	Descriptior
Name	VoltageSinusoidal1			
ACMAG	nan		nan	AC magnitude for small-signal .
ACPHASE	0		0	AC phase for small-signal anal
DC	0		0	DC voltage (Volts)
VO	0		0	Voltage offset from zero (Volts
VA	0		0	Voltage amplitude (Volts)
FREQ	1000000000		100...	Frequency (Hz)
TD	0		0	Delay to start of sine wave (se
ALPHA	0		0	Damping factor (1/seconds)
THETA	0		0	Phase delay
TONE	0		0	Frequency (Hz) to use for harn

- Click in the **Name** field to assign a name other than the default.
- Click the button in the **Noise Value** field. See Adding Noise Data to a Port Source for further information.
- When all data for the port voltage source have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.
- The **Configure ports and sources** window box now shows the new source.

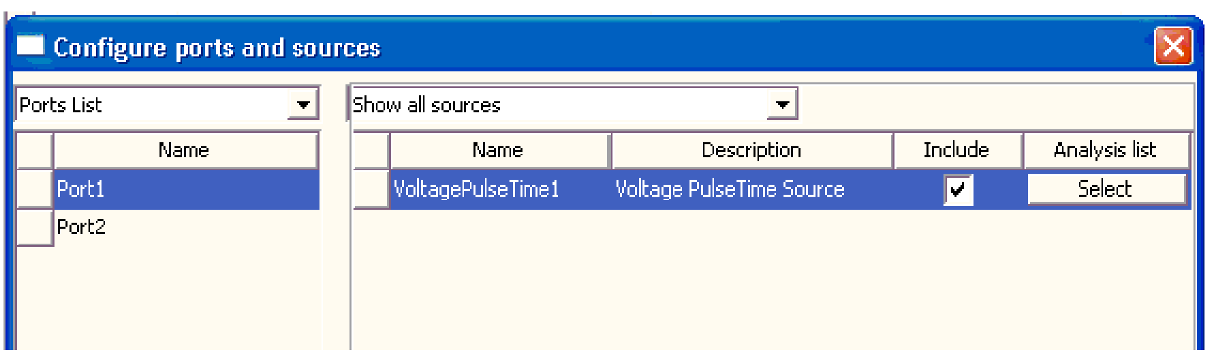


Pulse Port Voltage Source

In the **Add New Source** group box, select **Voltage** and **Pulsetime**. Click **Add to selected port**. Following are the parameters for the Pulse Voltage Source:

Name	Value	Unit	Ev...	Descriptio
Name	VoltagePulseTime1			
ACMAG	nan		nan	AC magnitude for small-signal
ACPHASE	0		0	AC phase for small-signal ana
DC	0		0	DC voltage (Volts)
V1	0		0	Initial and final voltage (Volts)
V2	0		0	Pulse voltage (Volts)
TD	0		0	(Positive) delay time to start of
TR	0		0	Risetime from V1 to V2 (secor
TF	0		0	Falltime from V2 to V1 (secon
PW	1e+100		1e+...	Pulse width (V2 hold time) (se
PER	1.5e+100		1.5...	Period of repetition for trapezc
TONE	0		0	Frequency (Hz) to use for har
CoSimulator	DefaultNetlist			
CosimDefinition				
Noise	Noise			
Netlist	V@ID %0 %1 *DC(DC=@DC) PULSE(?V1(@...			

- Click the button in the **Noise Value** field. See Adding Noise Data to a Port Source for further information.
  - When all data for the port voltage source have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.
- The **Configure ports and sources** window box now shows the new source.

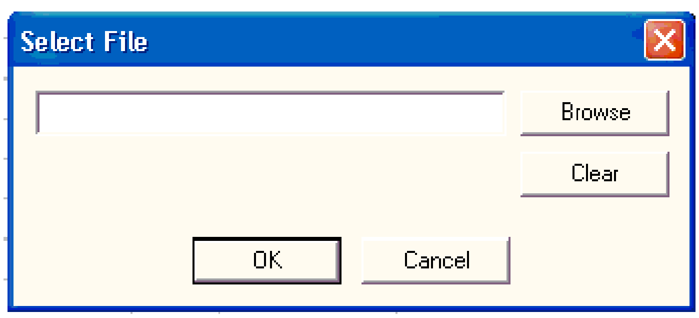


Piecewise Linear Port Voltage Source

Following are the parameters for the Piecewise Linear voltage source:

Name	Value	Unit	Ev...	Description
Name	VoltagePiecewise1			
ACMAG	nan		nan	AC magnitude for small-signal analysis (Volts)
ACPHASE	0		0	AC phase for small-signal analysis
DC	0		0	DC voltage (Volts)
R	nan		nan	Time value from which to repeat the function (s...
TD	0		0	Time delay to start of first PWL waveform and re...
TONE	0		0	Frequency (Hz) to use for harmonic balance an...
time1	0		0	Timepoint value where the corresponding volta...
val1	0		0	Value at the corresponding timepoint (Volts)
time2	0		0	Timepoint value where the corresponding volta...
val2	0		0	Value at the corresponding timepoint (Volts)
time3	0		0	Timepoint value where the corresponding volta...
val3	0		0	Value at the corresponding timepoint (Volts)
time4	0		0	Timepoint value where the corresponding volta...

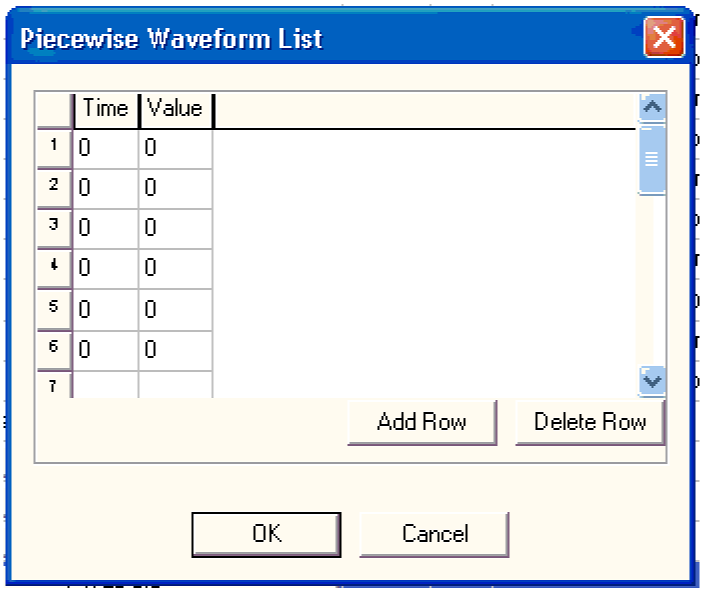
- Click the button in the **Noise Value** field. See Adding Noise Data to a Port Source for further information.
- If the PWL data is in a file, click the **PWL_FILE** button near the bottom of the property listing. The **Select File** window opens:



Use **Browse** to locate the file, then click OK to return to the Properties window.

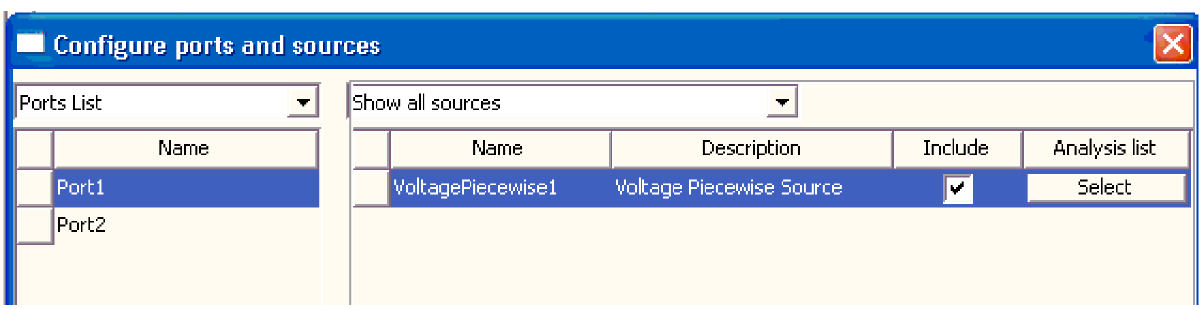
- The PWL data file should contain two columns of data: the first for time points and the second for corresponding data points.
- A pound sign (#) on the first line indicates a header which is used to set the units of measure for the time and data points. For example, the header “# ns mv” sets the time unit to nanoseconds and the data unit to millivolts. (The header is optional, however, and the default unit values are seconds and volts.)
- Comment lines are indicated by an exclamation point (!), asterisk (*), or semi-colon (;) and all data appearing on a comment line is ignored.
- Tabs or spaces are used to separate columns of data — commas are NOT allowed.

To enter the PWL data manually, click the **PWL_DATA** button to open a window for entering the PWL time points and corresponding current values:



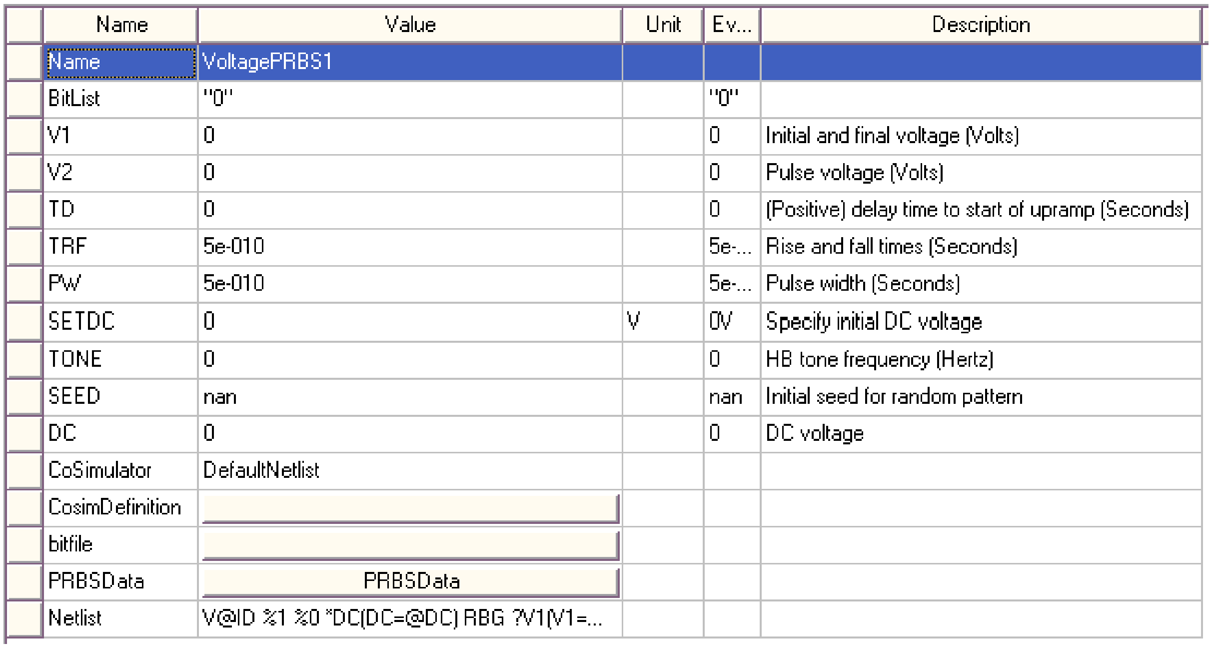
When all time points and values have been entered, click **OK** to return to the **Properties** window. When all parameters have been entered, click **OK** to open the **Select Analysis** window. Use the check boxes to associate the source with one or more analyses.

- The **Configure ports and sources** window box now shows the new source.



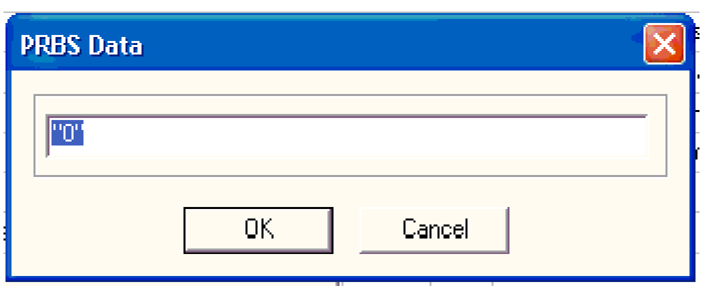
PseudoRandom Bit Source Port Voltage Source

In the **Add New Source** group box, select **Voltage** and **PRBS**. Click **Add to selected port**. The **Properties** window lists the PRBS source parameters, as in the following screenshot.



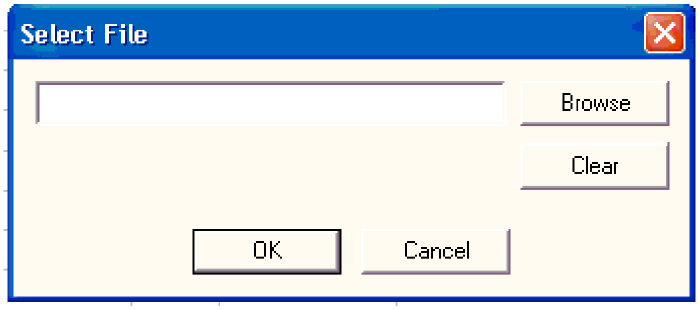
Name	Value	Unit	Ev...	Description
Name	VoltagePRBS1			
BitList	"0"		"0"	
V1	0		0	Initial and final voltage (Volts)
V2	0		0	Pulse voltage (Volts)
TD	0		0	(Positive) delay time to start of upramp (Seconds)
TRF	5e-010		5e-...	Rise and fall times (Seconds)
PW	5e-010		5e-...	Pulse width (Seconds)
SETDC	0	V	0V	Specify initial DC voltage
TONE	0		0	HB tone frequency (Hertz)
SEED	nan		nan	Initial seed for random pattern
DC	0		0	DC voltage
CoSimulator	DefaultNetlist			
CosimDefinition				
bitfile				
PRBSData	PRBSData			
Netlist	V@ID %1 %0 *DC(DC=@DC) RBG ?V1(V1=...			

- Click in the **Name** field to assign a name other than the default ("PowerIQ*n*").
- Assign new values or units to the source parameters in the **Parameters** window by clicking in the **Value** or **Unit** field to be changed.
- To assign the PRBS bitlist manually, click the **PRBS Data** button. The **PRBS Data** window opens:



Use the field in the group box to enter the data. Click **OK** to return to the Properties window.

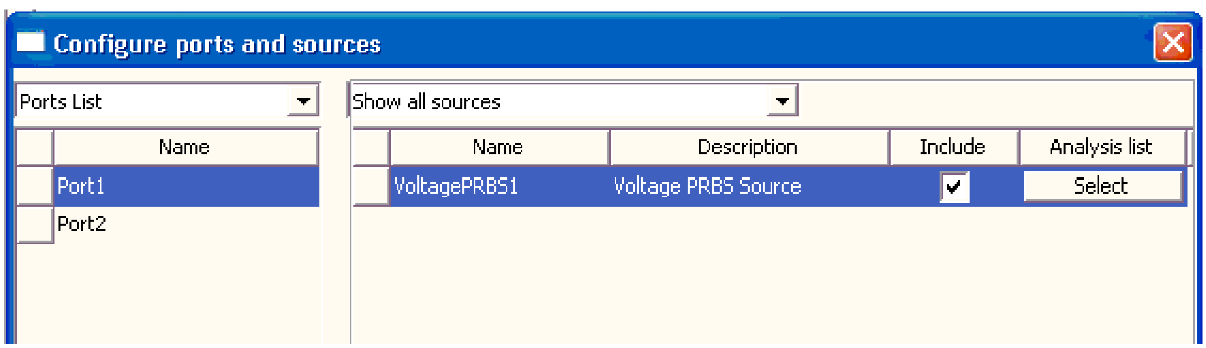
- If the bitlist data is in a file, click the **Bitfile** button near the bottom of the property listing. The **Select File** window opens:



- Click the **Browse** field to locate the data file. Click **OK** to return to the Properties window.
- When the PRBS data has been specified, click **OK** to close the **Properties** window and open the **Select Analysis** window.

Click the check boxes to associate the source with one or more analyses. Click **OK**.

- The **Configure ports and sources** window box now shows the new source.



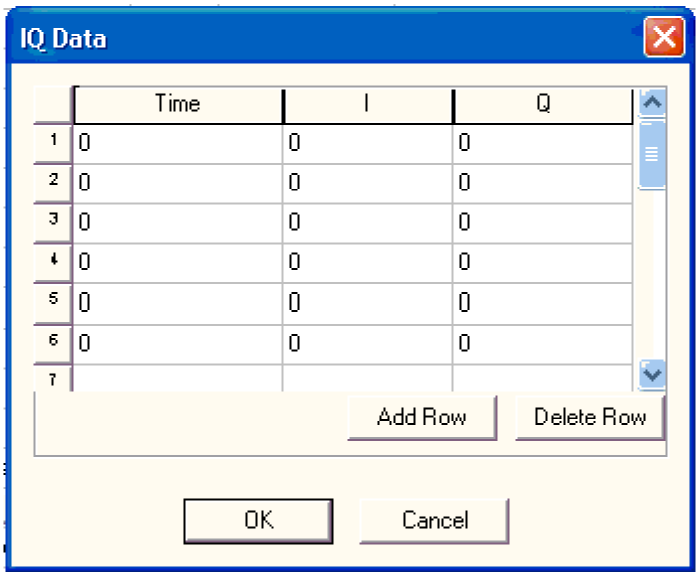
IQ-Modulated Port Voltage Source

In the **Add New Source** group box, select **Voltage** and **IQ**. Click **Add to selected port**. The **Properties** window lists the IQ source parameters, as in the following screenshot.

Name	Value	Unit	Ev...	Description
Name	VoltageIQ1			
FC	1000000000		100...	Carrier frequency (Hertz)
TS	nan		nan	Sampling time (Second)
DC	0		0	Source value at DC (Volt)
R	nan		nan	Repeat from time (Second)
TD	0		0	Time delay (Second)
V	1		1	Carrier amplitude, voltage-based (Volts)
VA	nan		nan	Carrier amplitude, power-based (Watts)
VO	0		0	Carrier amplitude offset (Volts or Watts)
RZ	50		50	Real carrier impedance (Ohm)
IZ	0		0	Imaginary carrier impedance (Ohm)
ALPHA	0		0	Damping factor (1/Seconds)
THETA	0		0	Phase delay
TONE	nan		nan	Source tone for HB analysis
time1	0		0	Timepoint value where the corresponding volta...
ival1	0		0	I value at the corresponding timepoint (Volts)
qval1	0		0	Q value at the corresponding timepoint (Volts)
time2	0		0	Timepoint value where the corresponding volta...

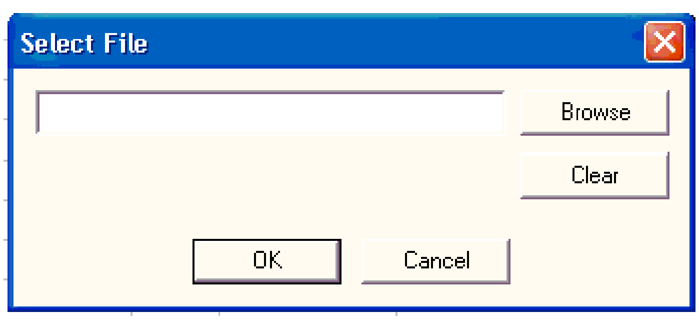
- Click in the **Name** field to assign a name other than the default (“PowerIQ*n*”).
- Assign new values or units to the source parameters in the **Parameters** window by clicking in the **Value** or **Unit** field to be changed.
- To assign the IQ time and data point manually, either enter the values in the *time*, *ival*, and *qval* parameter fields, or scroll down to the bottom of the property listing and click the**IQ**

Data button. The **IQ Data** window opens:



Use the field in the group box to enter the data. New rows are added automatically as you type. You can also add a row (to the end of the list) by clicking **Add Row**, or delete a selected row by clicking **Delete Row**. Click **OK** to return to the Properties window.

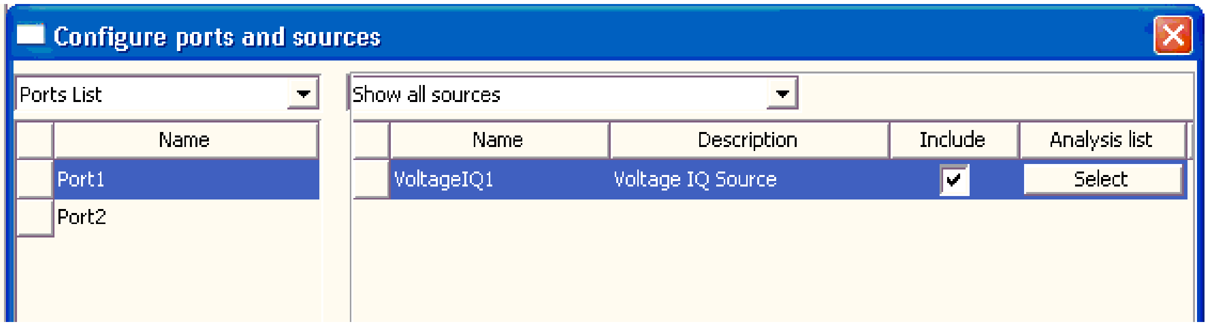
- If the IQ data is in a file, click the **File** button near the bottom of the property listing. The **Select File** window opens:



- Click the **Browse** field to locate the data file. Click **OK** to return to the Properties window.
- When the IQ data has been specified, click **OK** to close the **Properties** window and open the **Select Analysis** window. Click the check boxes to associate the source with one or more analyses.

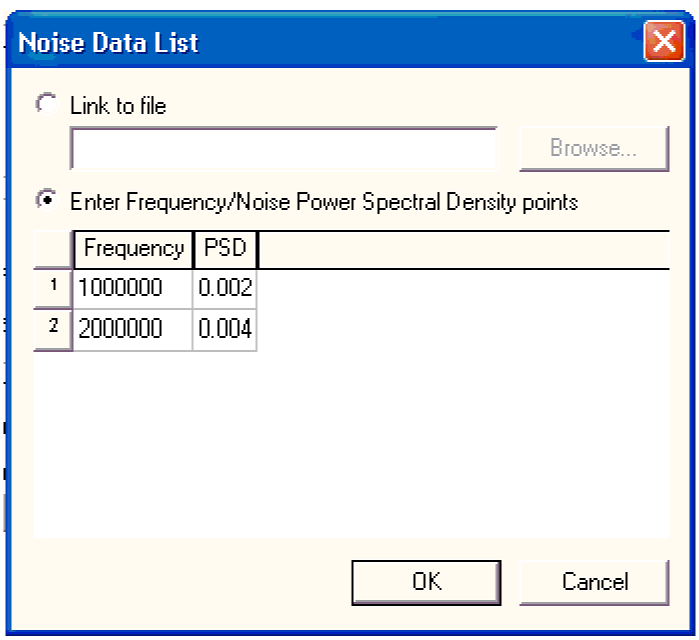
Click **OK**.

- The **Configure ports and sources** window box now shows the new source.

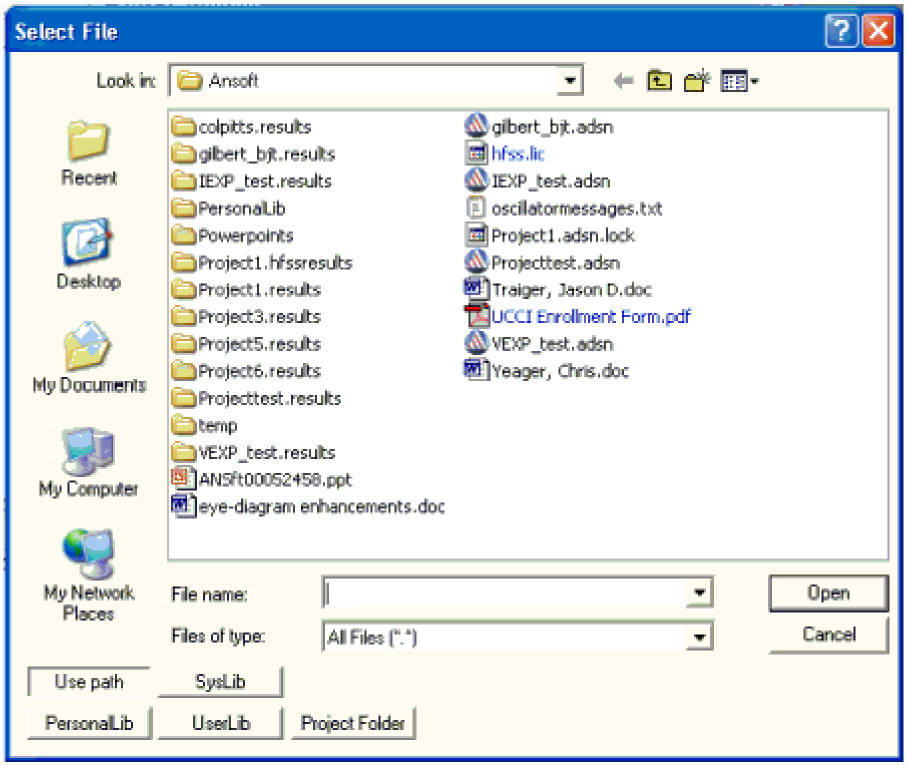


Adding Noise Data to a Port Source

Current and voltage interface port sources in schematics can specify the noise data for the noise analyses. Port power sources do not have the noise data parameter. To specify noise data, click in the **Noise Value** field on the **Parameters** tab of the **Source Selection** window for the current or voltage source. The **Noise Data List** window opens:



The default selection is **Link to File**. To use data from a file of noise data, click **Browse ...** to open the **Select File** window.



Click the **Look in** field to specify a directory, or locate the file by clicking **Use Path**, **PersonalLib**, **UserLib**, or **SysLib**. Within the folder, select the file from those listed, or type the name of the file in the **File Name** box.

- If you select **Use path**, type the name of the file in the **File Name** group box, or Click the **Look in** field to navigate to the file and record its name. Note that components or libraries imported with Use Path may not be portable when the project is moved to another machine.
- When **In project folder** is selected, references to the file in the design are relative to the directory where the project resides. In this case, the path is saved in the project **.adsn** file as a variable such as:

```
.lib '$PROJECTDIR/x_113854.lib'
```

- The variable $PROJECTDIR is expanded to the current location of the project when the design is converted to a netlist and run by an analysis tool. Otherwise, an absolute path is saved. If you move a project and its library files together to a new directory, preserve the file references by selecting the **Project Folder** option.
- The frequencies are in Hz and the corresponding noise power spectral densities are in Amp^2/Hz.

- To enter the noise data directly, click **Enter Frequency/Noise Power Spectral Density points**. Enter the frequencies (in Hz) and the corresponding PSDs (in Amp^2/Hz). Press **Tab** to advance to the next field.
- When all the noise data is as appropriate, click **OK** to close the **Noise Data List** window and return to the **Source Selection** window.

Noise data files have the format:

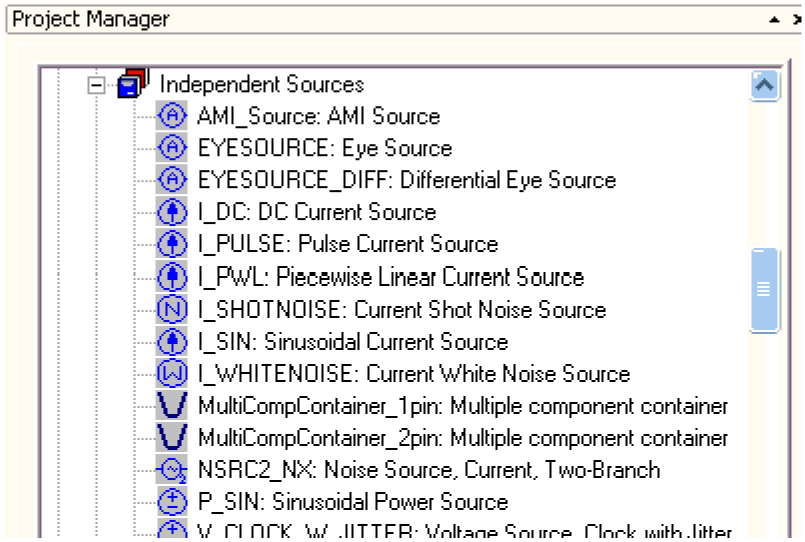
```
#Hz
 freq1 psd1
 freq2 psd2
  .
.
.
```

> **Note:** The **PersonalLib** (personal library) folder is located at **<Project Directory>/PersonalLib** ; the **userlib** (user library) folder is located at **<Installation Directory>/userlib** ; and the **syslib** (system library) folder is located at **<Installation Directory>/syslib**, where **<Project Directory>** is the location where the projects are saved and stored, and **<Installation Directory>** is the directory into which **Electronics Desktop** was installed during setup.
>
> **SysLib** is reserved for libraries supplied with **Electronics Desktop**.

Multiple Component Containers

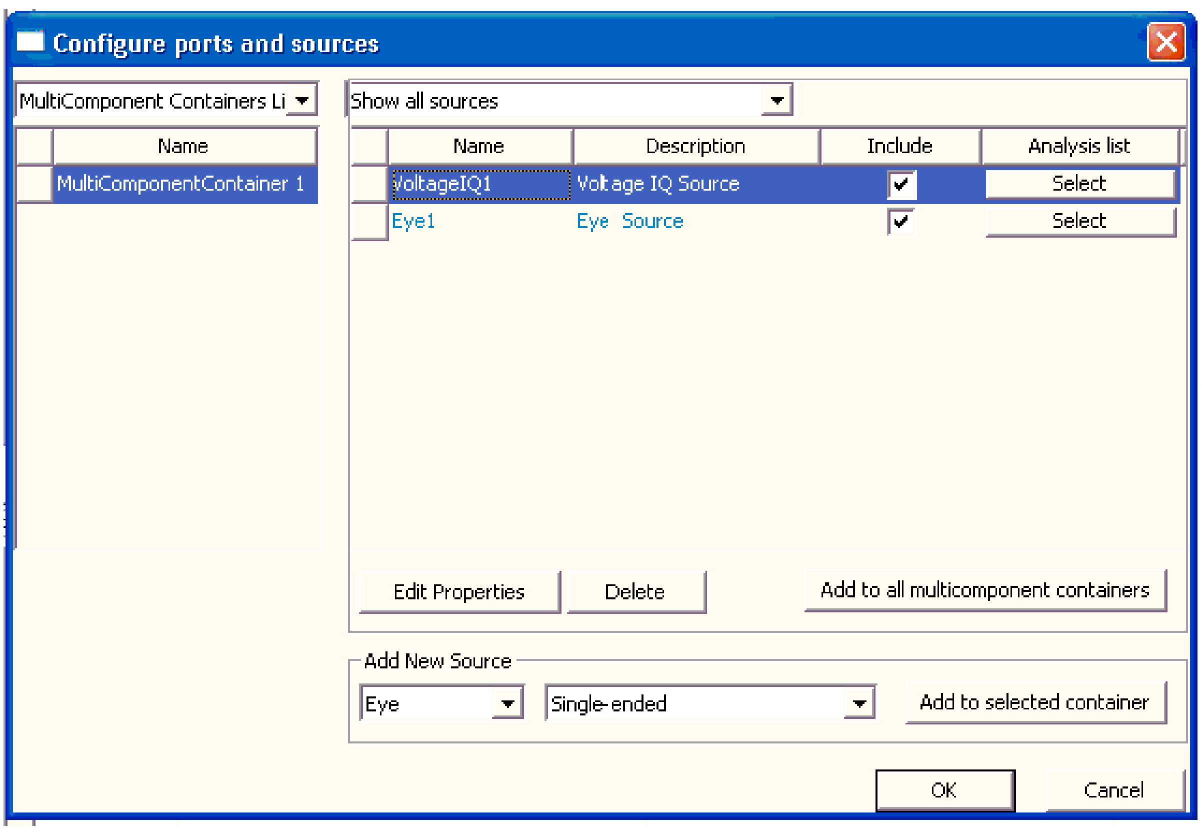
A multiple component container is a schematic object that can reference any number of sources, and allows easy reconfiguration of circuit sources and analyses.

To place a multiple component container, select the single-ended or differential version on the Independent Sources list:



Place the multiple component container on the node where the sources are to be applied.

Right-click the container and select Edit Source to open the Configure Sources window, open to the Multi Component Container tab:



Use the Add New Source group box to add Power, Current, Voltage, and Eye Sources to the list. As each source is defined, associate it with an available Analysis setup. The picture above shows two sources attached to one container. The two sources can both be included when they apply to different analyses.

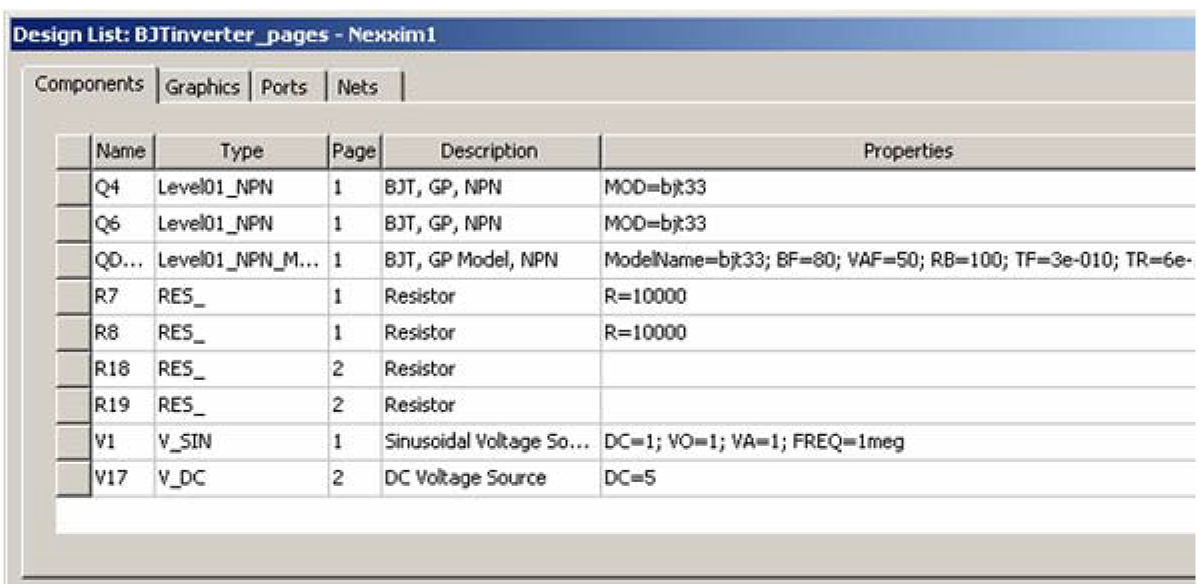
You can change the configuration of sources and analyses using this window. As long as the change does not affect a particular analysis, changing the configuration does not invalidate the results of that analysis.

Miscellaneous Schematic Operations

This topic describes a number of miscellaneous operations that can be performed using schematics including setting up mufti-page schematics, accessing include files and libraries, printing a schematic, generating a bill of materials, and adding ALTER blocks.

Schematic Design List window

The purpose of the **Design List** window is to provide a convenient way to view the components, graphics, ports and nets of a schematic. You can also use the window to delete or manipulate the properties related to a schematic. To access the window, on the menu, select **Schematic > List** .



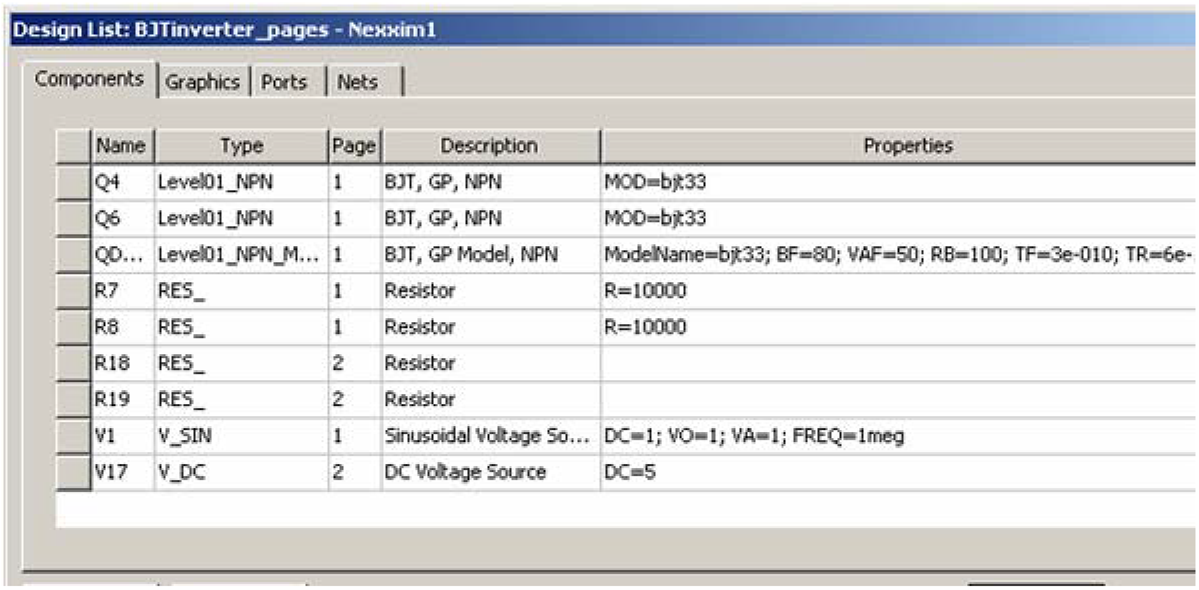
Name	Type	Page	Description	Properties
Q4	Level01_NPN	1	BJT, GP, NPN	MOD=bjt33
Q6	Level01_NPN	1	BJT, GP, NPN	MOD=bjt33
QD...	Level01_NPN_M...	1	BJT, GP Model, NPN	ModelName=bjt33; BF=80; VAF=50; RB=100; TF=3e-010; TR=6e-
R7	RES_	1	Resistor	R=10000
R8	RES_	1	Resistor	R=10000
R18	RES_	2	Resistor	
R19	RES_	2	Resistor	
V1	V_SIN	1	Sinusoidal Voltage So...	DC=1; VO=1; VA=1; FREQ=1meg
V17	V_DC	2	DC Voltage Source	DC=5

The **Schematic List** window contains four tabs: **Components**, **Graphics**, **Ports**, and **Nets**.

- Each tab contains a grid shows the items contained in the schematic.
- Each row in the grid contains one item, and each column displays an aspect of the item.

- Selecting a row zooms to the item in the schematic.
- Multiple items can be selected at once, which zooms/pans the schematic display so all selected items are visible.

Components Tab

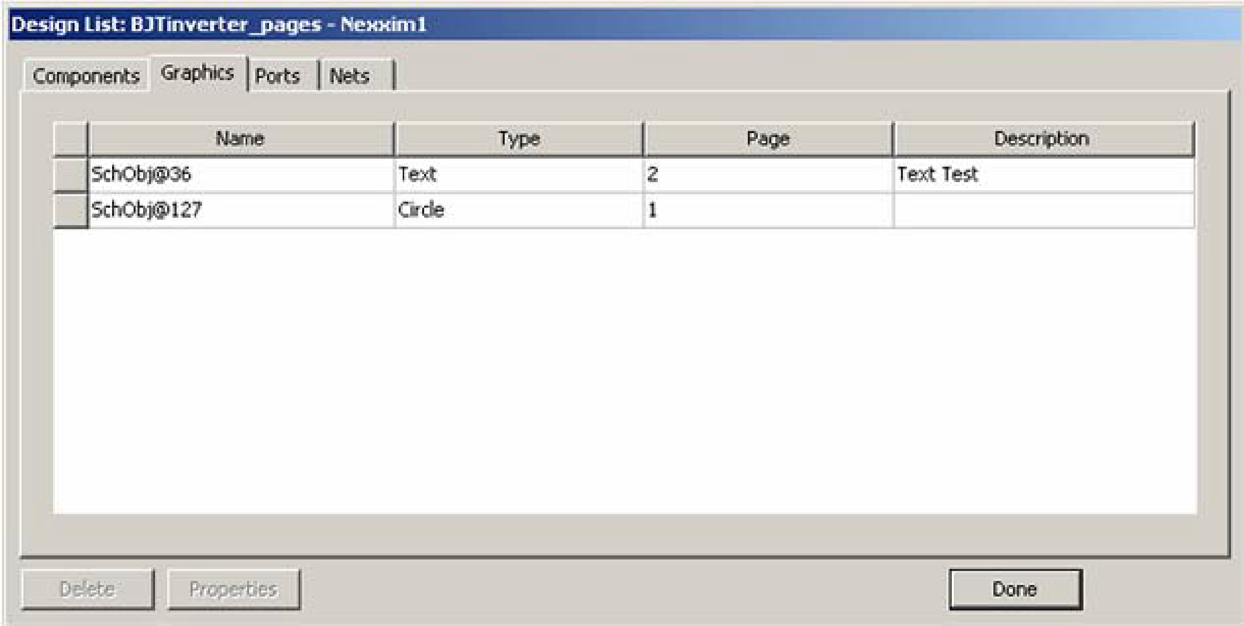

Design List: BJTinverter_pages - Nexxim1

Components | Graphics | Ports | Nets

Name	Type	Page	Description	Properties
Q4	Level01_NPN	1	BJT, GP, NPN	MOD=bjt33
Q6	Level01_NPN	1	BJT, GP, NPN	MOD=bjt33
QD...	Level01_NPN_M...	1	BJT, GP Model, NPN	ModelName=bjt33; BF=80; VAF=50; RB=100; TF=3e-010; TR=6e-
R7	RES_	1	Resistor	R=10000
R8	RES_	1	Resistor	R=10000
R18	RES_	2	Resistor	
R19	RES_	2	Resistor	
V1	V_SIN	1	Sinusoidal Voltage So...	DC=1; VO=1; VA=1; FREQ=1meg
V17	V_DC	2	DC Voltage Source	DC=5

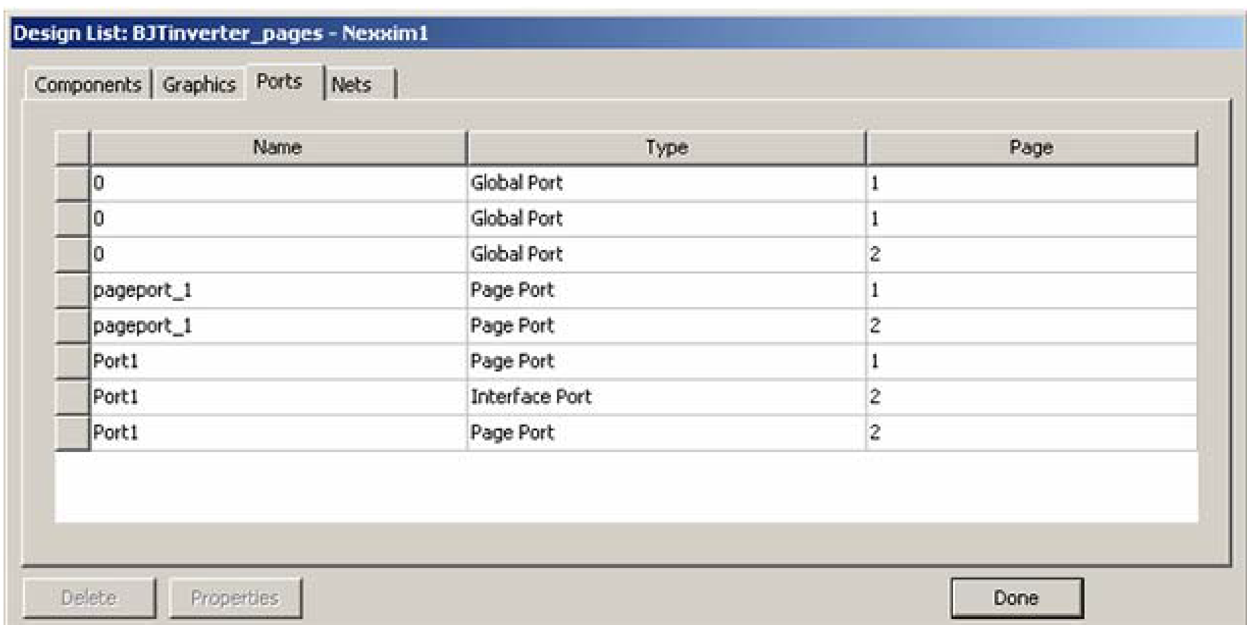
- **Name**: Shows the ID of the schematic component
- **Type**: Shows the component name from the library
- **Page**: Shows the page number on which that component is located
- **Description**: Shows the component description
- **Properties**: Shows the overridden properties of that component

Graphics Tab



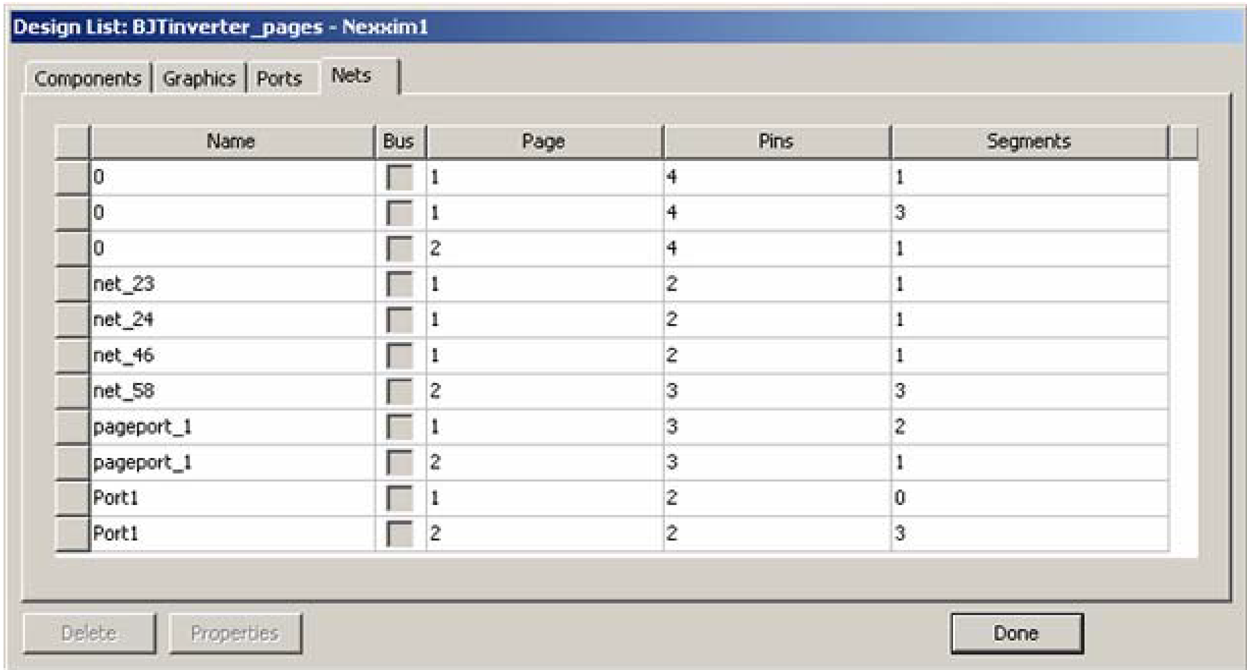
- **Name**: Shows the ID of the graphic
- **Type**: Shows the graphic type (Circle, Rectangle, Arc, Text, etc.)
- **Page**: Shows the page number on which that graphic is located
- **Description**: Shows the text for textual graphics, and is blank for non-textual graphics

Ports Tab



- **Name**: Shows the name of the port
- **Type**: Shows the port type (Interface, Page, or Global)
- **Page**: Shows the page number on which that port is located

Nets Tab



Name	Bus	Page	Pins	Segments
0		1	4	1
0		1	4	3
0		2	4	1
net_23		1	2	1
net_24		1	2	1
net_46		1	2	1
net_58		2	3	3
pageport_1		1	3	2
pageport_1		2	3	1
Port1		1	2	0
Port1		2	2	3

- **Name**: Shows the name of the net
- **Bus**: Is checked if the item is a bus, and is unchecked for normal wires
- **Page**: Shows the page number on which that net is located.
  Please note that a net may appear more than once if it spans multiple pages
- **Pins**: Number of pins connected to that net
- **Segments**: Number of individual line segments that comprise that net

Setting Up Multi-Page Schematics

Setting up multi-page schematics involves two activities: adding pages, and adding and naming page connectors to establish electrical connectivity between them as necessary.

To add a page or pages to a schematic:

1. Open the schematic to which you want to add a new page or pages.
2. Click the **New Page** tab.

You are now editing the new page, 2. Repeat as necessary to add additional pages.

To establish a common electrical connection between pages, place a page connector on each page that shares the connection. To do this on each page:

1. Select the page.
2. Do either of the following:
   - From the **Draw** menu, select **Page Connector**.
   - From the **Schematic Draw** toolbar, click the **Page connector** icon.

The cursor, now associated with a page connector symbol for placement, moves to the center of the schematic window. To place the connector, click at the appropriate location.

> **Tip:**
>
> You can rotate a page connector before placing it by repeatedly pressing **R** on your keyboard. Each press rotates the connector 90º counterclockwise.

If Multiple Placement is activated for page connectors in the **Schematic Options** window, place additional connectors by clicking at additional locations. To stop placing page connectors during multiple placement, do either of the following:

- Press **Enter**, the **Spacebar**, **Backspace**, or **Esc**.
- Right-click, then select **Place and Finish**, **Finish**, **Cancel**, or **Back**.

> **Note:**
>
> To ensure electrical connectivity among schematic elements, the pins of placed page connectors snap to a 100-mil (2.54-millimeter) grid. This snapping cannot be turned off, and the spacing of the connectivity grid cannot be adjusted.

Within a design and at the same hierarchical level, all page connectors with the same name act as a common electrical connection. Therefore, once you have finished placing page connectors that you want to be electrically common, you must ensure that they all have the same name. To change the name of a page connector, do either of the following:

- Open its **Properties** window by double-clicking the connector's schematic symbol, type a new name into the **Value** cell for the port's **Name** property, then click **OK**.
- Click the connector. In the **Parameters** tab of the **Properties** window, click the **Value** cell for the **NetName** parameter, type the new name, and press **Enter**.

> **Note:**
>
> Changing the name of any member of a group of same-named page connectors renames all members of the group with the new name.

Setting Up Hierarchical Schematics

A hierarchical schematic is a schematic with one or more levels of subcircuits. To create a hierarchical schematic, do either of the following:

- Start in an existing schematic and add a subcircuit to it.
- Copy an existing schematic into another schematic to make the first schematic a subcircuit of the second.

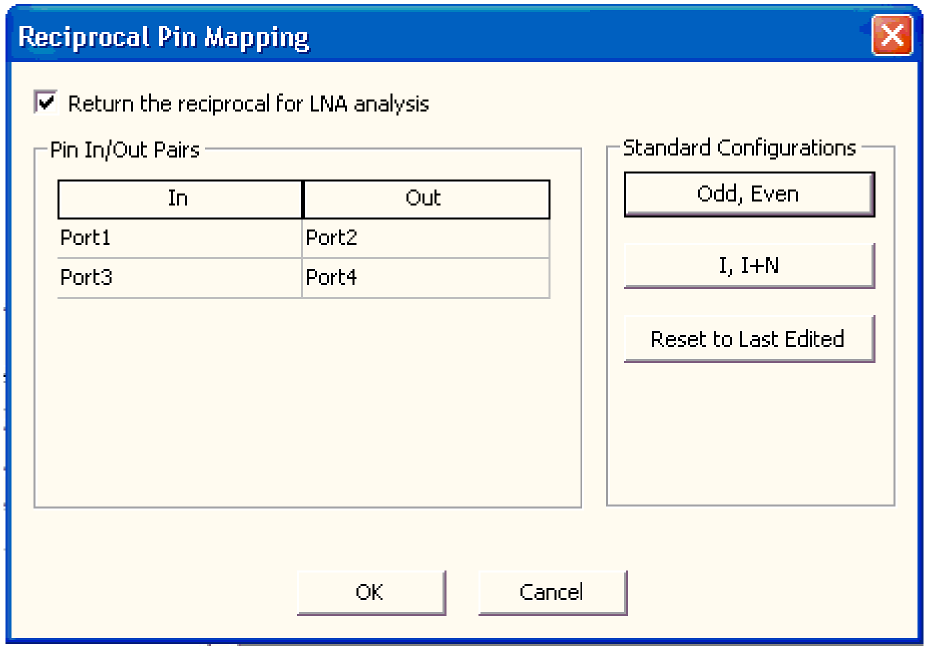
Adding a New SubCircuit to a Design

To add a new subcircuit to an existing design, complete the following steps.

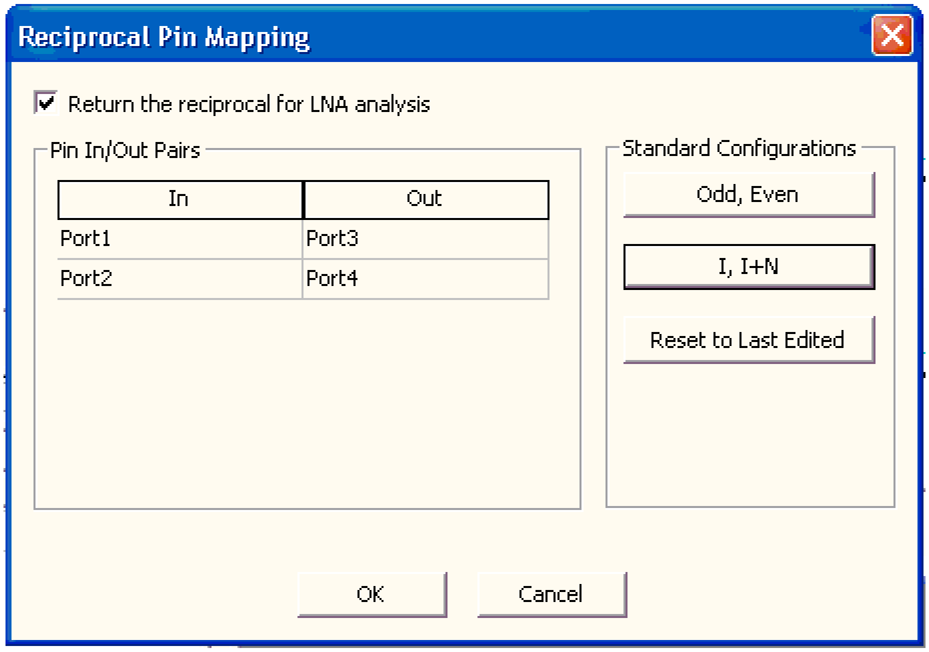
1. From the **Circuit** or **HFSS 3D Layout** menu, select **Add SubCircuit**. Then select the type of subcircuit to add.

> **Note:** To add an **HFSS Link**, **HFSS Extractor Link**, **Q3D Extractor Link**, or **SIwave Link**, the associated program must be installed on the system.

2. In a Circuit project, the **Schematic Editor** opens. In an HFSS 3D Layout project the **Layout Editor** opens. Use the editor to create the schematic or layout for the subcircuit. Add ports in the schematic or pins in the layout to provide connections to the outer circuit. An icon for the subcircuit appears in the **Project Manager** window (on the **Project Manager** window, expand the **Project Tree** and [*active design folder*]), with the designation "U*n*" (e.g., "U1" is the first subcircuit created). A symbol for the subcircuit also appears in the schematic/layout.
3. Connect the subcircuit to the rest of the circuit as required.
4. Optionally, specify that the subcircuit S-parameters is inverted (returning the reciprocal of the S-matrix) after running LNA, so the subcircuit can be used for de-embedding. Please note that *to invert correctly, the subcircuit MUST have an even number of ports or pins, arranged as input-output pairs. Subcircuits with odd numbers of ports show **Not Valid** as the value of the **LNA Reciprocal** property.*
   - Click the schematic symbol for the subcircuit to open the Property window.
   - Click in the **LNA Reciprocal Value** field. If the reciprocal operation is valid for the subcircuit, the button initially reads **LNA Reciprocal OFF**. The Reciprocal Pin Mapping window opens:



- To enable the reciprocal operation, click the box **Return the reciprocal for LNA analysis**.
- Click **Standard Configurations** to select the pin mapping configuration used by your subcircuit. The illustration above shows the **Odd, Even** configuration with four ports; the odd-numbered ports 1 and 3 are the inputs and the even-numbered ports 2 and 4 are the corresponding outputs.
- The following figure shows the **I, I+N** configuration; now the inputs are ports 1 and 2, while the corresponding outputs are ports 3 and 4.



- Click **Reset to Last Edited** to restore the configuration that was initially read on the schematic.
- Click **OK** to return to the schematic editor. The **Value** field of the **LNA Reciprocal** property now reads **LNA Reciprocal ON**.

Copying an Existing Subcircuit within a Design

To copy an existing subcircuit within a design:

1. Open the design that contains the subcircuit you want to copy.
2. Do either of the following:
   - From the **Project Manager** window, right-click the icon for the subcircuit and select **Copy**.
   - In the design, select the subcircuit symbol, then click **Copy** on the **Draw** menu.
3. Do either of the following:
   - From the **Edit** menu, select **Copy**.
   - From the keyboard, press **Ctrl** +**V**.

Within the schematic window, the cursor is associated with an *N*-port symbol for subcircuit placement. To place the subcircuit, click at the appropriate location.

> **Tip:**
>
> You can rotate the symbol for a subcircuit before placing it by repeatedly pressing **R** on your keyboard. Each press rotates the component 90º counterclockwise.

If Multiple Placement is activated for components in the **Schematic Options** window, place additional instances of the subcircuit by clicking at additional locations. To stop placing subcircuits during multiple placement, do either of the following:

- Press **Enter**, the **Spacebar**, **Backspace**, or**Esc** on your keyboard.
- Right-click and select **Place and Finish**, **Finish**, **Cancel**, or **Back**.

> **Note:**
>
> To ensure electrical connectivity among schematic elements, the pins of placed subcircuit symbols snap to a 100-mil (2.54-millimeter) grid. This snapping cannot be deactivated, and the spacing of the connectivity grid cannot be adjusted.

Moving Between Designs in a Schematic Hierarchy

You can move between a project's different designs and design instances by double-clicking their icons in the project tree. Specific menu commands are also available for moving up and down one level.

To move down one level:

1. In the schematic from which you want to move, select the symbol for the design to which you want to move.
2. From the **Schematic** menu, select **Push Down**.

To move up one level:

1. In the schematic from which you want to move, select the symbol for the design to which you want to move.
2. From the **Schematic** menu, select **Pop Up**.

Accessing Include Files on the Schematic Editor

A schematic design can reference an include file. For example, frequency-dependent model data is commonly written into a separate file rather than entered into a Property Window.

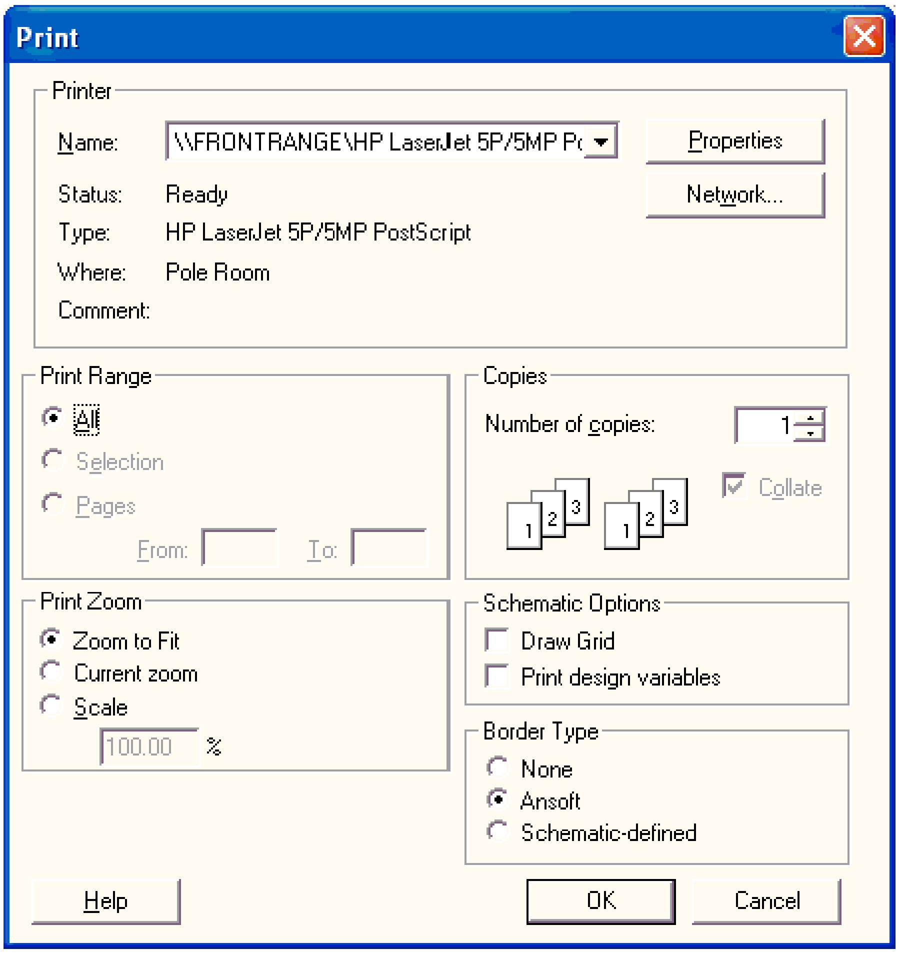
Accessing Library File Blocks on the Schematic Editor

A schematic design can reference a library file block.

Printing a Schematic

To print the schematic that is in the active window:

1. Right-click in the schematic window and select **Print** on the menu (or select **Print** on the **File** menu) to open the **Print** window:



2. Use the controls to specify how you wish the print to appear.
   - In the **Schematic Options** group box, specify whether or not to draw the grid and print the design variables.
   - In the **Border type** group box, select the type of border to print around the schematic.

     Selecting the **Ansys** border type places a single line box around the schematic and adds the current date and page number to the print.

     Selecting the **Schematic-defined** border type prints the border as defined through the

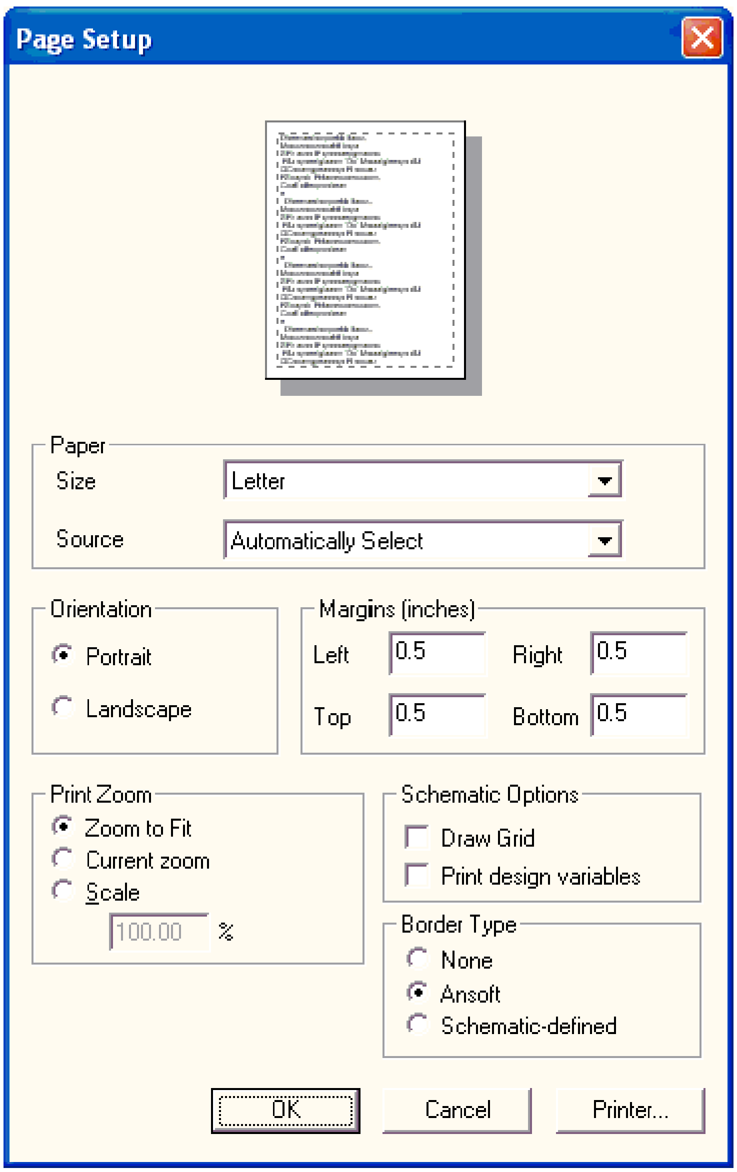
Schematic->Page Borders menu option at the size specified. Typically, select this option when printing out at a specific scale (e.g., 100%) to ensure that the border is visible. Use **File>Print Preview** to preview the effect of your choices.

- Make any appropriate selections in the **Print Range**, **Copies**, and **Print Zoom** group boxes.

3. Click **OK** to print the schematic, or click **Cancel** to cancel the operation.

To set up the page layout before printing:

1. Select **Page Setup** on the **File** drop-down menu to open the **Page Setup** window.



- Use the **Paper**, **Orientation**, **Margins**, and **Print Zoom** controls to specify how you wish the print to appear. In the **Schematic Options** group box, specify whether or not to draw the grid and print the design variables.
- In the **Border type** group box, select the type of border to print around the schematic.

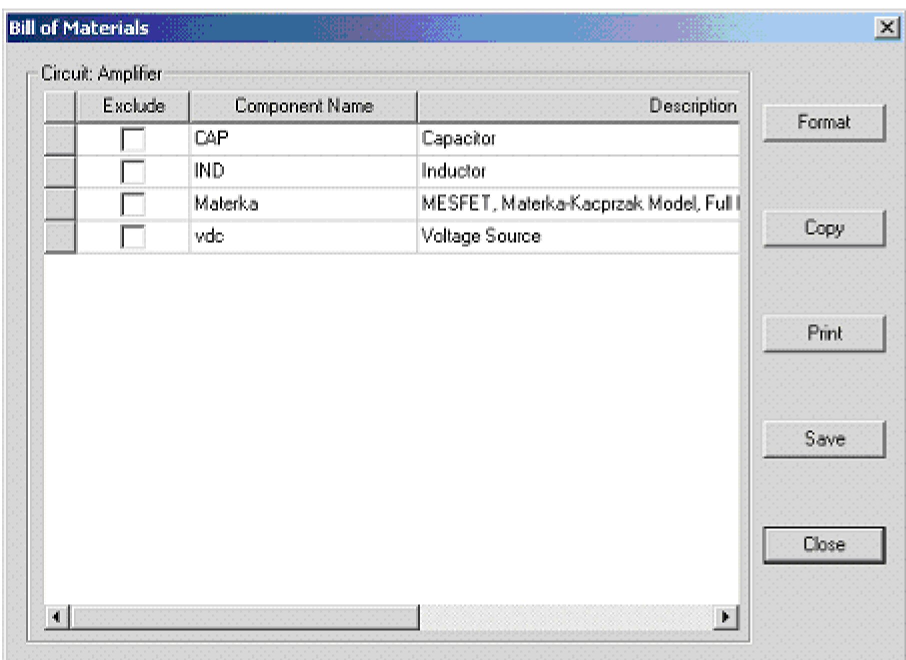
  — Selecting the **Ansys** border type places a single line box around the schematic and adds the current date and page number to the print.

  — Selecting the **Schematic-defined** border type prints the border as defined through the Schematic->Page Borders menu option at the size specified. Typically, select this option when printing out at a specific scale (e.g., 100%) to ensure that the border is visible. Use **File>Print Preview** to preview the effect of your choices.

3. Click **OK** to close the **Page Setup** window with your settings.
4. Select **Print Preview** on the **File** menu to see a preview of the print.

Bill of Materials

The Bill of Materials feature allows you to generate a configurable list of parts that are used in a schematic design. The list can be modified, viewed, printed, or exported to a file. To generate a Bill of Materials, with the design loaded into the schematic editor, select **Bill of Materials** on the **Product** menu.



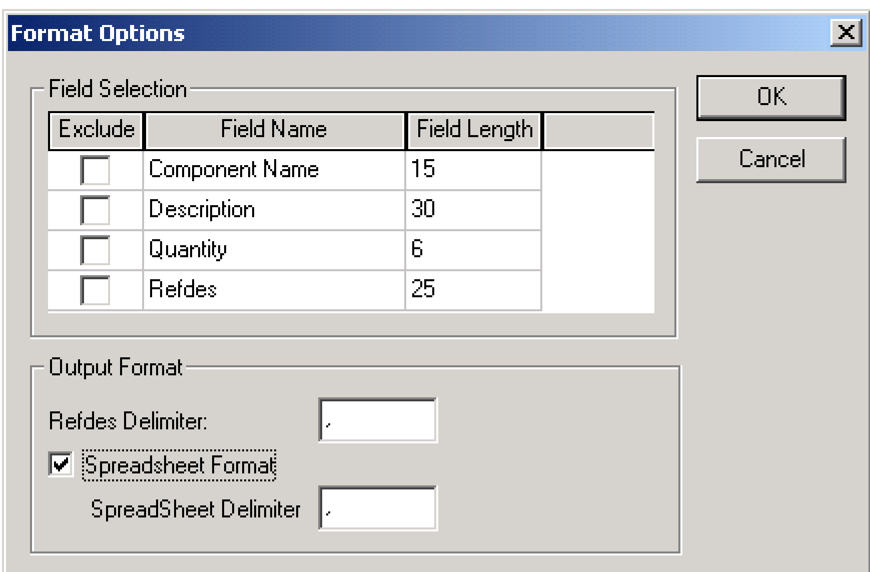
The **Bill of Materials** window displays the list of components used in a design. The following controls are available:

- **Format** allows you to format the display. In particular, the **Format** window allows you to **Display Titles**, specify a **Spreadsheet Format**, and selectively **Exclude** fields to be displayed.
- **Copy** copies the list to the Windows clipboard.
- **Print** opens the standard Windows print window.
- **Save** saves the list to a file. You choose to save the bill of materials in text format (using either a **.bom** filename extension or a **.txt** extension), or in spreadsheet format (using a **.csv** filename extension).
- **Exclude**, when selected, removes a component on the list.
- To modify a value in the display, click directly on the value.

The default fields displayed in the Bill of Materials include:

- Component Name
- Description
- Quantity
- Refdes (Reference Designator)

To change the default fields that are displayed, click **Format** to open the **Format Options** window.



- **Exclude**, when selected, removes a field to be displayed.
- **Display Titles**, when selected, specifies that titles and headers can be included in the display.
- **Refdes Delimiter** specifies the delimiter that is used for the reference designator.
- **Spreadsheet Format**, when selected, displays the list in a spreadsheet format.

- **SpreadSheet Delimiter**, when Spreadsheet Format is selected, specifies the delimiter that is used for the spreadsheet display.
- To modify a value in the display, click directly on the value.

Adding ALTER Blocks

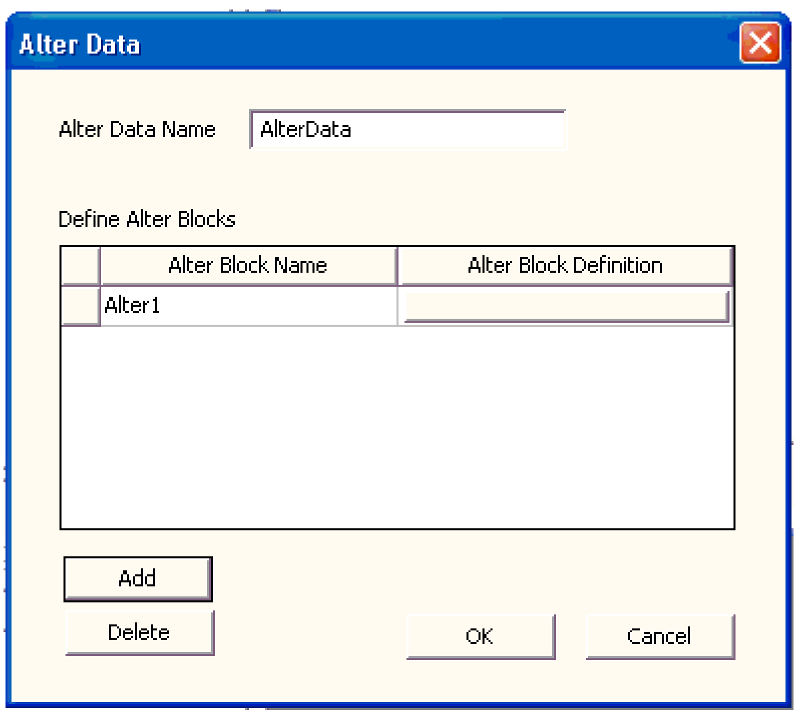
An ALTER statement directs Nexxim to resimulate a design using changed values for parameters. See ALTER Statements for details on the netlist representation of ALTER statements.

In a schematic, define one or more sets of ALTER blocks. ALTER data sets are added to the **Project Manager** window. Later, select one of the ALTER data sets on the **Project Manager** window to apply within a solution setup.

Defining a Set of ALTER Blocks in a Schematic

To create a set of ALTER blocks, right-click the icon for the Nexxim design in the **Project Manager** window and select **Add Alter Block**. (Alternatively, right-click the Analysis icon in the **Project Manager** window and select **Add Alter Block**).

The **Alter Data** window opens:

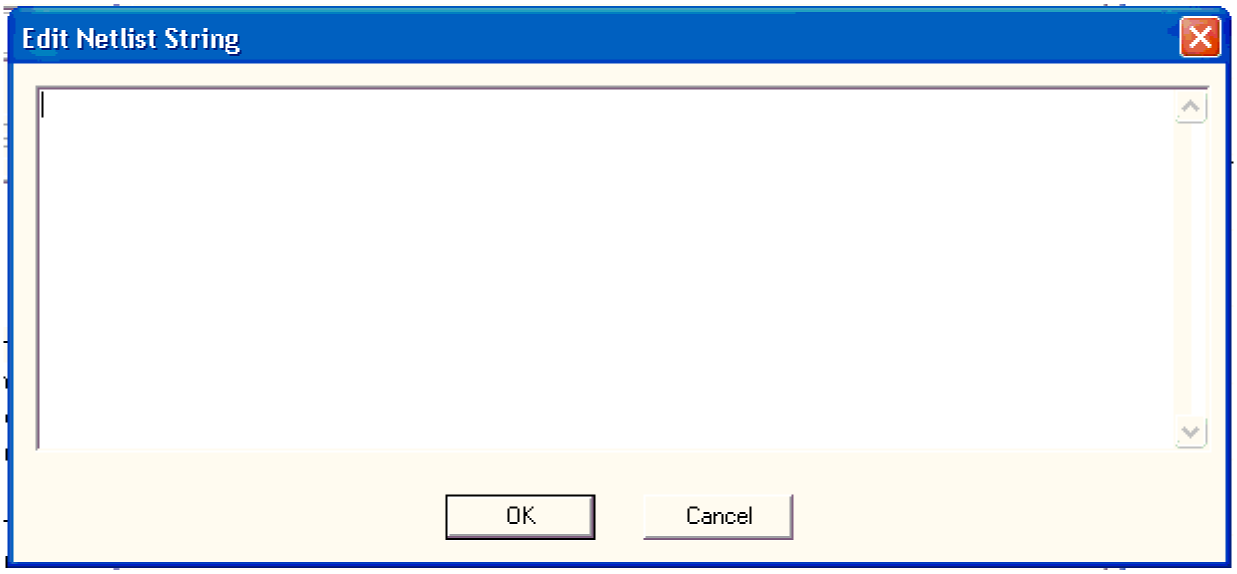


Click the **Alter Data Name** field to rename the ALTER dataset. The **Alter Data Name** is the name that appears for this ALTER dataset in the **Project Manager** window.

Initially, no ALTER blocks are defined. Click **Add** to start a new ALTER block (**Alter1** in the picture above).

- Click the name of the ALTER block to rename it if appropriate. The name of the ALTER block in the **Alter Data** window is the one that is written out to the netlist before simulation.

- Click in the **Definition Value** field to open the **Edit Netlist String** window:



- If the cursor does not appear immediately, click in the window to activate it.
- Type the text of the ALTER block one line at a time.

> **Note:** Start a new line by typing **Ctrl+Enter** at the end of the current line. Pressing **Enter** (without holding down **Ctrl**) ends the edit and closes the window..

When you are finished entering the ALTER block lines, click **OK** to return to the **Alter Data** window, or click **Cancel** to end the edit without defining anything.

- Use **Add** on the **Alter Data** window to add as many ALTER blocks to the set as appropriate. All the ALTER blocks in the set is written to the netlist when the set is selected for an analysis.
- Use **Delete** on the **Alter Data** window to delete ALTER block definitions that should not be used as part of the set being defined. Click the ALTER block to select it for deletion.
- When you have defined all the blocks in the set, click **OK** on the **Alter Data** window. The set of Alter blocks appears as an icon with the **Alter Data Name** entry under the Analysis icon in the **Project Manager** window.

> **Note:** nbThe names for the ALTER data sets in the **Project Manager** window are used for selection in the solution setup. They are independent of the names used for the individual ALTER blocks assigned in the **Alter Data** window. You can rename an Alter set by right-clicking on its icon in the **Project Manager** window and selecting **Rename** on the menu. Renaming the set does not change the names of any of the ALTER blocks defined for the set.

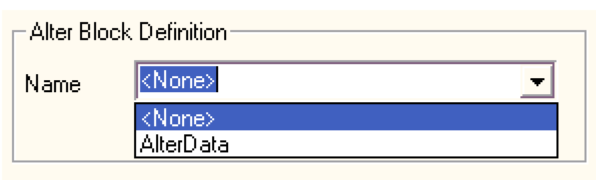
- To cancel the ALTER block entry process without creating any blocks, click **Cancel** on the **Alter Data** window.

Editing Alter Data Definitions

To review or edit the names or definitions in any ALTER dataset, double-click the icon for that set in the **Project Manager** window to open the **Alter Data** window.

Selecting an ALTER Data Set for Simulation

When one or more sets of ALTER blocks have been defined, the **Solution Setup** window for any analysis type has a field for selecting an Alter set to use as part of the simulation:

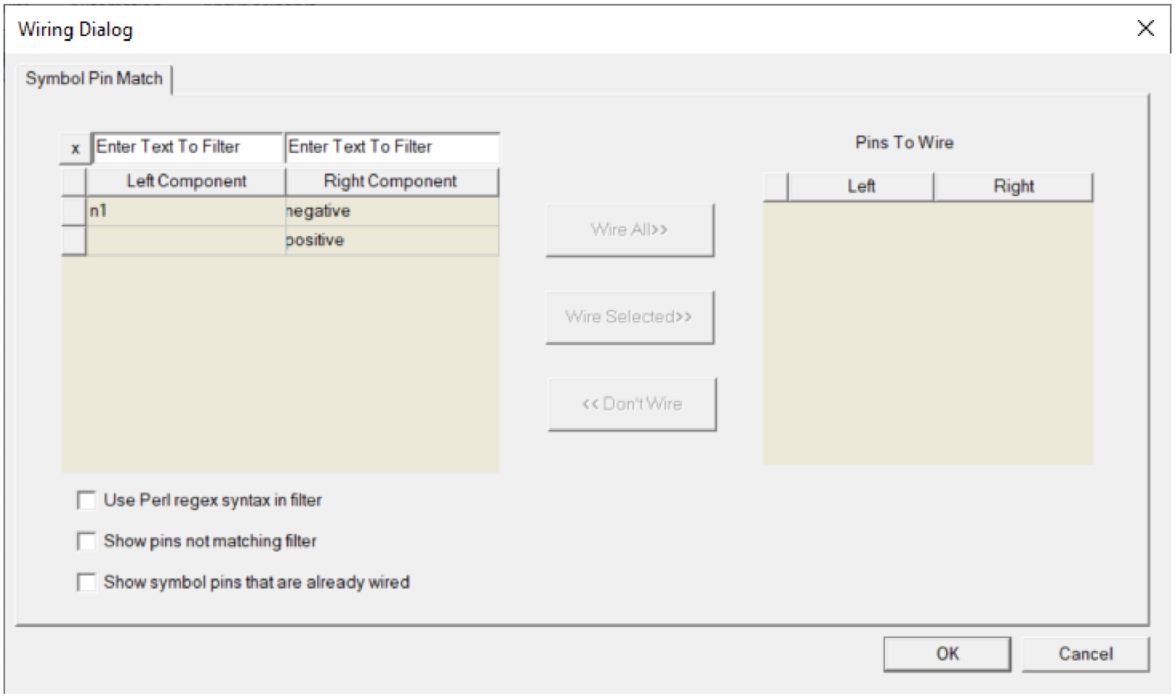


The sets listed are those available in the Project tree. When you complete the solution setup with an Alter set selected, all the ALTER blocks defined for that set are inserted at the end of the netlist that is generated internally for the schematic design with the solution setup. The blocks are inserted in the order they are defined in the set. To view the netlist, right-click the solution setup icon under the Analysis icon in the **Project Manager** window and select **Browse Netlist**.

When you run the analysis, Nexxim executes the statements in the ALTER blocks as described in the topic ALTER Statements. After simulation, create reports based on the settings in the ALTER statements.

Wiring window

In the **Schematic Editor**, hold Ctrl and select two component instances, then right-click and select **Wire**. This opens the **Wiring Dialog** window.



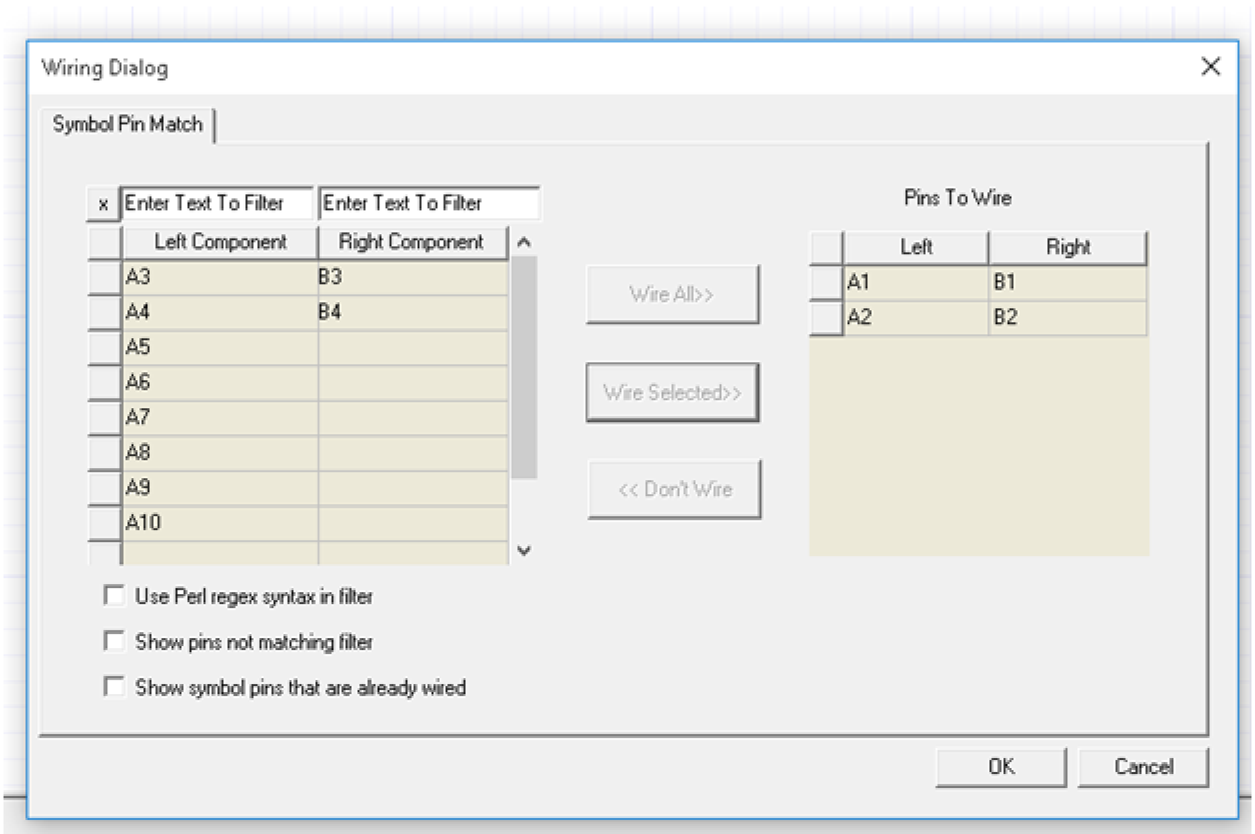
The pins of the selected component on the left are shown in the Left Component column. The pins of the selected component on the right are shown in the Right Component column. Pins are listed in alphanumeric order.

- As the user types into the text-box Filter for a column ("Enter Text To Filter"), the values in that column are filtered accordingly. A wild card ("*") matches zero or more characters.
- You can also filter the results by entering one of the following into the text-box Filter: <Odd>, <Even>, <Top Half>, <Bottom Half>.
- Click to select pins in both columns. Selected pins can be discontinuous. Multiple pins can be selected by holding down **Shift** or **Ctrl**.
- **Show symbol pins that are already wired** shows all ports on both components. But choosing to wire ports that are already wired may have a detrimental effect on the previous wiring.
- **Show pins not matching filter** shows all ports that do not match the filter entered.
- **Use Perl regex syntax in filter** shows all ports that match the entered Perl regular expression.
- **Wire Selected>>** puts the selected pairs of a pin in the **Pins To Wire** grid. If the number of pins in the left and right component are the same, click **Wire All>>** to move all pairs of pins to the **Pins To Wire** grid.
- You can select any pair of pins in the **Pins To Wire** grid. But if you decide not to wire a selected pair, click **Don’t Wire** to return the pins to their component lists.

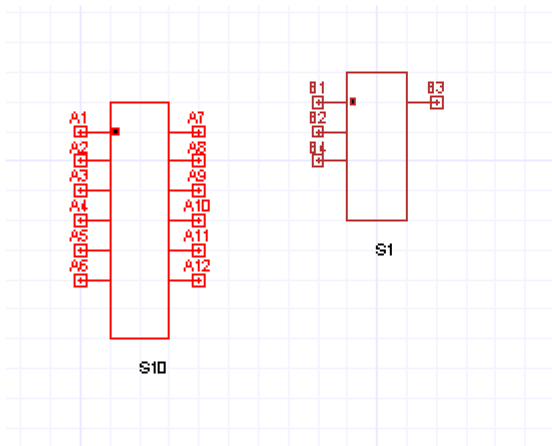
Click **OK** to confirm your selections or click **Edit>Undo Wiring** to undo the wiring operation and restore both symbols to their original states.

When you click **OK**, the two symbols is moved together and are positioned two squares apart (as long as one of them is not pre-wired to other components). For each pair selected, the pin on the left component is moved to the right side of the symbol, if it was not there before. Likewise, the pin on the right component is moved to the left side of the symbol, if necessary. Two pins is aligned horizontally and a wire connection is created between them. See the following example:

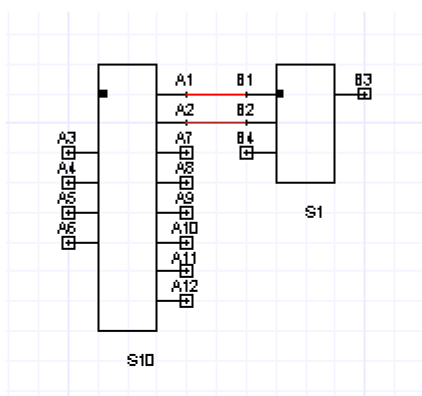
Wiring selection:



Before:

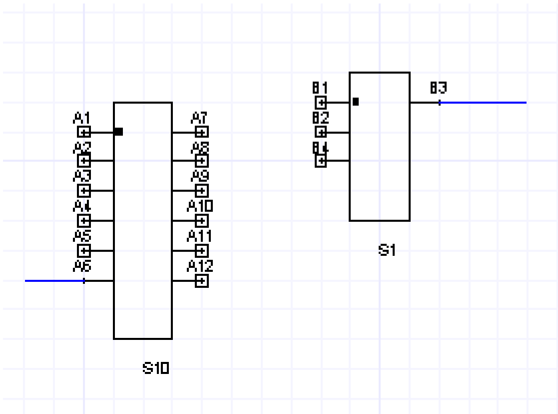


After:

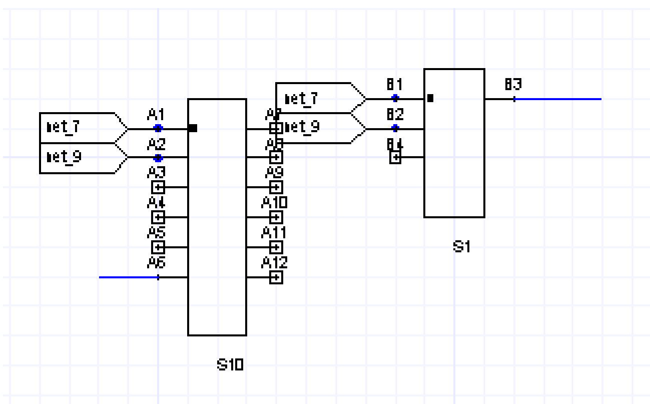


In the case where both component instances are pre-wired, one page-port is created for each pair of pins, each pin is connected to the page port, and the symbols and pins does not be moved. See the following example:

Before:



After:



Plot I-V Curves of Active Components

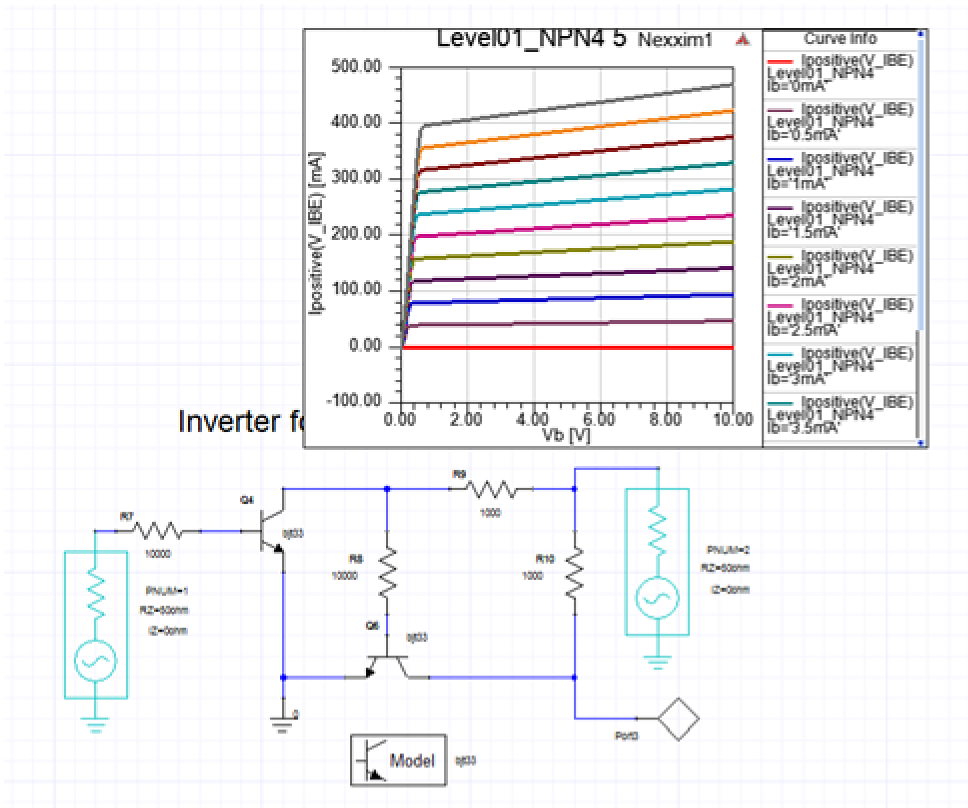
A current-voltage characteristic, or I-V curve, is the relationship between an electric current and its corresponding voltage. The electric current can be a circuit, device, or material. An I-V curve is typically represented as a chart or graph , then a plot of the current-voltage relationship can be traced. This is referred to as the I-V characteristic of the element. When the I-V characteristic of the element is in the load subcircuit, it is plotted via standard labeling. A resistor satisfies Ohm's law, so the I-V characteristic of the element passes through the origin and possesses slope.

DC-IV plots are supported for the following devices:

```
1.  BJT NPN
2.  BJT PNP
3.  BJT NPN 4 Pin with substrate
4.  BJT PNP  4 Pin with substrate
5.  BJT NPN 4 Pin with heat
6.  BJT PNP 4 Pin with heat
7.  BJT NPN 5 Pin
8.  BJT PNP 5 Pin
9.  Diode
10. MOSFET NMOS
11. MOSFET PMOS
12. MOSFET NMOS 4 Pin
13. MOSFET PMOS 4 Pin
14. MOSFET NMOS 5 Pin
15. MOSFET PMOS 5 Pin
16. MESFET NJF
17. MESFET PJF
18. MESFET NJF 4 Pin
19. MESFET PJF 4 Pin
20. JFET NJF
21. JFET PJF
22. GaAsFET NJF
23. GaAsFET PJF
```

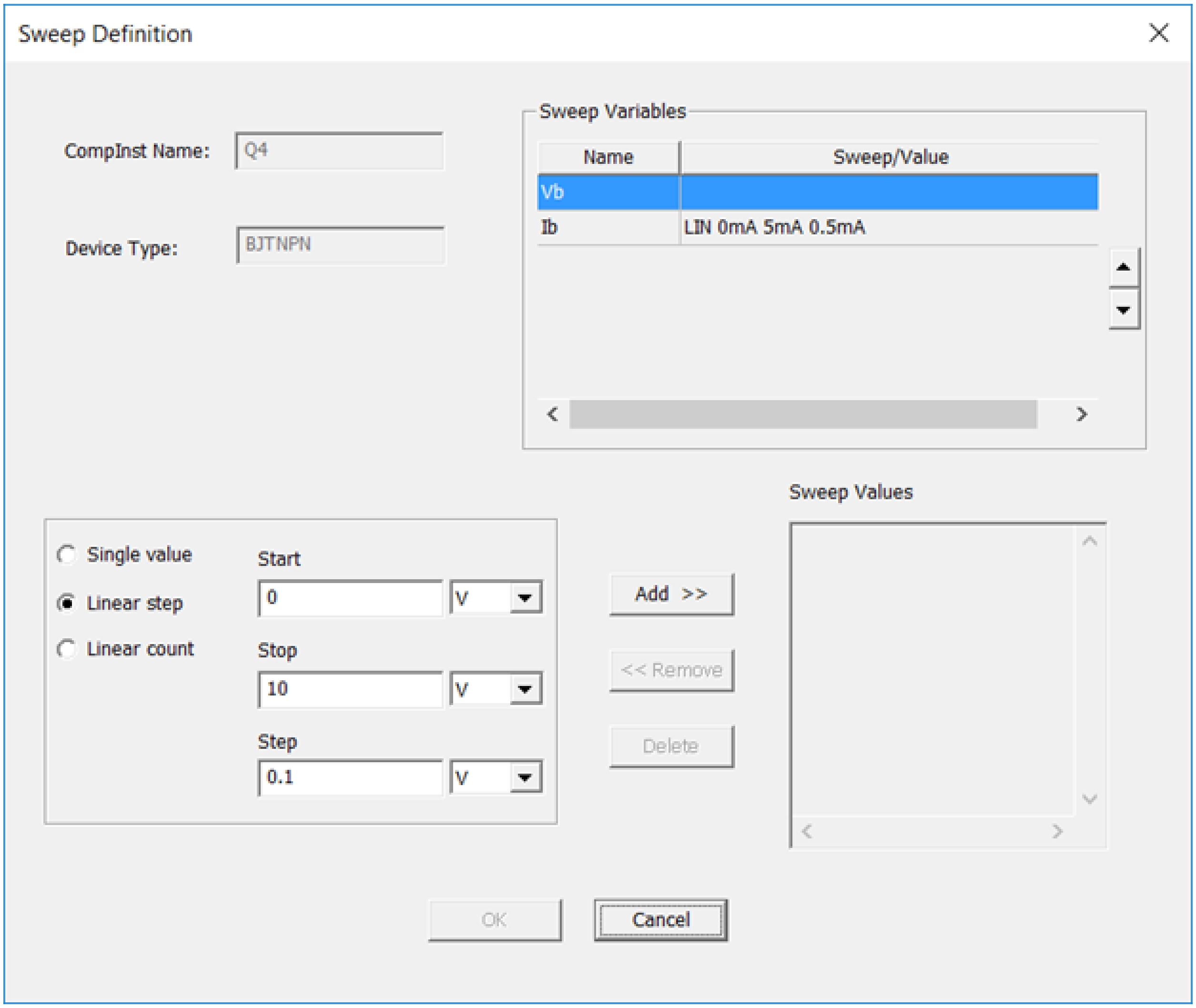
To generate a plot that displays I-V curves:

1. Open a project that contains one of the devices listed above; for example, BJT_inverter.aedt.
2. In the **Schematic Editor**, right-click the BJT device and select **Display IV Curves**. A schematic plot that shows I-V curves is generated and displayed.



A corresponding I-V solution for the device is created beneath **Analysis** in the **Project Manager** window.

3. To modify curves, select the I-V solution in the **Project Manager** window to open the **Sweep Definition** window.



The **Sweep Values** pane displays the values of the variable selected in the **Sweep Variables** pane.

— **To add a sweep value**: Select Start, Stop, and Step values at lower-left, then click **Add >>**.
— **To edit a sweep value**: Select a sweep on the Sweep Values list, then click **<< Remove**.
Then modify the Start, Stop, and Step values, then click **Add**.
— **To delete a sweep value**: Select a sweep on the Sweep Values list, then click **Delete**.
This deletes the I-V solution and the corresponding plot.

After making modifications, when you click **OK** the I-V curves are automatically updated.

Transmission Line Designer

The Transmission Line Designer (TRL) is an analysis/synthesis tool for single and coupled transmission lines. The TRL provides:

- window-based data entry
- Analysis results
- Log output for sweep data

The analysis algorithms are shared by the circuit simulator, and synthesis algorithms make use of closed-form equations whenever possible for computation efficiency.

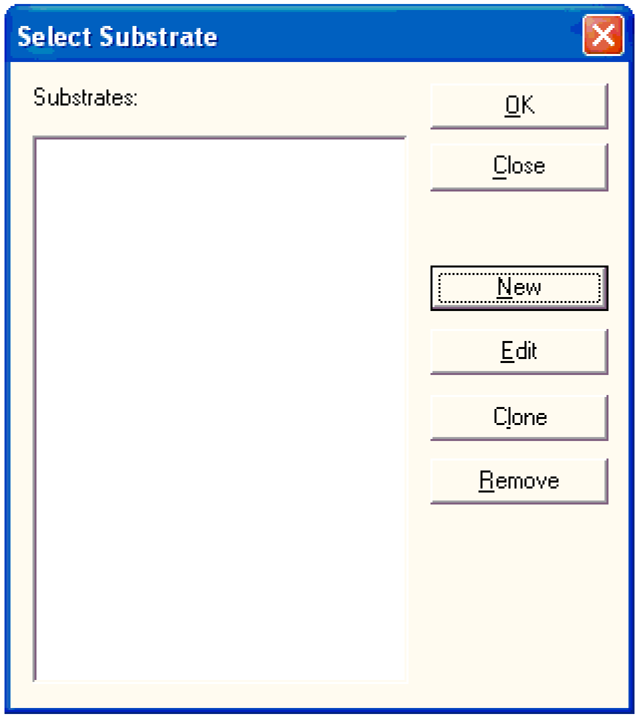
TRL Overview

The **TRL** is used to analyze and synthesize single and coupled transmission lines realized in various layered media. In addition, it allows the results for many of these transmission lines to be exported directly to your schematic.

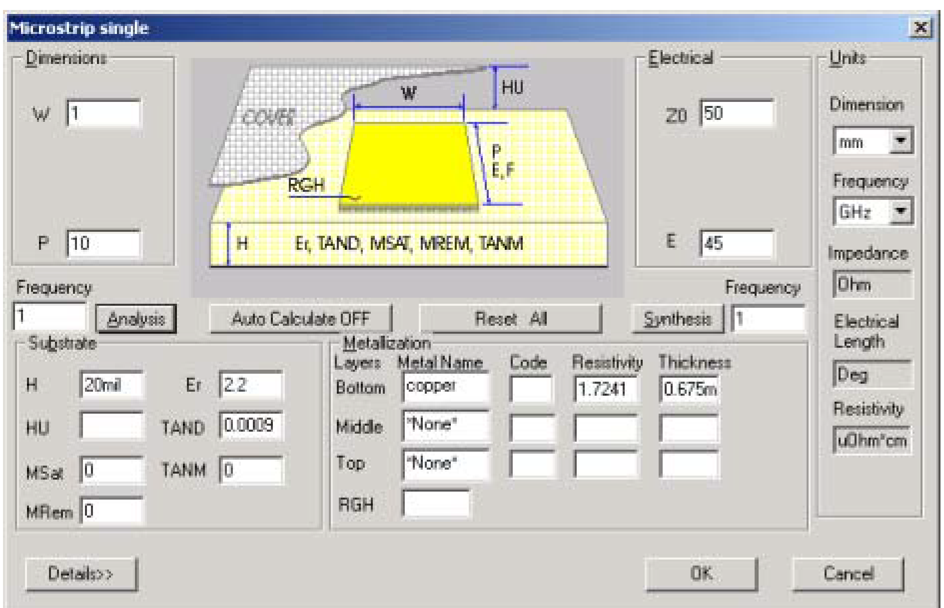
An **Auto Calculate** function allows you to specify whether calculations should be rerun automatically whenever any of the medium values are changed, or only run when you specifically request that synthesis or analysis be run. This feature is implemented for all dimensions and electrical properties of the transmission line, as well as for frequency.

Starting Transmission Line Synthesis

The Transmission Line Designer is run from within a Circuit (Schematic) design, so you must first insert a design before starting the **TRL**. You can insert a design by clicking **Insert Circuit Design** on the **Project** menu. The **Transmission Line Designer** can be started by choosing the appropriate transmission line type under the cascading **TRL** item on the **Circuit** menu. A **Select Substrate** window first appears where choose an existing substrate definition or create a new one.

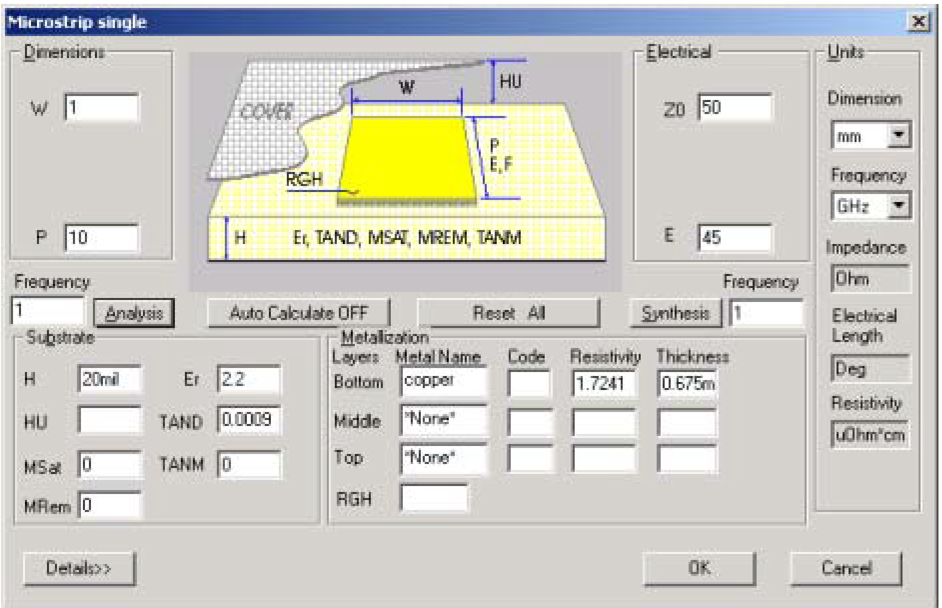


Once a substrate definition is chosen a Transmission Line window similar to the following appears to prompt you through a Transmission Line synthesis session.



Typical TRL window

Choose the appropriate transmission line type and select the substrate to open the main **TRL** window.



The **TRL** window features a menu bar which provides control, medium, and configuration selection. A rendition of the selected transmission line configuration and relevant dimensions is displayed at the center of the window. Dimensions of the line are arranged on the left, electrical parameters on the right. Two identical input boxes are provided for frequency selection; a number entered in one input box instantly appears in the second. Depending upon which box you change and the status of the **Auto Calculate** button, synthesis /analysis may run automatically.

The following window features are available:

- Click **Reset** underneath the picture to reset all data.
- Click **Auto Calculate** underneath the picture to toggle between ON and OFF, which functions as described:

Auto CalculateON – The program automatically runs the calculations each time you update dimensions, electrical parameters or frequency:

- If you change electrical parameters or the number in the **Frequency** group box located next to **Synthesis** , the program performs synthesis of the transmission line.
- If you change dimensions or the number in the **Frequency** group box located next to **Analysis**, the program performs analysis of the transmission line.

Auto CalculateOFF – You must click **Synthesis** or **Analysis** s to run the appropriate calculations after changing values.

- Common data and selections are in the lower right side of the window. The substrate group box has two sub sections for required and optional data.
- The **Units** group box has drop down window boxes for selection of the units for dimensions and frequency and displays the units for Impedance, Electrical length and Resistivity.
- An area is provided for defining metalization parameters. See Conductor Composition for details.
- Sweep results are shown in an output window at bottom, which displays messages on the simulation session. Data that appears in the output window is also stored into the session log file, the name of which is indicated at the beginning of the session.

Exporting Transmission Lines

Once you have designed a transmission line you may click **OK** to export the transmission line to the Ansys HFSS schematic and layout editors, or click **Cancel** to return to HFSS without exporting the transmission line.

Sweep Entries

Line width(s), gap spacing, and frequency may be analyzed over a list of values. This is accomplished by entering three values for the respective parameter: a start value, a stop value, and a step value. These parameters are swept during analysis on the start value to the stop value using the step increments. The following entry example defines a frequency sweep:

1GHZ, 5GHZ, 500MHZ

Or, with GHz selected as a global frequency unit:

1, 5, .5

The results of a sweep analysis are displayed in the output window and are saved in a log file. Physical to electrical length transformation is unavailable during sweep analysis.

Mandatory Entries

Certain data is required for **TRL** simulation. window fields are highlighted if required entries are omitted. The tables given for each medium type indicates which parameters are required.

Some physical dimensions have default values, and these values are described in the Designing Transmission Lines section and are displayed in the **TRL** window output window during analysis.

TRL Capabilities

The complete analysis and synthesis capabilities of **TRL** are summarized in the following topics.

TRL Analysis and Synthesis

The analysis and synthesis capabilities of the **TRL** are summarized in the following tables. The following keywords are used in the various drop-down menus of the **TRL** window to represent the function and medium types that are available.

ANA	analysis
SYN	synthesis
BCL	broadside-coupled lines
OCL	offset-coupled lines
CPL	edge-coupled lines
TRL	single transmission line
ACPL	asymmetric coupled lines

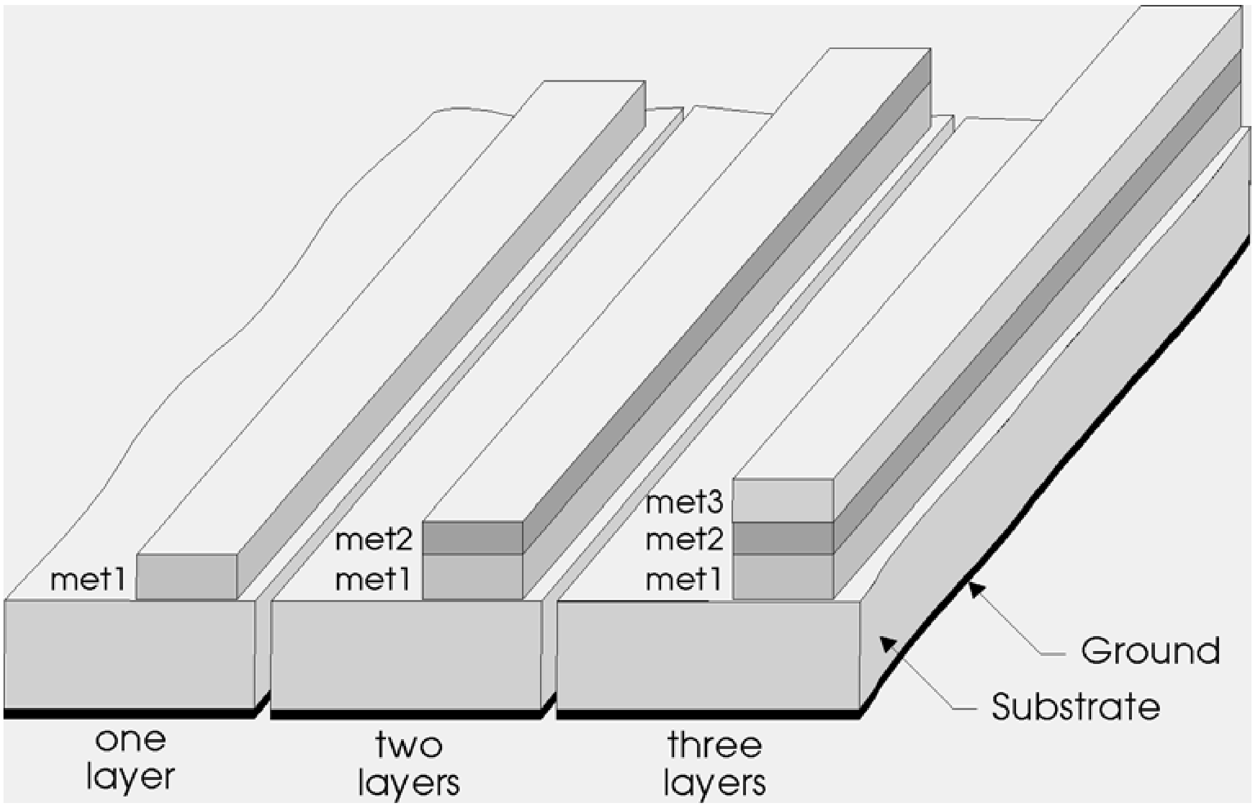
	Single Line	**Coupled Lines**	
Medium	**Capability**	**Keyword**	**Capability**
Microstrip	ANA/SYN	CPL ACPL Lange Coupler Design	ANA/SYN ANA ANA/SYN
Stripline	ANA/SYN	CPL BCL OCL	ANA/SYN ANA/SYN ANA/SYN
Offset Stripline	ANA/SYN		
Suspended substrate	ANA/SYN	CPL	ANA/SYN
Coplanar waveguide	ANA/SYN		
Grounded coplanar waveguide	ANA/SYN		
Coaxial	ANA/SYN		

TRL Data log

Both the output window and the log file contain all the relevant data that was used to select and dimension the structure as well as the associated results. The log file is for one session and captures the entries and the result of the calculations for all the opened documents. The log file contains sweep results that are produced when one or more parameters are swept during analysis.

TRL Conductor Composition

Conductor metallizations are selected on the Metalization group in the Main **TRL** window. A blank or selection of **none** corresponds to an ideal conductor with zero thickness. One to three metalization overlay layers can be specified by selecting the metal types and thickness.



METAL CODES AND RESISTIVITIES TABLE

Material	Metal Code	Resistivity ($\mu\Omega$ * cm)	Material	Metal Code	Resistivity ($\mu\Omega$ * cm)
aluminum	AL	2.65	rhodium	RH	4.51
chromium	CR	18 **	rolled copper	RC	1.673 ***
copper	CU	1.673	silver	AG	1.59
gold	AU	2.44 *	superconductor	SC	0
indium	IN	15.52	tantalum	TA	15.52
iridium	IR	5.30	tantalum nitride	TN	2.50
iron	FE	9.66	tin	SN	11.55
magnesium	MG	4.45	titanium	TI	55.0
molybdenum	MO	5.69	tungsten	W	5.6

nickel	NI	8.707	zinc	ZN	5.68
palladium	PD	10.69	zirconium	ZR	4.10
platinum	PT	10.62			

* Resistivity correction is applied when the metalization thickness is less than 0.03 μm.

** Resistivity correction is applied when the thickness is less than 0.1 μm.

*** Thickness of rolled copper is given in ounces of copper (1 oz = 0.00135 in).

> **Note:**
>
> If metalization material is specified but the conductor thickness is not, the conductor loss is zero.

Metalization surface roughness, **RGH**, can be specified for additional conductor loss calculations due to a nonideal metal surface. **RGH** is specified as the **RMS** variation from an ideal flat surface (in units of dimensions). **RGH** appears in each medium's **Metalization** control group as appropriate.

TRL Calculation Between Physical Length and Electrical Length

For each medium, TRL can compute physical length (**P)** from electrical length (**E)** for the transmission line if the **Frequency** is given. The reverse calculation is also possible. To compute **P**, enter the value for **E** and **Frequency** and click **Synthesize**. The **TRL** uses the effective dielectric constant to compute the transmission line length from **E** at **Frequency**. With a coupled line, the average of the effective dielectric constants for the even and odd mode is used. Similarly, to compute **E**, enter values for **P** and **Frequency** and click **Analysis**.

Designing Transmission Lines

The following transmission lines are available.

TRL Microstrip Transmission Line

TRL Edge-Coupled Symmetric Microstrip Transmission Lines

TRL Edge-Coupled Asymmetric Microstrip Transmission Lines

TRL Lange Coupler

TRL Stripline Transmission Line

TRL Edge-Coupled Stripline Transmission Lines

TRL Broadside-Coupled Stripline Transmission Lines

TRL Offset-Coupled Stripline Transmission Lines

TRL Offset Stripline Transmission Line

TRL Suspended Stripline Transmission Line

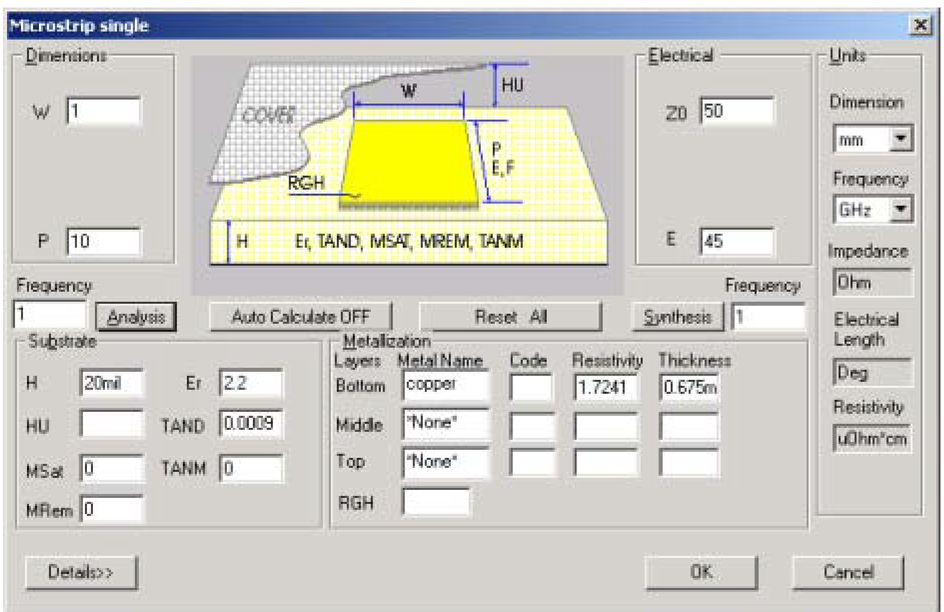
TRL Edge-Coupled Suspended Stripline Transmission Lines

TRL Coplanar Waveguide

TRL Grounded Coplanar Waveguide

TRL Coaxial Cable

TRL Microstrip Transmission Line



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
P	0	m	Physical length of transmission line
Z0	Req (Elec)	ohm	Transmission line impedance
E	0	deg	Electrical length of transmission line
Frequency	0	Hz	Analysis frequency
ER	Req		Relative dielectric constant
H	Req	m	Substrate thickness
HU	40*H	m	Cover height above substrate (default value has negligible effect)

TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness
MSAT	0	Gauss	Saturation magnetization (4πMs)
MREM	0	Gauss	Remnant magnetization (4πMr)
TANM	0		Magnetic loss tangent

Synthesis and Analysis

- For synthesis of microstrip transmission lines, the parameters **Z0**, **H**, and **ER** must be entered prior to clicking **Synthesis**. The width, **W**, is computed.
- For analysis, the parameters **W**, **H**, and **ER** must be entered prior to clicking **Analysis**. The impedance, **Z0**, is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed lossless unless **TAND** is specified and greater than zero.

Magnetic Substrates

- A magnetic substrate is defined when the **MSAT** and **MREM** parameters are given. The demagnetized substrate case occurs when **MSAT** > 0 and **MREM** = 0 (default). The partially magnetized substrate (including the fully magnetized, saturated, or latched substrate case) occurs when **MSAT** and **MREM** > 0, but **MREM** ≤**MSAT**.
- The direction of magnetic bias is assumed to be in the direction of wave propagation; that is, perpendicular to the transverse field.
- The magnetic loss tangent, **TANM**, can be specified to account for losses in the magnetic substrate. It is similar to the dielectric loss tangent, **TAND**, and its loss is added to find the total loss of the substrate.
- The minimum frequency of analysis must be greater than the gyromagnetic frequency:
- $f > 2.8$ (MHz/Gauss) $\times$ MSAT (Gauss)
- where the constant of 2.8 MHz/Gauss comes from the gyromagnetic ratio ($\gamma/2\pi$).
- References 23-25 document propagation on magnetic substrates. In particular, the formulations by Pucel and Massé have been utilized in the program.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the line's propagation characteristics. Up to three conductors of different metal and different thickness can be specified.

- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the microstrip transmission line are:

Frequency
Width, **W**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

To maintain accuracy, the following limitations should be followed:

$0.01 \leq W/H \leq 100$

$1 \leq \varepsilon r \leq 128$

$$F(GHz) \cdot H(mm) \leq 39$$

Example

To select the microstrip transmission line medium, select **TRL** on the **Product** menu, select **Microstrip**, and click **Single**. Select the units mm and GHz. Enter the following parameters for synthesis and analysis of the transmission line:

Line impedance, **Z0**: 50 ohms

Substrate thickness, **H**: 0.635 mm

Dielectric constant, **ER**: 9.8

Metal: CU 0.01 mm

Loss tangent, **TAND**: 0.0001

Click **Synthesis** to determine the width of the line:

Width: 0.605 mm

The output that scrolls in the lower window includes the effective dielectric constant, **Keff**, which is this case is computed as 6.4762.

Next, click **Analysis** to sweep the frequency from 0 to 30 GHz in steps of 2 GHz. Enter the sweep by typing **0,30,2** in the **Frequency** group box. The results of the swept-frequency analysis are as follows:

Single Line in Microstrip

Metals: 1.67 0.0100mm

H = 0.635mm ER = 9.80 TAND = 0.00010 T/H = 0.0157

Freq ghz	Width mm	W/H	Z0 Ohms	Keff	D Loss dB/mm	C Loss dB/mm	T LOSS dB/mm
0.0	0.605	0.953	50.00	6.476	0.0000	0.0003	0.0003
2.0	0.605	0.953	49.97	6.516	0.0000	0.0018	0.0018
4.0	0.605	0.953	49.97	6.581	0.0001	0.0025	0.0026
6.0	0.605	0.953	50.04	6.658	0.0001	0.0031	0.0032
8.0	0.605	0.953	50.21	6.741	0.0002	0.0036	0.0038
10.0	0.605	0.953	50.46	6.830	0.0002	0.0040	0.0042
12.0	0.605	0.953	50.81	6.922	0.0003	0.0043	0.0046
14.0	0.605	0.953	51.25	7.017	0.0003	0.0046	0.0050
16.0	0.605	0.953	51.77	7.113	0.0004	0.0049	0.0053
18.0	0.605	0.953	52.38	7.209	0.0004	0.0051	0.0056
20.0	0.605	0.953	53.06	7.304	0.0005	0.0054	0.0058
22.0	0.605	0.953	53.81	7.399	0.0005	0.0055	0.0061
24.0	0.605	0.953	54.62	7.491	0.0006	0.0057	0.0063
26.0	0.605	0.953	55.48	7.580	0.0006	0.0058	0.0065
28.0	0.605	0.953	56.39	7.667	0.0007	0.0060	0.0066
30.0	0.605	0.953	57.34	7.750	0.0007	0.0061	0.0068

Using this sweep analysis, determine how the impedance and **Keff** change with frequency, and the change in dielectric loss (**D Loss**), conductor loss (**C Loss**) and the total loss (**T Loss**) vary with frequency. At 0 GHz, the results reproduce the synthesized impedance of 50 ohms, and this impedance then rises at higher frequencies.

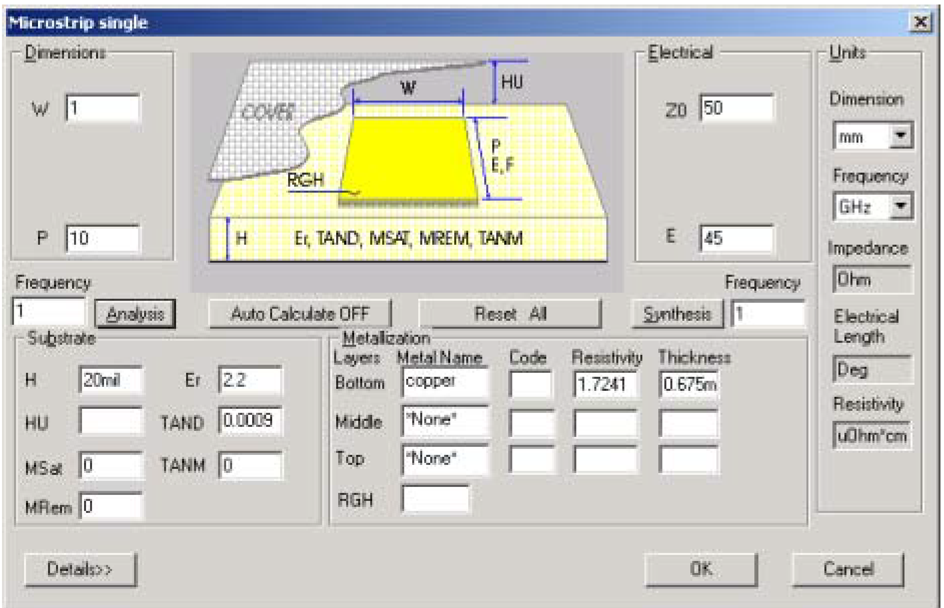
To convert from electrical length to physical length, enter the electrical length and the frequency:

E: 45 degrees
Frequency: 10 GHz

Click **Synthesis**. Since the frequency is now set to 10 GHz, a new width is calculated to maintain an impedance of 50 ohms. The equivalent physical length is also computed:

W: 0.617 mm
P: 1.432 mm

Edge-Coupled Symmetric Microstrip Transmission Lines



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
S	Req (Phys)	m	Spacing between conductors
P	0	m	Physical length of transmission line
Z0	Req (Elec Opt 1)	ohm	Impedance of the coupled lines
K	Req (Elec Opt 1)	dB	Coupling coefficient (positive)
Zo	Req (Elec Opt 2)	ohm	Odd-mode impedance of the coupled lines
Ze	Req (Elec Opt 2)	ohm	Even-mode impedance of the coupled lines
E	0	deg	Electrical length of the coupled lines
Frequency	0	Hz	Analysis frequency
ER	Required		Relative dielectric constant

H	Required	m	Substrate thickness
HU	40*H	m	Cover height above substrate (default value has negligible effect)
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

Two options exist for the synthesis of coupled microstrip transmission lines:

— Specify impedance, **Z0**, and coupling coefficient, **K**.

— Specify even-mode impedance, **Ze,** and odd-mode impedance, **Zo**.

- The substrate parameters **H**, and **ER** must be entered prior to clicking **Synthesis**. The width of the microstrip lines, **W**, and the spacing between them, **S**, is computed. Also, the alternate electrical option set is computed. For example, entering **Z0** and **K** computes **Ze** and **Zo**.
- For analysis, the parameters **W**, **S**, **H**, and **ER** must be entered prior to clicking **Analysis**. The electrical properties, **Z0**, **K**, **Zo**, and **Ze** is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and the frequency, **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the propagation characteristics. Up to three conductors of different metals and thicknesses can be specified.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the coupled microstrip transmission lines are:

Frequency
Width, **W**
Spacing, **S**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the **Sweep Entries** topic for further information.

Limitations

To maintain accuracy, the following limitations should be followed:

$0.01 \leq W/H \leq 100$

$1 \leq \varepsilon r \leq 128$

$$F(GHz) \cdot H(mm) \leq 39$$

Example

To select the coupled microstrip medium, select **TRL** on the **Product** menu, select **Microstrip**, and click **CPL**. Set the units to **mm** and **GHz**. Enter the following parameters for synthesizing a 6-dB coupler at 10 GHz:

Impedance, **Z0**: 50 ohms

Coupling, **K**: 6 dB

Frequency: 10 GHz

Substrate thickness, **H**: 0.635 mm

Dielectric constant, **ER**: 9.8

Metal: Cu 0.01 mm

Click Synthesis to determine the coupler parameters:

Width, **W**: 0.369 mm

Spacing, **S**: 0.055 mm

Odd-mode impedance, **Zo**: 28.82 ohms

Odd-mode **Keff**: 5.46

Even-mode impedance, **Ze**: 86.74 ohms

Even-mode **Keff**: 7.08

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the physical length:

Physical length, **P**: 2.99 mm
Notice that the average **Keff** for the even and odd modes is used in the calculation.

To perform a sweep analysis of the conductor spacing, type in:

Spacing, **S**: 0.03, 0.06, 0.01 mm

by typing **(0.03 0.06 0.01**); the space bar inserts a comma and indicate that a sweep is appropriate. Click **Analysis** to get the following results:

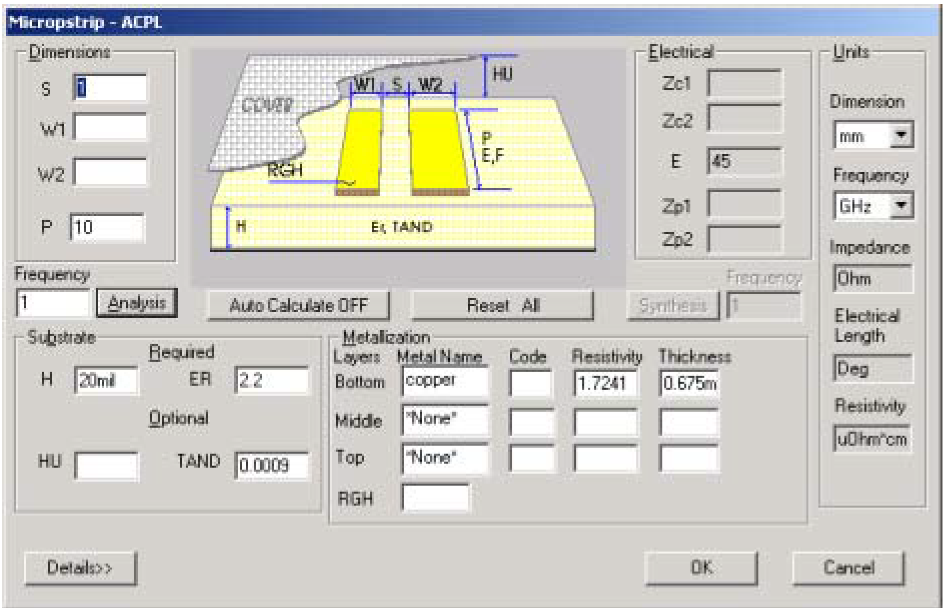
Coupled Lines in Microstrip

H = 0.635mm ER = 9.80 TAND = 0.00000 T/H = 0.0157

							Loss (dB/mm)			
Freq	S/H	W/H	ZE	ZO	KE	KO	Even Mode		Odd Mode	
ghz			Ohms	Ohms			Cond	Diel	Cond	Diel
10.0	0.047	0.581	88.0	24.8	7.020	5.441	0.0046	0.0000	0.0165	0.0000
10.0	0.063	0.581	87.5	26.6	7.040	5.446	0.0046	0.0000	0.0154	0.0000
10.0	0.079	0.581	87.0	28.1	7.056	5.450	0.0047	0.0000	0.0145	0.0000
10.0	0.094	0.581	86.5	29.4	7.070	5.456	0.0047	0.0000	0.0139	0.0000

You can also sweep frequency and width, **W**, to get analysis results as a function of these parameters.

TRL Edge-Coupled Asymmetric Microstrip Transmission Lines



Keyword	**Default**	Unit	**Description**
W1	Req (Phys)	m	Width of conductor 1
W2	Req (Phys)	m	Width of conductor 2
S	Req (Phys)	m	Spacing between conductors
P	0	m	Physical length of transmission line
Zc1	Req (Elec)	ohm	c-mode impedance of line 1 for even-like modes
Zc2	Req (Elec)	ohm	c-mode impedance of line 2 for even-like modes
Zp1	Req (Elec)	ohm	π-mode impedance of line 1 for odd-like modes
Zp2	Req (Elec)	ohm	π -mode impedance of line 2 for odd-like modes
E	0	deg	Electrical length of the coupled lines
Frequency	0	Hz	Analysis frequency
ER	Required		Relative dielectric constant
H	Required	m	Substrate thickness
HU	40***H**	m	Cover height above substrate (default value has negligible effect)
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Analysis

- For analysis, the parameters **W1**, **W2**, **S**, **H**, and **ER** must be entered prior to clicking **Analysis**. The electrical properties, **Zc1**, **Zc2**, **Zp1**, **Zp2** is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**. The average effective dielectric constant computed at **Frequency** is used for the calculation.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the propagation characteristics. Up to three conductors of different metal and different thickness can be selected.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the asymmetric coupled microstrip transmission lines are:

Frequency
Spacing, **S**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

To maintain accuracy, the following limitations should be followed:

$0.01 \leq W/H \leq 100$

$1 \leq \varepsilon r \leq 128$

$0.1 < S/H < 10.0$

Example

To select the asymmetric coupled microstrip medium, select **TRL** on the **Product** menu, select **Microstrip**, and click **ACPL**. Set the units to **mm** and **GHz**. Enter the following parameters to analyze an ACPL (synthesis is not implemented for ACPL):

Spacing, **S**: 0.1 mm

Width of line 1, **W1**: 0.2 mm

Width of line 2, **W2**: 0.3 mm

Substrate height, **H**: 0.635 mm

Dielectric constant, **ER**: 9.8

Frequency: 10 GHz

Click **Analysis** to determine the coupled line parameters:

c-mode impedance of line 1, Zc1: 115.97 ohms

c-mode impedance of line 2, Zc2: 93.12 ohms

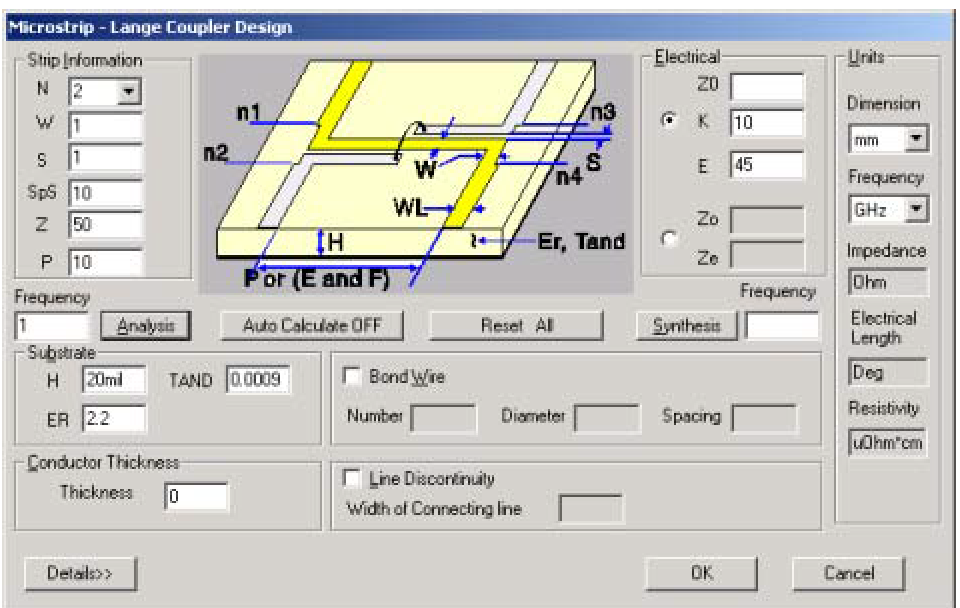
π-mode impedance of line 1, Zp1: 44.17 ohms

π-mode impedance of line 2, Zp2: 35.47 ohms

c-mode effective ER, Effc: 6.768

π-mode effective ER, Effp: 5.531

TRL Lange Coupler



Keyword	Default	Unit	Description
N	Req (Phys)	m	Number of strips (2, 4, or 6)
W	Req (Phys)	m	Spacing between conductors

S	Req (Phys)	m	Spacing between strips
Sps	10		Number of substrips used in the analysis. Substrips/strip × number of strips ≤ 60
Z	50	ohm	Connecting line impedance to account for impedance change from feed line to coupler
P	Req (Elec Opt 2)	m	Physical length of the coupler
K	Req (Elec Opt 1)	dB	Coupling coefficient (positive)
Z0	Req (Elec Opt 1)	ohm	Impedance of the coupled lines
Ze	Req (Elec Opt 2)	ohm	Even-mode impedance of the coupled lines
Zo	Req (Elec Opt 2)	ohm	Odd-mode impedance of the coupled lines
E	0	deg	Electrical length of the coupler
Frequency	0	Hz	The frequency at which the quasistatic calculations are performed and reused at other analysis frequencies
H	Required	m	Substrate thickness
Er	Required		Relative dielectric constant
Thickness	0	m	Thickness of metalization
TAND	0		Dielectric loss tangent
Bond Wire Number	Required when activated		Number of bond wires used to connect the interdigital strips
Keyword	**Default**	Unit	**Description**
Bond Wire Diameter	Required when activated	m	Diameter of bond wires
Bond Wire Spacing	Required when activated	m	Spacing between bond wire centers
Line Disc. Width	Required when activated	m	Width of the connecting line to account for the impedance discontinuity in the calculations

Synthesis and Analysis

Synthesis of a Lange coupler proceeds by first selecting the number of strips for the appropriate structure. Two options exist for synthesis:

— Specify impedance, **Z0**, and the coupling coefficient, **K**.

— Specify the even-mode impedance, **Ze**, and the odd-mode impedance, **Zo**.

- The substrate parameters **H**, and **ER** must be entered prior to clicking **Synthesis**. The width of the strips, **Width**, and the spacing between them, **Spacing**, is computed. Also, the alternate electrical option set is computed. For example, entering **Z0** and **K** computes **Ze** and **Zo**.
- For analysis, the parameters **Width**, **Spacing**, **H**, **ER** and either **PhysicalLength** or **Frequency** must be entered prior to clicking **Analysis**. The electrical properties, **Z0**, **K**, **Zo**, and **Ze** is computed.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering a value for **E** and the frequency, **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency** and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

The default conductor used for analysis is gold whose resistivity is 2.44 mohm-cm. The thickness of the metal can be specified using the **Thick** parameter in the substrate group. The calculation uses the thickness for loss and impedance calculations.

Frequency Sweep Options

A frequency sweep can be performed to determine the response of the coupler over frequency. Type *Start Stop Step* (e.g., **4 6 5**) and click **Analysis**.

Example

To select the Lange coupler, select **TRL** on the **Product** menu, select **Microstrip**, and click **Lange Coupler**. Set the units to **mil** and **GHz**. Enter the following parameters for synthesizing a 4 strip, 3-dB coupler at 10 GHz:

Number of Strips, **N**:	4
Impedance, Z0:	50 ohms
Coupling, K:	3 dB
Center Frequency:	10 GHz
Substrate thickness, H:	25 mil
Dielectric constant, ER:	10

Metal Thickness, Thick:	0.2 mil

Click **Synthesis** to determine the coupled line parameters:

Width, **W**	1.6545 mil
Spacing, S	2.0000 mil
Odd-mode impedance, Zo:	20.68 ohms
Even-mode impedance, Ze:	120.91 ohms

***** LANGE SYNTHESIS *****

COUPLING IMPEDANCES (OHMS) SUBSTRATE FULL STRIPS

DB EVEN ODD COUPLER ER H(mils) NO. T(mils) W(mils) S(mils)

3.00 120.91 20.68 50.00 10.00 25.0000 4 0.20000 1.6545 2.0000

To include the effects of a bond wire and connecting line parasitics, enter the following:

Bond Wire	Yes
Number	3
Diameter	.8 mil
Spacing	2 mil
Line Discontinuity	Yes
Width	24.5 mil

Clicking Analysis yields the following:

***** LANGE ANALYSIS *****

SUBSTRATE:

THICKNESS = 25.0000 mils

DIELECTRIC CONSTANT = 10.0000

LOSS TANGENT = 0.00000

STRIPS:

NUMBER = 4

WIDTH = 1.6545 mils

SPACING = 2.0000 mils

THICKNESS = 0.2000 mils

SUBSTRIPS = 10

PARASITICS:

NO. PARALLEL WIRES = 3

BOND WIRE DIAMETER = 0.8000 mils

WIRE SEPARATION = 2.0000 mils

WIDTH OF 50 OHM LINE = 24.5000 mils

W/H =0.066181 S/H =0.079999

W/H =0.071646 S/H =0.074534 EFFECTIVE VALUES DUE TO FINITE THICKNESS

*** DC CHARACTERISTICS

Z(0) = 50.00 Ohms

ZOE OHMS ZOO OHMS COUP DB Z(0) EFFKE EFFKO VE M/SEC VO M/SEC

131.56 22.81 3.04 54.79 6.378 5.502 1.1870E+8 1.2781E+8

*** CHARACTERISTICS WITH DISPERSION

CENTER FREQUENCY = 10000.0 MHZ

ZOE OHMS ZOO OHMS COUP DB Z(0) EFFKE EFFKO VE M/SEC VO M/SEC

134.08 23.18 3.03 55.75 6.380 5.510 1.1869E+8 1.2771E+8

EVEN MODE ATTENUATION ODD MODE ATTENUATION

db/in db/in

0.6087 2.1903 DIELECTRIC + CONDUCTOR

EFFECTIVE DIELECTRIC CONSTANT OF COUPLER = 5.9371

ESTIMATED COUPLING LENGTH = 121.098 mils

INDUCTANCE PER CROSSOVER LOCATION = 0.12526 nH

RESISTANCE PER CROSSOVER LOCATION = 0.04920 Ohms

*** RESPONSE WITH DISPERSION AND PARASITICS

FREQ. REFL. VSWR RET LOSS COUPL. PHASE THRU LOSS PHASE ISOLAT

(MHz) COEFF (dB) (dB) (Deg) (dB) (Deg) (dB)

10000 0.068 1.15 -23.3 3.23 -0.6 -3.11 -90.2 -24.66

DUE TO PARASITICS, THE CENTER FREQUENCY HAS SHIFTED DOWN FROM 10000 MHZ

To perform a frequency analysis from 4 GHz to 12 GHz in steps of 1 GHz, type **4 12 1** in the **Frequency** group box. Then add coupling length P = 121.098 mil. The last part of the output contains the analysis at each frequency. The 10 GHz center frequency analysis point is used to compute the quasistatic characteristics and reused at each analysis frequency.

FREQUENCY ANALYSIS OF COUPLER

FREQ. REFL. VSWR RET LOSS COUPL. PHASE THRU LOSS PHASE ISOLAT

(MHz) COEFF (dB) (dB) (Deg) (dB) (Deg) (dB)

4000 0.078 1.17 -22.1 5.99 42.2 -1.56 -45.6 -23.92

5000 0.081 1.18 -21.8 4.89 33.7 -2.02 -54.5 -23.42

6000 0.081 1.18 -21.8 4.17 26.0 -2.42 -62.5 -23.30

7000 0.079 1.17 -22.1 3.70 18.9 -2.74 -69.9 -23.41

8000 0.076 1.16 -22.4 3.41 12.2 -2.97 -76.9 -23.69

9000 0.072 1.16 -22.8 3.26 5.7 -3.10 -83.6 -24.11

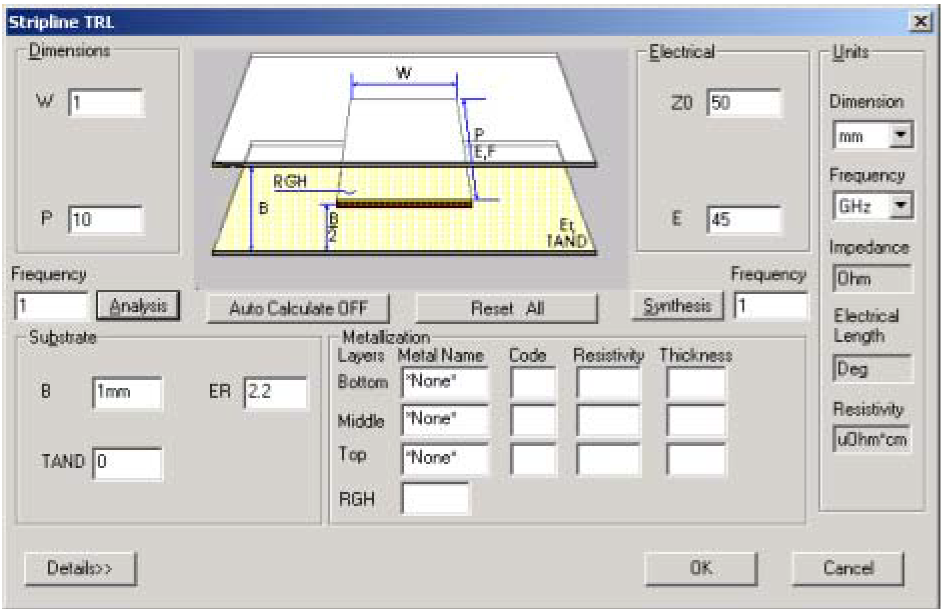
10000 0.068 1.15 -23.3 3.23 -0.6 -3.11 -90.2 -24.66

11000 0.065 1.14 -23.8 3.33 -7.1 -3.02 -96.9 -25.34

12000 0.060 1.13 -24.4 3.56 -13.8 -2.81 -103.8 -26.03

Electrical Length [Eeff=(Effo+Effe)/2] = 35.78 deg

TRL Stripline Transmission Line



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
P	0	m	Physical length of transmission line
Z0	Req (Elec)	ohm	Transmission line impedance
E	0	deg	Electrical length of transmission line
Frequency	0	Hz	Analysis frequency
ER	Req		Relative dielectric constant
B	Req	m	Substrate thickness
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

- For synthesis of stripline transmission lines, the parameters **Z0**, **B**, and **ER** must be entered prior to clicking **Synthesis**. The width, **W**, is computed.
- For analysis, the parameters **W**, **B**, and **ER** must be entered prior to clicking **Analysis**. The impedance, **Z0**, is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from

physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **B**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the line's propagation characteristics. Up to three conductors of different metals and thicknesses can be specified.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the stripline transmission line are:

Frequency
Width, W

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

Error less than 0.5% for **W/B** < 10.

Example

To select the stripline medium, pull down the **Structure** menu, select **Stripline**, and click **Single**. Set the units to **mm** and **GHz**. Enter the following parameters to synthesize a stripline:

Impedance, **Z0**: 50 ohms

Dielectric constant, **ER**: 2.2

Substrate thickness, **B**: 1.27 mm

Click **Synthesis** to determine the width, W, of the transmission line:

Width, **W**: 1.05 mm

Since stripline is a TEM structure and does not vary with frequency, perform a sweep analysis on the width of the strip instead. Type **0.5, 3.0, 0.5** into the **W** group box to sweep the line's width from 0.5 mm to 3.0 mm in steps of 0.5 mm. Be sure to supply information on the line's metalization and losses:

Metal: Cu 0.01 mm

TAND: 0.001

Clicking Analysis yields the following:

Single Stripline

B = 1.270mm ER = 2.20 TAND = 0.00100 T/B = 0.0079

Freq Width W/B Z0 D Loss C Loss T Loss

ghz mm Ohms dB/mm dB/mm dB/mm

0.0 0.500 0.394 74.86 0.0000 0.0002 0.0002

0.0 1.000 0.787 50.69 0.0000 0.0002 0.0002

0.0 1.500 1.181 38.41 0.0000 0.0001 0.0001

0.0 2.000 1.575 30.93 0.0000 0.0001 0.0001

0.0 2.500 1.969 25.90 0.0000 0.0001 0.0001

0.0 3.000 2.362 22.28 0.0000 0.0001 0.0001

Change the frequency to 10 GHz and click Analysis again:

Single Stripline

B = 1.270mm ER = 2.20 TAND = 0.00100 T/B = 0.0079

Freq Width W/B Z0 D Loss C Loss T Loss

ghz mm Ohms dB/mm dB/mm dB/mm

10.0 0.500 0.394 74.86 0.0014 0.0029 0.0043

10.0 1.000 0.787 50.69 0.0014 0.0024 0.0038

10.0 1.500 1.181 38.41 0.0014 0.0022 0.0035

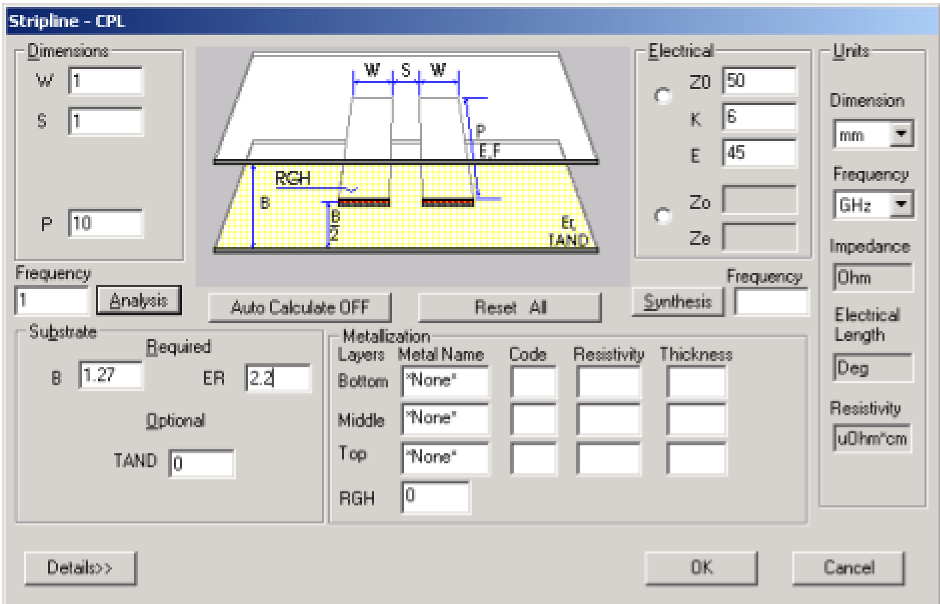
10.0 2.000 1.575 30.93 0.0014 0.0020 0.0034

10.0 2.500 1.969 25.90 0.0014 0.0019 0.0033

10.0 3.000 2.362 22.28 0.0014 0.0019 0.0032

Notice that the impedances are the same for DC or 10 GHz, but the conductor loss calculations differ significantly due to the skin effect.

TRL Edge-Coupled Stripline Transmission Lines



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
S	Req (Phys)	m	Spacing between conductors
P	0	m	Physical length of transmission line
Z0	Req (Elec,Opt 1)	ohm	Impedance of the coupled lines
K	Req (Elec,Opt 1)	dB	Coupling coefficient (positive)
Zo	Req (Elec,Opt 2)	ohm	Odd-mode impedance of the coupled lines
Ze	Req (Elec,Opt 2)	ohm	Even-mode impedance of the coupled lines
E	0	deg	Electrical length of the coupled lines
Frequency	0	Hz	Analysis frequency
ER	Required		Relative dielectric constant
B	Required	m	Substrate thickness
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

Two options exist for synthesis of edge-coupled stripline transmission lines:

— Specify impedance, **Z0**, and the coupling coefficient, **K**.

— Specify the even-mode impedance, **Ze**, and the odd-mode impedance, **Zo**.

- The substrate parameters **B**, and **ER** must be entered prior to clicking **Synthesis**. The width of the striplines, **W**, and the spacing between them, **S**, is computed. Also, the alternate electrical option set is computed. For example, entering **Z0** and **K** computes **Ze** and **Zo**.
- For analysis, the parameters **W**, **S**, **B**, and **ER** must be entered prior to clicking **Analysis**. The electrical properties, **Z0**, **K**, **Zo**, and **Ze** is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency,** and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **B**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the propagation characteristics. Up to three conductors of different metals and thicknesses can be specified.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for edge-coupled stripline are:

Frequency
Width, **W**
Spacing, **S**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See *Sweep Entries* topic for further information.

Limitations

To maintain accuracy, the following limitation should be followed:

T /**B** < 0.1, where **T** is the metal thickness

Example

To select the edge-coupled stripline medium, select **TRL** on the **Product** menu, select **Stripline**, and click **CPL**. Set the units to **mm** and **GHz**. Enter the following parameters to synthesize a 6-dB coupler at 10 GHz:

Impedance, **Z0**: 50 ohms

Coupling, **K**: 6 dB

Frequency: 10 GHz

Substrate thickness, **B**: 1.27 mm

Dielectric constant, **ER**: 2.2

Metal: Cu 0.01 mm

Click **Synthesis** to determine the coupled line's parameters:

Width, **W**: 0.625 mm

Spacing, **S**: 0.019 mm

Odd-mode impedance, **Zo**: 28.82 ohms

Even-mode impedance, **Ze**: 86.74 ohms

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the physical length:

Physical length, **P**: 5.05 mm

A sweep analysis can be performed on **W**, **S**, or **Frequency**. To perform a sweep analysis on **W**, type **0.3,0.7,0.05** in the **W** field to specify a sweep from 0.3 mm to 0.7 mm in steps of 0.05 mm. Clicking **Analysis** then yields the following results:

Coplanar Coupled Strip Lines

B = 1.270mm ER = 2.20 TAND = 0.00000 T/B = 0.0079

Freq S/B W/B ZE ZO Cond + Diel Loss (dB/mm)

ghz Ohms Ohms Even Mode Odd Mode

10.0 0.015 0.236 134.5 33.7 0.0028 0.0108

10.0 0.015 0.276 123.9 32.7 0.0027 0.0105

10.0 0.015 0.315 114.9 31.8 0.0026 0.0102

10.0 0.015 0.354 107.2 31.1 0.0025 0.0100

10.0 0.015 0.394 100.4 30.4 0.0024 0.0098

10.0 0.015 0.433 94.5 29.7 0.0024 0.0096

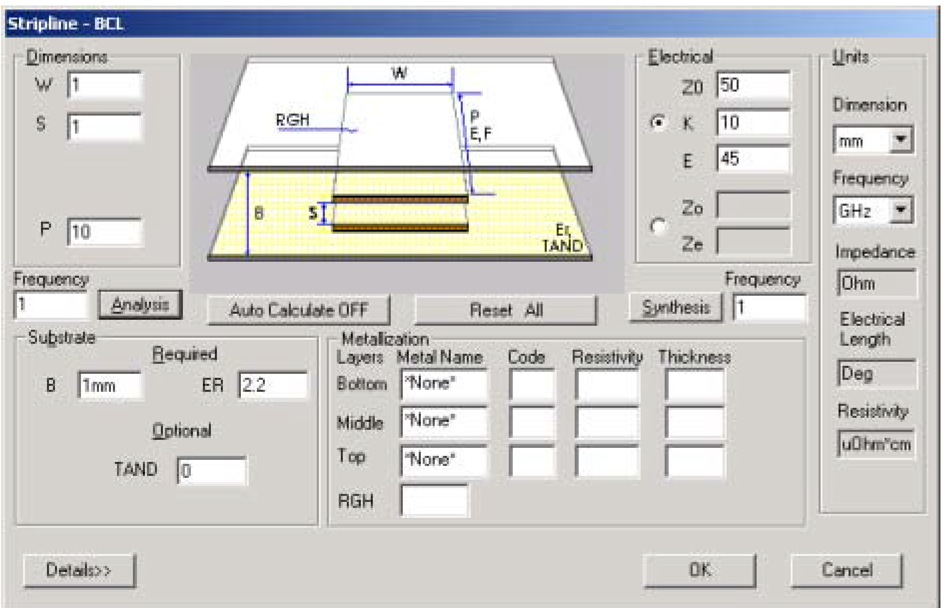
10.0 0.015 0.472 89.2 29.1 0.0023 0.0095

10.0 0.015 0.512 84.5 28.5 0.0023 0.0093

10.0 0.015 0.551 80.3 28.0 0.0022 0.0092

The variation of impedance and loss with conductor width is readily apparent.

TRL Broadside-Coupled Stripline Transmission Lines



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
S	Req (Phys)	m	Spacing between conductors
P	0	m	Physical length of transmission line
Z0	Req (Elec,Opt 1)	ohm	Impedance of the coupled lines
K	Req (Elec,Opt 1)	dB	Coupling coefficient (positive)
Zo	Req (Elec,Opt 2)	ohm	Odd-mode impedance of the coupled lines
Ze	Req (Elec,Opt 2)	ohm	Even-mode impedance of the coupled lines
E	0	deg	Electrical length of the coupled lines
Frequency	0	Hz	Analysis frequency
ER	Required		Relative dielectric constant

B	Required	m	Substrate thickness
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

Two options exist for synthesis of broadside-coupled stripline transmission lines:

— Specify impedance, **Z0**, and the coupling coefficient, **K**.

— Specify even-mode impedance, **Ze**, and odd-mode impedance, **Zo**.

- The substrate parameters **B**, and **ER** must be entered prior to clicking **Synthesis**. The width of the striplines, **W**, and the spacing between them, **S**, is computed. Also, the alternate electrical option set is computed. For example, entering **Z0** and **K** computes **Ze** and **Zo**.
- For analysis, the parameters **W**, **S**, **B**, and **ER** must be entered prior to clicking **Analysis**. The electrical properties, **Z0**, **K**, **Zo**, and **Ze** is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **B**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the structure’s propagation characteristics. Up to three conductors of different metals and thicknesses can be specified.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the broadside-coupled striplines are:

Frequency
Width, **W**
Spacing, **S**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the topic on *Sweep Entries* for further information.

Limitations

To maintain accuracy, the following limitations should be followed:

$$\frac{\mathbf{W}}{\mathbf{B}} \geq 0.35$$

and

$$\frac{\frac{\mathbf{W}}{\mathbf{B}}}{1-\frac{\mathbf{S}}{\mathbf{B}}} \geq 0.35$$

Example

To select the broadside-coupled stripline medium, select **TRL** on the **Product** menu, select **Stripline**, and click **BCL**. Set the units to **mm** and **GHz** and enter the following parameters to synthesize a 6-dB coupler at 10 GHz:

Impedance, **Z0**: 50 ohms

Coupling, **K**: 6 dB

Frequency: 10 GHz

Substrate thickness, **B**: 1.27 mm

Dielectric constant, **ER**: 2.2

Metal: Cu 0.01 mm

Click **Synthesis** to determine the coupler's parameters:

Width, **W**: 0.764 mm

Spacing, **S**: 0.294 mm

Odd-mode impedance, **Zo**: 28.82 ohms

Even-mode impedance, **Ze**: 86.74 ohms

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the line's physical length:

Physical length, **P**: 5.05 mm

A sweep analysis can be performed on **W**, **S**, or **Frequency**. To perform a sweep analysis on **W**, type **0.5,1.0,0.05** in the **W** field to specify a sweep from 0.5 mm to 1.0 mm in steps of 0.05 mm. Clicking **Analysis** then yields the following results:

Broadside Coupled Strip Lines

B = 1.270mm ER = 2.20 TAND = 0.00000 T/B = 0.0079

Freq S/B W/B ZE ZO Cond + Diel Loss (dB/mm)

ghz Ohms Ohms Even Mode Odd Mode

10.0 0.234 0.394 107.0 39.4 0.0016 0.0048

10.0 0.234 0.434 102.4 36.8 0.0015 0.0047

10.0 0.234 0.474 98.1 34.5 0.0014 0.0045

10.0 0.234 0.514 94.3 32.5 0.0014 0.0044

10.0 0.234 0.554 90.7 30.7 0.0013 0.0042

10.0 0.234 0.594 87.4 29.1 0.0012 0.0041

10.0 0.234 0.635 84.3 27.7 0.0012 0.0040

10.0 0.234 0.675 81.4 26.4 0.0012 0.0039

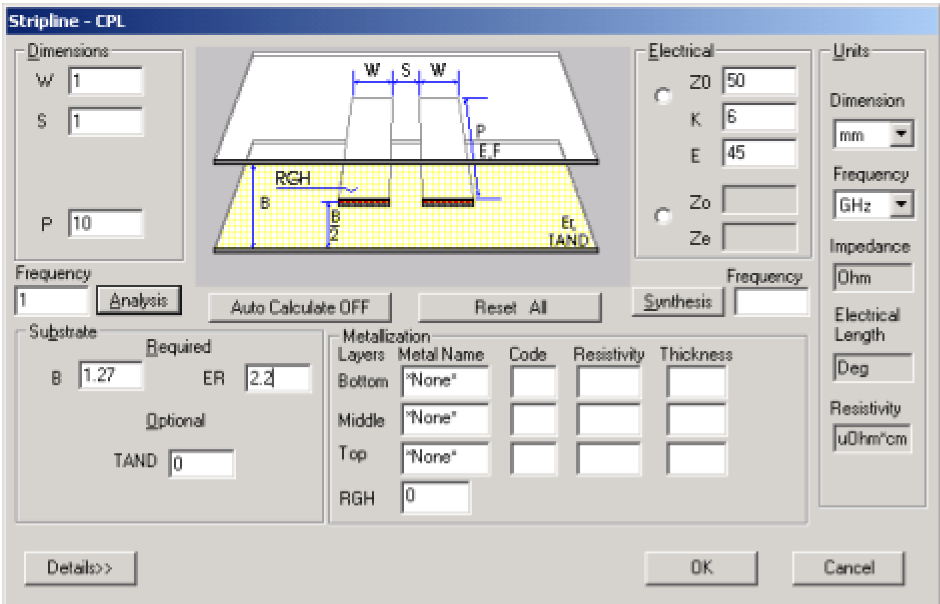
10.0 0.234 0.715 78.7 25.2 0.0011 0.0039

10.0 0.234 0.755 76.2 24.1 0.0011 0.0038

10.0 0.234 0.787 74.3 23.3 0.0010 0.0037

The variation of impedance and loss with conductor width is readily apparent.

TRL Suspended Stripline Transmission Line



Keyword	**Default**	Unit	**Description**
W	Req (Phys)	m	Conductor width
P	0	m	Physical length of transmission line
Z0	Req (Elec)	ohm	Transmission line impedance
E	0	deg	Electrical length of transmission line
Frequency	100 MHz	Hz	Analysis frequency
ER	Req		Relative dielectric constant
H	Req	m	Substrate thickness
HL	Req	m	Distance between bottom of substrate and ground
HU	Req	m	Cover height above substrate
A	inf.	m	Width between sidewalls
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

- To synthesize suspended substrate transmission lines, the parameters **Z0**, **H**, **HU**, **HL**, **A** and **ER** must be entered prior to clicking **Synthesis**. The width, **W**, is computed.
- For analysis, the parameters **W**, **H**, **HL**, **HU** and **ER** must be entered prior to clicking **Analysis**. The **A** parameter is assumed to be infinity. The impedance, **Z0**, is computed. The frequency is used if entered; otherwise, 100 MHz is used in the calculation.

- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the structure's propagation characteristics. Up to three conductors of different metals and thicknesses can be specified.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the suspended substrate transmission line are:

Frequency
Width, **W**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

Full-wave analysis is used; there are no limitations for practical dimensions.

Example

To select the suspended substrate medium, select **TRL** on the **Product** menu, select **Stripline**, and click **Single**. Set the units to **mm** and **GHz**. Enter the following parameters for synthesizing a 50-ohm transmission line:

Impedance, **Z0**: 50 ohms

Frequency: 10 GHz

Substrate thickness, **H**: 0.635mm

Lower height, **HL**: 1.0 mm

Upper height, **HU**: 1.0 mm

Sidewall width, **A**: 10.0 mm

Dielectric constant, **ER**: 9.8

Dielectric loss, **TAND**: 0.001

Metal: Cu 0.01 mm

Click **Synthesis** to determine the line's parameters:

Width, **W**: 1.79 mm

Keff: 2.25

Note that the impedance has changed slightly to 49.6 ohms. This is due to the synthesis being approximate for this medium

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the corresponding physical length:

Physical length, **P**: 5.0 mm

The dispersion of the medium can be observed by sweeping frequency. Sweep from 0 to 30 GHz in steps of 2 GHz by typing **0.1,30,2** in the **Frequency** field.

Single Line in Suspended Substrate Stripline

H = 0.635mm ER = 9.80 TAND = 0.00100 T/H = 0.0157

Freq Width W/H Z0 Keff C loss D Loss T Loss

ghz mm Ohms dB/mm dB/mm dB/mm

0.1 1.786 2.812 55.95 2.081 0.0002 0.0000 0.0002

2.1 1.786 2.812 55.63 2.088 0.0009 0.0001 0.0010

4.1 1.786 2.812 54.76 2.108 0.0013 0.0002 0.0015

6.1 1.786 2.812 53.41 2.142 0.0016 0.0004 0.0019

8.1 1.786 2.812 51.64 2.189 0.0019 0.0005 0.0024

10.1 1.786 2.812 49.54 2.251 0.0022 0.0007 0.0029

12.1 1.786 2.812 47.16 2.328 0.0025 0.0010 0.0034

14.1 1.786 2.812 44.58 2.422 0.0028 0.0013 0.0041

16.1 1.786 2.812 41.91 2.534 0.0032 0.0017 0.0048

18.1 1.786 2.812 39.27 2.667 0.0036 0.0021 0.0057

20.1 1.786 2.812 36.79 2.820 0.0040 0.0027 0.0067

22.1 1.786 2.812 34.59 2.995 0.0044 0.0034 0.0078

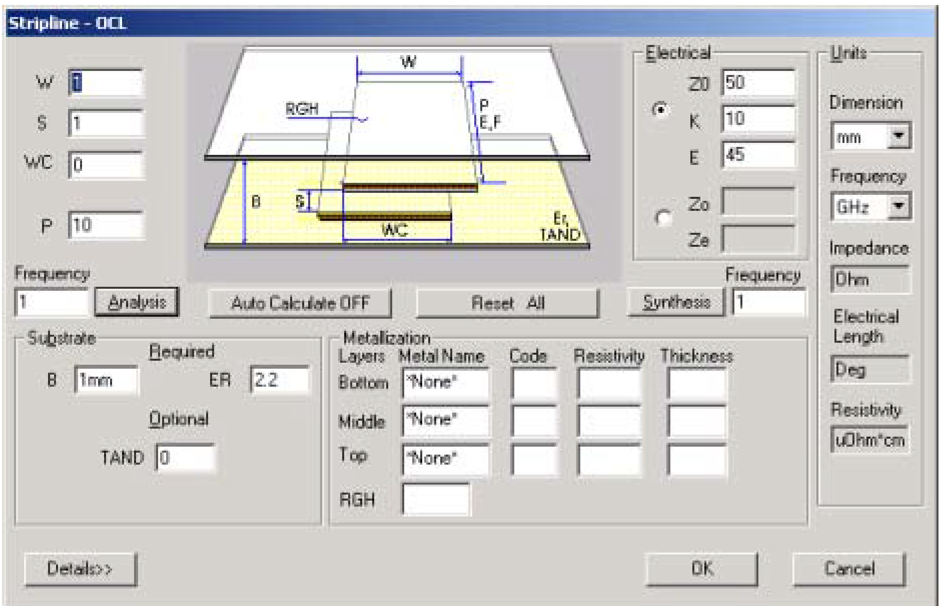
24.1 1.786 2.812 32.74 3.187 0.0048 0.0042 0.0090

26.1 1.786 2.812 31.22 3.394 0.0052 0.0050 0.0102

28.1 1.786 2.812 29.99 3.611 0.0056 0.0058 0.0114

30.0 1.786 2.812 29.02 3.822 0.0059 0.0065 0.0125

TRL Offset-Coupled Stripline Transmission Lines



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
S	Req (Phys)	m	Spacing between conductors
WC	0	m	Conductor offset
P	0	m	Physical length of transmission line
Z0	Req (Elec, Opt 1)	ohm	Impedance of the coupled lines
K	Req (Elec, Opt 1)	dB	Coupling coefficient (positive)
Zo	Req (Elec, Opt 2)	ohm	Odd-mode impedance of the coupled lines
Ze	Req (Elec, Opt 2)	ohm	Even-mode impedance of the coupled lines
E	0	deg	Electrical length of the coupled lines
Frequency	0	Hz	Analysis frequency
ER	Required		Relative dielectric constant

B	Required	m	Substrate thickness
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

Two options exist for synthesis of offset-coupled stripline transmission lines:

— Specify impedance, **Z0**, and the coupling coefficient, **K**.

— Specify even-mode impedance, **Ze**, and odd-mode impedance, **Zo**.

- The conductor spacing, **S**, and substrate parameters **B**, and **ER** must be entered prior to clicking **Synthesis**. The width of the striplines, **W**, and the offset between the lines, **WC**, is computed. Also, the alternate electrical option set is computed. For example, entering **Z0** and **K** computes **Ze** and **Zo**.
- For analysis, the parameters **W**, **S**, **B**, and **ER** must be entered prior to clicking **Analysis**. **WC** can also be entered, but defaults to 0. The electrical properties, **Z0**, **K**, **Zo**, and **Ze** is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency** and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **B**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the structure’s propagation characteristics. Up to three conductors of different metals and thickness can be specified.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for coupled offset stripline are:

Frequency
Width, **W**
Spacing, **S**
Offset, **WC**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

The approximate limitations on strip dimensions are:

(1) **Tight coupling case** (when the two strips overlap, **WC** > 0):

$$\frac{\mathbf{W}}{\mathbf{B}-\mathbf{S}} \geq 0.35$$

and

$$\frac{\mathbf{WC}}{\mathbf{S}} \geq 0.7$$

(2) **Loose coupling case** (when the two strips do not overlap, **WC** < 0):

$$\frac{\mathbf{W}}{\mathbf{B}-\mathbf{S}} \geq 0.35$$

and

$$\frac{2(\mathbf{W}-\mathbf{WC})}{\mathbf{B}+\mathbf{S}} \geq 0.85$$

Analyzing very loose coupling (e.g., coupling < 36 dB) requires a large number of iterations and the results may not be accurate. Synthesis, however, is accurate.

Example

To select the offset-coupled stripline medium, select **TRL** on the **Product** menu, select **Stripline**, and click **OCL**. Set the units to **mm** and **GHz**. Enter the following parameters to synthesize a 6-dB coupler at 10 GHz:

Impedance, **Z0**: 50 ohms

Coupling, **K**: 6 dB

Frequency: 10 GHz

Substrate thickness, **B**: 1.27 mm

Dielectric constant, **ER**: 2.2

Metal: Cu 0.01 mm

Spacing, **S**: 0.2 mm

Click **Synthesis** to determine the coupler's parameters:

Width, **W**: 0.728 mm

Offset, **WC**: 0.324 mm

Odd-mode impedance, **Zo**: 28.82 ohms

Even-mode impedance, **Ze**: 86.74 ohms

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the corresponding physical length:

Physical length, **P**: 5.05 mm

A sweep analysis can be performed on **W**, **S**, **WC**, or **Frequency**. To perform a sweep analysis on **W**, type **1,1.5,0.05** in the **W** field to specify a sweep from 1.0 mm to 1.5 mm in steps of 0.05 mm. Clicking **Analysis** then yields following results:

Offset Parallel Coupled Strip Lines

B = 1.270mm ER = 2.20

Freq WC WC/B S S/B W W/B ZE ZO

ghz mm mm mm Ohms Ohms

10.0 0.324 0.255 0.200 0.157 1.000 0.787 67.08 25.93

10.0 0.324 0.255 0.200 0.157 1.051 0.828 64.65 25.16

10.0 0.324 0.255 0.200 0.157 1.102 0.868 62.38 24.44

10.0 0.324 0.255 0.200 0.157 1.153 0.908 60.59 23.68

10.0 0.324 0.255 0.200 0.157 1.204 0.948 58.60 23.04

10.0 0.324 0.255 0.200 0.157 1.255 0.988 56.73 22.44

10.0 0.324 0.255 0.200 0.157 1.306 1.028 54.98 21.86

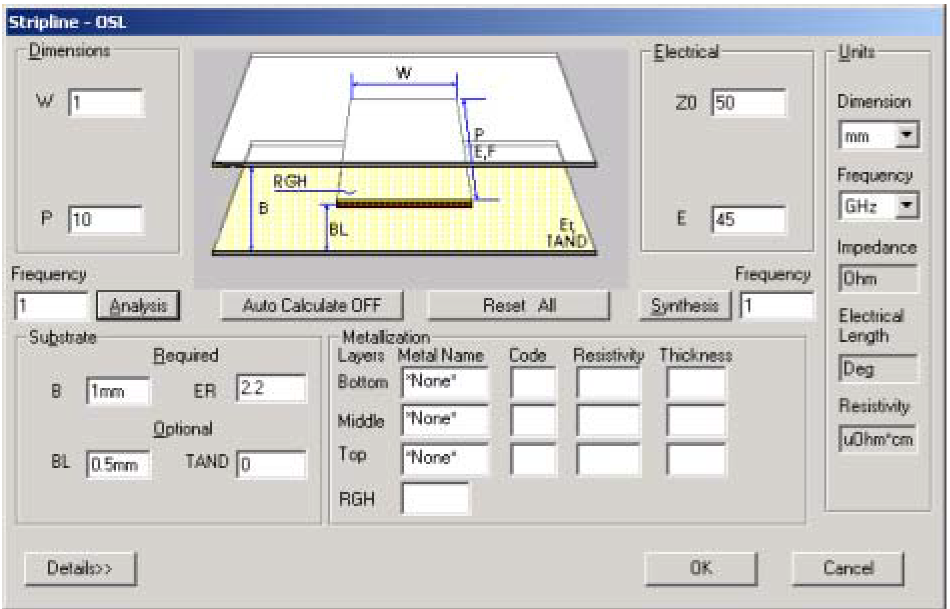
10.0 0.324 0.255 0.200 0.157 1.357 1.069 53.33 21.32

10.0 0.324 0.255 0.200 0.157 1.408 1.109 51.77 20.80

10.0 0.324 0.255 0.200 0.157 1.459 1.149 50.31 20.31

10.0 0.324 0.255 0.200 0.157 1.500 1.181 49.22 19.89

TRL Offset Stripline Transmission Line



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
P	0	m	Physical length of transmission line
Z0	Req (Elec)	ohm	Transmission line impedance
E	0	deg	Electrical length of transmission line
Frequency	0	Hz	Analysis frequency
ER	Req		Relative dielectric constant
B	Req	m	Substrate thickness
BL	Req	m	Offset spacing
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

- To synthesize offset stripline transmission lines, the parameters **Z0**, **B**, **BL,** and **ER** must be entered prior to clicking **Synthesis**. The width, **W**, is computed.
- For analysis, the parameters **W**, **B**, **BL**, and **ER** must be entered prior to clicking **Analysis**. The impedance, **Z0**, is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from

physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **B**, **BL**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the structure's propagation characteristics. Up to three conductors of different metals and thicknesses can be selected.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the offset stripline transmission line are:

Frequency
Width, **W**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

Error less than 0.5% for **W** /**B** < 10.

Example

To select the stripline medium, select **TRL** on the **Product** menu, select **Stripline**, and click **OSL**. Set the units to **mm** and **GHz**. Enter the following parameters to synthesize an offset stripline:

Impedance, **Z0**: 50 ohms

Dielectric constant, **ER**: 2.2

Substrate thickness, **B**: 2 mm

Offset Spacing, **BL**: 0.5mm

Click **Synthesis** to determine the width, W, of the transmission line:

Width, **W**: 1.1982 mm

Since offset stripline is a **TEM** structure and does not vary with frequency, perform a sweep analysis on the width of the strip instead. Type **(0.5, 3.0, 0.5)** into the **W** field to sweep the width from 0.5 mm to 3.0 mm in steps of 0.5 mm. Be sure to supply metal and loss information:

Metal: Cu 0.01 mm

TAND: 0.001

Click Analysis. The following warning message appears:

W/(B-T-BL) >0.7

Clicking Yes yields the following results:

Offset Stripline

B = 2.000mm BL = 0.500mm ER = 2.20 TAND =0.00100 T/B =0.0050

Freq Width W/B Z0 D Loss C Loss T Loss

ghz mm Ohms dB/mm dB/mm dB/mm

0.0 0.500 0.250 79.93 0.0000 0.0004 0.0004

0.0 1.000 0.500 55.40 0.0000 0.0002 0.0002

0.0 1.500 0.750 42.74 0.0000 0.0002 0.0002

0.0 2.000 1.000 34.86 0.0000 0.0001 0.0001

0.0 2.500 1.250 29.45 0.0000 0.0001 0.0001

0.0 3.000 1.500 25.50 0.0000 0.0001 0.0001

Change the frequency to 10 GHz and click analysis again:

Offset Stripline

B = 2.000mm BL = 0.500mm ER = 2.20 TAND =0.00100 T/B =0.0050

Freq Width W/B Z0 D Loss C Loss T Loss

ghz mm Ohms dB/mm dB/mm dB/mm

10.0 0.500 0.250 79.93 0.0014 0.0062 0.0076

10.0 1.000 0.500 55.40 0.0014 0.0035 0.0048

10.0 1.500 0.750 42.74 0.0014 0.0025 0.0039

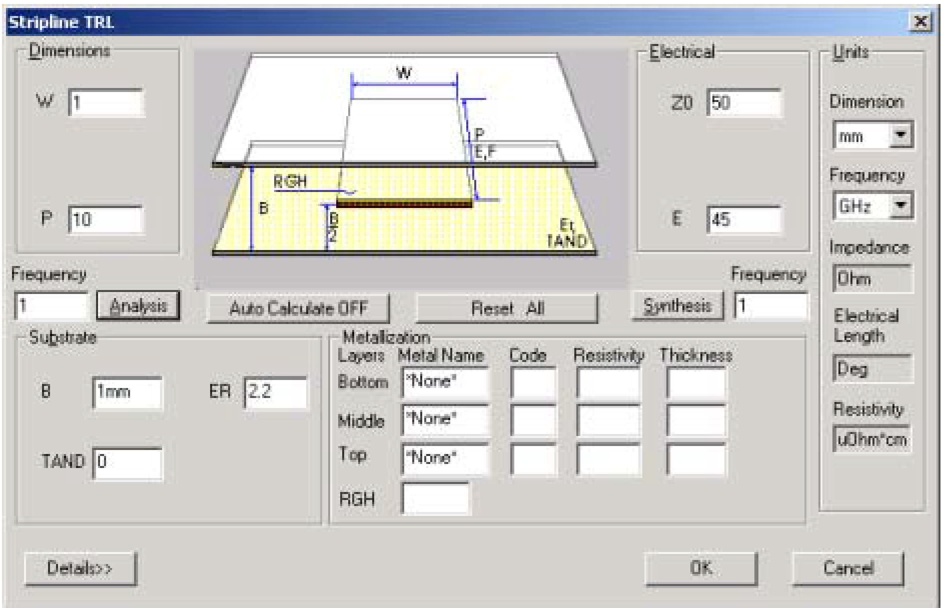
10.0 2.000 1.000 34.86 0.0014 0.0021 0.0034

10.0 2.500 1.250 29.45 0.0014 0.0018 0.0031

10.0 3.000 1.500 25.50 0.0014 0.0016 0.0030

Notice that the impedances are the same for DC or 10 GHz, but the conductor loss calculations differ significantly due to the skin effect.

TRL Edge-Coupled Suspended Stripline Transmission Lines



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width
S	Req (Phys)	m	Spacing between conductors
P	0	m	Physical length of transmission line
Z0	Req (Elec, Opt 1)	ohm	Impedance of the coupled lines
K	Req (Elec, Opt 1)	dB	Coupling coefficient (positive)
Zo	Req (Elec, Opt 2)	ohm	Odd-mode impedance of the coupled lines
Ze	Req (Elec, Opt 2)	ohm	Even-mode impedance of the coupled lines
E	0	deg	Electrical length of the coupled lines
Frequency	100 MHz	Hz	Analysis frequency
ER	Required		Relative dielectric constant
H	Req	m	Substrate thickness
HL	Req	m	Distance between bottom of substrate and ground
HU	Req	m	Cover height above substrate
A	inf.	m	Width between sidewalls
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

Two options exist for synthesis of edge-coupled suspended substrate transmission lines:

— Specify impedance, **Z0**, and the coupling coefficient, **K**.

— Specify the even-mode impedance, **Ze**, and the odd-mode impedance, **Zo**.

- The substrate parameters **H**, **HU**, **HL**, **A**, and **ER** must be entered prior to clicking **Synthesis**. The width of the strips, **W**, and the spacing between them, **S**, is computed. Also, the alternate electrical option set is computed. For example, entering **Z0** and **K** computes **Ze** and **Zo**.
- For analysis, the parameters **W**, **HU**, **HL**, and **ER** must be entered prior to clicking **Analysis**. The electrical properties, **Z0**, **K**, **Zo**, and **Ze** is computed. The frequency is used if entered; otherwise, 100 MHz is used in the calculation.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency** and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- The selection of the conductor is performed in the Metalization group. The conductor isn’t specified, in this case the conductor loss is zero and no thickness corrections are made to the propagation characteristics. Up to three conductors of different metal and different thickness can be selected.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for edge-coupled suspended substrate lines are:

Frequency
Width, **W**
Spacing, **S**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

Full-wave analysis is used; there are no limitations for practical dimensions.

Example

To select the edge-coupled suspended substrate medium, select **TRL** on the **Product** menu, select **Stripline**, and click **CPL**. Set the units to **mm** and **GHz**. Enter the following parameters to synthesize a 6-dB coupler at 10 GHz:

Impedance, **Z0**: 50 ohms

Coupling, **K**: 6 dB

Frequency: 10 GHz

Substrate thickness, **H**: 0.635 mm

Lower height, **HL**: 1.0 mm

Upper height, **HU**: 1.0 mm

Sidewall width, **A**: 10.0 mm

Dielectric constant, **ER**: 9.8

TAND 0.001

Metal: Cu 0.01 mm

Click **Synthesis** to determine the coupler's parameters:

Width, **W**: 1.122 mm

Spacing, **S**: 0.112 mm

Odd-mode impedance, **Zo**: 28.8 ohms

Even-mode impedance, **Ze**: 86.7 ohms

Odd-mode Keff, **KO**: 4.54

Even-mode Keff, **KE** 2.11

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the corresponding physical length:

Physical length, **P**: 4.12 mm

A sweep analysis can be performed on **W**, **S**, or **Frequency**. To perform a sweep analysis on **W**, type **0.8,1.2,0.05** in the **W** field to specify a sweep from 0.8 mm to 1.2 mm in steps of 0.05 mm. Clicking **Analysis** then yields the following results:

Coupled Lines in Suspended Substrate Stripline

H = 0.635mm ER = 9.80 TAND = 0.00100 T/H = 0.0157

Freq S/H W/H ZE ZO KE KO C Loss D Loss T Loss

ghz Ohms Ohms (dB/mm) (dB/mm) (dB/mm)

10.0 0.176 1.260 101.4 31.0 2.283 4.815 0.0102 0.0025 0.0127

10.0 0.176 1.339 98.8 30.6 2.251 4.770 0.0099 0.0025 0.0123

10.0 0.176 1.417 96.3 30.2 2.222 4.726 0.0096 0.0024 0.0120

10.0 0.176 1.496 93.9 29.8 2.194 4.682 0.0093 0.0024 0.0117

10.0 0.176 1.575 91.6 29.5 2.168 4.638 0.0091 0.0024 0.0114

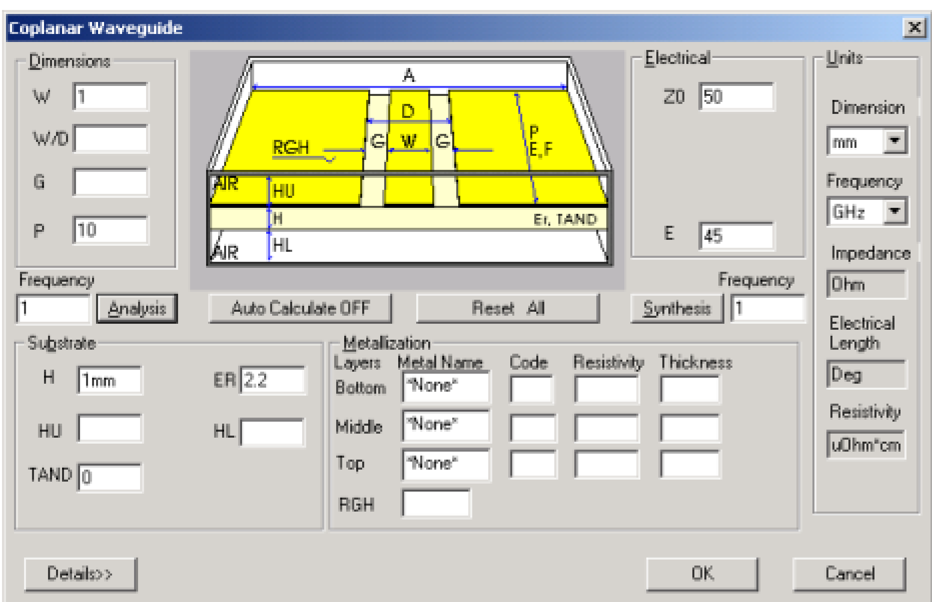
10.0 0.176 1.654 89.5 29.2 2.144 4.595 0.0088 0.0024 0.0112

10.0 0.176 1.732 87.5 28.9 2.122 4.553 0.0086 0.0023 0.0110

10.0 0.176 1.811 85.5 28.6 2.101 4.511 0.0084 0.0023 0.0107

10.0 0.176 1.890 83.7 28.4 2.082 4.469 0.0082 0.0023 0.0105

TRL Coplanar Waveguide Transmission Lines



Keyword	Default	Unit	Description
W	Req (Phys)	m	Conductor width

W/D			**W** /(**W** +2**G**)
G	Req (Phys)	m	Gap width
P	0	m	Physical length of transmission line
Z0	Req (Elec)	ohm	Characteristic impedance
E	0	deg	Electrical length of transmission line
Frequency	Req	Hz	Frequency
ER	Req		Relative dielectric constant
H	Req	m	Substrate thickness
HL	inf.	m	Distance between bottom of substrate and ground
HU	inf.	m	Cover height above substrate
A	inf.	m	Width between sidewalls
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

- To synthesize coplanar waveguide, the parameters (**Z0**, **G**, **H**, and **ER)** or **(Z0, W, H** and **ER)** must be entered prior to clicking **Synthesis**. The width, **W**, and **W/D** is computed if **G** is entered and **W** is left empty, while **G** is computed if **W** is entered and **G** is left empty.
- For analysis, the parameters (**W, G**) or (**W/D, G**), **H**, **Frequency** and **ER** must be entered prior to clicking **Analysis**. The impedance, **Z0**, is computed.
- For now, the parameter **A** is not taken into consideration, and thus has no effect on the results. A specific formula is used to take into account the effect of dispersion on the line's effective dielectric constant and characteristic impedance.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrates

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If a conductor is not specified, the conductor loss is zero and no thickness corrections are made to the line's propagation characteristics. Up to three conductors of different metal types and thicknesses can be specified. See the *Metalization* topic for further information.

- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for a coplanar waveguide analysis are:

Frequency
Normalized width ratio, **W/D**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

To maintain accuracy, the following limitations should be followed:,

$$\frac{H}{\lambda_g} < 0.05$$

where λ_g is the guide wavelength

Example

To select the coplanar waveguide medium, select **TRL** on the **Product** menu and select **Coplanar Waveguide**. Set the units to **mm** and **GHz**. Enter the following parameters for synthesizing a 50-ohm line:

Impedance, **Z0**: 50 ohms

Frequency: 10 GHz

Gap, **G**: 0.5 mm

Substrate thickness, **H**: 0.635 mm

Lower height, **HL**: 5.0 mm

Upper height, **HU**: 5.0 mm

Dielectric constant, **ER**: 9.8

Dielectric loss, **TAND**: 0.001

Metal: Cu 0.01 mm

Click **Synthesis** to determine the line’s parameters:

Width, **W**: 1.62241 mm

Keff: 4.012

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the corresponding physical length:

Physical length, **P**: 3.74157 mm

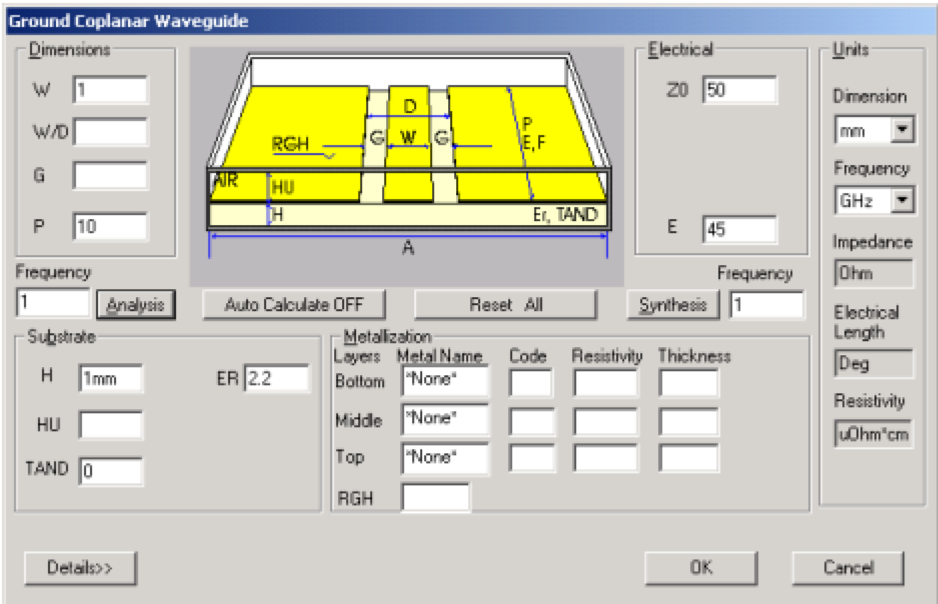
The loss of the medium can be observed by sweeping frequency. Sweep from 10 to 50 GHz in steps of 5 GHz by typing **10,50,5** in the **Frequency** field. Click **Analysis**. The results of the frequency sweep are as follows:

Conventional Coplanar Waveguide Analysis

G = 0.500mm H/G = 1.3 ER = 9.80 TAND = 0.00100 T/G = 0.0200

Freq	W/(W+2G)	W	Z0	Keff	C Loss	D loss	T Loss
ghz		mm	ohm		(db/mm)	(db/mm)	(db/mm)
10.0	0.619	1.622	50.03	4.012	0.0027	0.0015	0.0043
15.0	0.619	1.622	48.94	4.194	0.0034	0.0024	0.0058
20.0	0.619	1.622	47.69	4.416	0.0041	0.0033	0.0073
25.0	0.619	1.622	46.39	4.667	0.0047	0.0043	0.0090
30.0	0.619	1.622	45.12	4.934	0.0052	0.0054	0.0106
35.0	0.619	1.622	43.92	5.208	0.0058	0.0065	0.0124
40.0	0.619	1.622	42.80	5.482	0.0064	0.0078	0.0141
45.0	0.619	1.622	41.79	5.750	0.0069	0.0090	0.0160
50.0	0.619	1.622	40.89	6.008	0.0075	0.0104	0.0178

TRL Grounded Coplanar Waveguide Transmission Lines



Keyword	**Default**	Unit	**Description**
W	Req (Phys)	m	Conductor width
W/D			**W** /(**W** +2**G**)
G	Req (Phys)	m	Gap width
P	0	m	Physical length of transmission line
Z0	Req	ohm	Characteristic impedance
E	0	deg	Electrical length of transmission line
Frequency	Req	Hz	Analysis frequency
ER	Req		Relative dielectric constant
H	Req	m	Substrate thickness
HL	inf.	m	Distance between bottom of substrate and ground
HU	inf.	m	Cover height above substrate
A	inf.	m	Width between sidewalls
TAND	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness

Synthesis and Analysis

- To synthesize grounded coplanar waveguide, the parameters (**Z0**, **G**, **H**, and **ER)** or **(Z0, W, H** and **ER)** must be entered prior to clicking **Synthesis**. The width, **W**, and **W/D** is computed if **G** is entered and **W** is left empty, while **G** is computed if **W** is entered and **G** is left empty.

- For analysis, the parameters (**W, G**) or (**W/D, G**), **H**, **Frequency** and **ER** must be entered prior to clicking **Analysis**. The impedance, **Z0**, is computed.
- For now, the frequency is not taken into consideration, and thus it has no effect on the results other than the electrical length.
- Conversion from electrical length, **E**, to physical length, **P**, can be performed by entering values for **E** and **Frequency**. Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, and click **Analysis** to compute **E**.

Dielectric Substrate

A dielectric substrate is defined by the parameters **H**, **ER**, and **TAND**. The substrate is assumed to be lossless unless **TAND** is specified to be greater than zero.

Conductor Metalization

- Conductor specification is performed in the **Metalization** control group. If the conductor is not specified, the conductor loss is zero and no thickness corrections are made to the line’s propagation characteristics. Up to three conductors of different metals and thicknesses can be specified. See the *Metalization* topic for further information.
- Metalization rms surface roughness, **RGH**, can be specified for additional conductor losses due to imperfect metal surfaces. **RGH** is specified in terms of rms variation from an ideal flat surface.

Sweep Options

Parameters that can be swept for the grounded coplanar waveguide are:

Frequency
Normalized width ratio, **W/D**

The order shown is the order used to generate output data when multiple parameters swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

To maintain accuracy, the following limitations should be followed:,

$$\frac{\mathbf{H}}{\lambda_g} < 0.01$$

where λ_g is the guide wavelength

Example

To select the grounded coplanar waveguide medium, select **TRL** on the **Product** menu and select **Grounded Coplanar Waveguide**. Set the units to **mm** and **GHz**. Use the following parameters for synthesizing a 50-ohm line:

Impedance, **Z0**: 50 ohms

Frequency: 10 GHz

Gap, **G**: 0.5 mm

Substrate thickness, **H**: 0.635 mm

Upper height, **HU**: 5.0 mm

Dielectric constant, **ER**: 9.8

Dielectric loss, **TAND**: 0.001

Metal: Cu 0.01 mm

Click **Synthesis** to determine the line’s parameters:

Width, **W**: 0.522178 mm

Keff: 5.871

To specify a 1/4 wavelength line, enter:

Electrical length, **E**: 90

Click Synthesis again to determine the corresponding physical length:

Physical length, **P**: 3.09306 mm

The loss of the medium can be observed by sweeping frequency. Sweep from 5 to 50 GHz in steps of 5 GHz by typing **5,50,5** in the **Frequency** field. Click **Analysis**. The results of the frequency sweep are as follows:

Grounded Coplanar Waveguide Analysis

G = 0.500mm H/G = 1.3 ER = 9.80 TAND = 0.00100 T/G = 0.0200

Freq W/(W+2G) W Z0 Keff C Loss D loss T Loss

ghz mm Ohm (db/mm) (db/mm) (db/mm)

5.0 0.343 0.522 49.95 5.871 0.0033 0.0010 0.0043

10.0 0.343 0.522 49.95 5.871 0.0046 0.0020 0.0067

15.0 0.343 0.522 49.95 5.871 0.0057 0.0031 0.0087

20.0 0.343 0.522 49.95 5.871 0.0065 0.0041 0.0106

25.0 0.343 0.522 49.95 5.871 0.0073 0.0051 0.0124

30.0 0.343 0.522 49.95 5.871 0.0080 0.0061 0.0141

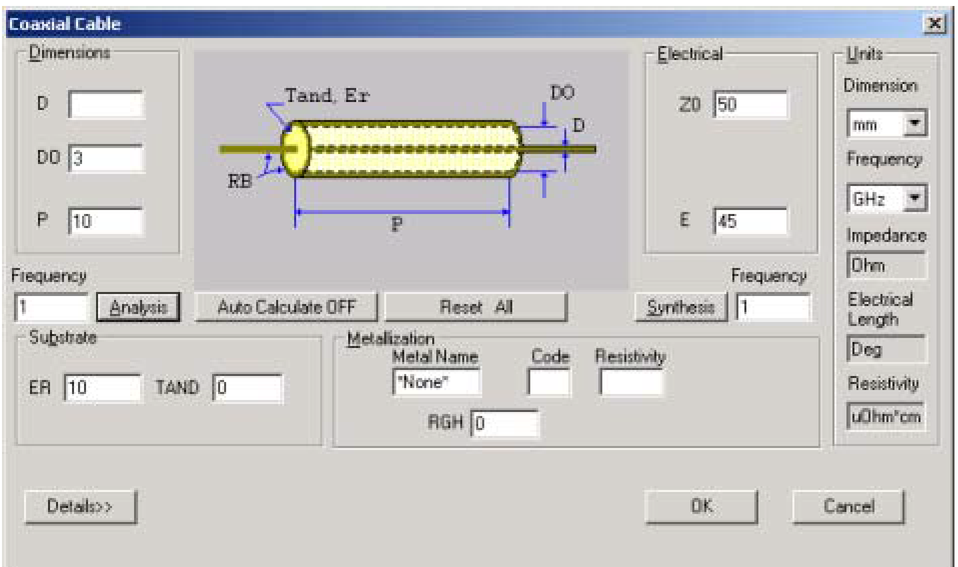
35.0 0.343 0.522 49.95 5.871 0.0086 0.0071 0.0158

40.0 0.343 0.522 49.95 5.871 0.0092 0.0082 0.0174

45.0 0.343 0.522 49.95 5.871 0.0098 0.0092 0.0190

50.0 0.343 0.522 49.95 5.871 0.0103 0.0102 0.0205

TRL Coaxial Cable



Keyword	**Default**	Unit	**Description** cc
D	Req (Phys)	m	Diameter of the inner conductor
DO	Req (Phys)	m	Diameter of the outer conductor
P	0	m	Physical length of coaxial cable
Z0	Req (Elec)	ohm	Characteristic impedance of coaxial cable
Er	Req		Relative dielectric constant
Frequency	0	Hz	Analysis frequency
Tand	0		Dielectric loss tangent
RGH	0	m	Rms surface roughness
E	0	deg	Electrical length of coaxial cable

Synthesis and Analysis

- To synthesize coaxial cable, the parameters **Z0, D0** and **ER** must be entered prior to clicking **Synthesis**. The inner diameter **D** is computed.
- For analysis, the parameters **D, D0,** and **ER** must be entered prior to clicking **Analysis**. The characteristic impedance, **Z0** is computed. The frequency is used if entered; otherwise, 0 Hz is used in the calculation.
- Conversion from electrical length **E,** to physical length **P,** can be performed by entering values for **E** and **Frequency.** Click **Synthesis** to compute **P**. Similarly, to convert from physical length to electrical length, enter values for **P** and **Frequency**, then click **Analysis** to compute **E.**

Dielectric

The line dielectric is defined by the parameters **ER** and **TAND**. The dielectric is assumed to be lossless unless **TAND** is specified and greater than zero.

Conductor Metalization

The conductor loss is specified by the resistivity, metalization rms surface roughness **RGH**, and the cable dimensions.

Sweep Options

Parameters that can be swept for coaxial cable are:

Frequency

Inner diameter, **D**

The order shown is the order used to generate output data when multiple parameters are swept simultaneously. See the *Sweep Entries* topic for further information.

Limitations

$\varepsilon r \geq 1$

DO > **D**

Example

To select coaxial cable, select **TRL** on the **Product** menu and select **CoaxialCable**. Select the units for mm and GHz. Enter the following parameters for synthesis and analysis of the coaxial cable:

Line impedance, **Z0**: 50 ohms

Diameter of outer conductor, **DO**: 5.0 mm

Dielectric constant, **ER**: 2.2

Frequency: 1 GHZ

Electric Length, **E**: 90 degrees

Dielectric loss, **TAND**: 0.001

Metal: Au

Click the **Synthesis** button to determine the diameter of the cable’s inner conductor:

Diameter of inner conductor, **D**: 1.453 mm

Physical length, **P**: 50.53 mm

Next, use analysis to sweep the diameter of the inner conductor from 1 to 2 mm. Enter the sweep by typing **1,2,0.1** in the **D** group box. Clicking **Analysis** then yields:

Electrical Length= 90.00 deg

Coaxial Cable Analysis

Do = 5.000mm ER = 2.20 TAND = 0.00100 rb = 2.4400 [micro-Ohm-cm]

Freq d Do/D Zo D Loss C Loss T Loss

ghz mm Ohms dB/mm dB/mm dB/mm

1.0000 1.0000 5.0000 65.1050 0.0001 0.0003 0.0004

1.0000 1.1000 4.5455 61.2495 0.0001 0.0002 0.0004

1.0000 1.2000 4.1667 57.7297 0.0001 0.0002 0.0004

1.0000 1.3000 3.8462 54.4918 0.0001 0.0002 0.0004

1.0000 1.4000 3.5714 51.4940 0.0001 0.0002 0.0004

1.0000 1.5000 3.3333 48.7031 0.0001 0.0002 0.0004

1.0000 1.6000 3.1250 46.0924 0.0001 0.0002 0.0004

1.0000 1.7000 2.9412 43.6400 0.0001 0.0002 0.0004

1.0000 1.8000 2.7778 41.3278 0.0001 0.0002 0.0004

1.0000 1.9000 2.6316 39.1407 0.0001 0.0003 0.0004

1.0000 2.0000 2.5000 37.0658 0.0001 0.0003 0.0004

You can also sweep the analysis frequency. You note that the impedance does not change with frequency since there is no dispersion for the coaxial cable model.

TRL References

The following topics list various TRL references according to topic.

TRL Microstrip References

Thomas G. Bryant and Jerald A. Weiss, “Parameters of Microstrip Transmission Lines and of Coupled Pairs of Microstrip Lines,” *IEEE Transactions onMicrowave Theory and Techniques*, vol. MTT-16, No. 12, pp. 1021- 1027; Dec. 1968.

K. C. Gupta, Ramesh Garg and I. J. Bahl, *Microstrip Lines and Slotlines*, Artech House, Inc., MA: Dedham, 1979.

K. C. Gupta, Ramesh Garg and Rakesh Chadha, *Computer-Aided Design of Microwave Circuits*, Artech House, Inc., MA: Dedham, 1981.

R. L. Ramey and T. S. Lewis, “Properties of Thin Metal Films at Microwave Frequencies,” *Journal of Applied Physics*, vol. 39, pp. 1747-1752; Feb. 15, 1968.

S. P. Morgan, “Effect of Surface Roughness on Eddy Current Losses at Microwave Frequencies,” *Journal of Applied Physics*, vol. 20, pp. 352-358; Apr. 1949.

J. D. Welch and H. J. Pratt, “Losses in Microstrip Transmission Systems for Microwave Circuits,” *1966 NEREM Record,* Boston, pp. 100-101; Nov. 1966.

M. V. Schneider, “Dielectric Loss in Integrated Microwave Circuits,” *Bell System Technical Journal*,” vol. 48, pp. 2325-2332; Sep/Oct. 1969.

M. Caulton, “Film Technology in Microwave Integrated Circuits,” *Proceedings of the IEEE,* vol. 59, pp. 1481-1489; Oct. 1971.

H. Sobol, “Application of Integrated Circuit Technology to Microwave Frequencies,” *Proceedings of the IEEE,* vol. 59, pp. 1202-1213; Aug 1971.

H. A. Wheeler, “Formulas for the Skin-Effect,” *Proceedings of the IRE*, vol. 30, pp. 412-424; Sep. 1942.

Robert A. Pucel, Daniel J. Massé and Curtis P. Hartwig, “Losses in Microstrip,” *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-16, pp. 342-350; Jun. 1968. (See also MTT-16, p. 1064; Dec 1968).

A. A. Milgram and C. S. Lu, “Preparation and Properties of Chromium Films,” *Journal of Applied Physics,* vol. 39, pp. 2851-2856; May. 1968.

E. Hammerstad and O. Jensen, “Accurate Models for Microstrip Computer-Aided Design,” *1980 IEEE MTT Symposium Digest*, pp. 407-409; Jun 1980.

W. J. Getsinger, “Microstrip Dispersion Model,” *IEEE Transactions onMicrowave Theory and Techniques*, vol. MTT-21, pp. 34-39; Jan. 1973.

G. Kompa and R. Mehran, “Planar Waveguide Model for Calculating Microstrip Components,” *Electronics Letters,* vol. 11, pp. 459-460; Sep. 18, 1975.

R. P. Owens, “Predicted Frequency Dependence of Microstrip Characteristic Impedance Using the Planar Waveguide Model,” *Electronics Letters,* vol. 12, pp. 269-270; May. 27, 1976.

H. F. Pues and A. R. van de Capelle, “Accurate Formulas for Frequency Dependence of Microstrip Parameters,” *Electronics Letters,* vol. 16, pp. 870-872; Nov 6 1980.

W. J. Getsinger, “Dispersion of Parallel-Coupled Microstrip,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-21, pp. 144-145; Mar. 1973.

S. L. March, “Microstrip Packaging: Watch the Last Step,” *Microwaves*, vol. 21, pp. 83-94; Dec. 1981.

W. J. Getsinger, “Measurement of the Characteristic Impedance of Microstrip Over A Wide Frequency Range,” *1982 IEEE MTT-S International Symposium Digest*, pp. 342-349; Jun. 1982.

R. Garg and I. J. Bahl, “Characteristics of Coupled Microstriplines,” *IEEETransactions on Microwave Theory and Techniques*, vol. MTT-27, pp. 700-705; Jul. 1979.

E. J. Delinger, “Losses of Microstrip Lines,” *IEEE Transactions on MicrowaveTheory and Techniques*, vol. MTT-28, pp. 513-522; Jun. 1980.

Edgar J. Denlinger, “A Frequency Dependent Solution for Microstrip Transmission Lines,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-19, pp. 30-39; Jan. 1971.

Robert A. Pucel and Daniel J. Massé, “Microstrip Propagation on Magnetic Substrates – Part I: Design Theory,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-20, pp. 304-308; May. 1972.

Daniel J. Massé and Robert A. Pucel, “Microstrip Propagation on Magnetic Substrates – Part II: Experiment,” *IEEE Transactions on Microwave Theoryand Techniques,* vol. MTT-20, pp. 309-313; May. 1972.

TRL Lange References

Julius Lange, “Interdigitated Stripline Quadrature Hybrid,” *IEEE Transactions on Microwave Theory and Techniques,* Vol. MTT-17, Dec. 1969, pp. 1150-1151.

Donald D. Paolino, “Design More Accurate Interdigitated Couplers,” *Microwaves*, May 1976, pp. 34-38.

Vittorio Rizzoli and Alessandro Lipparini, “The Design of Interdigitated Couplers for MIC Applications,”*IEEE Transactions on Microwave TheoryTech.*, Vol. MTT-26, No. 1, Jan. 1978, pp. 7-15.

R. M. Osmani, “Synthesis of Lange Couplers,” *IEEE Transactions onMicrowave Theory Tech.*, Vol. MTT-29, No. 2, Feb. 1981, pp. 168-170.

TRL Stripline References

S. B. Cohn, “Characteristic Impedance of the Shielded-Strip Transmission Line,” *IRE Transactions on Microwave Theory and Techniques,* vol. MTT-2, pp. 52-57; Jul. 1954.

Howe, Jr., *Stripline Circuit Design*, Artech House, Inc., MA: Dedham, 1974.

H. A. Wheeler, “Transmission Line Properties of a Stripline Between Parallel Planes,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-26, pp. 866-876; Nov. 1978.

S. B. Cohn, “Shielded Coupled-Strip Transmission Line,” *IRE Transactions onMicrowave Theory and Techniques,* vol. MTT-3, pp. 29-38; Oct. 1955.

S. B. Cohn, “Characteristic Impedances of Broadside Coupled Strip Transmission Lines,” *IRE Transactions on Microwave Theory and Techniques*, vol. MTT-8, pp. 633-637; Nov. 1960.

S. B. Cohn, “Thickness Corrections for Capacitive Obstacles and Strip Conductors,” *IRE Transactions on Microwave Theory and Techniques,* vol. MTT-8, pp. 638-644; Nov. 1960.

J. P. Shelton, “Impedances of Offset Parallel Coupled Strip Transmission Lines,” *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-14, pp.7-15; Jan 1966. (See also MTT-14, p. 149; May. 1966.)

R. Levy, “Transmission Line Directional Couplers for Very Broadband Operation,” *Proceedings of the IEEE,* vol. 112, pp. 469-476; Apr. 1965.

I. J. Bahl and P. Bhartia, “The Design of Broadside-Coupled Stripline Circuits,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-29, pp. 165-168; Feb. 1981.

TRL Suspended Substrate Stripline References

Eikichi Yamashita and Kazuhiko Atsuki, “Analysis of Thick-Strip Transmission Lines,” *IEEE Transactionson Microwave Theory and Techniques,* vol. MTT-19, pp. 120-122; Jan. 1971.

E. Yamashita and K. Atsuki, “Stripline with Rectangular Outer Conductor and Three Dielectric Layers,” *IEEE Transactions on Microwave Theory andTechniques*, vol. MTT-18, pp. 238-244; May. 1970.

J. B. Davies and D. Mirshekar-Syahkal, “Spectral Domain Solution of Arbitrary Coplanar Transmission Lines with Multilayer Substrate,” *IEEETransactions on Microwave Theory and Techniques,* vol. MTT-25, pp. 143-146; Feb. 1977.

F. E. Gardiol, “Careful MIC Design Prevents Waveguide Modes,” *Microwaves,* pp. 188-191; May. 1977.

D. Mirshekar-Syahkal and J. B. Davies, “Accurate Solution of Microstrip and Coplanar Structures for Dispersion and for Dielectric Conductor Losses,” *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-27, pp. 694-699; Sep. 1979.

TRL Coplanar Waveguide References

C. P. Wen, “Coplanar Waveguide: A Surface Strip Transmission Line Suitable for Non-reciprocal Gyromagnetic Device Application,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-17, pp. 1087-1090; Dec. 1969.

C. P. Wen, “Attenuation Characteristics of Coplanar Waveguides,” *Proceedings of the IEEE,* vol. 58, pp. 141-142; Jan. 1970.

J. B. Knorr and J. D. Kuchler, “Analysis of Coupled Slots and Coplanar Strips on Dielectric Substrate,” *IEEE Transactions on Microwave TheoryandTechniques,* vol. MTT-23, pp. 541-548; Jul. 1975.

T. Kitazawa, Y. Hayashi and M. Suzuki, “A CoplanarWaveguide with Thick Metal Coating,” *IEEE Transactions on Microwave Theory and Techniques,* vol. MTT-24, pp. 604-608; Sep. 1976.

V. F. Hanna, “Finite Boundary Corrections to Coplanar Stripline Analysis,” *Electronics Letters,* vol. 15, pp. 88-90; Feb 1, 1979.

B. E. Spielman, “Dissipation Loss Effects in Isolated and Coupled Transmission Lines,” *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-25, pp. 648-656; Aug. 1977.

B. E. Spielman, “Dissipation Loss Effects in Isolated and Coupled Transmission Lines,” *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-25, pp. 648-656; Aug. 1977.

G. Ghione and C. Naldi, “Analytical formulas for Coplanar lines in hybrid and monolithic MICs,” *Electronics Letters,* vol. 20, Feb. 1984 pp. 179-181 From subroutine CPW1 in CPWG.

K. C. Gupta, Ramesh Garg and I. J. Bahl, “Microstrip Lines and Slotlines,” pp. 281-287

TRL Grounded Coplanar Waveguide References

G. Ghione and C. Naldi, “Parameters of Coplanar Waveguides with Lower Ground Plane,” *Electronics Letters*, vol. 19, pp. 734-735 Sept. 1, 1983,

G. Ghione and C.U. Naldi, “Coplanar Waveguides for MMIC Applications: Effect of Upper Shielding, Conductor Backing, Finite-Extent Ground Planes, and Line to- Line Coupling,” *IEEE Transactions on MicrowaveTheory and Techniques,* vol. MTT-35, pp. 260-267; Mar. 1987.

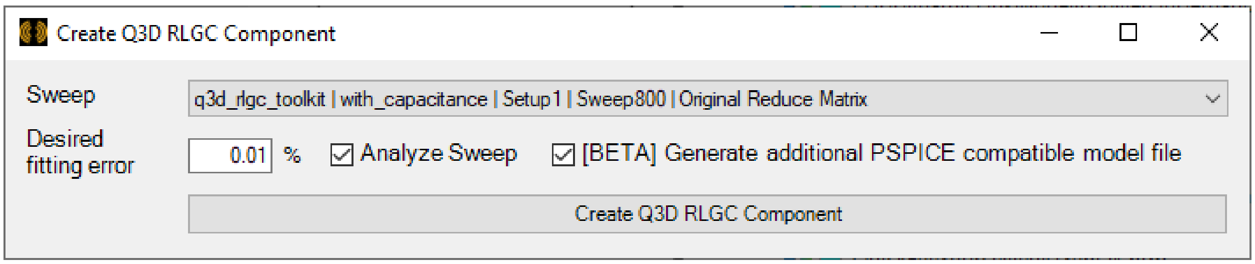
Q3D RLGC Component Toolkit

Nexxim supports dynamic links with Q3D through state-space fitting on the S-parameter, but it suffers accuracy issues for some areas of applications. The Q3D RLGC Component toolkit supports Nexxim Q3D links through state-space fitting on RLGC data to improve accuracy.

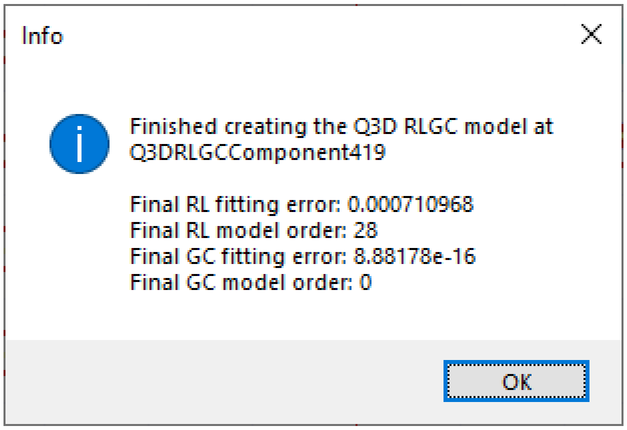
Creating a Component

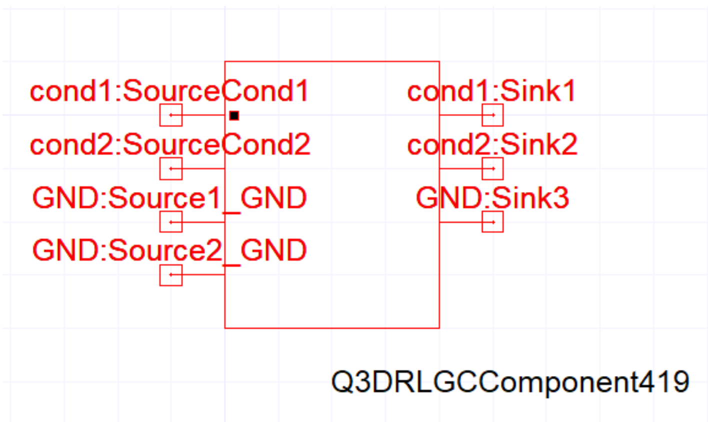
Complete the following steps to create a component.

1. Open a project with a Q3D design.
2. Insert a circuit design.
3. You can access the toolkit by two ways:
   i. Select the Circuit design and browse to the Automation ribbon. Click on **Create Q3D RLGC Component**.
   ii. From the Circuit menu, select **Toolkit** > **Q3D RLGC Component** to open the **Create Q3D RLGC Component** window.

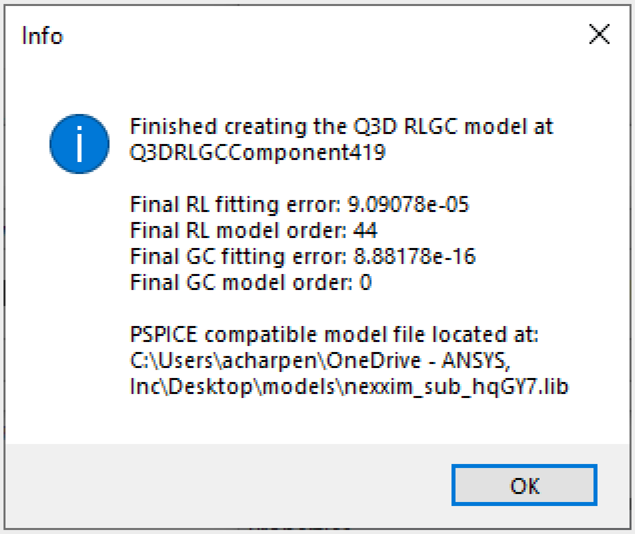


4. Choose a Q3D solution from the **Sweep** drop-down menu.
5. In the **Desired fitting error** field, enter an acceptable error for the RLGC data state space fitting.
6. Check the **Analyze Sweep** box, if appropriate.
7. Check the **[BETA] Generate additional PSPICE compatible model file** box, if appropriate. The PSPICE-compatible model is saved in the project directory. RLGC sub-circuit terminals are named after Q3D terminals.
8. Click **Create Q3D RLGC Component**. The toolkit invokes the state space fitting algorithm to fit the RLGC data and writes a Nexxim subcircuit netlist to generate a schematic component. When it is done, the **Info** window opens to report success or failure and the new Nexxim subcircuit component appears in the Schematic editor. Use the component to create circuits for Nexxim simulations.


Info
Finished creating the Q3D RLGC model at
Q3DRLGCComponent419
Final RL fitting error: 0.000710968
Final RL model order: 28
Final GC fitting error: 8.88178e-16
Final GC model order: 0
OK


cond1:SourceCond1
cond2:SourceCond2
GND:Source1_GND
GND:Source2_GND
cond1:Sink1
cond2:Sink2
GND:Sink3
Q3DRLGCComponent419

Note: If the **[BETA] Generate additional PSPICE compatible model file** box is checked, the toolkit opens an explorer window to the directory where the PSPICE-compatible model is stored. The file path is also shown in the **Info** window.



9. Use this component to create circuits for Nexxim simulations.

The procedure is complete.

Automatic Causality Correction

Q3D may return non-causal RL data in certain circumstances. Non-causal data can cause a poor fit when generating a time domain circuit model. Automatic Causality Correction chooses the best fit to the Q3D data between the original S-Parameter data (S-data) and the causal-corrected S-data. A fitting error threshold of 5% on the original fit is used to trigger the correction routine. The threshold is adjustable when generating the model via command line from the *.rlgcdata* file. Refer to the **causality_thresh** entry in "Running State-Space Fitting on the Command Line" on page 25-90.

Passivity Enforcement (Beta Feature)

In simulations where non-passivity of an RLGC PSPICE circuit model export is suspected, the Passivity Enforcement feature can be activated from the command line. Refer to the **passive_rlfit** entry in "Running State-Space Fitting on the Command Line" on page 25-90.

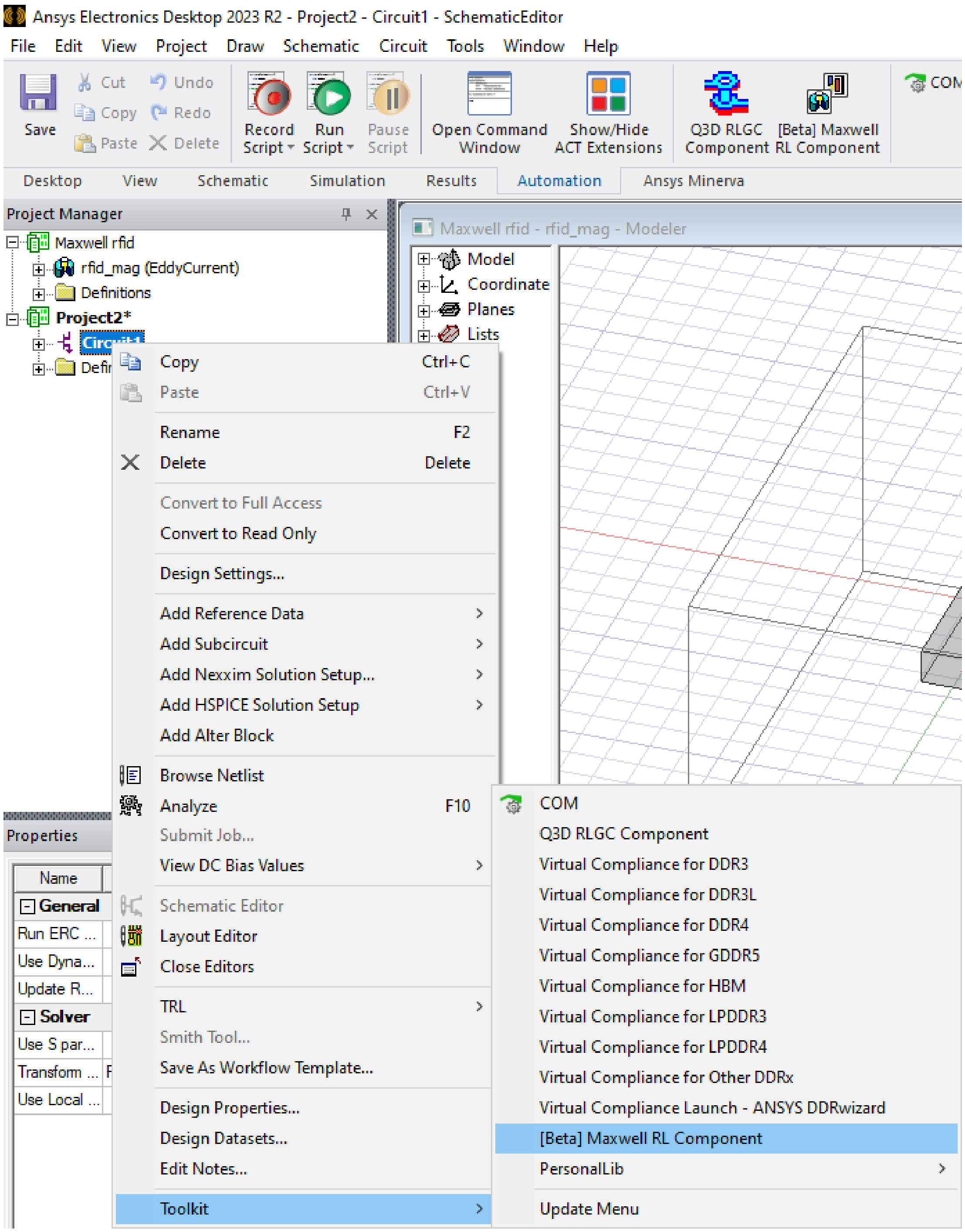
Maxwell RL Component Toolkit

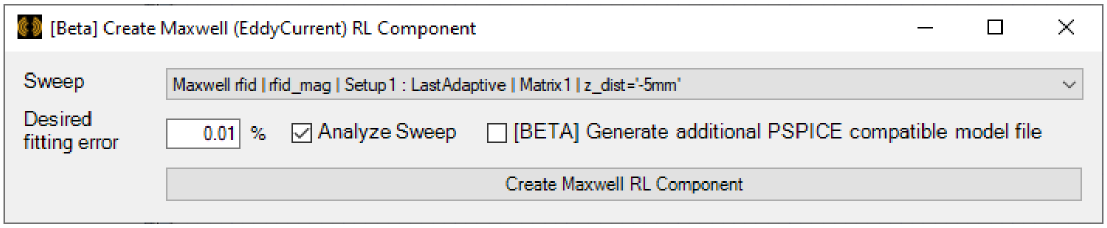
The Maxwell RL Component toolkit supports Circuit-Maxwell links through state-space fitting on RL data. This feature is currently in Beta mode.

Creating a Component

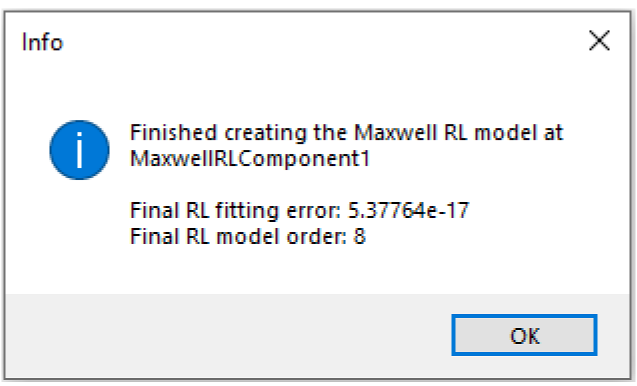
Complete the following steps to create a component.

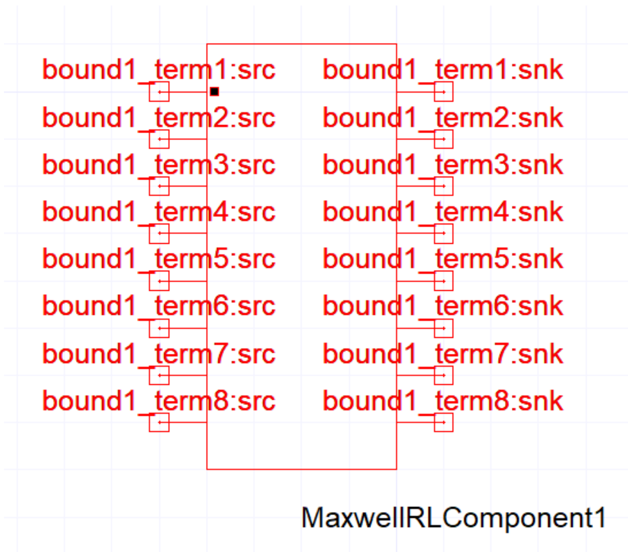
1. Open a project with a Maxwell EddyCurrent design.
2. Insert a circuit design.
3. You can access the toolkit by two ways:
   i. Select the Circuit design and browse to the Automation ribbon. Click on **[Beta] Maxwell RL Component**
   ii. From the Circuit menu, select **Toolkit** > **[Beta] Maxwell RL Component** to open the **[Beta] Create Maxwell (EddyCurrent) RL Component** window.


Ansys Electronics Desktop 2023 R2 - Project2 - Circuit1 - SchematicEditor
File Edit View Project Draw Schematic Circuit Tools Window Help
Save
Cut
Copy
Paste
Undo
Redo
Delete
Record Script
Run Script
Pause Script
Open Command Window
Show/Hide ACT Extensions
Q3D RLGC Component
[Beta] Maxwell RL Component
Desktop View Schematic Simulation Results Automation Ansys Minerva
Project Manager
Maxwell rfid
rfid_mag (EddyCurrent)
Definitions
Project2*
Maxwell rfid - rfid_mag - Modeler
Model
Coordinate
Planes
Lists
Copy Ctrl+C
Paste Ctrl+V
Rename F2
Delete Delete
Convert to Full Access
Convert to Read Only
Design Settings...
Add Reference Data
Add Subcircuit
Add Nexxim Solution Setup...
Add HSPICE Solution Setup
Add Alter Block
Browse Netlist
Analyze F10
Submit Job...
View DC Bias Values
Schematic Editor
Layout Editor
Close Editors
TRL
Smith Tool...
Save As Workflow Template...
Design Properties...
Design Datasets...
Edit Notes...
Toolkit
COM
Q3D RLGC Component
Virtual Compliance for DDR3
Virtual Compliance for DDR3L
Virtual Compliance for DDR4
Virtual Compliance for GDDR5
Virtual Compliance for HBM
Virtual Compliance for LPDDR3
Virtual Compliance for LPDDR4
Virtual Compliance for Other DDRx
Virtual Compliance Launch - ANSYS DDRwizard
[Beta] Maxwell RL Component
PersonalLib
Update Menu
Properties
Name
General
Run ERC ...
Use Dyna...
Update R...
Solver
Use S par...
Transform ...
Use Local ...

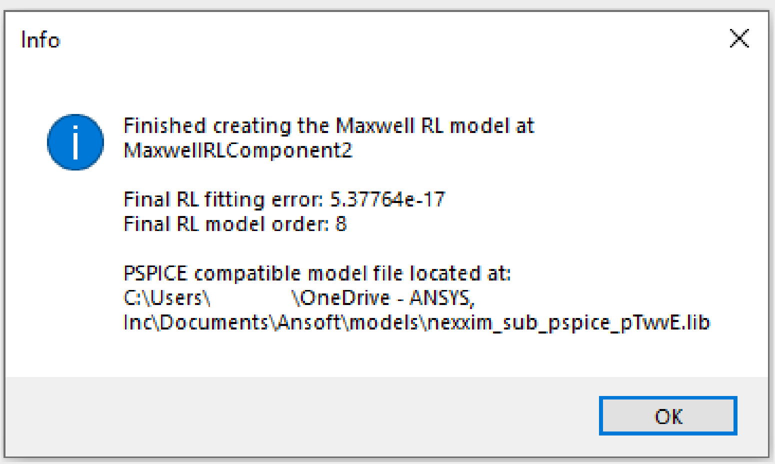


4. Choose a Maxwell solution from the **Sweep** drop-down menu.
5. In the **Desired fitting error** field, enter an acceptable error for the RL data state space fitting.
6. Check the **Analyze Sweep** box, if appropriate.
7. Check the **[BETA] Generate additional PSPICE compatible model file** box, if appropriate. The PSPICE-compatible model is saved in the project directory. RLGC sub-circuit terminals are named after Maxwell terminals.
8. Click **Create Maxwell RL Component**. The toolkit invokes the state space fitting algorithm to fit the RL data and writes a Nexxim subcircuit netlist. It then uses the Nexxim subcircuit netlist to generate a schematic component. When it is done, the **Info** window opens to report success or failure and the new Nexxim subcircuit component appears in the Schematic editor. Use the component to create circuits for Nexxim simulations.





Note: If the **[BETA] Generate additional PSPICE compatible model file** box is checked, the toolkit opens an explorer window to the directory where the PSPICE-compatible model is stored. The file path is also shown in the **Info** window.



9. Use this component to create circuits for Nexxim simulations.

The procedure is complete.

10 - Circuit Netlist Operations

The Circuit Netlist Editor is an easy-to-use tool for viewing and editing netlists. The input conventions are the same as those of standard text editors, and the Netlist Editor also contains special functions that integrate it with the desktop environment.

Typically, a netlist is a coded textual representation that describes the connections, subcircuits, ports, and other components in a circuit. The netlist contains electrical information and optional references to substrate definitions. Netlist contributions from circuit components such as coupled lines and transmission lines can contain information about the physical layout of the component.

- The Circuit simulator supports both netlist-based designs and schematic-based designs. A netlist-based design consists of just a netlist definition, but with a schematic-based design, a netlist is created internally to represent the circuit.
- The Planar EM tool does not generate a netlist, but simulates directly on the geometry and substrate information.

Related Topics

Starting the Netlist Editor

Netlist Editor Operation

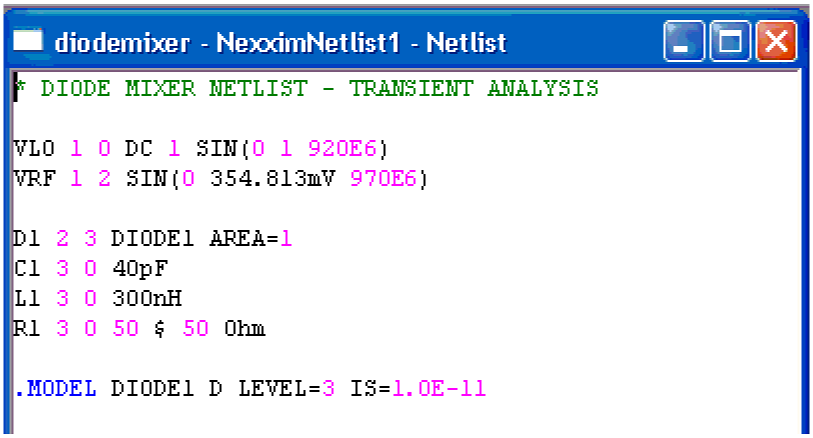
Starting the Netlist Editor

The Netlist Editor appears automatically when you create a new netlist design. You can also load a netlist-based design from an external file and edit it using the Netlist Editor. In schematic-based circuit designs, use the Netlist Editor to view the derived netlist without making any changes. See Netlist Format for further information.

Creating a New Netlist

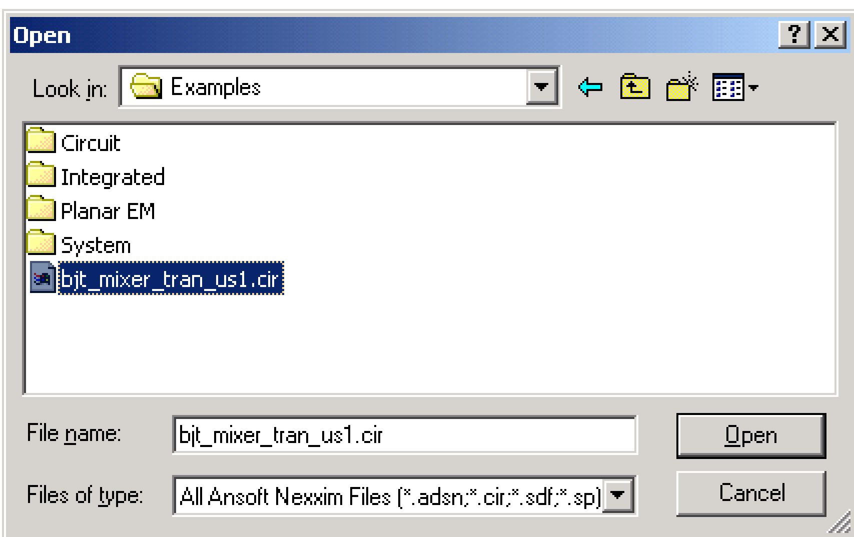
The Netlist Editor is automatically started when you insert a new netlist design. To create and edit a new Circuit Netlist design:

1. Click **File** > **New** to create a new Project.
2. Do one of the following:
   - From the **Project** menu, select **Insert Circuit Netlist**
   - From the **Project Manager** window, right-click the **Project** folder and select **Insert** > **Insert Circuit Netlist**
3. A new Netlist Editor window opens in the design area.

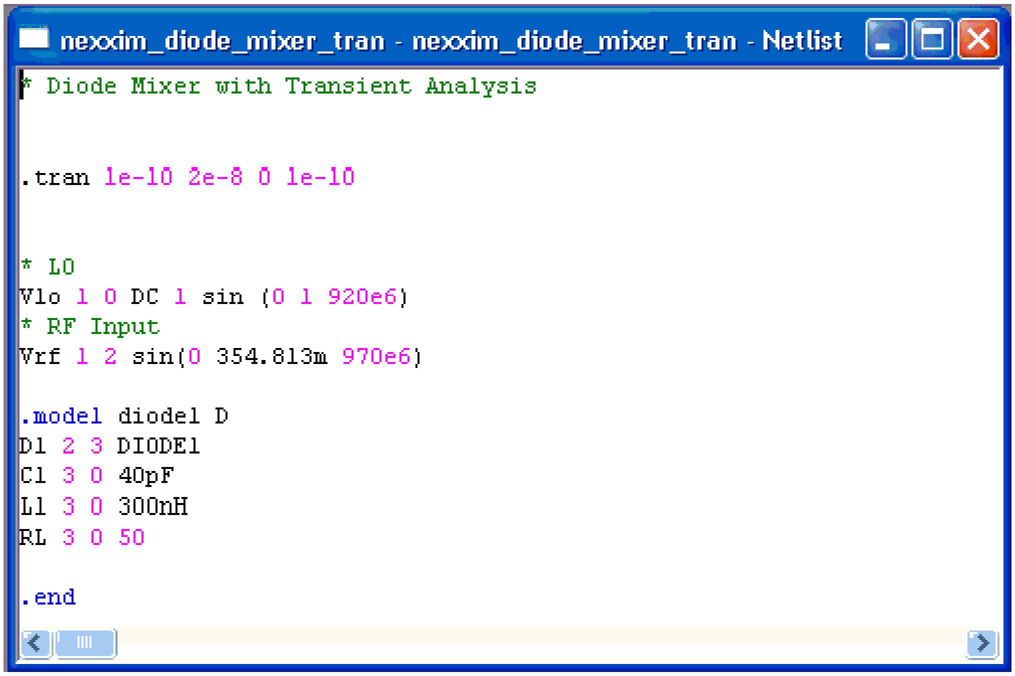


Loading a Netlist from a File

To load a design from an existing Circuit netlist file, Click the **File > Open** command. Select the “All Ansoft Nexxim Files” extension in the **Files of type** window of the explorer window.



The file opens in the Netlist Editor window.




```
* Diode Mixer with Transient Analysis


.tran 1e-10 2e-8 0 1e-10


* LO
Vlo 1 0 DC 1 sin (0 1 920e6)
* RF Input
Vrf 1 2 sin(0 354.813m 970e6)

.model diode1 D
D1 2 3 DIODE1
C1 3 0 40pF
L1 3 0 300nH
RL 3 0 50

.end
```

The Netlist Editor display is colored to provide easy identification of variables, values, comments, bookmarks, and other elements of the netlist.

Viewing Netlists in Schematic Designs

Circuit designs typically use the **Schematic Editor** to create circuits. In these designs, the schematic defines the circuit, and the netlist may be viewed for reference. Internally, the simulators generate a netlist on the schematic, and update it on the schematic right before performing any simulation.

You can view the netlist in any schematic design by selecting **Browse Netlist** on the **Product** menu. Or, in any simulator design, right-click the name of the design in the **Project** window, and select **Browse Netlist**.

> **Note:**
>
> The netlist in a schematic-based design is for reference only, and should not be modified in the Netlist Editor view. The Netlist Editor allows you to modify the netlist, and asks if you want to save the changes. However, any modifications you make is overwritten the next time the netlist is generated on the schematic.

Netlist Editor Operation

The Netlist Editor is a text editor that allows you to perform many common editor operations in order to enter, select, modify, and delete netlist text. The Netlist Editor provides additional commands that allow you to:

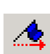
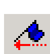
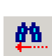
- Search text
- Add bookmarks
- Search and replace text
- Go to a numbered line in the netlist

Netlist Editor commands can be accessed on the toolbar, on the Edit menu, or by using keyboard shortcuts.

Netlist Editor Toolbar

When the Netlist Editor is the active window, the **Netlist Editor Toolbar** is displayed in the toolbar area.

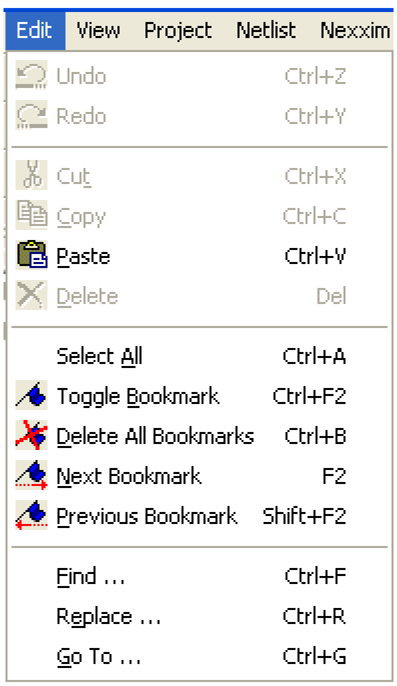
The following table summarizes the Netlist Editor Toolbar commands.

Command Function	Toolbar Commands and Description	
Toggle Bookmark		Inserts a bookmark at the active cursor position, or turns off an existing bookmark.
Delete All Bookmarks		Deletes all bookmarks in the text.
Next Bookmark		Moves the cursor to the next bookmark in the text.
Previous Bookmark		Moves the cursor to the previous bookmark in the text.
Find Window		Displays the text string you have entered to find. Click the drop-down menu to display previously searched for text strings.
Find Forward		Searches forward in the netlist for text entered in the Find Window.
Find Backward		Searches backward in the netlist for text entered in the Find Window.

Command Function	Toolbar Commands and Description	
Find Forward, Case-sensitive		Searches forward in the netlist for text entered in the Find Window with case-sensitive control.
Find Backward, Case-sensitive		Searches backward in the netlist for text entered in the Find Window with case-sensitive control.

Netlist Edit Menu

When the Netlist Editor is the active window, the Edit menu on the menu includes Netlist Editor commands.



The menu also identifies keystroke shortcuts that may be used in place of the mouse-oriented command methods.

- The **Replace** and **Go To** commands are not available on the Netlist Editor Toolbar. These commands must be executed on the Edit menu or with a keystroke shortcut.

For more information, search the Help or use the Help Index to find information on any of the commands that appear on the **Netlist Edit Menu**.

Using Bookmarks

A bookmark is a colored overlay that highlights a line in the netlist text that is significant to you. Once created, netlist editor bookmarks allow quick access to text sections.

Inserting a Bookmark

To insert a bookmark, position the cursor anywhere in the line to be marked, then do one of the following:

- Click the Toggle Bookmark command icon  on the toolbar
- Select **Edit** > **Toggle Bookmark**
- Press **Ctrl** +**F2**

A colored overlay indicates that the line is bookmarked. The color used for bookmarks can be specified as a Netlist Editor option. See Netlist Editor Options for further information.

Deleting a Bookmark

To delete a single bookmark, position the cursor anywhere in the bookmarked line, then do one of the following:

- Click the Toggle Bookmark icon  on the toolbar
- Select **Edit** > **Toggle Bookmark**
- Press **Ctrl**+**F2** to close the overlay.

Deleting All Bookmarks

To delete all bookmarks in the netlist, do one of the following:

- Click the Delete All Bookmarks icon .
- Select **Edit** > **Delete All Bookmarks**
- Press **Ctrl** +**B**

Moving Forward to the Next Bookmark

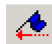
To move the cursor forward to the next bookmark, do one of the following:

- Click the Next Bookmark icon
  
- Select **Edit** > **Next Bookmark**
- Press **F2**

If no bookmarks have been inserted, the **NextBookmark** command does not move the cursor. If the cursor is on or past the last bookmark in the netlist, the **NextBookmark** command moves the cursor to the first bookmark in the netlist.

Moving Backward to the Previous Bookmark

To move the cursor backward to the previous bookmark, do one of the following:

- Click the Previous Bookmark icon .
- Select **Edit** > **Previous Bookmark**
- Press **Shift** +**F2**

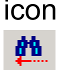
If no bookmarks have been inserted, the **PreviousBookmark** command does not move the cursor. If the cursor is on or before the first bookmark in the netlist, the **PreviousBookmark** command moves the cursor to the last bookmark in the netlist.

Searching the Netlist

The Netlist Editor can locate an item of test in the netlist. You can specify the search parameters via the toolbar icons, on the Edit menu, or with a keystroke shortcut.

Searching on the Toolbar

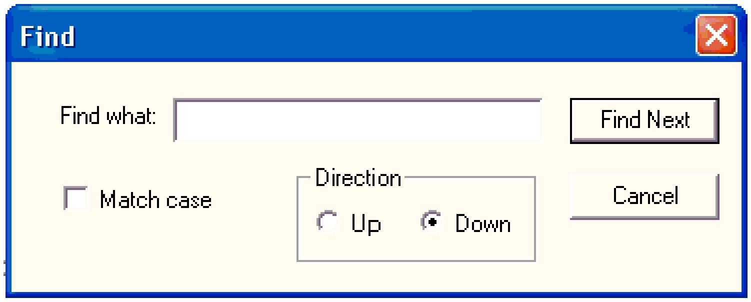
To search for an item of text in the netlist, enter the search text in the **Find** window in the toolbar:


- To search forward using case-insensitive matching, click the **Find Forward** toolbar icon
  
- To search backward using case-insensitive matching, click the **Find Backward** toolbar icon
  
- To search forward using case-sensitive matching, click the **Find Forward**, **case-sensitive** toolbar icon
  
- To search backward using case-sensitive matching, click the **Find Backward**, **case-sensitive** toolbar icon
  

In all search cases, if the cursor is past the location of the search text, the search wraps on the beginning of the netlist. If any search fails to find the target text, the editor notifies you with a message.

Searching from the Edit Menu or by Keyboard Shortcut

Start a text search by selecting **Edit** > **Find** or by pressing **Ctrl+F** on your keyboard. Either of these operations opens the **Find** window:

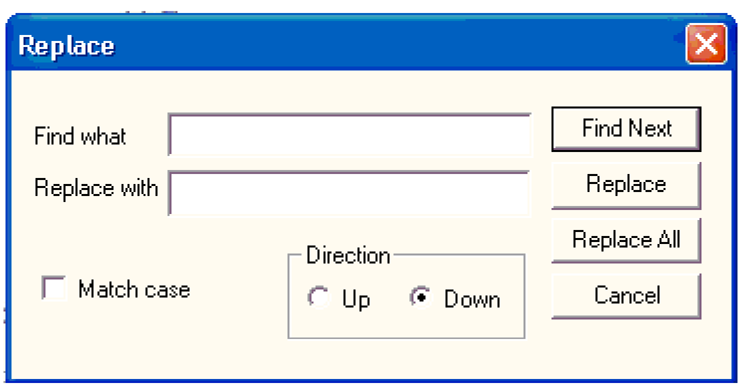


1. Enter the search text in the **Find what** field.
2. Select the **Direction** for the search. **Up** searches on the cursor toward the front of the netlist, wrapping on the front to the back if the text is not found. **Down** searches on the cursor toward the back of the netlist, wrapping from back to front is the text is not found.
3. Toggle **Match case** to enable case-sensitive text matching.
4. Click **Find Next** to start the search.

If any search fails to find the target text, the editor notifies you with a message.

Searching for and Replacing Text

To search for a text item and replace it with new text, select **Edit** > **Replace** or press **Ctrl +R**. Either of these operations opens the **Replace** window:

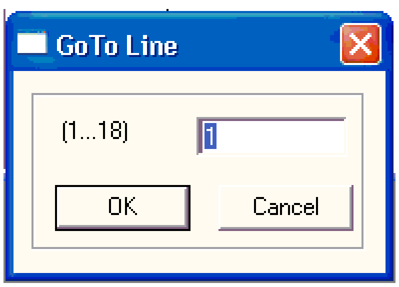


1. Enter the search text in the **Find what** field.
2. Enter the replacement text in the **Replace with** field.

3. Select the **Direction** for the search. **Up** searches on the cursor toward the front of the netlist, wrapping on the front to the back if the text is not found. **Down** searches on the cursor toward the back of the netlist, wrapping from back to front is the text is not found.
4. Toggle **Match case** to enable case-sensitive text matching.
5. To find and replace text item-by-item, click **Find Next** to start the search. When the next instance of the search text is highlighted, click **Replace** to replace the search text with the replacement text, or click **Find Next** again to leave the instance unchanged. Repeat this step until all replacements have been made.
6. To replace all instances of the search text with the replacement text globally, click **Replace All**.
7. When all replacements have been made, click **Cancel** to close the **Replace** window.

Going to a Numbered Line

Errors and warnings on the netlist parser See line numbers in the netlist. Line numbers are not displayed in the Netlist Editor, but the cursor can be moved to a line by specifying its number. To move the cursor to a numbered line, select **Edit > Go to** or press **Ctrl +G**. Either operation opens the **Go To Line** window box.



The parenthesized display shows the range of available line numbers in the netlist that is currently being edited:

- Enter the number of the line and click **OK**.

The cursor moves to the beginning of that line.

Netlist Editor Options

To set option defaults that affect all designs, click **Tools>Options >General Options** to open the **Options** window, then select **Netlist & Script Editor** on the left group box.

The following controls are available:

- **Display Line Numbers**, when checked, displays line numbers in the Netlist and Script Editors.
- Click the **Bookmark** color-display group box to select a new color on the color palette.

- The **Font Name** drop-down menu lists many available font styles from which to choose.

- **Size** allows you to vary the size of the selected **Font Name** style.
- An example of the **Font Name** style is shown in the display window.
- Click **OK** to save your changes and close the window, or click **Cancel** to close without saving changes.

11 - Circuit Time Domain Analyses

Nexxim provides a variety of analysis methods for simulating circuit designs in the time domain. Any of these analyses can include a sweep of circuit parameters.

DC Operating Point Analysis

DC analysis provides the DC operating point voltages and currents. In turn, the DC operating point provides the initial values for DC sweep analysis, harmonic balance, and transient analyses. It also provides the large signal bias operating point for small signal AC analysis, noise analysis, and linear network analysis.

See Nexxim DC Analysis.

Transient Analysis in Nexxim

Transient analysis computes the response of a circuit over time, using a system of differential/algebraic equations derived from information provided by the circuit topology and by the device models. Transient analysis can calculate its initial values by running a DC operating point simulation, or can be set to use initial values supplied in the netlist.

See Nexxim Transient Analysis.

Statistical Eye Diagram Analyses

Statistical eye analysis is commonly used to perform bit error rate (BER) simulations on high-speed communications channels. Statistical eye analysis uses statistical post-processing to calculate the performance of a serial channel given the channel parameters including jitter.

Nexxim Quick Eye analysis uses simplifying assumptions to calculate the BER from a transient analysis of single transitions. Nexxim VerifEye analysis uses a fully statistical approach to calculate the BER.

See VerifEye And Quick Eye Analyses.

AMI Analysis

The IBIS Algorithmic Modeling Interface (AMI) allows time-domain simulation of a linear channel using customer-supplied models for the transmitter and receiver. Reports include eye diagrams and bit-error-rate (BER) contours like those available for QuickEye.

For details on Nexxim AMI Analysis, see AMI Analysis.

HSPICE Transient Analysis

Ansys **Electronics Desktop** Circuit projects offer direct integration with HSPICE® analysis software. This integration does not require the Ansys RSM Service. HSPICE is a trademark of Synopsys, Inc.

See HSPICE in Circuit Designs.

Nexxim DC Analysis

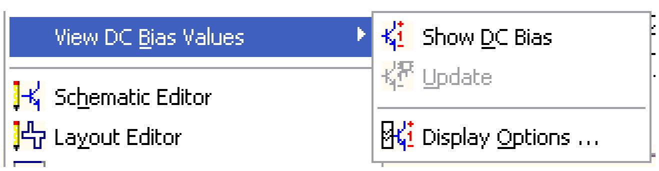
DC (direct current) analysis first initializes the circuit, then solves the circuit equations to derive the DC operating point. The DC operating point consists of the voltages at all nodes and currents through all branches and includes the DC bias voltage applied to semiconductor devices. DC operating point analysis also provides the initial values used as the starting point for DC sweep analysis, harmonic balance, and transient analysis, unless these are set by the user. It also provides the large-signal bias operating point for small-signal AC analysis, noise analysis, and linear network analysis. Successful and accurate calculation of the DC operating point is essential for these simulations.

> **Note:**
>
> In particular, the accuracy of the DC operating point solution strongly affects the accuracy of the small-signal analyses.

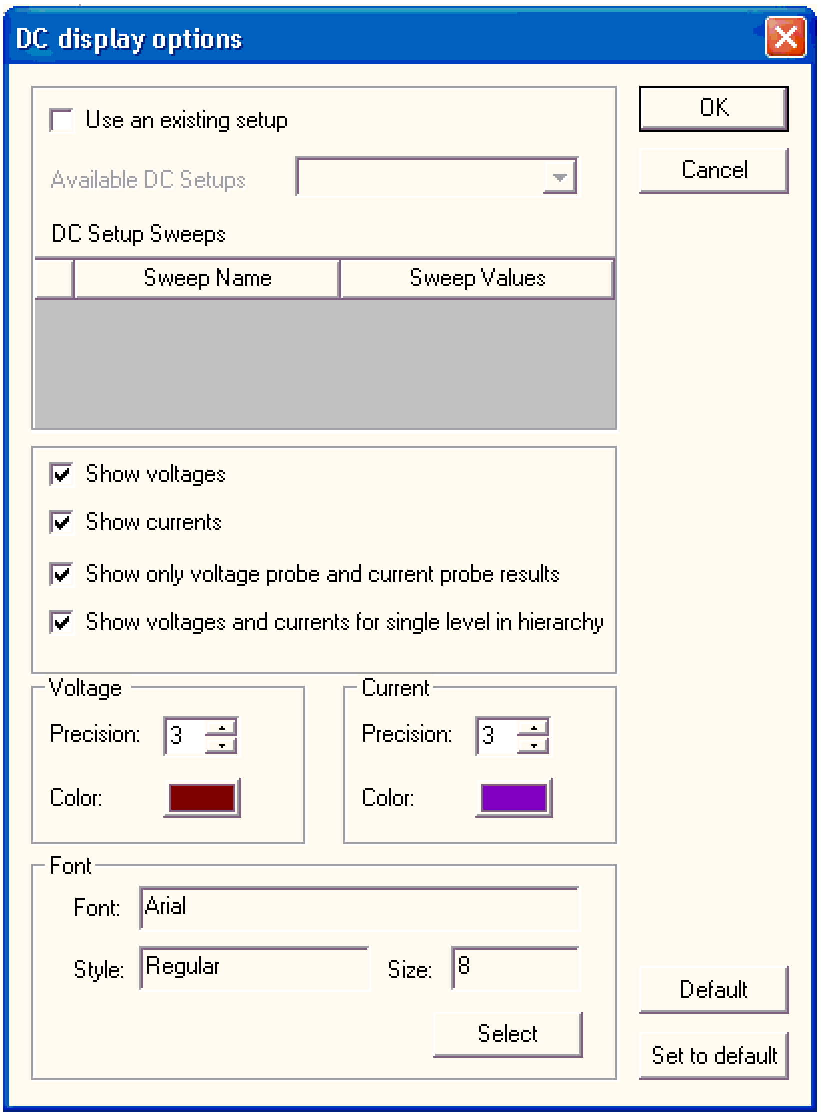
For details on DC sweep simulations, See the separate topic on Variable Sweep.

Viewing DC Bias Voltages and Currents in a Schematic

To run a DC bias analysis on a Circuit design on the **Schematic Editor**, right-click **Circuit** and select **View DC Bias Values** to open a window.



- Select **Show DC Bias** to run the analysis and display the results on the schematic. To remove the DC bias information on the schematic, toggle **Show DC Bias**.
- After the first DC bias analysis has been run, select **Update** to recalculate the bias values after making changes to the circuit.
- Select **Display Options** to open the DC display options window:



- Click **Use an existing setup** to use the setup information from a DC solution setup that you have already created. Select the setup on the **Available DC Setups** drop-down menu.
- Click the **DC Setup Sweeps** group box to select a sweep on the existing setup, and select one of the sweep values for the DC bias analysis. (To run the sweep through all the values, you must perform the DC analysis as described in Running DC Analysis on the **Schematic Editor**.)

By default, the DC bias display shows values only for voltage and current probes. However, if the circuit does not contain any probes, Nexxim advises you to adjust the options. Deselect **Show only voltage probe and current probe results** to enable DC analysis to display values for all nodes.

- Click the **Show voltages** and **Show currents** options to display voltages only, currents only, or both voltages and currents.

By default, the DC bias information for only a single level of hierarchy is displayed. Toggle this option off to show the information for all levels of a hierarchical design.

- Click the **Voltage** and **Current** group boxes to control the precision (number of decimal points) and color used to open these values.
- Click the **Font** group box to control the font used for the display. Click **Select** to open the Font window. Use the sliders to select the appropriate **Font**, **Style**, and **Size**, then click **OK** to return to the **DC display options** window.

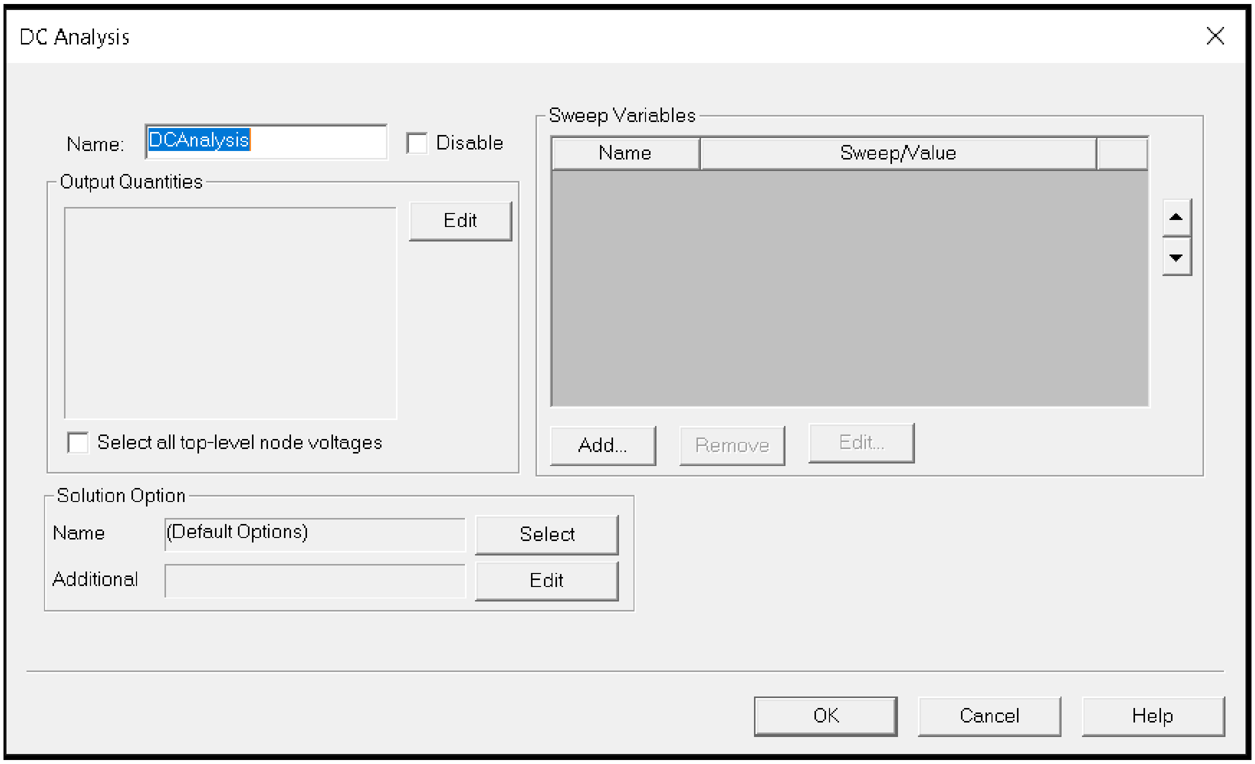
Click **Default** to restore the default option values in the window. Click **Set to default** to make the currently set options the defaults.

Click **OK** to enable the specified options, or click **Cancel** to close the window without changing any options.

Running DC Analysis on the Schematic Editor

From the **Schematic Editor**, perform the following steps to set up and run a DC analysis.

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **DC Analysis** to open the **DC Analysis** window.



2. Type an **Analysis Name** (or accept the default name, for example “DCAnalysis”).
3. To add a sweep, locate the **Sweep Variable** area in the **DC Analysis**window.

- From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
- In the **Variable** list, select Temp or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value, Linear step, Linear count, Decade count**, or **Octave count**.
- Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
- For details on sweeps, see Variable Sweep.
- Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **DC Analysis** window.

4. Optionally, click **Edit Quantities** to open the **Output** window.

- Click the check boxes on the **Output** window to add net voltages and branch currents to the data generated by the DC analysis.
- Click **OK** to add the selected output values and return to the **DC Analysis** window.

5. The **Select all top level node voltages** checkbox is a short cut to automatically save node voltage output quantities for all schematic nets which aren't being saved otherwise.

Caution -For larger designs this could result in large solution data files and an increase in total simulation time.

6. Optionally, use the fields in the **Solution Options** group box to select or add DC analysis options and other Nexxim options to the design.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **DC Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Select the **DC Options** tab. Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window box. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **DC Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

The line can contain multiple options settings. Use spaces to separate the options settings.

To modify or delete an option once it has been added, just edit or delete its entry in the list.

To clear all options, delete the entire line.

Click **OK** to return to the **DC Analysis** window.

The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

- For more information, see DC Analysis Options.

7. Run the simulation:

- From the **Circuit** menu, select **Start Analysis**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
- The **Message Manager** window signals success or failure.
- For details on creating and modifying reports, see Generating Reports and Post-processing.

DC Operating Point Statement in a Netlist

To perform a DC operating point analysis on a Nexxim circuit in the Netlist Editor, the netlist should contain a .DC statement. The syntax is:

.DC

or

.OP

By default, all node voltages and branch currents are output to the results file. For large circuits, significant simulation time can be saved by specifying a set of values to output, rather than the default of calculating variables for every node or current in the circuit.

In the **Schematic Editor**, Click the **Edit Output Quantities** feature in the **Solution Setup** window to specify particular outputs.

The netlist can use one or more .PRINT statements to specify the outputs from DC analysis. If you do not add a .PRINT statement, all node voltages and branch currents are output to the results file. The netlist syntax for printing DC analysis results is:

`.PRINT DC`*`output1`*` [`*`output2`*`] ...`

For more information see Controlling Nexxim Output.

DC Analysis Options

The DC analysis options are listed in the DC Analysis Options Reference. In the **Schematic Editor**, these option values can be changed by selecting **Solution Options** on the **Solution Setup** window, then selecting the **DC Options** tab.

In the netlist editor, set option values by specifying them on an .OPTIONS statement. The syntax for the .OPTIONS statement is:

`.OPTIONS [dc.]`*`option1`*`=`*`value1`*` [[dc.]`*`option2`*`=`*`value2`*` ... ]`

Use the prefix **dc.** to associate an option with DC analysis only. Options with no prefix apply to all analyses that recognize the option. Options that are set in the **Solution Options** window under **DC Solution Setup** in the **Schematic Editor** automatically receive the **dc.** prefix in the generated netlist.

- See Analysis-Specific Options for a listing of options that are used by more than one Nexxim analysis.
- See DC Analysis Technical Notes for an explanation of the DC controls and options.
- See Convergence, Speed, and Accuracy for a discussion of the effect of changing these options.

Table 15: Nexxim DC Analysis Options

Option	Default Value	Description
abstol	1e-9	Absolute current tolerance (Amps). The DC simulator starts with this options at an initial value.
alpha	1e-13	Minimum conductance of the resistor placed in parallel with each branch (mho, equivalent to GMINDC)
autoconverge	1	1 = Enable automatic cyclic progression through convergence methods starting with **conv** setting, upon convergence failure. 0 = Turns off automatic cycling of methods.
beta	1e-12	Minimum conductance of the resistor placed between each node and ground (mho, equivalent to GSHUNT)
conv	1	1 = Beta and device continuation 2 = Device continuation 3 = Pseudotransient 4 = Modified beta/device continuation 5 = Alpha, beta, and device continuation
create_ic_file	0	1=Create initial conditions (.ic) file from DC operating point.
dcic	1	Flag to ignore initial conditions
gmax	1e2	Maximum conductance of the resistor placed in parallel with each branch (mho). Used for HSPICE compatibility in nodesets and initial conditions
limiting	1	1=allow limiting on Newton steps
limiting_ method	Standard Limiting	Standard Limiting = Damped Newton-Raphson. Modified Limiting = Newton-Raphson with enhanced limiting algorithm. Faster convergence rate for moderately nonlinear circuits.
limiting_step	0.3 for nonlinear circuits. Very large (unavailable) for linear circuits	Maximum absolute voltage update allowed per N-R iteration (Volts). The DC simulator starts with this options at an initial value.
max_ continuation_ iterations	500	Maximum number of continuation steps allowed per strategy
max_newton_ iterations	200	Maximum number of Newton-Raphson (N-R) iterations per continuation step. The DC simulator starts with this

Option	Default Value	Description
		options at an initial value.
max_newton_ iterations_limit	None	Maximum post-lambda pseudo-transient continuation
max_pt_ continuation_ iterations (Used only when **conv=3** has been specified.)	5000	Maximum iterations of pseudotransient continuation
min_lambda_ step	1e-4	Minimum lambda step allowed in continuation
noisemodel	external	external = use external data if present, else use internal noise model internal = use internal noise model. External data is ignored none = no noise calculation Noisemodel applies to frequency domain analyses, not to time domain.
output_ic_file_ name	None	Name of the initial condition file
reltol	1e-3	Relative current and voltage tolerance. The DC simulator starts with this options at an initial value.
result_as_ initial_guess	1	Turned off, when set to 0, otherwise the use of the result from each step in a sweep as the initial guess for the next sweep step.
vntol	50e-6	Absolute voltage tolerance (Volts). The DC simulator starts with this options at an initial value.

Nexxim Monte Carlo Analysis

The Nexxim DC and transient analyses can be performed multiple times while using randomly generated values of one or more component parameters. The random-value method is referred to as Monte Carlo analysis. For DC, Monte Carlo analysis can be performed with or without specifying a parameter sweep. For details on the syntax used to specify sweeps, see Variable Sweep.

A Monte Carlo analysis requires three steps:

1. Define a netlist parameter using one of the Monte Carlo distributions.
2. Use the netlist parameter directly or indirectly as the value of an element parameter or model parameter.
3. Add the Monte Carlo analysis to the DC or TRAN solution setup in the netlist.

> **Note:**
>
> Monte Carlo analysis is supported only in netlists. There is no setup dialog in the schematic editor.

Related Topics

Running Monte Carlo Analyses

Viewing a Monte Carlo Analysis in a Report

Monte Carlo Analysis Netlist Syntax

Monte Carlo Analysis Options

Monte Carlo Distributions

Running Monte Carlo Analyses

If you use the Netlist Editor to create a design, the netlist should contain the information for running the Monte Carlo analysis. No Solution Setup is required.

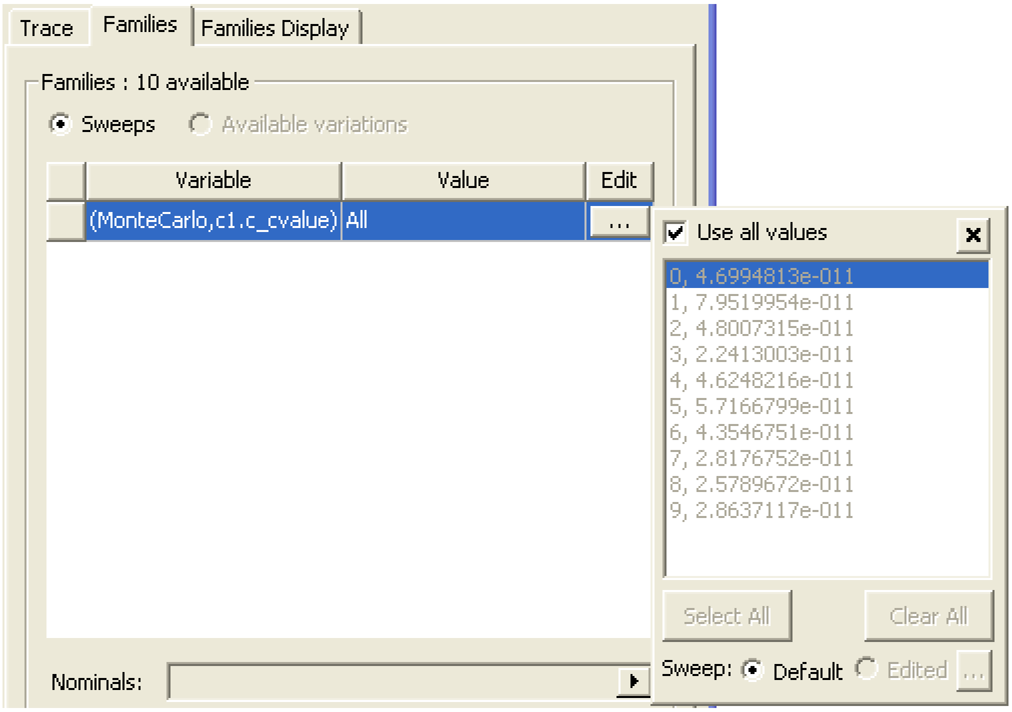
1. Specify the Monte Carlo analysis as part of simulation statement (.DC or .TRAN), using the syntax described in the section Monte Carlo Analysis Netlist Syntax.
2. Run the simulation:
   - On the **Circuit** menu, click **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a green progress bar appears.
   - The **Message Window** signals success or failure.

Viewing a Monte Carlo Analysis in a Report

After running a Monte Carlo Analysis, you can plot any calculated or derived quantity against the sequence of Monte Carlo parameter values.

1. Right-click the Results icon in the Project tree, and select **Create Standard Report >Rectangular Plot** from the menus.
2. The **Reports** dialog opens. The Solution, Domain, and Plotting Values fields have all been filled in with the correct data.

3. Click on the **Families** tab. Select the **Monte Carlo** item to display the sweep values:



4. Select the **Use all values** box.
5. Click **New Report**. The report is displayed in the window.

Monte Carlo Analysis Netlist Syntax

The netlist syntax for Monte Carlo analysis has three parts:

- The netlist parameter definition.

  See Defining a Monte Carlo Netlist Parameter.

- The use of the netlist parameter as the value of a component or model parameter.

  See Setting a Component or Model to Use Monte Carlo Analysis.

- The addition of Monte Carlo analysis to the solution setup statement.

  See Monte Carlo Solution Setups.

Defining a Monte Carlo Netlist Parameter

The netlist uses the .PARAM statement to define a netlist parameter with a Monte Carlo distribution. Five distributions are supported:

- Uniform distribution with absolute bounds
- Uniform distribution with relative bounds
- Gaussian distribution with absolute spread
- Gaussian distribution with relative spread
- Limit or min/max distribution

See Monte Carlo Distributions for explanations of these distributions.

Uniform Distribution with Absolute Bounds

To define a Monte Carlo netlist parameter with uniform distribution and absolute bounds, use the following syntax:

.PARAM *name*=**AUNIF(***nominal_value*,*absolute_bound* [,*multiplier*]**)**

Monte Carlo values will be generated from a uniform distribution in the range:

(*nominal_value-absolute_bound*) to (*nominal_value* + *absolute_bound*)

If a *multiplier* is specified, that many values will be generated, the value that has the highest absolute deviation from the nominal value will be used in the simulation, and the other values will be discarded. The default is to do one selection.

For example, to define a uniform distribution with a nominal value of zero and an absolute range of plus or minus five:

```
.PARAM D1=AUNIF(0, 5)
```

Uniform Distribution with Relative Bounds

To define a Monte Carlo netlist parameter with uniform distribution and relative bounds, use the following syntax:

.PARAM *name*=**UNIF(***nominal_value*,*relative_bound* [,*multiplier*]**)**

Monte Carlo values will be generated from a uniform distribution in the range:

-(*nominal_value*×*relative_bound*) to +(*nominal_value*×*relative_bound*)

If a *multiplier* is specified, that many values will be generated, the value that has the highest absolute deviation from the nominal value will be used in the simulation, and the other values will be discarded. The default is to do one selection.

For example, to define a uniform distribution with a nominal value of 10 and a relative range of plus or minus 5%:

```
.PARAM D1=UNIF(10, 0.05)
```

A relative distribution cannot have a nominal value of zero. To create an actual distribution with relative range and nominal value zero, you could use a secondary variable:

```
.PARAM D1=UNIF(10, 0.05)
.PARAM ACTUAL_D1='D1 - 10'
```

Alternatively, you could use the expression to set the instance or model parameter:

```
R21 Vin 0 R='1000 + (D1 - 10)'
```

Gaussian Distribution with Absolute Spread

To define a Monte Carlo netlist parameter with Gaussian distribution and absolute spread, use the following syntax:

.PARAM *name*=**AGAUSS(***mean*,*abs_spread*, num_*sigma*[,*multiplier*]**)**

Monte Carlo values will be generated from a Gaussian distribution with standard deviation:

σ = *abs_spread/num_sigma*

If a *multiplier* is specified, that many values will be generated, the value that has the highest absolute deviation from the mean will be used in the simulation, and the other values will be discarded. The default is to do one selection.

For example, to define a Gaussian distribution with a mean of zero and a standard deviation of 1 (a “normal” distribution) using an absolute spread:

```
.PARAM D1=AGAUSS(0, 3, 3) // Normal distribution
```

Gaussian Distribution with Relative Spread

To define a Monte Carlo netlist parameter with Gaussian distribution and relative spread, use the following syntax:

.PARAM *name*=**GAUSS(***mean*,*rel_spread*, num_*sigma*[,*multiplier*]**)**

Monte Carlo values will be generated from a Gaussian distribution with standard deviation:

σ = (*mean* × *rel_spread*)/*num_sigma*

If a *multiplier* is specified, that many values will be generated, the value that has the highest absolute deviation from the mean will be used in the simulation, and the other values will be discarded. The default is to do one selection.

For example, to define a Gaussian distribution with a mean of 10 and a standard deviation of 50% of the mean, using a relative spread:

```
.PARAM D1=GAUSS(10, 0.5, 1)
```

A relative distribution cannot have a mean of zero. To create an actual distribution with relative range and zero mean, you could use a secondary variable:

```
.PARAM D1=GAUSS(10, 0.5, 1)
.PARAM ACTUAL_D1='D1 - 10'
```

Alternatively, you could use the expression to set the instance or model parameter:

```
R21 Vin 0 R='1000 + (D1 - 10)'
```

Limit or Min/Max Distribution

To define a Monte Carlo netlist parameter with a limit or min/max distribution, use the following syntax:

.PARAM *name*=**LIMIT(***nominal_value*,*absolute_offset*]**)**

Either the minimum value (*nominal_value-absolute_offset*) or the maximum value (*nominal_ value* + *absolute_offset*) will be selected, with equal probability.

For example, to create a sample set consisting of the two values 5 and 15, you could use:

```
.PARAM D1=LIMIT(10, 5)
```

Setting a Component or Model to Use Monte Carlo Analysis

When the netlist specifies the name of a Monte Carlo netlist parameter as the value of an instance or model parameter, Nexxim will generate a random variable for it, and will recalculate the random variable when the instance or model parameter is referenced.

For both the instance and the model usage, the reference to the Monte Carlo distribution can be individual or grouped.

Instance Parameter with Individual Reference

In an individual reference, the instance parameter references the Monte Carlo netlist parameter directly.

In the following example, the two resistors will be simulated using separate random distributions for their resistance values:

```
.PARAM indiv=AUNIF(0, 0.05)
...
Rtest1 Vin 0 R='0.1 + indiv'
Rtest2 Vin 0 R='0.1 + indiv'
```

Instance Parameter with Group Reference

In a group reference, the instance parameter references a netlist parameter that in turn references the Monte Carlo netlist parameter.

In the following example, both resistors will be simulated using the same set of random values for resistance.

```
.PARAM indiv=AUNIF(0, 0.05)
.PARAM group='10*indiv'
...
```

```
Rtest1 Vin 0 R='0.1 + group'
Rtest2 Vin 0 R='0.1 + group'
```

Model Parameter with Individual Reference

In an individual reference, the model parameter references the Monte Carlo netlist parameter directly.

In the following example, the two MOSFET instances will be simulated using the random distribution for model parameter TOX specified for the model.

```
.PARAM indiv_model=AGAUSS(0, 5, 3)
...
Mtest1 Vcc Vin 0 MonteModel
Mtest2 Vcc Vin 0 MonteModel
.MODEL MonteModel NMOS LEVEL=1 TOX='1.0e-7+indiv_model*1e-8'
```

Model Parameter with Group Reference

In a group reference, the model parameter references netlist parameter that in turn references the Monte Carlo netlist parameter.

In this example, all instances of either MonteModel1 or MonteModel2 will use the same values of model parameter TOX.

```
.PARAM indiv_model=AGAUSS(0, 5, 3)
.PARAM group_model='indiv_model*1e-8'
...
Mtest1 Vcc Vin 0 MonteModel1
Mtest2 Vcc Vin 0 MonteModel2
.MODEL MonteModel1 NMOS LEVEL=1 TOX='1.0e-7 + group_model'
.MODEL MonteModel2 NMOS LEVEL=1 TOX='1.0e-7 + group_model'
```

Combining Instance and Model Distributions

The netlist can specify one distribution on an instance parameter, and a different distribution on a model parameter. The two distributions must be referenced by different parameters. Nexxim then creates two distributions, one for the model and another for the element. For example, a MOSFET instance could use a Monte Carlo distribution for channel length variations (that might differ from instance to instance of MOSFETs on a wafer) and the model could use a different distribution for oxide thickness (that could differ from wafer to wafer).

Monte Carlo Solution Setups

To run a Monte Carlo Analysis with a simple DC analysis, use the following syntax:

.DC SWEEP MONTE=*mval* [**FIRSTRUN=***firstval*]

To run a Monte Carlo analysis with a DC sweep, use the following syntax:

```
.DC variable sweep_specSWEEP MONTE=mval [FIRSTRUN=firstval]
```

To run a Monte Carlo analysis with Transient analysis, use the following syntax:

```
.TRANstepstoptimeMONTE=mval [FIRSTRUN=firstval]
```

In the Monte Carlo setup, the value (*mval*) specified for the **MONTE** keyword sets the number of Monte Carlo trials to be run. The random value generator can be set to start at a definite seed value by setting the **SEED** option (See Monte Carlo Analysis Options).

Here are some examples. Suppose the Monte Carlo parameters have been set up with the following statements (from a previous example):

```
.PARAM indiv=AUNIF(0, 0.05)
...
Rtest1 Vin 0 R='0.1 + indiv'
Rtest2 Vin 0 R='0.5 + 10*indiv'
```

Now, to run a DC analysis using ten random resistance values for each resistor:

```
.DC SWEEP MONTE=10
```

To run a set of Monte Carlo analyses during each iteration of a DC sweep:

```
.DC Vin V LIN 0.1 0.5 5 SWEEP MONTE=10
```

The previous example would execute five iterations of the voltage sweep, with ten Monte Carlo variations in each iteration, for a total of 50 analyses.

To run a transient analysis with the same Monte Carlo parameters:

```
.TRAN 1e-3 10e-2 SWEEP MONTE=10
```

The **FIRSTRUN** parameter runs the Monte Carlo simulation starting with random value number *firstval*, and continuing through random value number (*mval* minus 1). Random values 1 through (*firstrun*-1) are discarded. The **FIRSTRUN** parameter can be used with the **SEED** option to partition a Monte Carlo analysis over separate runs. Any Monte Carlo analysis that starts with a given **SEED** setting runs through the same sequence of random values. So, for example, you could run ten trials with the following netlist:

```
.OPTION SEED=5
...
.TRAN 0.1 10 SWEEP MONTE=10
```

And later, modify the netlist as follows to do the next ten trials:

```
.OPTION SEED=5
...
.TRAN 0.1 10 SWEEP MONTE=21 FIRSTVAL=11
```

The **FIRSTVAL** entry causes Nexxim to run the Monte Carlo analysis, discarding the first ten random values, then execute transient simulations using Monte Carlo values 11 through 20.

If a solution setup does not specify a Monte Carlo distribution, the nominal or mean value will be used as the value of any Monte Carlo netlist parameter that is referenced by an instance or model parameter, when that solution is run.

You can specify a Monte Carlo distribution for one or more analyses, but use the nominal values of Monte Carlo netlist parameters for other analyses set up in the netlist. For example, using the netlist from the previous examples:

```
.PARAM indiv=AUNIF(0, 0.05)
...
Rtest1 Vin 0 R='0.1 + indiv'
Rtest2 Vin 0 R='0.5 + 10*indiv'
.DC
.TRAN 1e-3 10e-2 SWEEP MONTE=10
```

The DC analysis uses the nominal value (zero) for the Monte Carlo parameter indiv. The transient analysis sweeps the Monte Carlo parameter over the specified distribution.

Monte Carlo Analysis Options

Nexxim Monte Carlo analysis recognizes one option, the **SEED** option, set with the following syntax:

```
.OPTION SEED=seed_value
```

The *seed_value* is used by the random-value generator to initialize the sequence of pseudo-random values. All Monte Carlo analyses that use a given *seed_value* generate the same sequence of parameter values. The *seed_value* should be a positive integer.

In the absence of a **SEED** option setting, Nexxim Monte Carlo analysis calculates a seed internally.

Monte Carlo Distributions

Nexxim Monte Carlo analysis supports three distributions from which random values are selected—uniform, Gaussian, and limit (min/max).

Uniform Distribution

A uniform distribution generates values within a range centered on a nominal value. All values within the range have equal likelihood of selection.

The range for a uniform distribution can be specified using absolute bounds or relative bounds.

- With an **absolute bound**, the range is the nominal value plus or minus the absolute bound. For example, if the nominal value is zero and the absolute bound is 2.5, the range of values is -2.5 to +2.5.
- With a **relative bound**, the range is the nominal value plus or minus the relative bound times the nominal value. For example, if the nominal value is 10 and the relative bound is 0.1, the range of values is 9 to 11. Note that a relative range cannot have a nominal value of zero.

Selection from a uniform distribution can be set up to do multiple selections and simulate with the value that has the highest absolute deviation from the nominal value. For example, suppose the nominal value is zero and the range is from -2.5 to +2.5. With a multiplier of 5, the Monte Carlo analyzer might select the five values -1.1, 2.2, 0.7, -2.4, and -0.6. The most extreme value, -2.4, would be used in the simulation and the other selections would be discarded. The resulting output distribution is somewhat bimodal. If no multiplier is specified, the default is one selection.

Gaussian Distribution

A Gaussian distribution generates values from plus infinity to minus infinity ($-\infty$ to $+\infty$) centered on the mean or nominal value. A value within the distribution has a probability of selection determined by the standard deviation.

The standard deviation for a Gaussian distribution is specified using an absolute spread or a relative spread that represents a specified number of standard deviations.

- With an **absolute spread**, the standard deviation is the absolute spread divided by the number of standard deviations it represents. For example, if the absolute spread is 2 and it represents 3 standard deviations, the standard deviation is 2/3 or 0.6666.
- With a **relative spread**, the standard deviation is the relative spread times the mean, divided by the number of standard deviations represented by the product. For example, if the mean is 10, the relative spread is 0.2 and the product represents 3 standard deviations, the standard deviation is $(10\times0.2)/3$ or 0.6666. Note that a relative spread cannot have a mean of zero.

Selection from a Gaussian distribution can be set up to do multiple selections and simulate with the value that has the highest absolute deviation from the nominal value. For example, suppose the mean value is zero, the standard deviation is 1.0, and the range is from the range is -1.5 to +1.5. With a multiplier of 5, the Monte Carlo analyzer might select the five values -0.1, 1.2, 0.7, -0.4, and 0.6. The most extreme value, 1.2, would be used in the simulation and the other selections would be discarded. The resulting output values would tend to be somewhat bimodal. If no multiplier is specified, the default is one selection.

Limit or Min/Max Distribution

A limit or min/max distribution generates two values offset an absolute distance from the nominal value. The two values have equal likelihood of selection. The two values are:

- The nominal value plus the absolute offset.
- The nominal value minus the absolute offset.

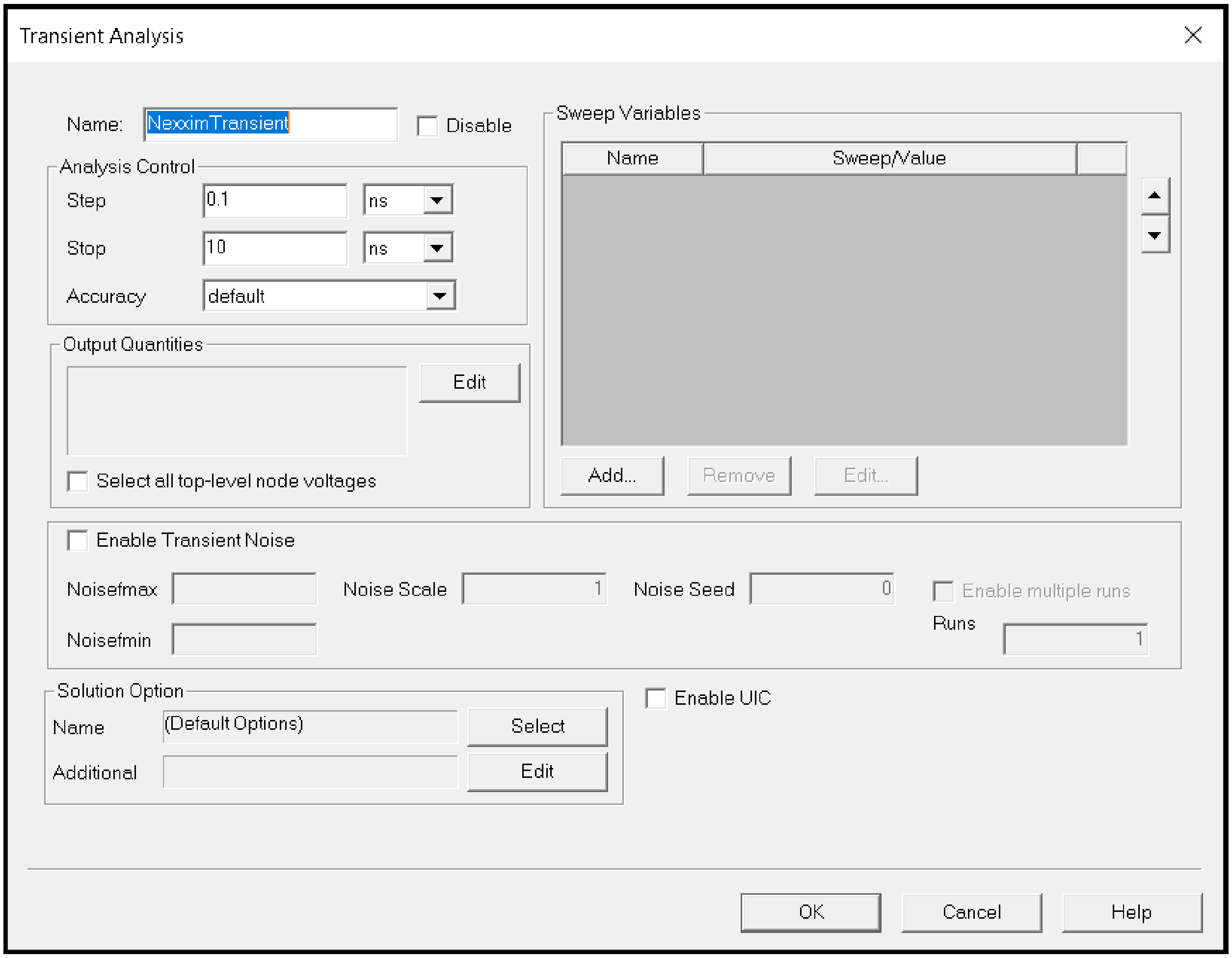
Circuit Transient Analysis

Transient analysis computes the time-domain response of a circuit by numerically integrating a system of differential/algebraic equations. The equations are derived on the circuit topology and from information provided by the circuit device models. Transient analysis breaks continuous time into discrete timesteps and uses numerical integration methods (such as the trapezoidal rule) and Newton-Raphson iterations to solve the circuit equations at each timestep.

Transient Analysis Setup from the Schematic Editor

From the **Schematic Editor**, perform the following steps to set up and run a Nexxim transient analysis:

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Transient Analysis** to open the **Transient Analysis** window.



2. Type an **Analysis Name** (or accept the default name, "NexximTransient").
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
4. Under **Analysis Control**, specify the **Step** time, the **Stop** time, and the **Accuracy** level using the drop-down menu.
5. To add a sweep, locate the **Sweep Variable** area in the **Transient Analysis** window.
   - From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
   - In the **Variable** list, select **Temp** or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value, Linear step, Linear count, Decade count**, or **Octave count**.

- Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
- For details on sweeps, see Variable Sweep.
- Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Transient Analysis** window.

6. Optionally, click **Edit** in the **Output Quantities** group box to open the **Output** window.

- Click the check boxes on the **Output** window to add net voltages and branch currents to the data generated by the transient analysis.
- Click **OK** to add the selected output values and return to the **Transient Analysis** window.

7. The **Select all top level node voltages** checkbox is a short cut to automatically save node voltage output quantities for all schematic nets which aren't being saved otherwise.

   Caution -For larger designs this could result in large solution data files and an increase in total simulation time.

8. Optionally, click the **Enable Transient Noise** group box to activate the fields in the transient noise group box. Refer to the *Transient Noise* topic in the "Transient Analysis Netlist Syntax " on page 11-25 for details on these fields.
9. Optionally, use the fields in the **Solution Options** group box to select or add Transient or DC analysis options and other Nexxim options to the design.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.
- Check the **Enable UIC** checkbox to enable UIC.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **Transient Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Select the **DC Options** or **Transient Options** tab. Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **Transient Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

The line can contain multiple options settings. Use spaces to separate the options settings.

To modify or delete an option once it has been added, just edit or delete its entry in the list.

To clear all options, delete the entire line.

Click **OK** to return to the **Transient Analysis** window.

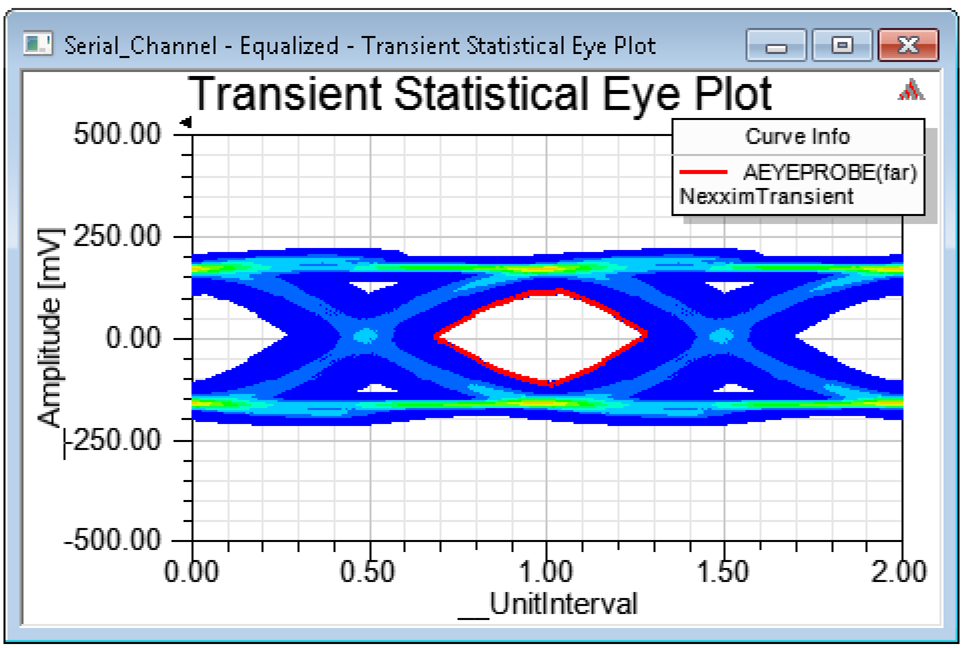
The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

- For more information, see Transient Analysis Options.

10. Run the simulation:
    - From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
    - The **Message Manager** window signals success or failure.

Generating a Transient Sampled Eye Report

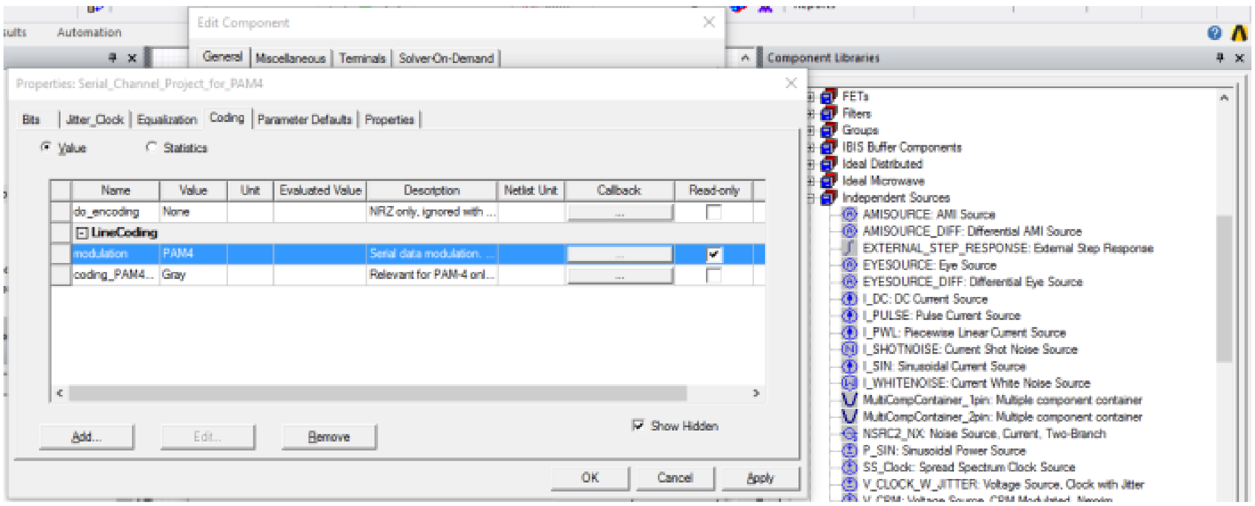
A sampled or statistical eye diagram samples the transient data to generate the eye. To generate a sampled or statistical eye report after transient analysis has run to completion, click the Results icon and select **Create Statistical Eye Report** > **Statistical Eye Plot**. Select your Transient analysis as the Solution, **UI** as the Domain, **Eye** as the Category, your Eye Probe as the Quantity, and **<none>** as the Function. Click **New Report** to generate the plot. Here is an example:



PAM-4 Transient Analysis

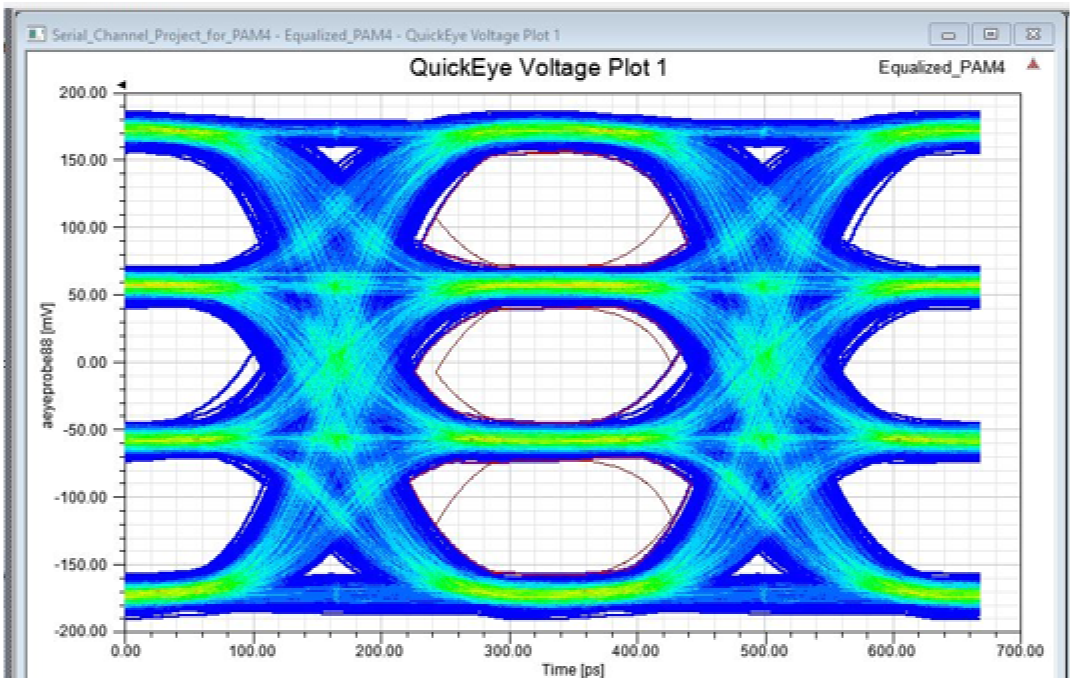
PAM-4 signaling is advantageous because it doubles the number of bits in serial data transmissions. PAM-4 allows you to double the bit rate in the channel without doubling the required bandwidth. This is accomplished by increasing the number of levels of pulse-amplitude modulation, but does so at the cost of noise susceptibility.

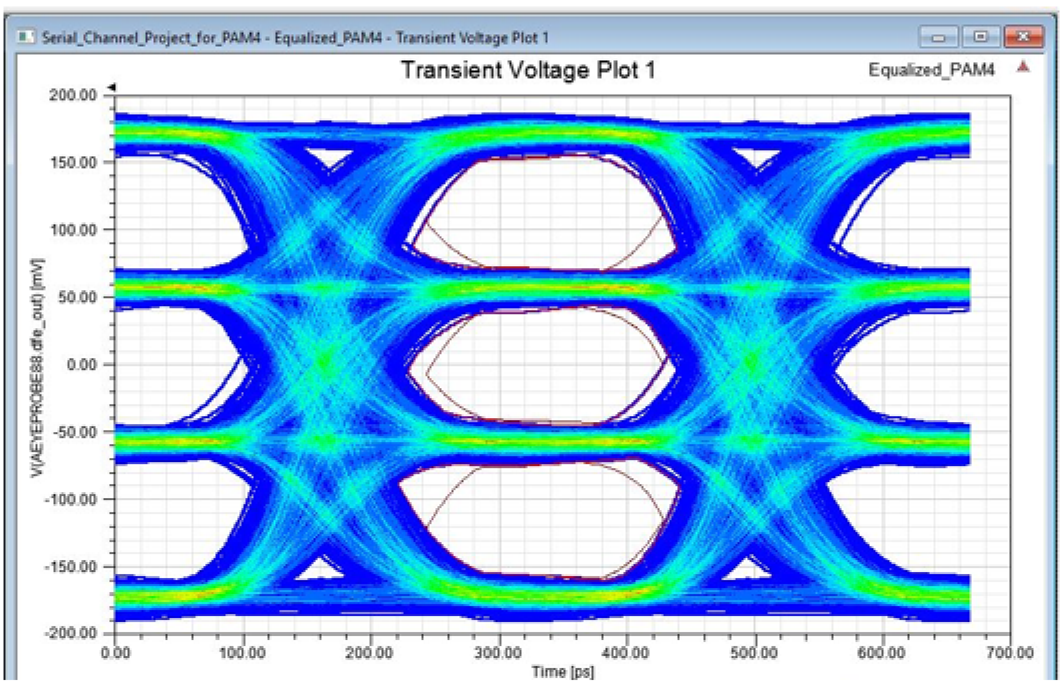
To set up a PAM-4 Transient Analysis, select and edit the Eye Source listed in the Components Window. Then, in the Coding tab of the **Properties** window, click the modulation Value and select **modulation PAM-4**.

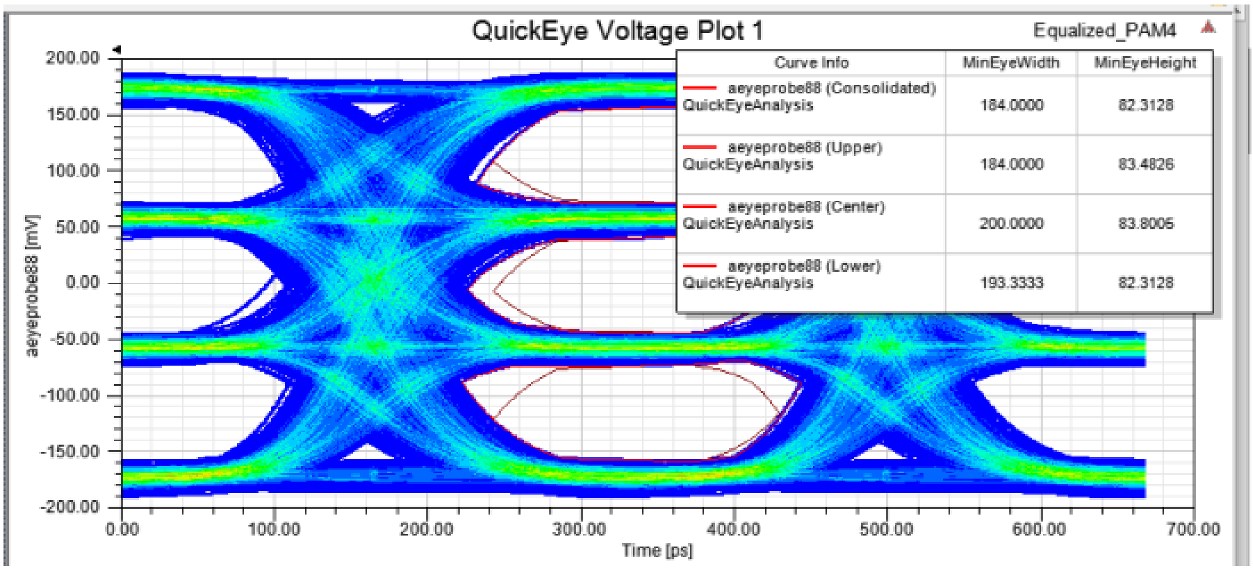


Voltage Plots of PAM-4 Results

When you generate a Rectangular Eye Diagram from PAM-4 data, the waveform sub-view can be zoomed in to show the boundaries of the unit intervals (black lines) and the inner-eye contour overlay. Following are some examples of PAM-4 Plot Results.


Serial_Channel_Project_for_PAM4 - Equalized_PAM4 - QuickEye Voltage Plot 1
QuickEye Voltage Plot 1
Equalized_PAM4
aeyeprobe88 [mV]
200.00
150.00
100.00
50.00
0.00
-50.00
-100.00
-150.00
-200.00
0.00
100.00
200.00
300.00
400.00
500.00
600.00
700.00
Time [ps]


Serial_Channel_Project_for_PAM4 - Equalized_PAM4 - Transient Voltage Plot 1
Transient Voltage Plot 1
Equalized_PAM4
V(AEYEPROBE88.dfe_out) [mV]
200.00
150.00
100.00
50.00
0.00
-50.00
-100.00
-150.00
-200.00
0.00
100.00
200.00
300.00
400.00
500.00
600.00
700.00
Time [ps]



For more information, see 2D and Circuit PAM-4 Eye Plot.

Running Transient Analysis on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist contains the information for running the linear network analysis, as described later in this topic. No Solution Setup is required.

Run Circuit Transient Analysis

To run the transient analysis setup:

1. From the **Circuit** menu, select **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
2. The **Message Manager** window signals success or failure.

For details on creating and modifying reports, see Generating Reports and Post-processing.

Transient Analysis Netlist Syntax

The transient analysis setup in a netlist has a basic form plus optional arguments for control of accuracy, access to file data, and calculation of noise.

TRAN Statement

To execute transient analysis simulation on a circuit, add a .TRAN statement to the netlist. The syntax for the basic .TRAN statement is:

```
.TRANstep stoptime
```

All transient analyses start at time 0. The *step* argument is a suggested initial step size and also influences the maximum step size that is used. The *stoptime* argument sets the end of the transient solution interval. For example, to execute a transient analysis on an interval from time zero to one microsecond with a suggested initial step size of one nanosecond:

```
.TRAN 1e-9 1e-6
```

Transient Analysis Accuracy Syntax

Two optional arguments on the transient analysis statement provide broad control over the trade-off between speed and accuracy:

```
.TRANstepstoptime
+ ERRPRESET=relaxed|moderate|strict
 + RELREF=pointlocal|allocal|sigglobal|allglobal
```

The **ERRPRESET** argument overrides the settings for several options that control the trade-off between speed and accuracy.

Errpreset Controls

Option	errpreset			
	None	**relaxed**	**moderate**	**strict**
reltol	x1	x10	x1	/10
trtol	7	7	7, 8, and 9	10
relref	alllocal	sigglobal	sigglobal	alllocal
method	trap	trap	trap	ndf2
update_jacobian_ period	3	3	3	1

Errpreset can be set to **relaxed**, **moderate**, **strict**, **pspice**, or **pspice2**:

- **Relaxed** lowers accuracy to speed simulation. Set the following values and they will not be overridden by errpreset.
  - **Local truncation error factor (trtol)** = **< 7**
- **Moderate** balances accuracy and speed. To avoid long simulation time caused by high accuracy settings, or low accurate results due to fast settings, the following values will be automatically set.
  - Reset **Error reference (relref)**= **sigglobal** if the customized value is **alllocal** or **pointlocal**
  - Reset **Integration method** = **trapezoidal** if the customized value is **ndf2**

  - Reset **Local truncation error factor (trtol)**= **7** if the customized value is **< 7**
  - Reset **Local truncation error factor (trtol)**= **9** if the customized value is **> 9**
- **Strict** provides results of high accuracy but slows simulation. Set the following values and they will not be overridden by errpreset.
  - **Error reference (relref)**= **pointlocal**
  - **Local truncation error factor (trtol)** = **>10**
  - **Update Jacobian period** = **1**
- **Pspice** is only available if the circuit contains pspice components. Pspice will be overridden by **relaxed**, **moderate**, or **strict** settings if **Accuracy** is customized. The following values are automatically set to improve the chance of convergence:
  - Reset **Integration method** = **ndf2** if the customized value is **trapezoidal**
  - Reset **Update Jacobian period** = **1** if the customized value is **>1**
  - Reset **Maximum Newton-Raphson iterations** = **30** if the customized value is **<30**
- **Pspice2** must be enabled manually via the **Solution Option** group box, by entering **tran.errpreset=pspice2**. **Pspice2** takes precedence over all aforementioned errpreset settings and automatically sets the following value.
  - Set **Integration method** = **ndf2**
  - Increase **Max time step size** to **stop time**
  - Set **Update Jacobian period** = **3**
  - Set **Relative error tolerance for current (reltol)** = **0.0025**

The **Relref** argument controls how the norms are computed for the delta check, function check, and LTE checks. *Point-local* means the norm is local to the signal and the timestep. *Local* means the norm is local to the signal but considers the maximums over all time-steps. *Global* means the norm is global for the entire circuit over all time-steps. The settings for **relref** are:

- **pointlocal** = Point-local for delta, function, and LTE
- **alllocal** = Local for delta, function, and LTE
- **sigglobal** = Local for function, global for delta & LTE
- **allglobal** = Global for delta, function, and LTE

Transient Analysis File Arguments

Optional arguments on the transient analysis statement enable Nexxim to read initial state information from an external data file, and write initial or final state information to an external data file.

```
.TRANstepstoptime
+ READ_STATE=file_reference
+ WRITE_FINAL_STATE=file_reference
+ WRITE_INITIAL_STATE=file_reference
```

- **READ_STATE** reads the initial values of node voltages on the specified file.
- **WRITE_FINAL_STATE** writes the final values of node voltages to the specified file.
- **WRITE_INITIAL_STATE** writes the initial values of node voltages to the specified file.

See File References for details on references to external files.

Transient Analysis Noise Syntax

To enable transient noise analysis, the .TRAN statement takes additional optional arguments.

```
.TRANstepstoptime
+ NOISEFMAX=val
+ NOISESCALE=val
+ NOISESEED=val
+ NOISEFMIN=val
```

Table 2: Transient Noise Arguments

Argument	Description	Unit	Default
NOISEFMAX	The highest frequency to analyze for pseudorandom noise. Must be non-zero to enable transient noise analysis. Maximum transient timestep is set to 1/NOISEFMAX.	Hz	0
NOISESCALE	Scale factor	None	1
NOISESEED	Random number seed. Using the same seed generates the same set of random values	None	None
NOISEFMIN	Lowest frequency to analyze for noise. Under NOISEFMIN, noise power density is constant. 1/NOISEFMIN must be less than the transient analysis stoptime.	Hz	NOISEFMAX
NOISESTART	Noise start time	Second	0

Netlist Syntax for Sources and Probes

The complete netlist syntax for the Eye Source is provided in the Eye Source component topic.

The complete netlist syntax for the Eye Probe is provided in the Eye Probe component topic.

The complete netlist syntax for the Eye Scope is provided in the Eye Scope component topic.

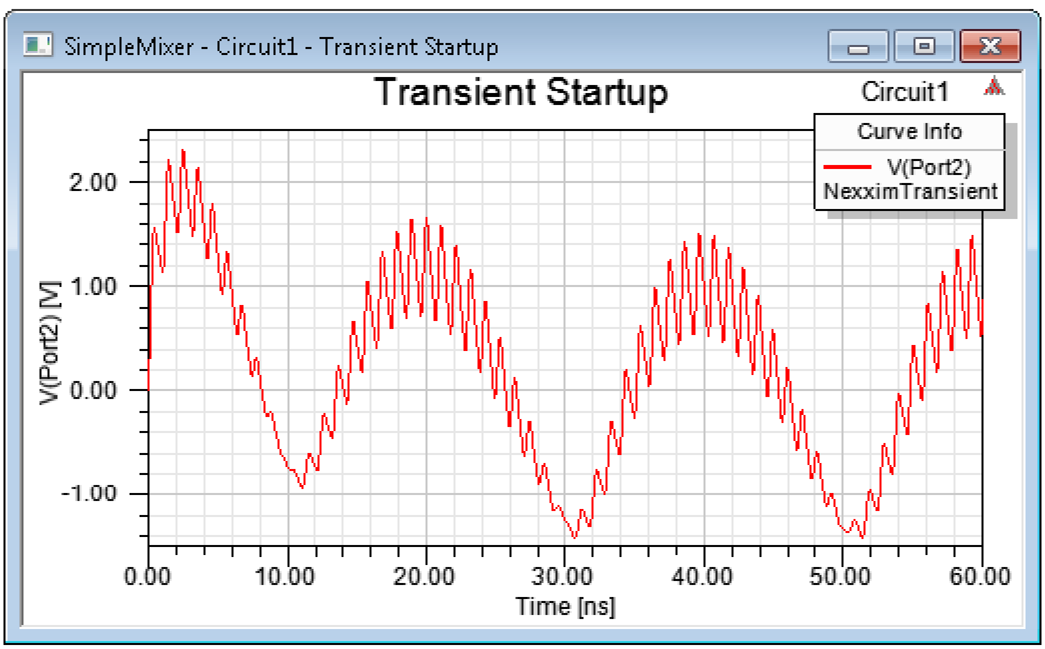
The complete netlist syntax for the MATLAB Probe is provided in the MATLAB Probe component topic.

Transient Analysis Outputs

The graphical plots and reports described in this topic help you to visualize the relationship between design parameters and transient analysis results.

1. To view the Transient results in the Time domain, click on the **Results** icon and select **Create Standard Report>Rectangular Plot**.
2. Selections in the report window are context-sensitive. Whatever Eye Probe/Scope settings are chosen determines the relevant output quantities from which you may select.

   Select Transient analysis as the Solution, **Time** as the Domain, **Voltage** as the Category, a port or net voltage as the Quantity, and **<none>** as the Function.
3. Click **New Report** to generate the plot.



For details on creating and modifying reports, see Generating Reports and Post-processing.

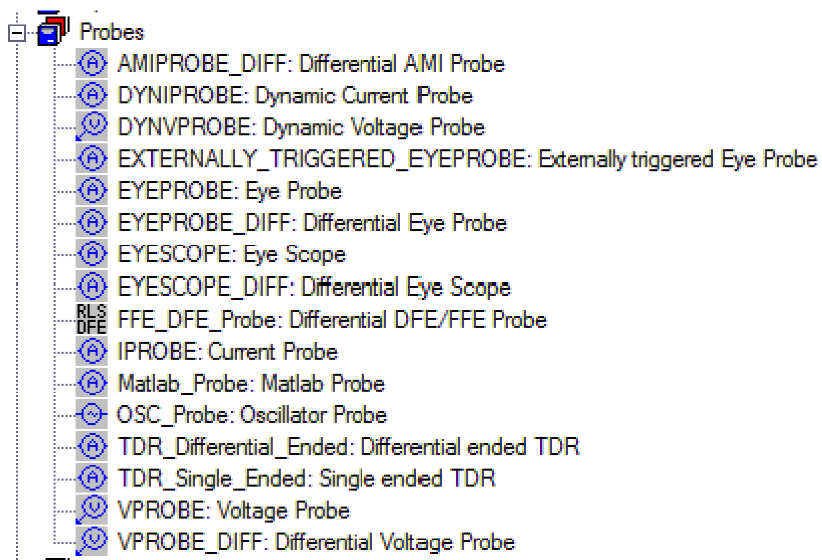
Outputs for Circuits with Eye Source and Eye Probe

The Eye Source and Eye Probe may be used with Transient analysis. In this way a single source and probe set can drive transient, QuickEye, and VerifEye analyses.

The Eye Source and Eye Probe setups for transient analysis are the same as for Quick Eye analysis. The outputs are also similar. See Display Quick Eye Analysis Outputs for statistical eye and bathtub plots for transient analysis.

Adding a MATLAB Probe

A MATLAB probe allows the user to pass transient analysis data to a MATLAB script (MATLAB™ is a trademark of The Mathworks, Inc.). In a schematic, select the MATLAB Probe on the list of Probes:



To set up the MATLAB probe, right-click and select **Edit Properties**:

	Name	Value	Unit	Ev...	Description
	scriptfile				Filename of matlab script file to be run
	bitsource				Bit source
	step_size	0		0	Value of step size to sample time with (double)
	bitwidth	0		0	Bit width (double)
	type	0		0	0 or 1 (0 for generic t.v and 1 for fibre channel)
	CosimDefinition	Edit			
	COMPONENT	MATLAB_...			
	param1				
	param2				
	param3				
	param4				
	param5				
	param6				
	param7				
	param8				
	param9				

- Set **scriptfile** to the name of the file containing the MATLAB script.
- Set **bitsource** to the netlist name of the source of the bit pattern for transient to use.
- Set **bitwidth** to the bit width or unit interval.
- Set **type** to 0 for generic time/voltage data format, 1 for FibreChannel data format.
- For FibreChannel data only, set **step_size** to a non-zero value (fixed step size to interpolate the transient data).
- Enter values for up to ten parameters. Each parameter can be a single numeric or string value, or a vector of numeric or string data.

Transient Parallel Bus Speedup

Highly non-linear circuits do not simulate well with QuickEye or VerifEye, so transient analysis must be used. To speed up transient solutions for large non-linear designs that require simulating millions of bits, Nexxim offers the Transient Parallel Bus Speedup (TPBS) option.

To enable the TPBS feature globally for all analyses, Click the **HPC and Analysis Options** window.

- From **Electronics Desktop**, select **Tools>Options>HPC and Analysis Options**.
- From the **HPC and Analysis Options** window, select the **Options** tab.
- Select **Circuit Design** on the **Design type** menu.

- Click **Enable HPC for transient parallel bus speedup**. Use the menu to select **True**.
- Click **OK** to close the **HPC and Analysis Options** window.

To enable TPBS for just a single transient analysis:

- Open the **Transient Analysis** setup window.
- In the **Solution Options** group box, click **Edit** on the **Additional** field.
- Enter tran.tpbs=1 num_threads=*val*, where *val* is the number of threads to be used.
- Click **OK** to close the Transient Analysis setup window.

Parallel bus speedup requires that multiple processors are available for the analysis—10 CPU cores minimum, 16 recommended. Parallel speedup also requires at least 20GB of free memory, 30GB recommended. Due to the overhead required, transient parallel speedup should be used only for bit-pattern problems requiring from a few hundred to millions of bits to determine. TPBS supports RGB, PRBS, Eye, and LFSR bit-pattern sources. A bare minimum of 25 bits per thread is required to switch to TPBS; the minimum increases as the size of the circuit increases.

Transient parallel speedup operates by dividing the simulation length into equal-length overlapping windows; simulation of windows is performed in parallel. After calculating the response within each window, the overlaps are removed using an iterative error-minimizing algorithm.

For the right kind of problem, parallel speedup can achieve a significant speedup of the transient simulation. Parallel speedup is less efficient for small examples and for large ones with relatively small simulation time. Networks with high time constants reduce TPBS efficiency; examples of such networks are transmission lines with large propagation delays and RC networks.

> **Note:** Enabling TPBS turns on passivity enforcement. Click the **Additional** options field in the **Transient Analysis** setup window to set **auto_enforce_passivity=0** if required to simulate.

Circuit Transient Analysis Options

The transient analysis options and their default values are listed in the Transient Analysis Options Reference. See also Transient Analysis Technical Notes for an explanation of the controls and options.

The values can be changed with a .OPTIONS statement. The syntax is:

```
.OPTIONS tran.option1=value1 [tran.option2=value2 ... ]
```

The **tran.** prefix causes the option to apply only to transient analysis. Without the prefix, an option in the netlist applies to all analyses that recognize it.

See Analysis-Specific Options for a listing of options that are used by more than one Nexxim analysis.

You can also specify options for DC analysis to control the DC calculation done to obtain the initial conditions for transient analysis. See DC Analysis Options.

You can also specify the statistical eye resolution for circuits with Eye Source and Eye Probe. See Resolutions of Eye Diagram for further information.

Some options (in particular, **abstol** and **reltol**) can affect both DC analysis and transient analysis. To specify different settings for the DC initialization and the subsequent transient analysis, use the prefix **dc.** for the DC settings and the prefix **tran.** for the transient analysis settings. In general, option settings with the prefix **dc.** affect only the DC initialization, and settings with the prefix **tran.** affect only the transient analysis. An option setting with no prefix affects both the DC and transient analyses. However, **dc.abstol** or **dc.reltol** is applied to transient analysis when **tran.abstol** or **tran.reltol** is not given explicitly.

Here is an example:

```
.option dc.conv=3 $ Applied to DC initialization only
.OPTION dc.abstol=4e-9 $ Applied to DC initialization only
.OPTION tran.abstol=1e-9 $ Applied to transient only
.OPTION reltol=2e-4 $ Applies to both DC and transient
.OPTION tran.UI-bins=400 tran.ampl-bins=400 $ for eye resolution
(rest of netlist)
.TRAN 1e-3 1
```

Transient Analysis Error Messages

When Nexxim transient analysis is unable to complete its analysis successfully, it displays an error message. This topic provides an analysis of some of the error messages.

Unable to satisfy convergence tolerances at t=*time*

Nexxim was unable to achieve the accuracy specified by the RELTOL, ABSTOL, and VNTOL options at time *time*. During transient analysis, a variant of Newton's method is used to attempt to solve the nonlinear algebraic problem generated at each time point. Convergence is achieved only when both a delta check and a residue check are satisfied. (The delta check and residue check are explained in the Nexxim DC Analysis topic; see Nexxim DC Analysis Convergence Criteria.) Convergence problems may indicate that some model parameters specified in the netlist are inconsistent or are not correctly interpreted.

- Verify the values of device parameters that are set by device instance statements or MODEL statements in the netlist.

Unable to satisfy local error tolerances on voltage at time *time*, node *nodename*

Although Nexxim was able to achieve convergence at time *time*, the resulting solution did not satisfy the local error tolerance. This usually indicates that there is a discontinuity or instability in the solution that the transient analysis engine could not step over.

- The best remedy is to change the circuit so as to eliminate the problem behavior.
- A second possible remedy is to adjust the local error tolerance itself. The local error tolerance is looser than the convergence tolerances specified by RELTOL, ABSTOL and VNTOL by a factor of TRTOL. You can relax all transient tolerances by increasing TRTOL to 10 from its default of 7.

See Timestep Control for more information on TRTOL and the local error tolerance.

VerifEye and Quick Eye Analyses

The eye diagram is a convenient way to analyze the performance of a serial communications channel. In a traditional eye diagram, copies of the waveform generated by transient analysis are overlaid at a spacing of one unit interval (UI). The width of the eye is affected by timing variations such as jitter and by variations in setup and hold times. The maximum allowable shrinkage in eye width is called the *jitter budget*. The height of the eye is affected by voltage variations or noise. The maximum allowable closure in eye height is called the *noise margin*. The jitter budget and noise margin depend on the maximum allowable bit error rate (BER), the ratio of correctly received bits to the total number of bits sent. In high-speed channels, the required BER becomes very small, on the order of one error in 10^{12} bits. Generating a useful eye diagram requires transient simulation over multiple terabits.

In practice, however, designers are interested only in the *probability* that a waveform violates the eye window. For this purpose, faster statistical techniques can be used instead of the full transient simulation.

- Quick Eye analysis uses simplifying assumptions to calculate an eye diagram from a transient analysis of single transitions.
- VerifEye analysis uses a fully statistical approach to calculate the BER directly, similar to the public domain Stat Eye.

Quick Eye Analysis

Quick Eye analysis employs pattern-dependent convolution. QuickEye uses a brief transient simulation to calculate the channel's step responses to a single rising edge and a single falling edge. QuickEye combines the step responses with the data stream to approximate the output of the channel. The eye diagram is formed by overlaying UIs in the usual manner. With QuickEye, the effect of inter-symbol interference (ISI) on the bit error rate is easy to calculate.

VerifEye Analysis

VerifEye analysis is a statistical eye-analysis algorithm. VerifEye also calculates the rising and falling edge responses. VerifEye calculates the probability density function (PDF) for the receiver voltage, taking into account the conditional probabilities for the various kinds of transitions. The BER at a given sample time is the proportion of the voltage PDF that results in an incorrect bit being read. A plot of the BERs over all sample times generates the familiar bathtub curve.VerifEye simulates transmit jitter by adding the jitter distribution to the PDFs for the edge responses. With edge responses, random jitter is truly random (timings of edges are independent) and duty cycle distortion (DCD) is easy to calculate. QuickEye simulates transmit jitter using Monte Carlo techniques.

Channel Equalization

Both QuickEye and VerifEye have the ability to simulate the effects of equalization on the channel. Both analyses can add feed-forward equalization (FFE) and decision-feedback equalization (DFE).

Peak Distortion Analysis

Peak distortion analysis (PDA) to find the worst case bit pattern is automatically performed for both VerifEye and QuickEye analyses. For Quick Eye analyses, the worst case bit pattern can be pushed to all eye sources, overriding any bit lists set in the eye sources. For both QuickEye and VerifEye analyses, the worst case bit pattern can be displayed in reports.

Running VerifEye Analysis on the Schematic Editor

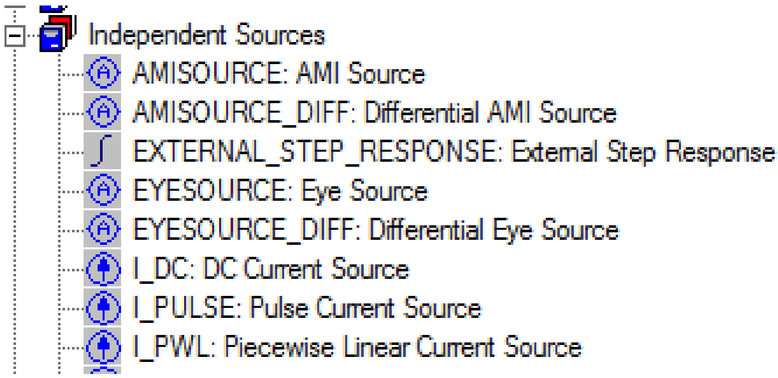
To run VerifEye analysis, the design must include at least one Eye Source and a corresponding Eye Probe. Channel parameters including equalization are set up in the source and probe. Other parameters can be specified in the setup window.

Add an Eye Source for VerifEye

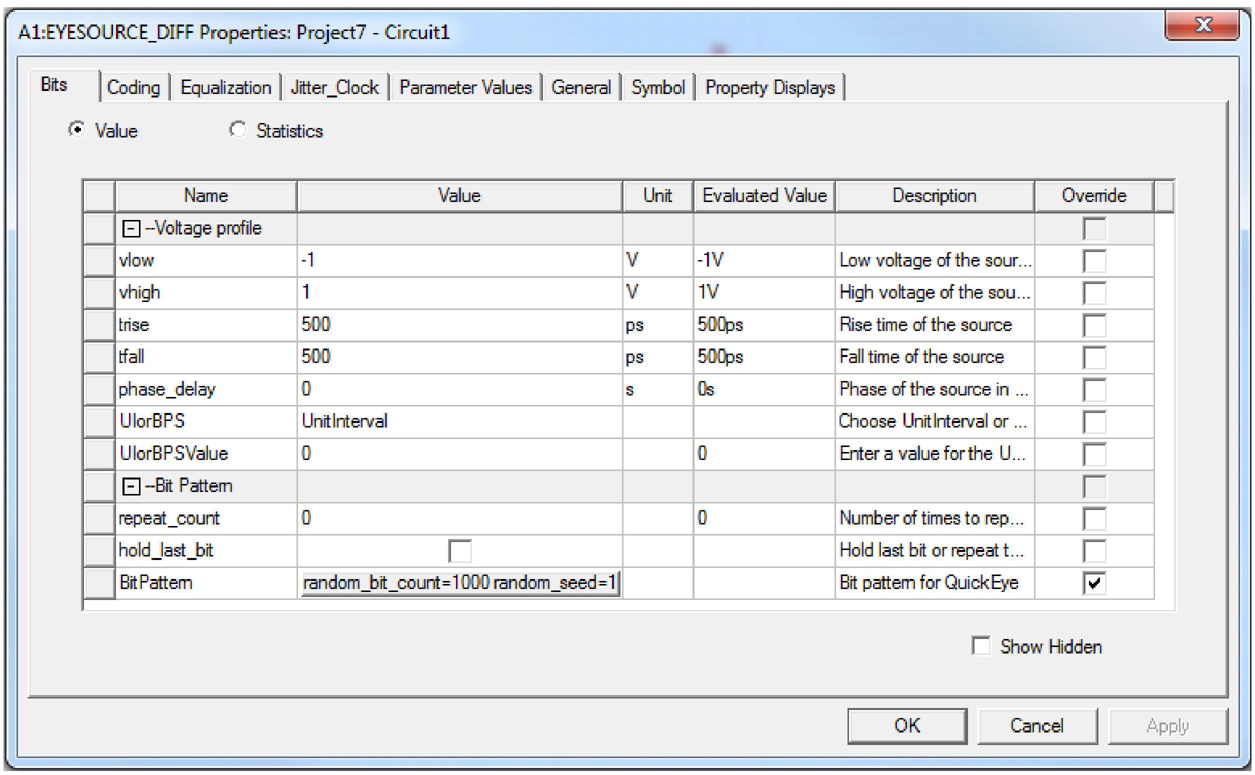
To run VerifEye analysis, the design must include one Eye Source and a corresponding Eye Probe. Note that the Eye sources are internally grounded. All terminals should be connected to signal lines — not directly to ground.

- For Quick Eye analysis, all Eye Source parameters apply except spread spectrum clocking.
- For VerifEye analysis, all parameters apply except the bit pattern generators and spread spectrum clocking.
- For Transient analysis, all parameters apply.

To add an Eye Source, select the source on the Independent Sources list on the Components tab:



Both single-ended and differential versions are available. Click the Eye Source to open the Properties list:



Bits Tab

The **Properties** window opens with the Bits tab displayed.

- The logic low, **vlow**. The default is 0 volt.
- The logic high, **vhigh**. The default is 1 volt.
- The rise time of input (**trise**). The default is 5.0e-10 seconds (500 picoseconds).

- The fall time of input (**tfall**). The default is 5.0e-10 seconds (500 picoseconds).
- The **phase_delay** for this source. The default is 0 seconds.
- The data rate: Click the **UIorBPS** drop-down to select **UnitInterval** or **BitsPerSecond**. Then Click the **UIorBPS Value** field to enter the unit interval size or number of bits per second.

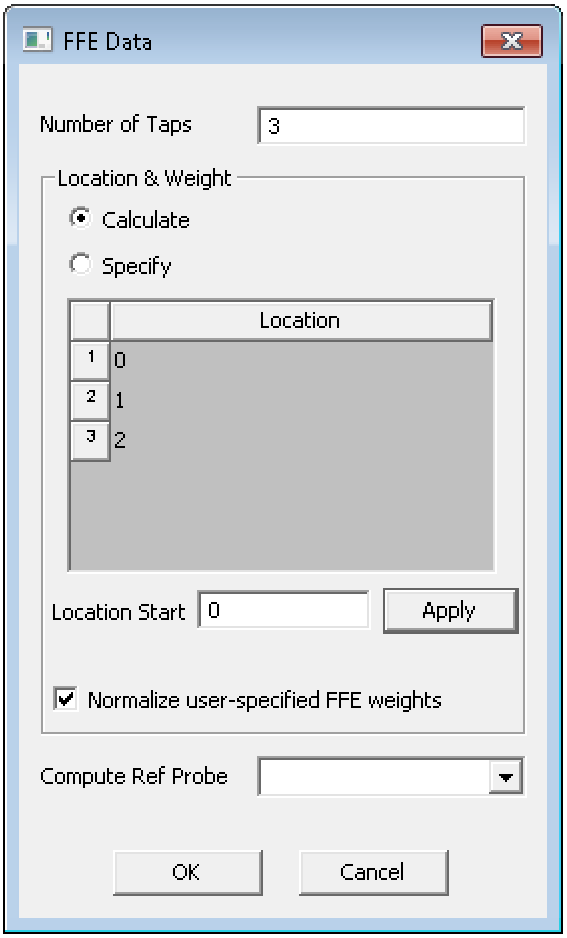
Coding Tab

The Coding tab contains parameters to set up bit encoding and select NRZ or PAM4 modulation.

- The **do_encoding** parameter controls 8b10b, 64b66b, 128b130b, or 128b132b encoding of the transmitted bit stream. The default is no encoding (None). Select 8b10b encoding, 64b66b encoding, 128b130b encoding, or 128b132b encoding on the drop-down menu. Note: Option **do_encoding** is valid only when **modulation=NRZ**. **do_encoding** is ignored when **modulation=PAM4**
- The **modulation** to be used. The default **NRZ** uses two voltage levels (binary data), **PAM4** uses four voltage levels to represent two-bit symbols. See **coding**.
- In the window, the **coding** parameter is labeled **coding_PAM4_only** as a reminder that it is ignored for NRZ transmissions. The **coding** parameter defines how digital bits are sent as analog waveforms in PAM-4 transmissions. In PAM-4, two bits ("00"/"01"/ "10"/"11") are transmitted at a time; in Gray coding, "00" is transmitted as "vlow," "01" as "vlow + (vhigh-vlow)/3," "11" as (vhigh - (vhigh-vlow)/3), "10" as "vhigh". In Linear coding, these voltage levels (smallest to largest) are used to transmit bits "00," "01,"10," and "11," respectively. The **coding** parameter has no effect in NRZ transmissions. In NRZ, one bit ('0' or '1') is transmitted at a time; '0' is transmitted as "vlow," and 1 as "vhigh."

Equalization Tab

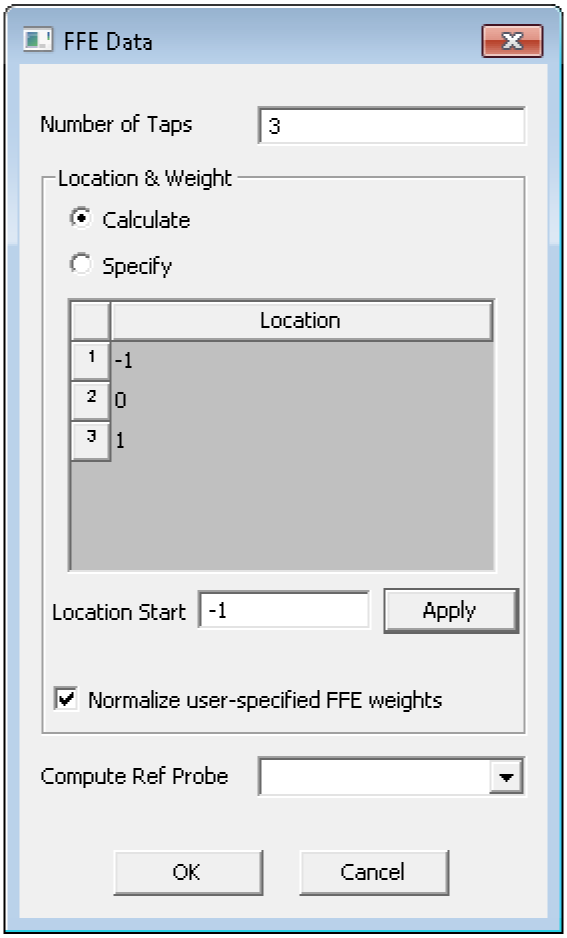
The Equalization tab allows you to set up Feed-Forward Equalization in the Eye Source. To apply Feed-Forward Equalization, click **FFE_data** and open the window:



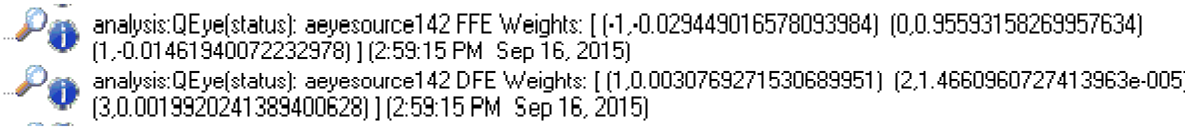
- To enable Feed-Forward Equalization, set the **Number of taps** in the **FFE** field to a positive, non-zero integer value. The default is zero taps (no FFE). When one or more taps have been specified, a list of tap numbers and cursor locations appears. In the example above, three taps have been specified. QuickEye applies the step response to transitions to generate the eye diagram. Each FFE tap has an associated weight that adjusts the applied step response in order to correct for the effect of the bit sampled at that location in the original step response.
- Tap numbers are always numbered from 1 to N, the number of taps you specify. The cursor locations are sample points in the step response over the number of unit intervals (UI) equal to the number of taps. Cursor 0 is the main cursor, whose weight affects the value of the current bit. By default, Tap 1 is at cursor location 0, the main cursor. Taps at positive locations are postcursors, bits that occur later in time than the main cursor bit. Taps at negative locations are precursors, bits that occur earlier in time than the main cursor bit. In the example above, Tap 1 is at main cursor 0. Taps 2 and 3 are postcursor bits.

  Note that the list of cursor locations must include the main cursor, location 0, and when negative tap locations are specified, it may be necessary to add a delay (of at least one UI) to the transient analysis, if there is no inherent delay in the system.

- Use **Location Start** to set Tap 1 to 0 or to a positive or negative cursor location. Click **Apply** to apply the new starting location. The locations of the taps in the display change to reflect the selected starting location. In the following example, three taps of FFE are applied, with one precursor, the main cursor, and one postcursor.



- By default, Nexxim calculates the tap weights on the (unequalized) step response. The algorithm always places the largest tap weight at location 0. For weights calculated by Nexxim, normalization is always performed such that the sum of the absolute values of all the weights is equal to one. For VerifEye, the tap weights are applied to the step response before analyzing. The tap weights calculated by QuickEye are displayed in the **Message Manager** window after the Quick Eye analysis has completed.



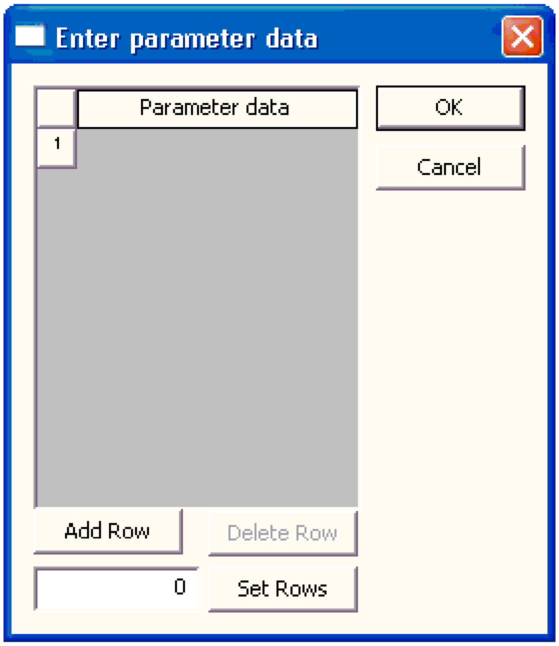
- To specify the tap weights directly, select **Specify** and enter the weights. For weights supplied by the user, normalization of the tap weights to 1 is enabled by checking the **Normalize FFE weights** check box. Normalization of user-supplied weights is enabled by default in the Eye Source **Properties** window. Uncheck the box to disable normalization of user-supplied weights.
- Make a selection on the Eye Probe drop-down menu to be the **Compute Ref Probe** for the FFE calculation. When FFE has been enabled, a **Compute ref probe** is required.
- The **Compute ref probe** field is automatically filled in with the name of the Eye Probe when the Source and Probe are paired on the GUI. For more information, see Pairing an Eye Source and an Eye Probe in the Schematic.
- Click **OK** to close the **FFE_data** window and return to the Properties window. For more information on FFE, see Feed-Forward Equalization.

Jitter_Clock Tab

The Jitter_Clock tab contains parameters to set up DCD, transmit jitter, and Spread Spectrum clocking.

- Duty cycle distortion: Click the **DCDFractionorTime** dropdown menu to select **Fraction** (the default) or **Time**. For **Fraction**, the DCD value is a decimal fraction of the UI between 0.0 and 1.0. For **Time**, the DCD value is a time between 0.0 and UI in seconds. A time unit such as ps may be used to set the DCD time. The default for both Fraction and Time is 0.
- For more information, see Duty Cycle Distortion in the QE/VE Technical Notes.

For **txrj**, **txpj**, and **txuj**, clicking within the **Value** field allows you to specify one or more values, generating multiple jitter sources:



- For the random transmit jitter (**txrj**) the value is the standard deviation for the Gaussian distribution. The default is 0 seconds.
- For the Periodic random transmit jitter (**txpj**) the value is the amplitude. The default is 0 seconds.
- For the Uniform random transmit jitter (**txuj**) the value is the amplitude. The default is 0 seconds.
- For the User-defined transmit jitter (**txcj**) the value is the name of the file containing the time (seconds and probability density function (PDF) data. There is no default.
- For more information on transmit jitter parameters, see Transmit Jitter in the Technical Notes.

Parameter Values Tab

The Parameter Values tab contains parameters that are seldom changed.

- The **resistance** to be placed in series with the eye source. The default is 50 Ohm.
- The number of unit intervals to be used in computing the step response (**step_resp_ num_ui**). The default is 100.
- Click **EYESOURCE** in the **Value** field of the **Info** property to open the help for the Eye Source component.

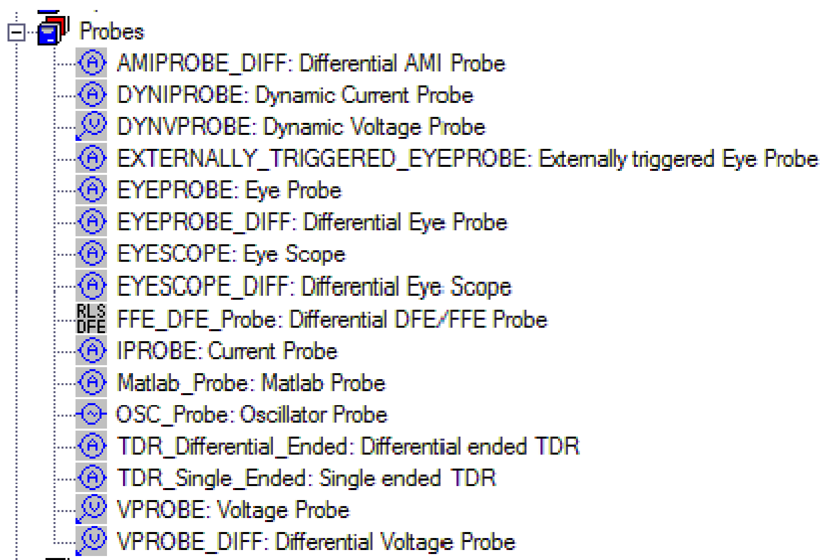
The other properties on the Parameter Values tab are primarily for use by the Component Library editors.

General, Symbol, and Property Display Tabs

The parameters on these tabs are primarily for use by the Component Library editors.

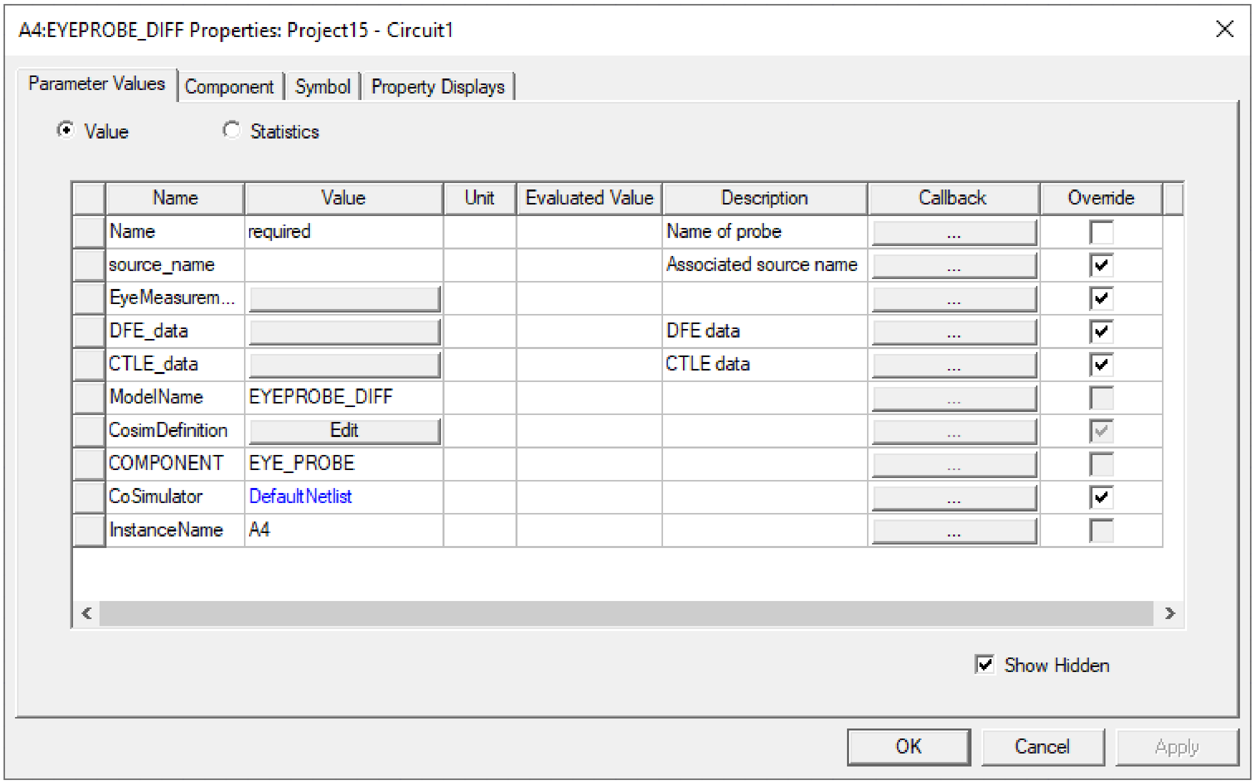
Add an Eye Probe for VerifEye

The Eye Probe may be selected on the Probes list on the Components tab:



Both single-ended and differential versions are available.

Click the Eye Probe to open its parameter list:

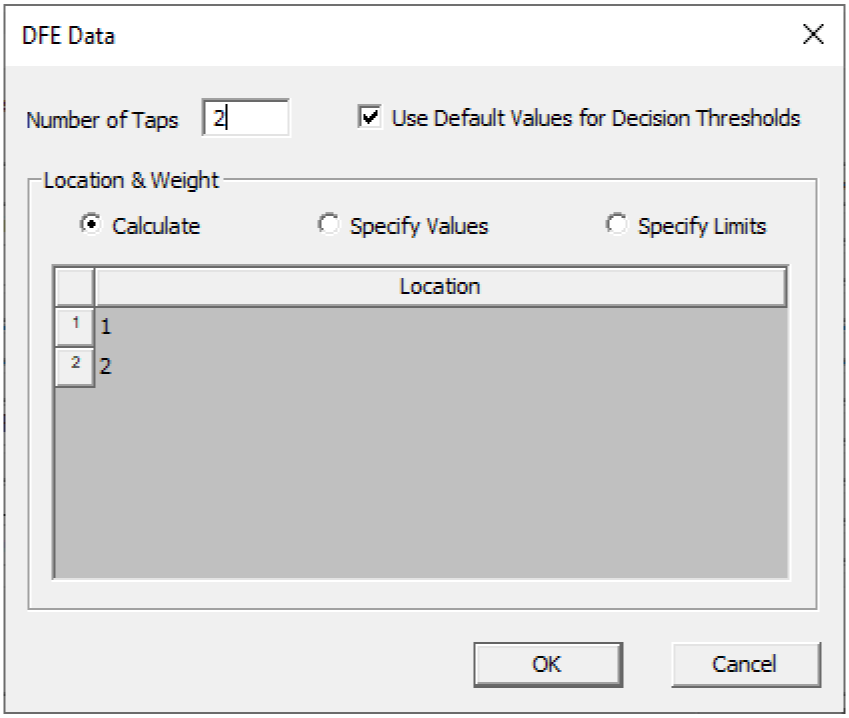


The Parameter Values tab is the default.

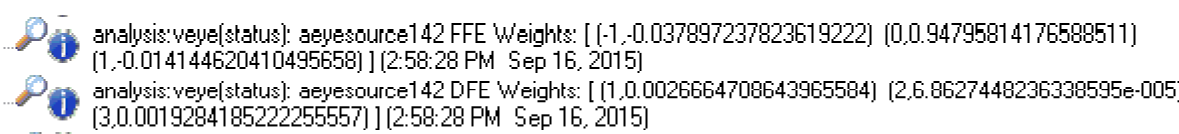
- Specify a name for the Eye Probe.
- Select the Eye Source for this probe on the **source_name** drop-down menu. A channel can have multiple sources, but only the source actually at the transmitter should be the reference source.

The **source_name** field is automatically filled in with the name of the Eye Source when the Source and Probe are paired on the GUI. See Pairing an Eye Source and an Eye Probe in the Schematic.

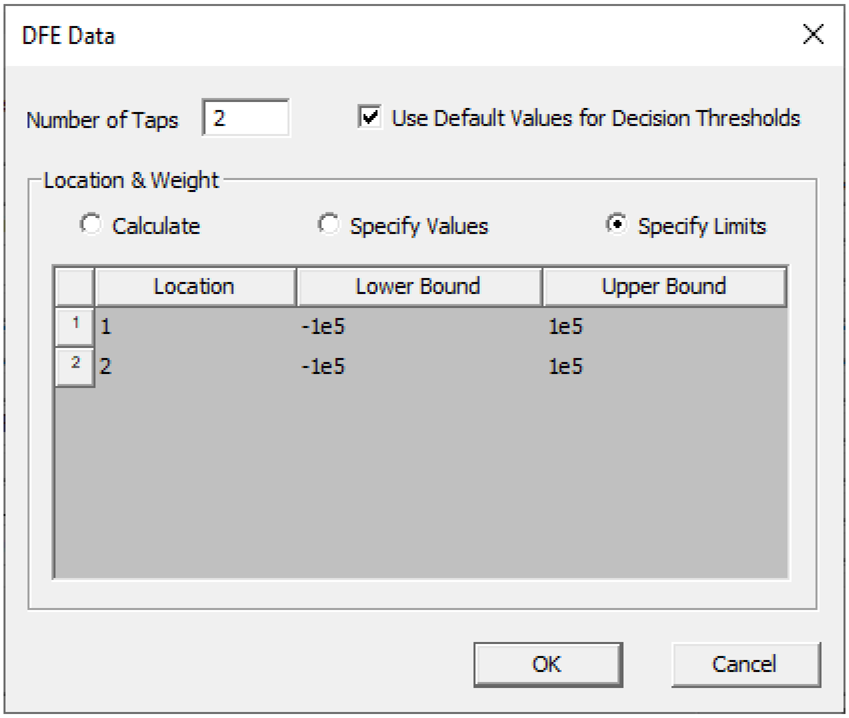
- To apply decision-feedback equalization to the received bits, click **DFE_data** to open the window:



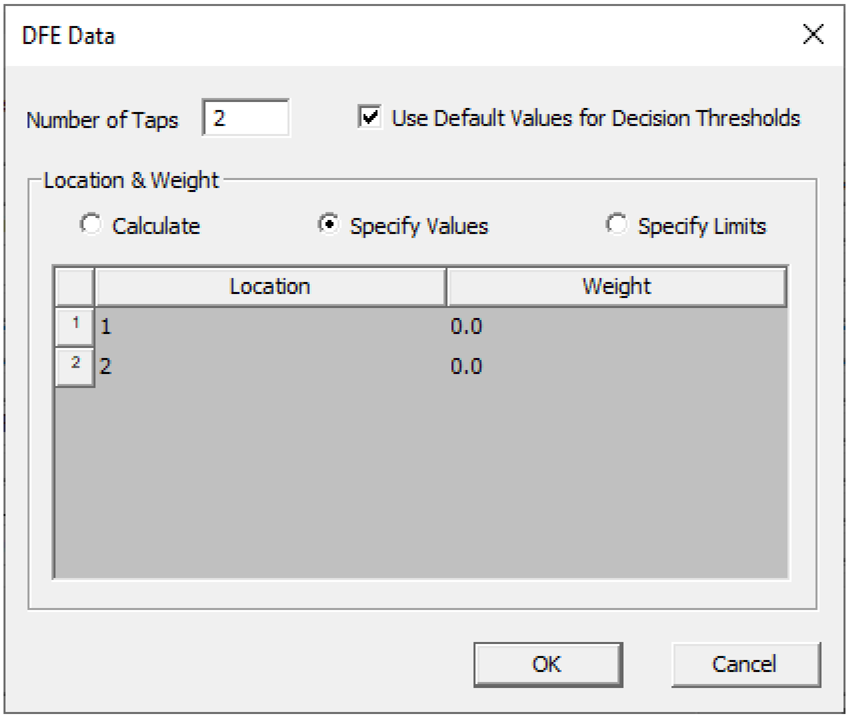
- To enable Decision-Feedback Equalization at the probe, set the **Number of taps** in the **DFE** field to a positive, non-zero value. The default is zero taps (no DFE). When one or more taps have been specified, the list of tap numbers and locations appears. The tap locations are those of previous bits. DFE corrects for inter-symbol interference (ISI) on the current bit due to previous bits.
- DFE applies its tap weights to the decision threshold for each bit to correct for inter-symbol interference (ISI) in the transition from previous bits to the current bit. A DFE with M taps is like a buffer of M bits. Tap 1 is applied to the earliest bit to arrive in the DFE buffer. Tap 2 is applied to the next later bit, et cetera.
- The value of each bit is decoded, and the difference between the actual voltage and the ideal voltage is calculated. Tap weights compensate for the difference. In VerifEye, the weights are applied to the step response.
- By default, the Circuit solver calculates the weights to apply to the DFE taps. The tap weights calculated by VerifEye are displayed in the **Message Manager** window after the VerifEye analysis has completed.



- To automatically calculate the tap weights, choose **Specify Limits** and set the limits. The tap weights are calculated in the bounds you specify. The default value for bounds are +/-1e5 for taps. There are no requirements on the limits.



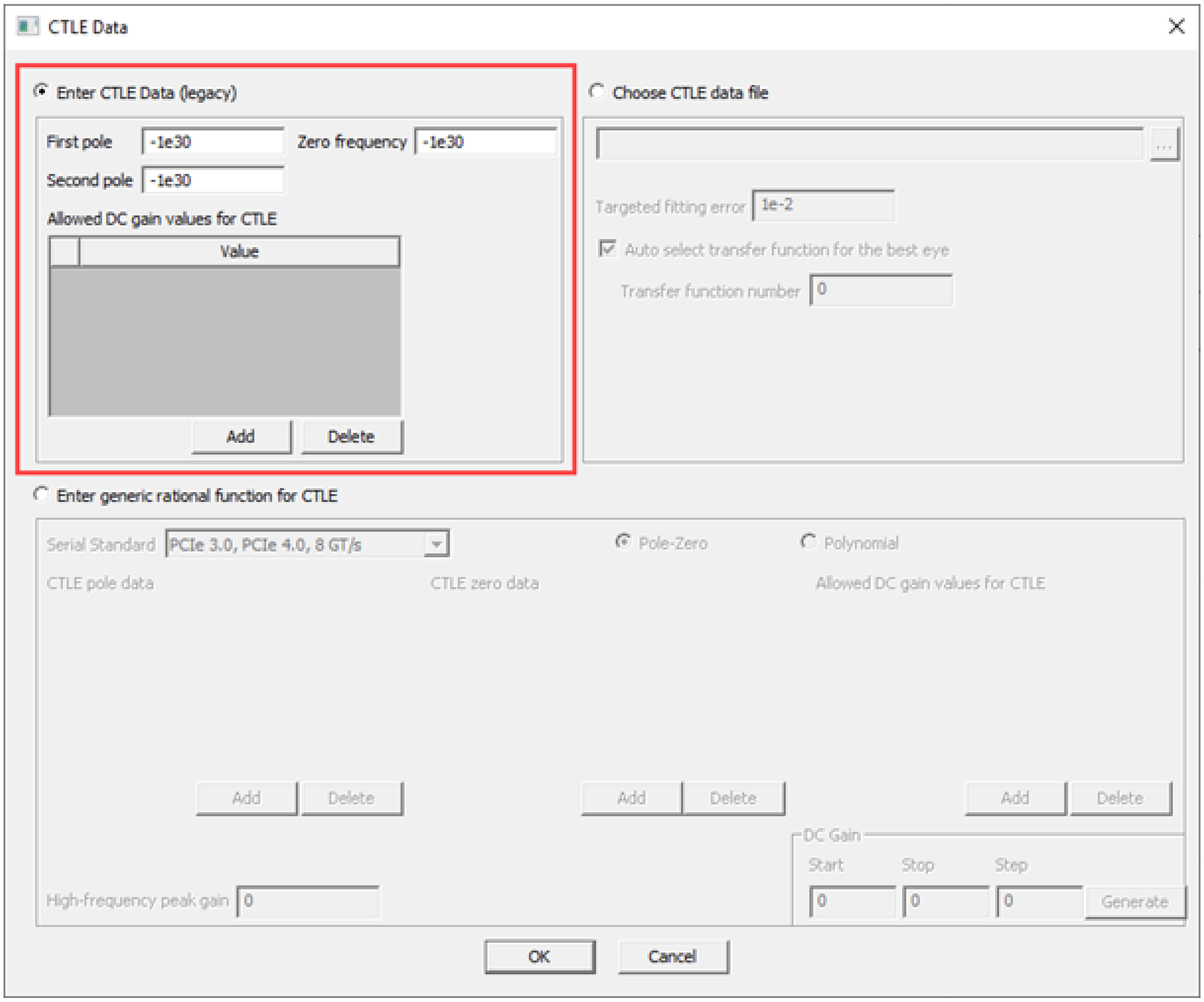
- To specify your own set of tap weights, select **Specify Values** and click the weights to specify them.



- To decode the received bits, mark **Use Default Values for Decision Thresholds** which causes Nexxim to calculate the thresholds.
- For more information on DFE, see Decision-Feedback Equalization.
- Click **OK** to close the **DFE_data** window and return to the Properties window.
- Click **CTLE_data** to open the **CTLE Data** window. (See Continuous Time Linear Equalization in the QE/VE Technical Notes for details).

Select **Enter CTLE Data (legacy)**, **Choose CTLE data file**, or **Enter generic rational function for CTLE**.

If you select **Enter CTLE Data**, only the CTLE data group box is activated:

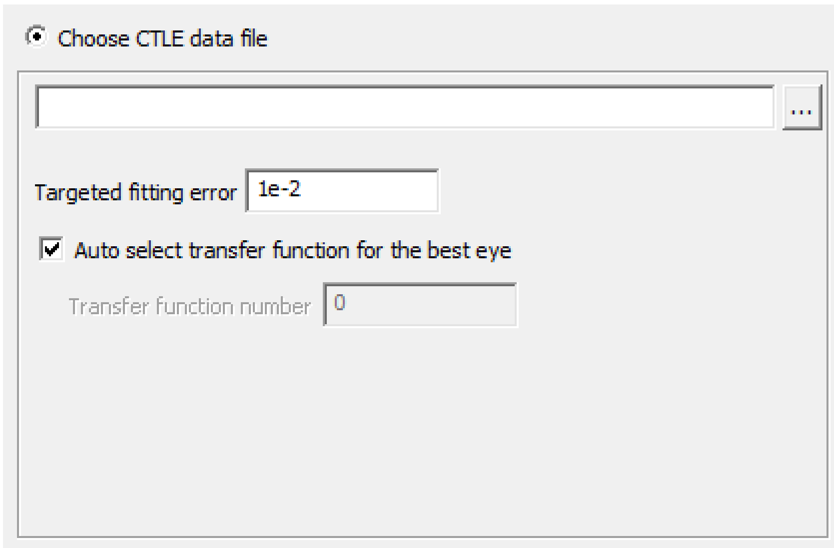


- Specify the first pole frequency in Hz (positive). The first pole frequency typically is the lower bound of the high passband.
- Optionally, specify a second pole frequency. The second pole frequency typically is the higher bound of the passband.
- Optionally, specify a zero frequency, typically where the response curve begins to rise.

> **Note:**
> 1. The default pole and zero frequency values of -1e30 are equivalent to "None."
> 2. If the zero frequency is specified, it must be greater than 0.0.
> 3. See Continuous Time Linear Equalization in the QE/VE Technical Notes for an illustration of the two-pole, one zero filter.

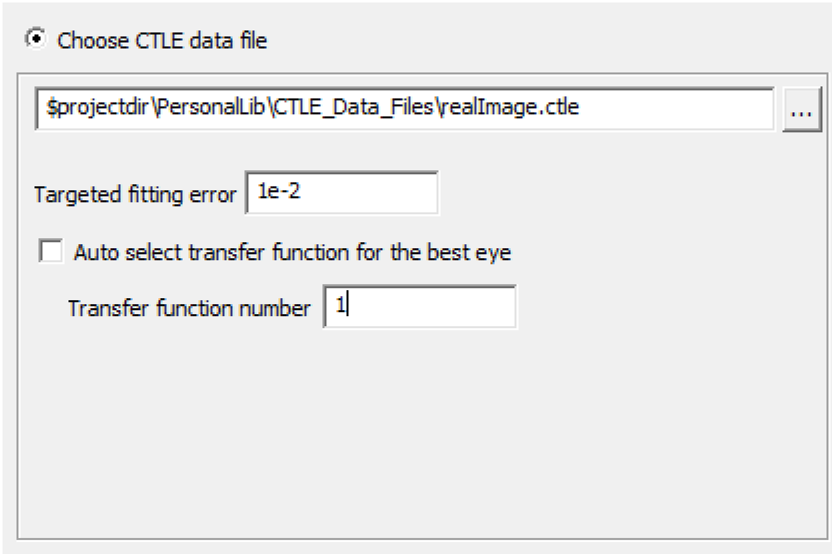
- Optionally, click **Add Row** and **Delete Row** to specify one or more allowable values for the DC gain factor in dB for the CTLE calculation. Nexxim varies the gain to find the optimum equalization (maximum eye height). If no values are provided, the Circuit solver calculates the optimum CTLE gain (between -infinity dB to 0 dB).

If you select **Choose CTLE Data File**, only the CTLE data file group box is activated. Here is that portion of the window:



- Use **Browse** to access the **File Open** window. Browse to the target directory and select the file with the CTLE transfer function data (**.ctle** extension). Nexxim fits a rational function to the data.
- Optionally, Click the **Targeted fitting error** field to specify a tolerance to apply to the CTLE rational function calculation using the transfer function data.
- By default, **Auto select transfer function for the best eye** is enabled. Nexxim tries all the transfer functions on the file, and uses the one that yields the maximum eye height. Optionally, uncheck the **Auto select** group box, then enter the number of a transfer function among the multiple transfer functions in the data file.

In the following figure, transfer function 1 is specified

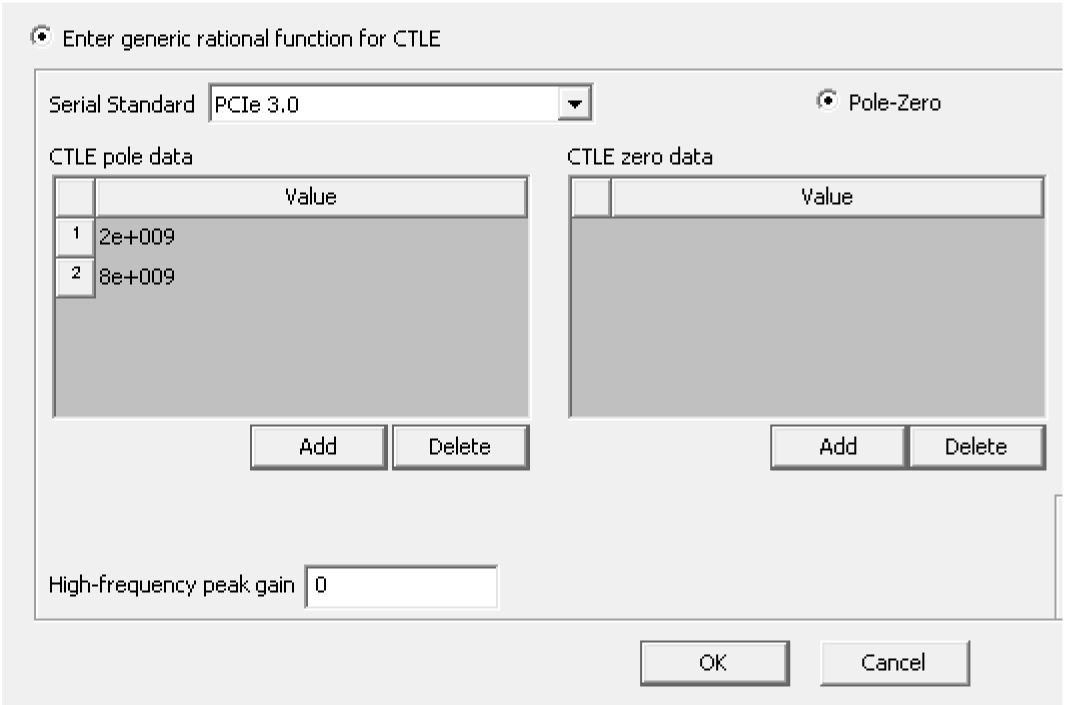


- Values for the transfer function number run from 0 (the **Auto select** default) to the number of transfer functions in the file. The first transfer function is number 1, the second is number 2, et cetera. If the number given for this parameter is less than 0 or greater than the number of transfer functions in the file, an error occurs.

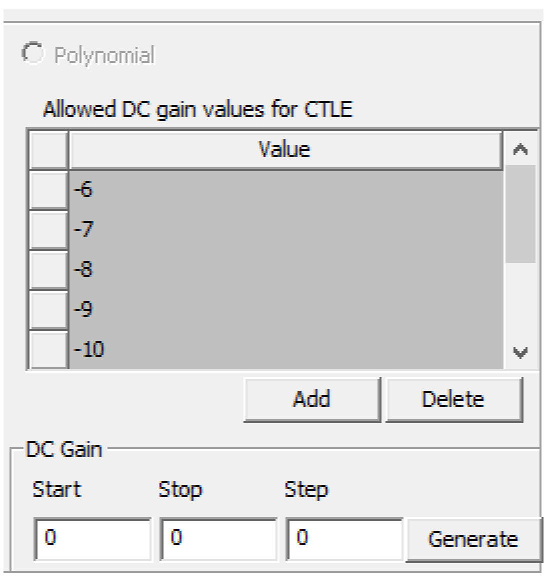
When you select **Enter generic rational function for CTLE**, only that group box is activated.

There are two formats for specifying a generic rational function, chosen by selecting **Pole-Zero** or **Polynomial**. By default, the **Pole-Zero** format is selected. With this format, the rational function is specified by a set of pole frequencies, a set of zero frequencies, the high-frequency peak gain, and a list of allowable DC gain values.

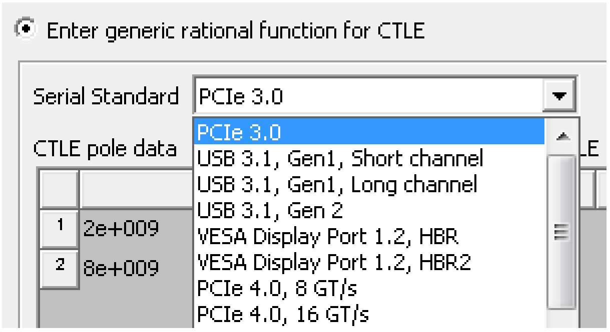
Here are the entry fields for the pole and zero frequencies and the High frequency peak gain:



Here is the entry field for the DC gain values:



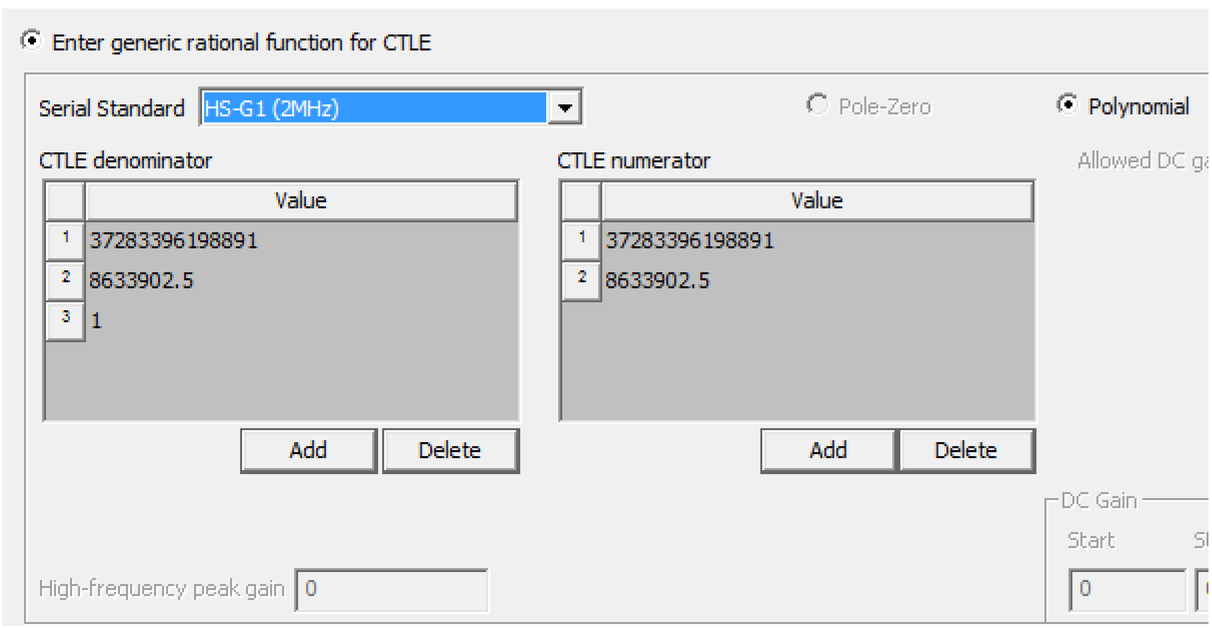
The **Serial Standard** field has a drop-down menu with a selection of preset standard CTLEs. The PCIe, USB, and VESA preset CTLEs use only the **Pole-Zero** format.



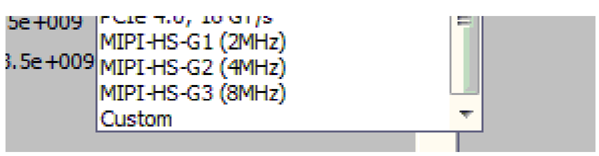
> **Note:** This illustration may not include later additions to the preset standards.

Alternatively, select **Polynomial** format. In this format, the rational function is specified by giving the coefficients for the terms in the numerator and denominator of the polynomial fraction. The **High-frequency peak gain** and **Allowed DC gain** fields are inactivated in the **Polynomial** format.

Here are the entry fields for the **Polynomial** format coefficients:



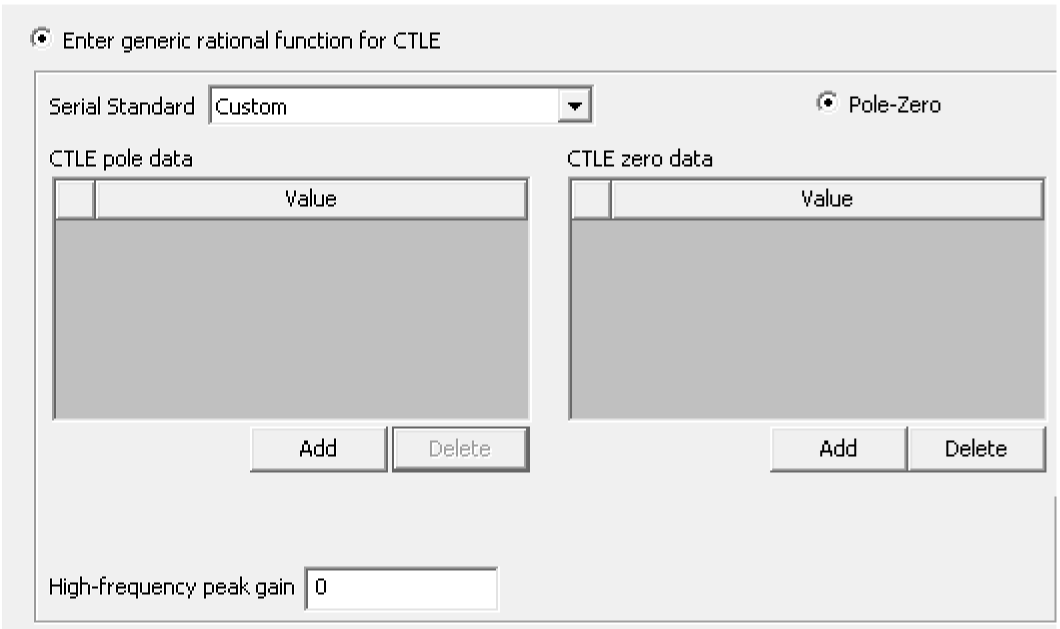
The MIPI-HS-G1, MIPI-HS-G2, and MIPI-HS-G3 CTLE presets use only the **Polynomial** format.



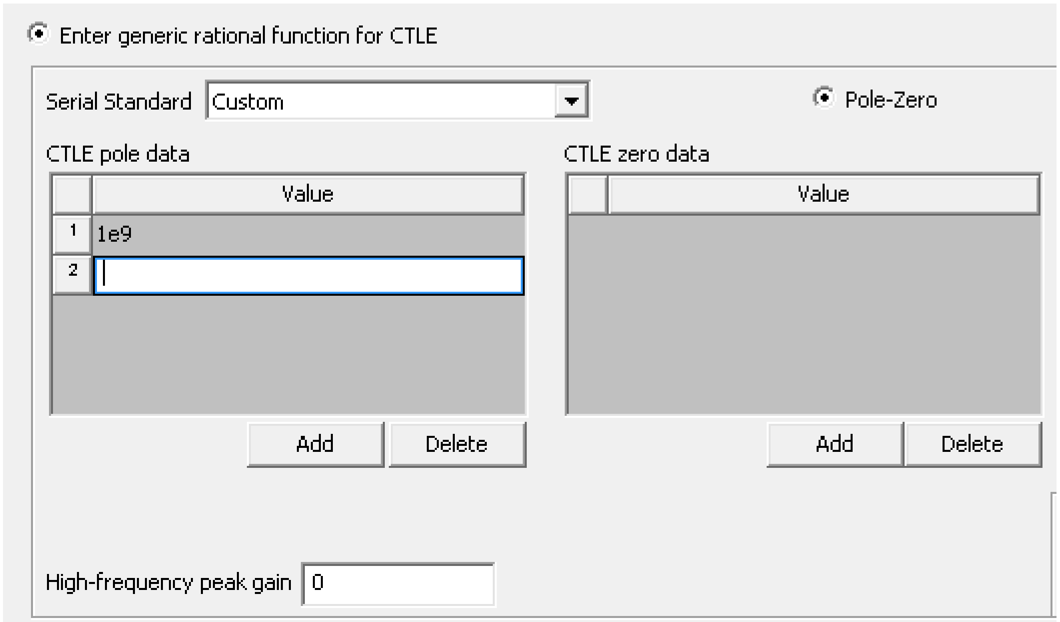
> **Note:** This illustration may not include later additions to the preset standards.

To create your own rational function, select **Custom** on the menu. The **Custom** entry allows you to specify a generic rational transfer function in either **Pole-Zero** format or **Polynomial** format.

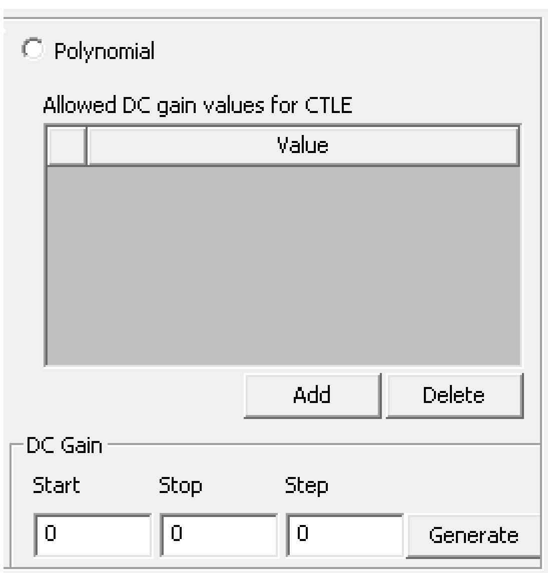
Here are the **CTLE pole data**, **CTLE zero data**, and **High-frequency gain** entry fields for the **Custom** entry in **Pole-Zero** format:



Click **Add** to add a row in the pole or zero data list. Click in the **Value** field to enter the pole or zero frequency in Hz. Click **Return** to generate a new row. Click **Delete** in either field to delete unwanted rows. Set the **High-frequency peak gain** to a appropriate value in dB.

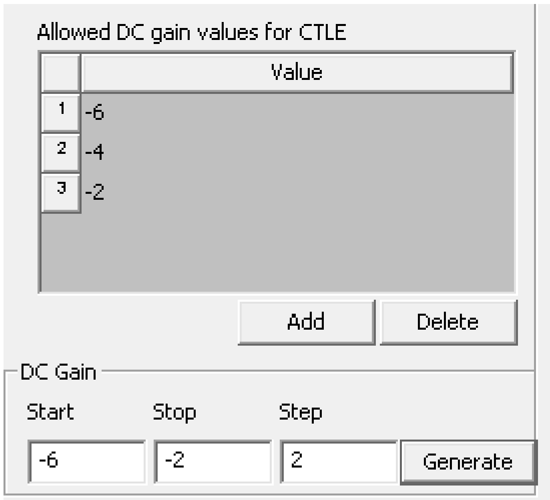


Use the **Allowed DC gain** group box to specify a range of DC gain values in dB.

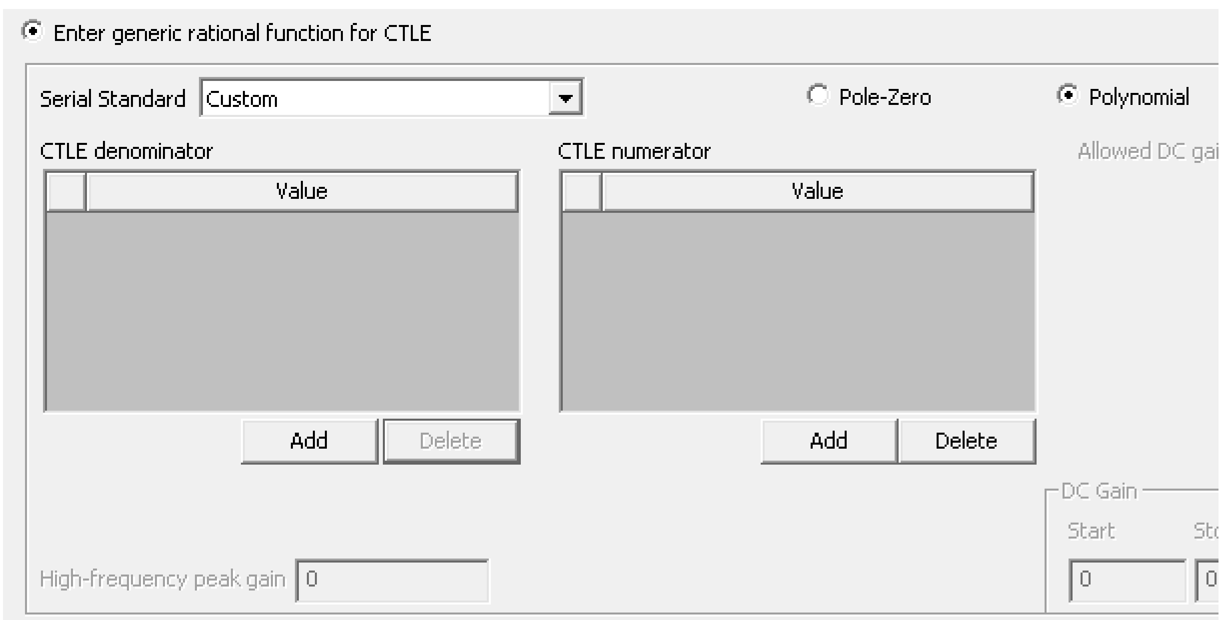


Click **Add** and **Delete** to specify one or more allowable values for the DC gain factor in dB for the CTLE calculation. Nexxim varies the gain to find the optimum equalization (maximum eye height). If no values are provided, the Circuit solver calculates the optimum CTLE gain (between -infinity dB to 0 dB).

To have Nexxim generate a sequence of DC values, enter the **Start**, **Stop**, and **Step** values in the **DC Gain** fields and click **Generate**. In this window, Start=-6, Stop=-2. and Step=2:



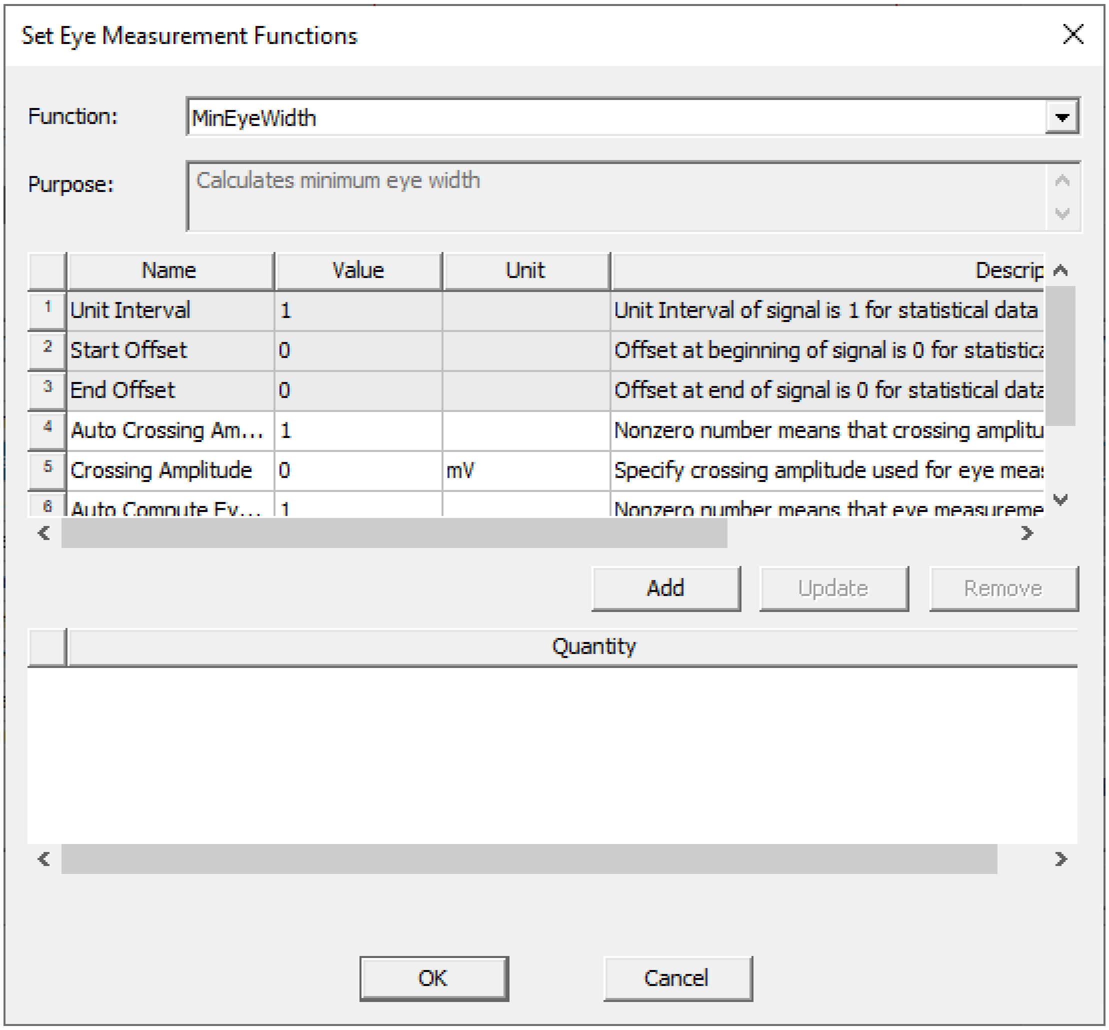
Alternatively, a **Custom** rational function CTLE can Click the **Polynomial** format.



Click **Add** to add a row in the CTLE denominator or CTLE numerator list. Click in the **Value** field to enter the (unitless) coefficient. Click **Return** to generate a new row. Click **Delete** in either field to delete unwanted rows. Click **OK** to close the **CTLE Data** window.

For more information see Continuous Time Linear Equalization.

Click **EyeMeasurementFunctions** to open the **Set Eye Measurement Functions** window.



- In the **Function** field, select **MinEyeWidth** or **MinEyeHeight**.
- Set the **Unit Interval** parameter to the Unit Interval (UI) of the Eye diagram.
- Set the **Start Offset** parameter if an initial time offset is appropriate.
- Set the **End Offset** parameter if trailing time offset is appropriate.
- Leave **Auto Crossing Amplitude** set to 1 (or any non-zero number) to have the Circuit solver compute the crossing amplitude by averaging the histograms for the high and low voltages. The crossing amplitude sets the voltage at which the minimum eye width is calculated.
- When **Auto Crossing Amplitude** is 0, the Circuit solver uses the voltage set in the **Crossing Amplitude** parameter as the crossing voltage amplitude for the minimum eye width calculation.
- Leave **Auto Compute Eye Measurement Point** set to 1 (or any non-zero number) to have the Circuit solver compute the eye measurement point by averaging the times of the

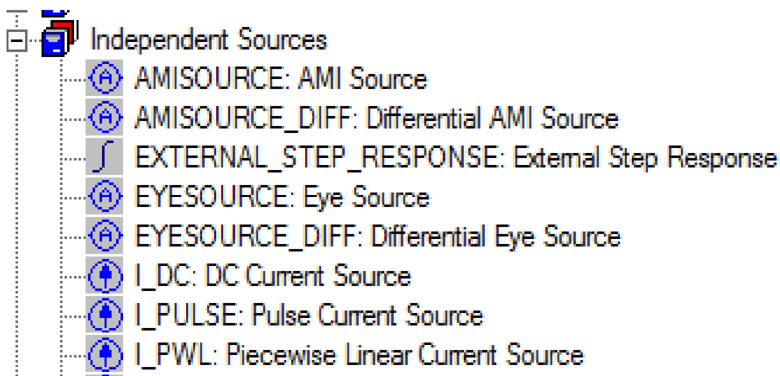
crossing points. The eye measurement point sets the time at which the minimum eye height is calculated.

- When **Auto Compute Eye Measurement Point** is 0, the Circuit solver uses the voltage set in the **Eye Measurement Point** parameter as the time for the minimum eye height calculation.
- Click **Add** to add the measurement to the analysis. Add any number of measurements. The window above has added a **MinEyeWidth** measurement using the default settings.
- Click **OK** to close the **Set Eye Measurement Functions** window.
- For more information, see Eye Measurements in the Reports topic.

Click **OK** to close the Properties window of the probe.

Add an External Step Response to a VerifEye Analysis

Instead of having the Circuit solver compute the step response, supply the step response via a data file, using the External Step Response component. First, place Eye Sources and Probes in the circuit. Select the External Step Response on the Independent Sources list on the Components tab:



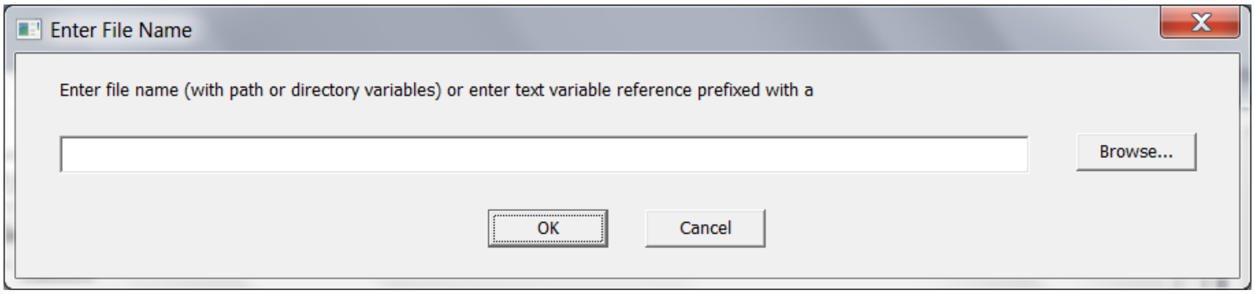
The External Step Response component does not physically connect to the circuit.

Click the component to open the Properties list:

source_name				Eye source name
probe_name				Eye probe name
step_response_file_rise				Rising step response fil...
step_response_file_fall				Falling step response fil...

Select the source name and the probe name. The source and probe names appear automatically when they are present in the circuit. Each step response source applies to one source and one probe.

Specify the files that contain the rising and falling step responses by clicking the buttons in the **Value** field. The same window applies to both files.



If only one file is supplied, the rise and fall are treated as symmetrical.

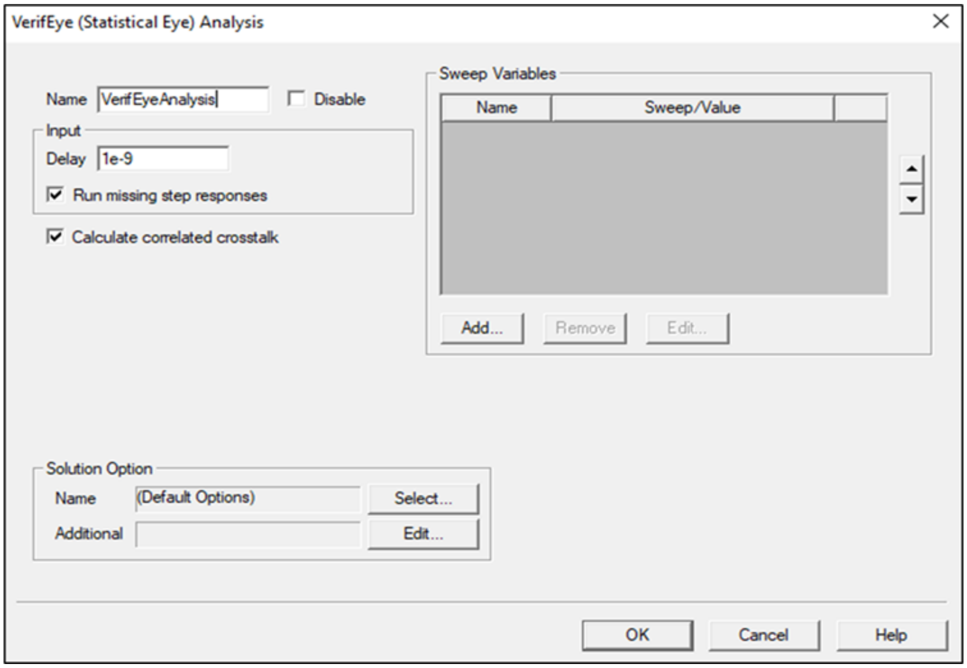
> **Note:** If no step response file is specified (or found), the default is for analysis to fail. Setting the VE/QE analysis parameter **run_missing_step_response=1** runs transient analysis to generate the missing step response.

The step response data files have a two-column format: *time voltage*. Linear interpolation is used between time points.

Set Up the VerifEye Analysis

From the **Schematic Editor**, perform the following steps to set up and run a VerifEye analysis.

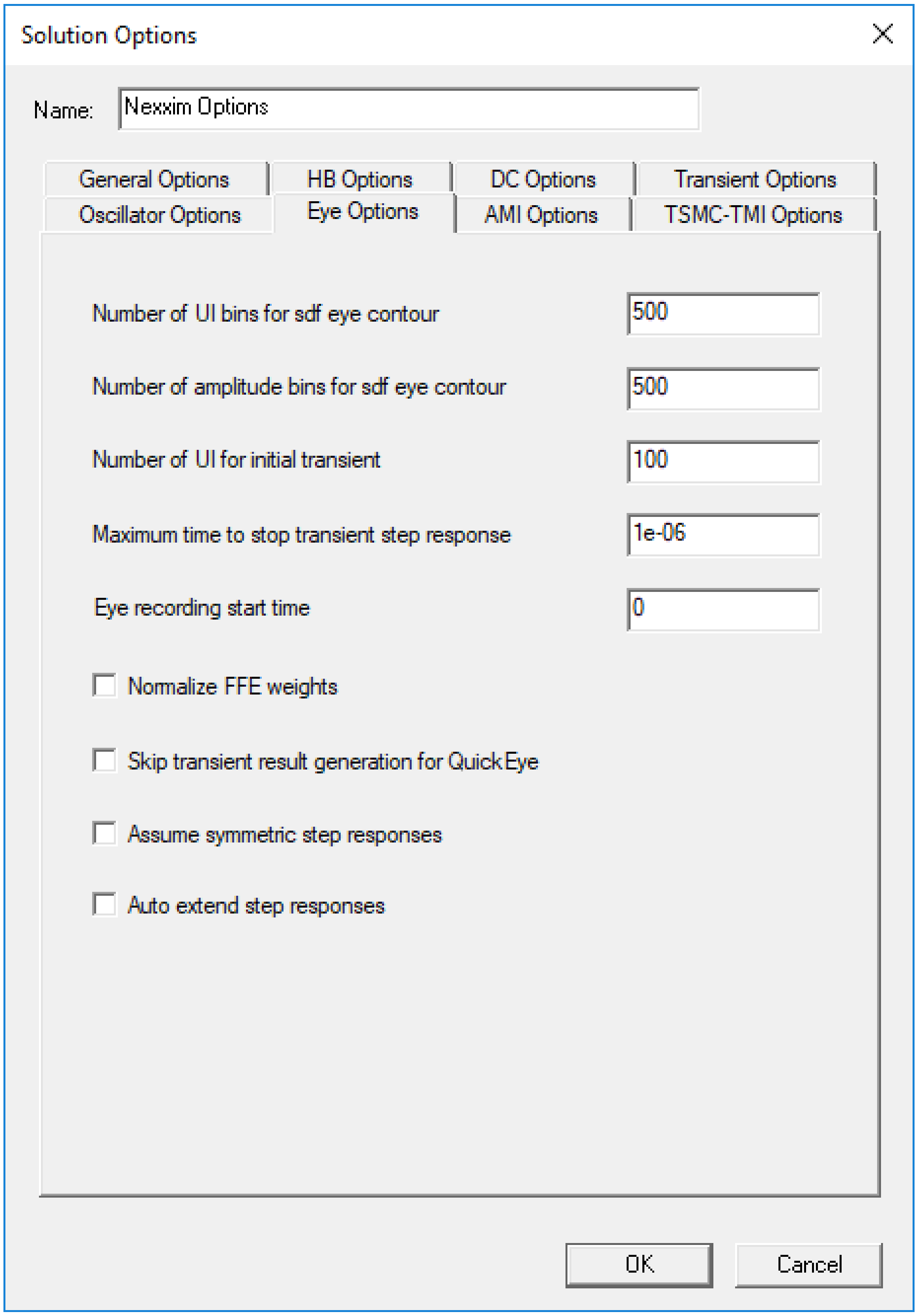
1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **VerifEye Analysis** to open the **VerifEye (Statistical Eye) Analysis** window.



2. Type an **Analysis Name** (or accept the default name, for example “VerifEyeAnalysis”).
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use.
4. From the **Input** group box:
   - Specify the **Delay** to be applied before the step response is applied (default is 0 seconds)
   - **Run missing step response** allows simulation to proceed when an External Step Response component is present in the design but no step response data file can be found.
5. **Calculate correlated crosstalk** is checked (i.e., activated) by default. To compute uncorrelated crosstalk, uncheck the **Calculate correlated crosstalk** box.
6. To add a sweep, locate the **Sweep Variable** area in the **VerifEye Analysis** window.
   - From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
   - In the **Variable** list, select **Temp** or the name of a variable (when a variable has been defined for the design), then select one of the following: Single value, Linear step, Linear count, Decade count, or Octave count.
   - Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
   - For details on sweeps, see Variable Sweep.
   - Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **VerifEye Analysis** window.

6. Optionally, use the fields in the **Solution Options** group box to add **VerifEye/QuickEye** options.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.
- To create a new option set, click **New** to open the **Solution Options** window.



- Click the **Name** field to name the new option set.
- Click the **Eye Options** group box fields and check boxes to set any appropriate options.

Number of UI bins sets the number of UI histogram bins to use for generating the 3D and contour data. The default is 500. Option: **eye.ui_bins**.

Number of amplitude bins sets the number of amplitude histogram bins to use for generating the 3D and contour data. The default is 500. Option: **eye.ampl_bins**.

Number of UI sets the number of unit intervals (UI) over which the initial transient step response should run after any channel or source delay. The default is 100. Option: **eye.num_initial_ui**.

Maximum time sets the maximum stop time for the transient step response calculation. The default is 1µs. Option: **eye.resp_tmax**.

NOTE: VerifEye uses the minimum of (**num_initial_ui** x UI) and (**resp_tmax**) to set the transient final time for the step response calculation.

Eye recording start time adds a delay in seconds to the start of sampling for the Statistical Eye diagrams. Default is 0 seconds. Option: **eye.eye_start_time**.

Normalize FFE weights sets the solver to normalize FFE tap weights (sum of absolute values of all tap weights equals 1). This option applies only to weights supplied by the user. The Circuit solver always normalizes the taps weights that it calculates. By default, the option to normalize user-supplied weights is not enabled in the **Solution Options** window. Option: **eye.normalize_ffe_weights**.

Skip transient result turns off QuickEye storage of the eye diagram and the transient-like result. By default, QuickEye stores the eye diagram and the transient-like result. Option: **eye.qe_only_cmf**.

Assume symmetric step responses sets QE/VE to use the rise time as the fall time. The default is to use different rise and fall times. Option: **eye.sym_step_resp**.

Auto extend step responses extends the step response beyond the simulation time. By default, the step response is extended only to the end of simulation time. Option: **eye.auto_extend_step_resp**.

- For more information, see VerifEye and Quick Eye Analysis Options.
- Click **OK** to return to the **VerifEye (Statistical Eye)Analysis** window.

Run the VerifEye Analysis and Display Results

1. From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.

- The **Message Manager** window signals success or failure.

For information on viewing the results of the VerifEye analysis, see "Display VerifEye Analysis Outputs " on page 11-64. For details on creating and modifying reports, see Generating Reports and Post-processing.

Running VerifEye Analysis on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist contains the information for running the envelope analysis. No Solution Setup is required.

VerifEye Source Netlist Syntax

The netlist syntax for the VerifEye Source is:

```
AEYESOURCExxxxn1 [n2] COMPONENT=EYE_SOURCE
 + RESISTANCE=val
+ VLOW=valVHIGH=val TRISE=val TFALL=val
+ UI=valBPS=val
+ STEP_RESP_NUM_UI=val
+ DCD=valDCD_TIME=val
+ TXRJ=[val ...]TXPJ=[val ...]TXUJ=[val ...]
 + TXCJ='filename'
 + FFE_TAPS=val
+ FFE_LOCS=[locations]
 + FFE_WEIGHTS=[weights]
 + NORMALIZE_FFE=val
+ FFE_COMPUTE_PROBE='probe_name'
```

n1 is the node connected to the single-ended source. *n2* is the second node on a differential source.

> **Note:**
>
> The Eye sources are internally grounded. All terminals should be connected to signal lines, not directly to ground.

The entry **COMPONENT=EYE_SOURCE** is required.

See *Eye Source* in the Components help for details on these properties.

VerifEye Probe Netlist Syntax

The Eye Probe contains the equalization (CTLE and DFE) setups. The netlist syntax for the VerifEye Probe is:

```
AEYEPROBExxxxn1 [n2] COMPONENT=EYE_PROBE
 + SOURCE_NAME='source_name'
 + DFE_TAPS=val
+ DFE_LOCS=[locations]
 + DFE_WEIGHTS=[weights]
+ DFE_WEIGHT_lbS=[lower bounds]
```

```
+ DFE_WEIGHT_ubS=[upper bounds]
+ DECISION_HIGH=val
+ DECISION_LOW=val
+ DECISION_THRESHOLD=val
+ CTLE_FIRST_POLE=val
+ CTLE_SECOND_POLE=val
+ CTLE_ALLOWED_GAIN=[val ...]
 + CTLE_FILE="file_reference"
 + CTLE_RELTOL=val
+ TF_NUM=val
+ CTLE_POLES=[val ... ]
 + CTLE_ZEROS=[val ... ]
 + CTLE_AC_GAIN=val
+ CTLE_ALLOWED_GAIN=[val ... ]
 + CTLE_NUMERATOR=[val ... ]
 + CTLE_DENOMINATOR=[val ... ]
```

n1 is the node connected to the single-ended source. *n2* is the second node on a differential probe. The entry **COMPONENT=EYE_PROBE** is required.

See *Eye Probe* in the Components help for details on these properties.

VerifEye External Step Response Netlist Syntax

The netlist syntax for the Eye External Step Response is:

```
AxxxxCOMPONENT=EXTERNAL_STEP_RESPONSE
 + SOURCE_NAME='eye_source '
 + PROBE_NAME='eye_probe '
 + STEP_RESPONSE_FILE_RISE='file_reference '
 + STEP_RESPONSE_FILE_FALL='file_reference '
```

The entry **COMPONENT=EXTERNAL_STEP_RESPONSE** is required**.**

See External Files for details on references to external files.

See *Eye External Step Response* in the Components help for details on these properties.

VerifEye Analysis Netlist Format

VerifEye analysis is invoked on the netlist using the .EYE statement:

```
.EYE
LOW=val HIGH=valTRISE=val TFALL=val BPS=val|UI=val
STEP_RESP_NUM_UI=val
FFE_TAP=val FFE_LOCS=[val val...] FFE_WEIGHTS=[val val...]
NORMALIZE_FFE_WEIGHTS=val
DFE_TAP=val DFE_LOCS=[val val...] DFE_WEIGHTS=[val val...]
```

(or DFE_WEIGHT_LBS=[*val val* ...] DFE_WEIGHT_UBS=[*val val* ...])
DECISION_THRESHOLD=*val* DECISION_HIGH=*val* DECISION_LOW=*val*
DELAY=*val* RUN_MISSING_STEP_RESPONSE=*val*

The VerifEye analysis statement can specify channel parameters and equalization settings. These values is applied to any Eye Sources and Probes that do not specify them. When a parameter is set on the source or probe and in the analysis statement, the source/probe setting is used.

VerifEye Analysis Statement Parameters

Parameter	Description	Unit	Default
LOW	Logic low voltage level	Volt	
HIGH	Logic high voltage level	Volt	
TRISE	Low-to-high rise time	Second	
TFALL	High-to-low fall time	Second	TRISE
BPS	Bit rate for the source	1/Second	None
UI	Duration of the unit interval. When both UI and BPS are given, UI is used.	Second	None
STEP_RESP_ NUM_UI	Number of unit intervals to allow step response to reach steady state	None	
FFE_TAP	Number of FFE taps	None	0
FFE_LOCS	Bit locations of FFE taps	None	None
FFE_WEIGHTS	FFE weights	None	None
NORMALIZE_ FFE_WEIGHTS	1=Normalize user-supplied FFE tap weights. (Sum of absolute values of all tap weights equals 1) 0=No normalization	None	0
DFE_TAP	Number of DFE taps	None	None
DFE_LOCS	Bit locations of DFE taps	None	None
DFE_WEIGHTS	DFE weights	None	None
DFE_WEIGHT_ LBS	DFE weight lower bounds	None	None
DFE_WEIGHT_ UBS	DFE weight upper bounds	None	None
DECISION_HIGH	Decision high value	Volt	
DECISION_LOW	Decision low value	Volt	None
DECISION_ THRESHOLD	Decision threshold	Volt	
DELAY	Delay offset for transition	Second	

Parameter	Description	Unit	Default
RUN_MISSING_STEP_RESPONSE	0=analysis fails when no step response file is found for an External Step Response component 1=run transient for any missing step response data.	None	None

Run VerifEye Analysis

Run the simulation:

1. From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
2. The **Message Manager** window signals success or failure.

For details on viewing results of VerifEye analysis, see VerifEye Analysis Outputs.

For details on creating and modifying reports, see Generating Reports and Post-processing.

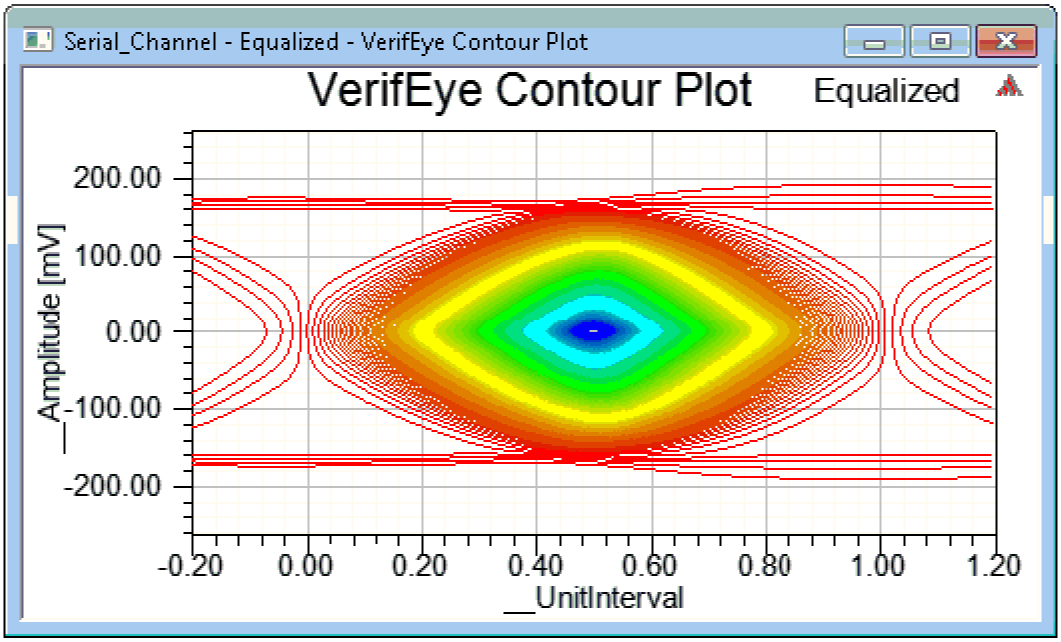
Display VerifEye Analysis Outputs

To open the results of a Quick Eye analysis, click the **Results** icon in the **Project Manager** window and select one of the **Create Report** options. There are many ways to plot the data; this topic gives examples of a Statistical (Sampled) Eye Diagram, a 2D Bathtub plot,standard and statistical Eye diagrams with PAM4 modulation, and a 2D Bathtub plot with PAM4 modulation.

For bathtub curve of QE, distinguish simulated/extrapolated regions using a partitioning horizontal line via the ‘SimulatedLimit’ option that is in available the ‘Quantity’ column. Such processing allows for the robust extrapolation of bathtub curves arising out of QE and AMI analyses.

2D Contour Plot

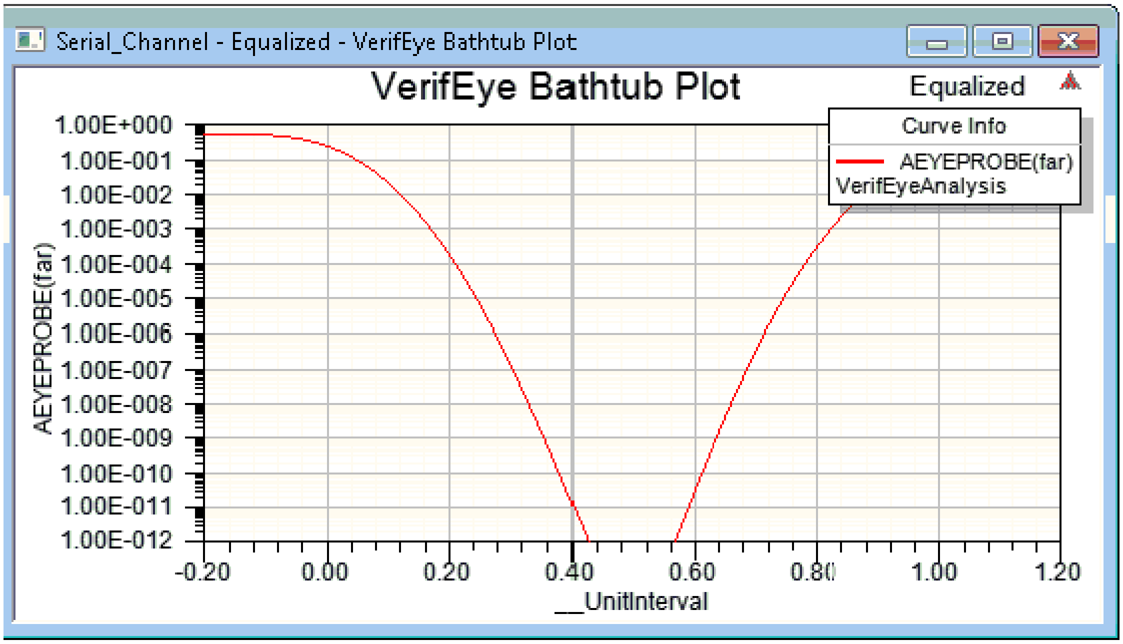
The contour plot of the bit error rate at times within the unit interval. To view the data as a contour plot of the bit error rate, select **Create Standard Report** > **Rectangular Contour Plot**. From the Report window, select **UI** as the Domain and **Eye** as the Category. Click **New Report**.



The contours for AEYEPROBE represent the bit error rates using logarithmic scaling. Amplitude represents the voltage transitions and UnitInterval is the location of the data relative to the unit interval.

2D Bathtub Plot

To view the VerifEye data as a 2D bathtub curve, select **Create Standard Report** > **Rectangular Plot**. From the Report window, select **UI** as the Domain and **Bathtub** as the Category. Click **New Report**.



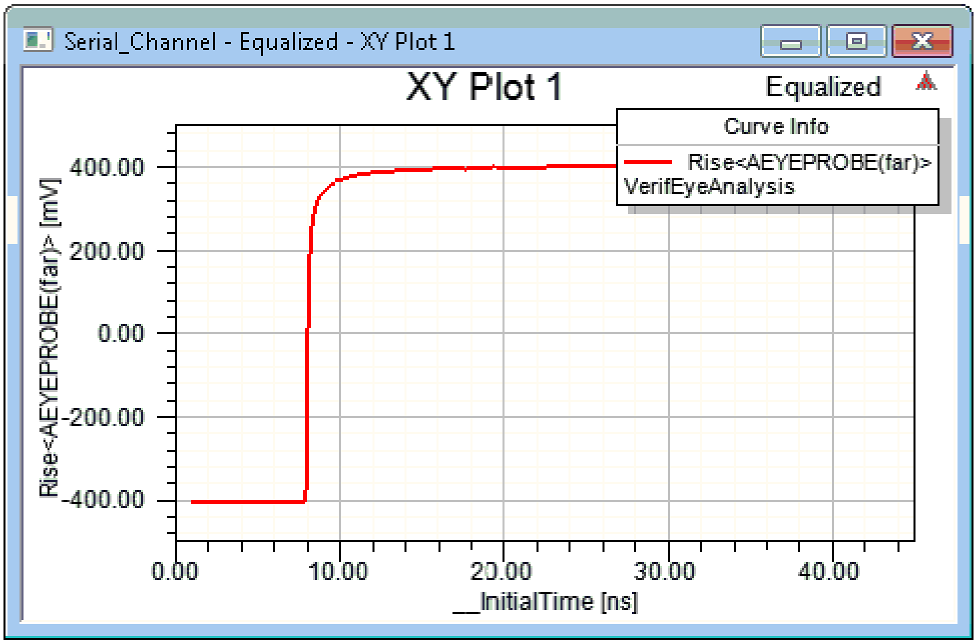
This graph shows the BER at various locations in the unit interval. The amplitude is set at the midpoint of its value range. The BER values are on the Y-axis using logarithmic scaling and a minimum of 1e-16 ~ e-21.

2D Step Response Plot

To open the initial step response of the channel as calculated by VerifEye:

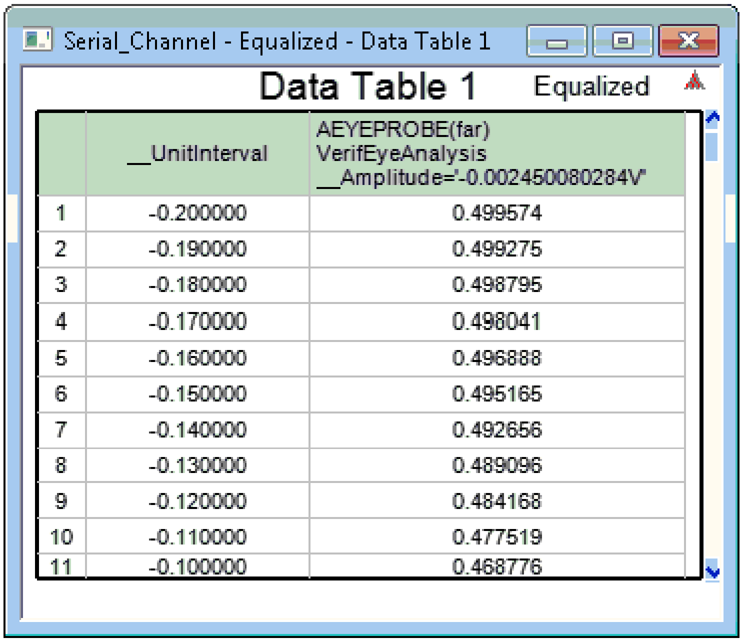
- Under Reports, select **Create Standard Report** > **Rectangular Plot**.
- Click the **Domain** drop-down menu to select **Initial Response**. Click **New Report**. The

report shows the initial response of the channel in the time domain



VerifEye Data Table

To open the VerifEye data as a table, select **Create Standard Report** > **Data Table** on the Results menu. Select **Bathtub** as the Category, the name of the Eye probe as the Quantity, and <None> as the Function. Click **New Report**.



Serial_Channel - Equalized - Data Table 1

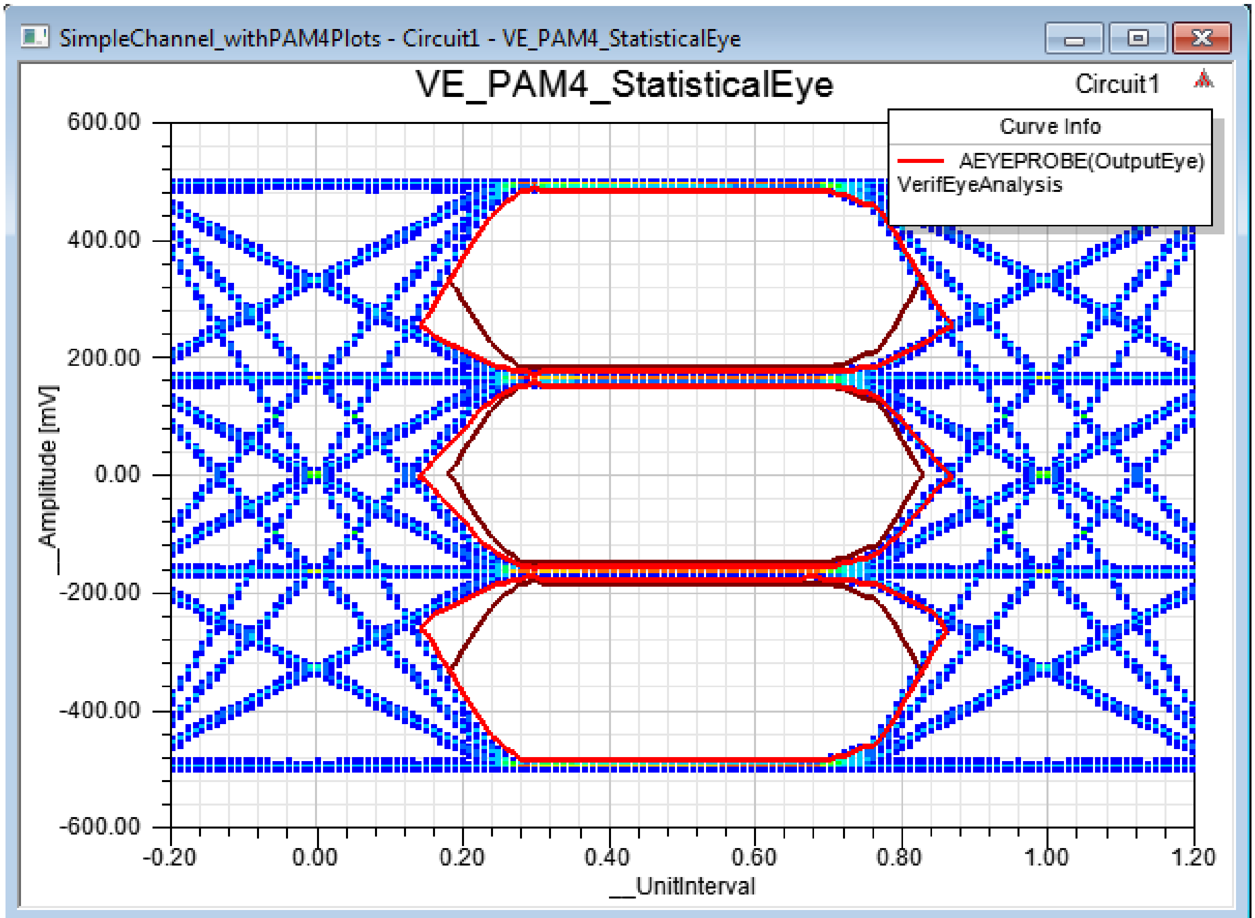
Data Table 1 Equalized

	__UnitInterval	AEYEPROBE(far) VerifEyeAnalysis __Amplitude='-0.002450080284V'
1	-0.200000	0.499574
2	-0.190000	0.499275
3	-0.180000	0.498795
4	-0.170000	0.498041
5	-0.160000	0.496888
6	-0.150000	0.495165
7	-0.140000	0.492656
8	-0.130000	0.489096
9	-0.120000	0.484168
10	-0.110000	0.477519
11	-0.100000	0.468776

2D PAM4 Statistical Eye Diagrams

To enable PAM4 modulation, set the **modulation** parameter in the Eye source to **PAM4**. Select **Gray** or **Linear** for the **coding** parameter as applicable. When analysis completes, select **Create Statistical Eye Diagram** > **Rectangular Plot**.

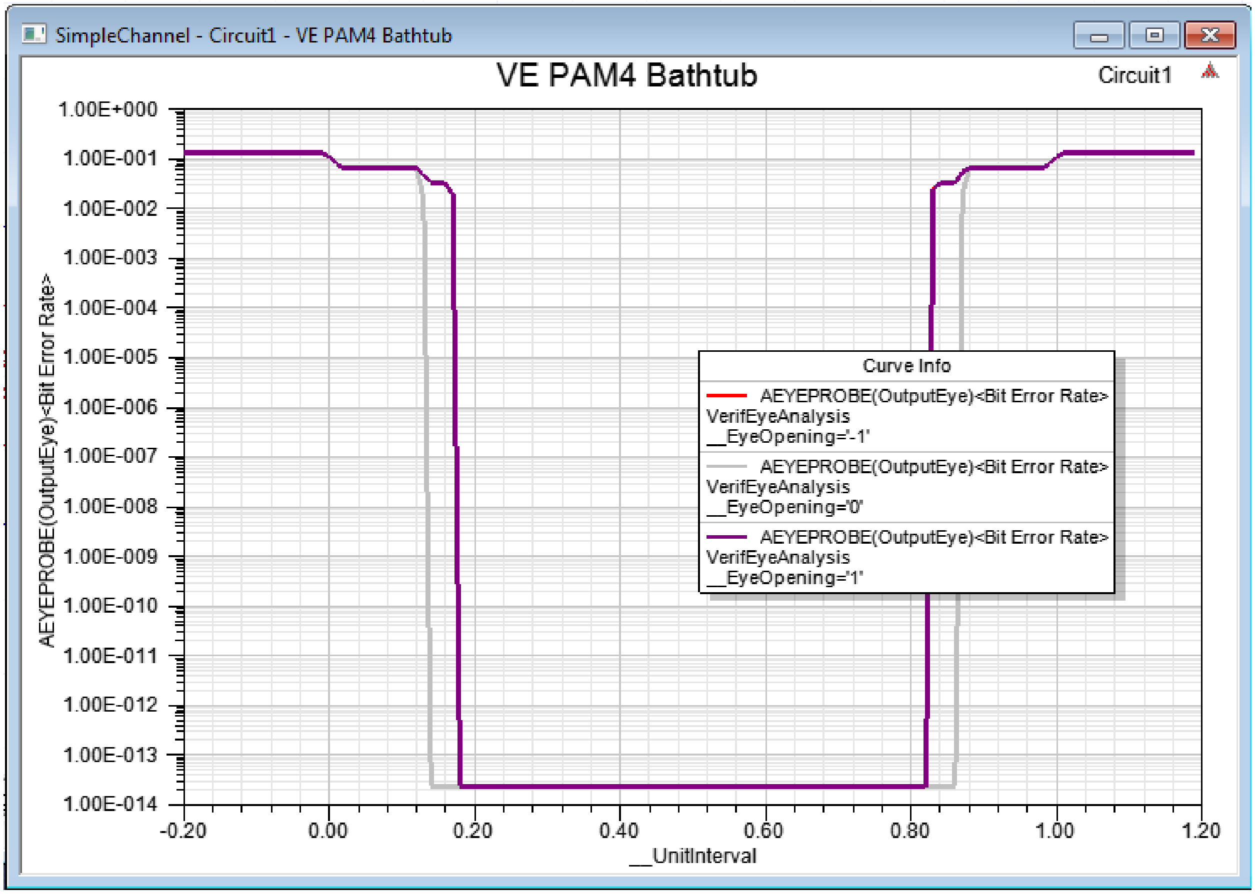
The plot shows four voltage levels, each one corresponding to a transition between two bits. This example uses the default Gray coding: the lowest level represents 00, the next higher level represents 01, the next higher level represents 11, and the highest level represents 10. The four voltage levels form three eyes in the diagram.



The red lines show the eye openings. The dark lines show the intersection of all three eye openings.

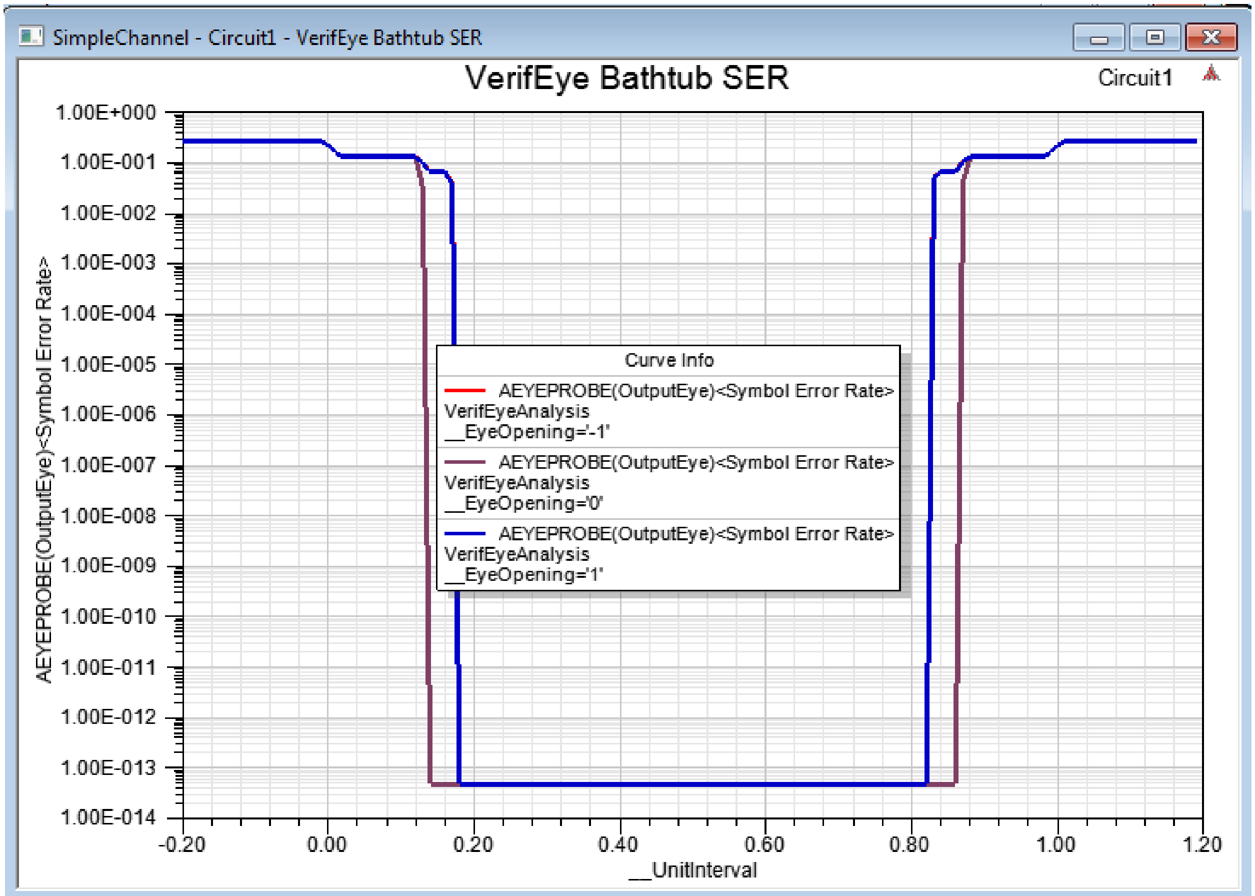
2D PAM4 Bathtub Plots

To enable PAM4 modulation, set the **modulation** parameter in the Eye source to **PAM4**. Select **Gray** or **Linear** for the **coding** parameter as applicable. When analysis completes, select **CreateStandard Report** > **Rectangular Plot**. From the Report setup, click the Families tab and edit the _EyeOpening family. The entries are -101 (combined error rate), -1 (upper eye error rate), 0 (middle eye error rate) and 1 (lower eye error rate). From the Report window, select **UI** as the Domain and **Bathtub** as the Category. Select the **<Bit Error Rate>** trace to see the bit error rate.



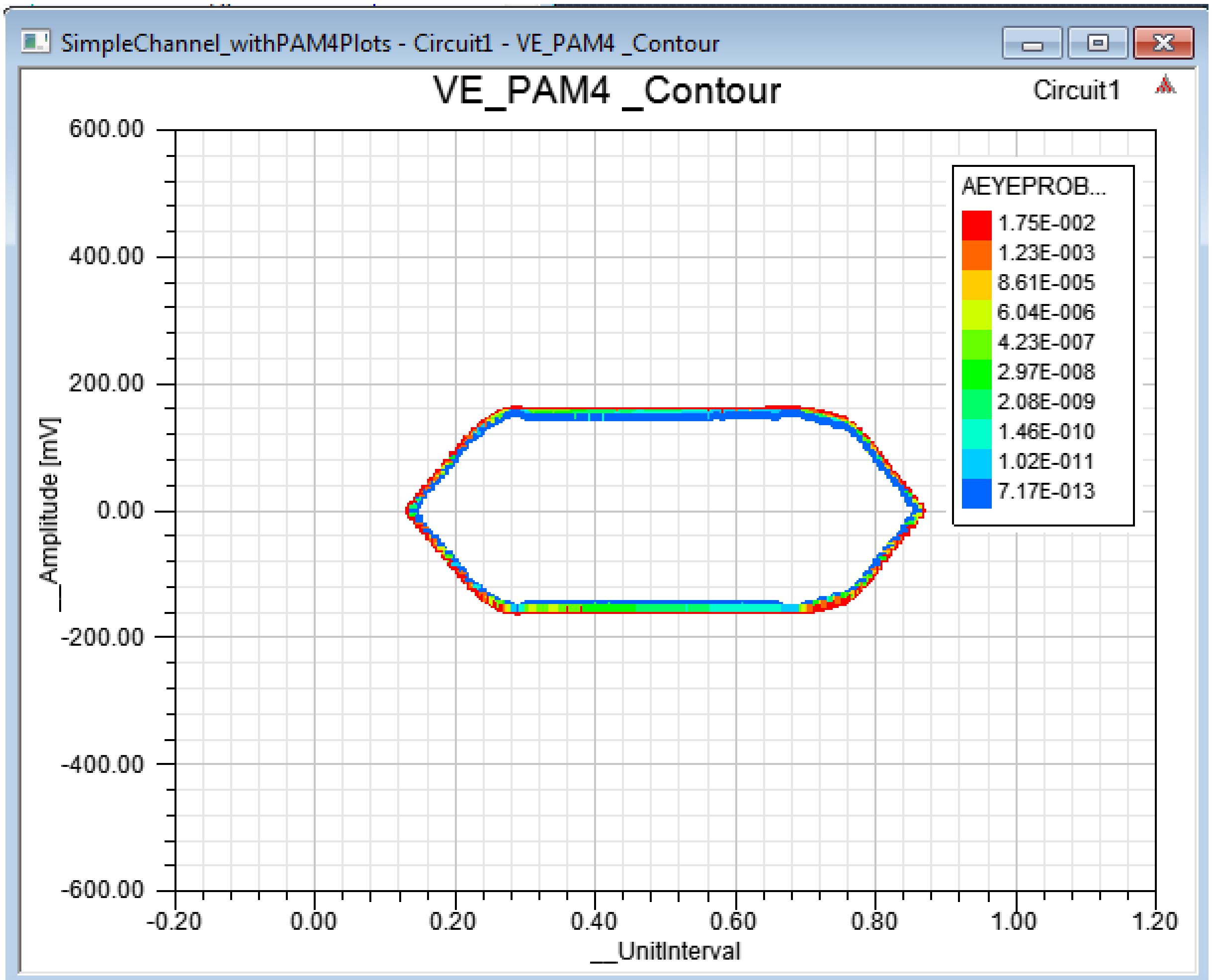
The example plot shows the separate bit error rates for the three PAM4 eyes (-1, 0, and 1 selected).

The bathtub curve for a PAM4 simulation can show the symbol error rate instead of the bit error rate. With PAM4, a symbol is two bits. The report setup is the same, but now select the **<Symbol Error Rate>** trace.



2D PAM4 Contour Plots

To enable PAM4 modulation, set the **modulation** parameter in the Eye source to **PAM4**. Select **Gray** or **Linear** for the **coding** parameter as applicable. When analysis completes, select **CreateStandard Report** > **Rectangular Contour Plot**. From the Report setup, click the Families tab and edit the _EyeOpening family. The entries are -101 (combined error rate), -1 (upper eye error rate), 0 (middle eye error rate) and 1 (lower eye error rate).



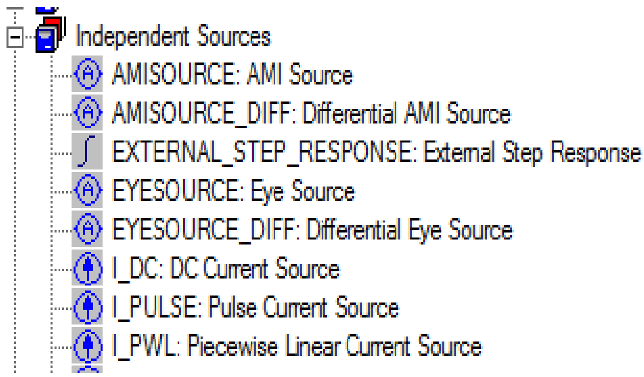
The example plot shows the separate bit error rates for the three PAM4 eyes (-1, 0, and 1 selected).

Running Quick Eye Analysis on the Schematic Editor

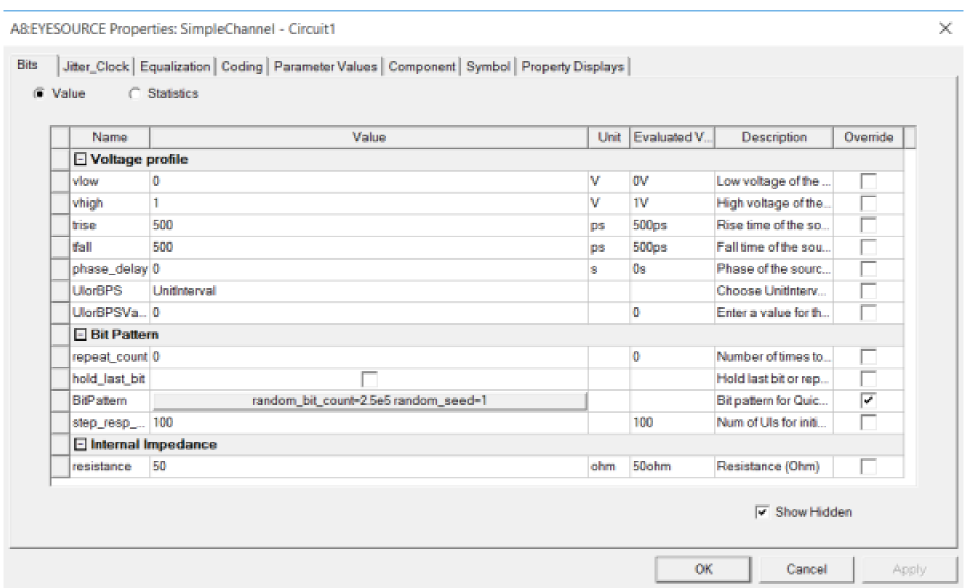
The eye diagram is a convenient way to analyze the performance of a serial communications channel. In a traditional eye diagram, copies of the waveform generated by transient analysis are overlaid at a spacing of one unit interval (UI). To run Quick Eye analysis, the design must include one or more Eye Sources and corresponding Eye Probes. Transmitter and receiver parameters including equalization are set up in the sources and probes. Other parameters can be specified in the setup window.

Add a QuickEye Source to a Schematic

The eye diagram is a convenient way to analyze the performance of a serial communications channel. In a traditional eye diagram, copies of the waveform generated by transient analysis are overlaid at a spacing of one unit interval (UI). The Eye Source may be used as the transmitter in QuickEye, VerifEye, and Transient analyses. The Eye Source may be selected on the Independent Sources list on the Components tab.



Both single-ended and differential versions are available. In a schematic, double-click the Eye Source, or right-click the Eye Source and select **Properties**, to open the Properties window.

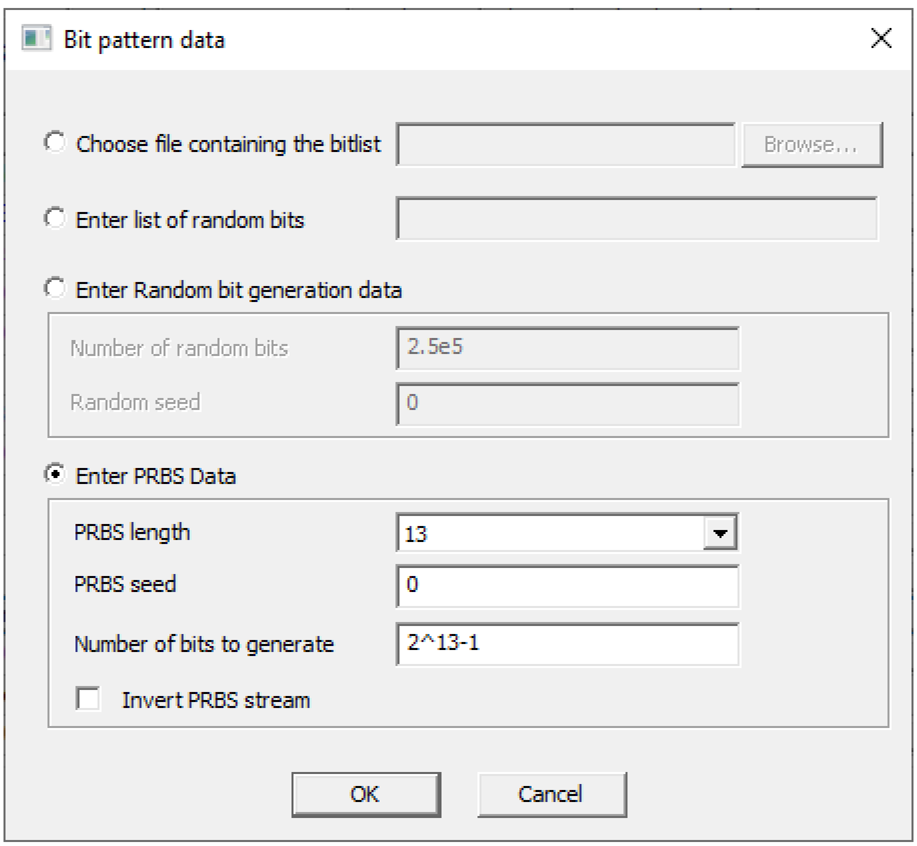


- For Quick Eye analysis, all Eye Source parameters apply except spread spectrum clocking.
- For VerifEye analysis, all parameters apply except the bit pattern generators and spread spectrum clocking.
- For Transient analysis, all parameters apply.

Bits Tab

The **Properties** window opens with the Bits tab displayed.

- The logic low, **vlow**. The default is 0 volt.
- The logic high, **vhigh**. The default is 1 volt.
- The rise time of input (**trise**). The default is 5.0e-10 seconds (500 picoseconds).
- The fall time of input (**tfall**). The default is 5.0e-10 seconds (500 picoseconds).
- The **phase_delay** for this source. The default is 0 seconds.
- The data rate: Click the **UIorBPS** drop-down to select **UnitInterval** or **BitsPerSecond**. Then Click the **UIorBPS Value** field to enter the unit interval size or number of bits per second.
- Click **Bit Pattern** to open the window.



- Select ONE of the methods for generating the QuickEye or Transient bit pattern.
- To link to a file with the bit values, click **Choose file**, then Click the **Browse** window to locate and select the data file.

- To specify a bit list, click **Enter list** and type in the list of 1s and 0s.
- To have the Eye source generate a random sequence of bits, click **Enter random**, then enter the **Number of random bits to be generated** and the **Random seed** value. To guarantee that multiple Eye sources always use different random seeds, set the **Random seed** to **-1** or **-2**. All Eye sources with setting **-1** get sequential random seeds starting with 1000, the same on every run. All Eye sources with setting **-2** get random seeds derived on the clock, changing from run to run.
- To generate a pseudo-random sequence of bit patterns of various lengths, click **Enter PRBS Data**. Select the length of the pattern, **PRBS_length**, (2 to 31 bits) on the drop-down menu. Enter a PRBS seed value, and specify the total number of bits to be generated. To invert the PRBS bit stream, check the check box at the lower-left of the PRBS Data group box.

Click **OK** to close the **Bit pattern** window and return to the Eye Source parameter list.

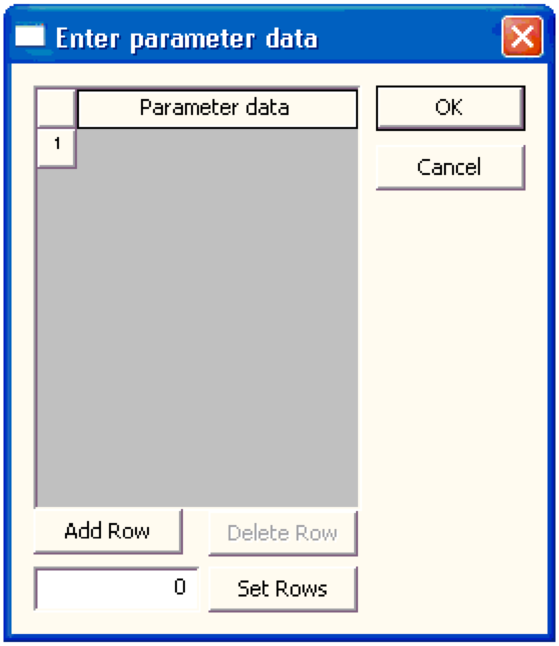
- To specify one or more repeats of the bits in the bit list, bit file, random sequence, or PRBS, set the **repeat_count** parameter on the Eye Source to the number of repeats. For more information, see Eye Source Bit Data in the Technical Notes.
- The **Hold Last Bit** check box is for sources with different bit sequences. The shorter bit list can be repeated or the last bit value can be held, until simulation ends. The default is to repeat (check box not checked).

Jitter_Clock Tab

The Jitter_Clock tab contains parameters to set up DCD, transmit jitter, and Spread Spectrum clocking.

- Duty cycle distortion: Click the **DCDFractionorTime** dropdown menu to select **Fraction** (the default) or **Time**. For **Fraction**, the DCD value is a decimal fraction of the UI between 0.0 and 1.0. For **Time**, the DCD value is a time between 0.0 and UI in seconds. A time unit such as ps may be used to set the DCD time. The default for both Fraction and Time is 0.
- For more information, see *Duty Cycle Distortion* in the QE/VE Technical Notes.

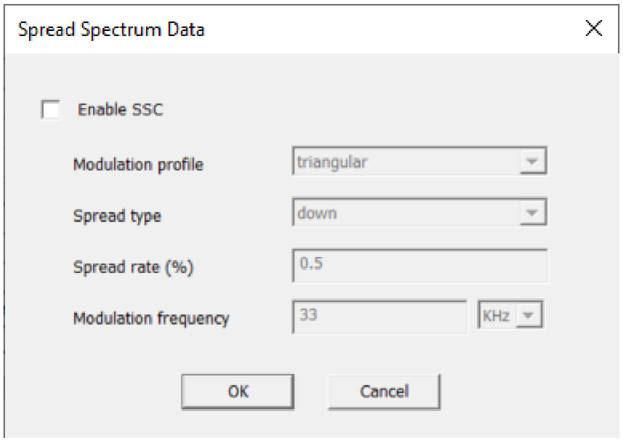
For **txrj**, **txpj**, and **txuj**, clicking in the **Value** field allows you to specify one or more values that allow you to configure the generation of multiple jitter sources.



- For the random transmit jitter (**txrj**) the value is the standard deviation for the Gaussian distribution. The default is 0 seconds.
- For the Periodic random transmit jitter (**txpj**) the value is the amplitude. The default is 0 seconds.
- For the Uniform random transmit jitter (**txuj**) the value is the amplitude. The default is 0 seconds.
- For the User-defined transmit jitter (**txcj**) the value is the name of the file containing the time (seconds and probability density function (PDF) data. There is no default.
- For more information on transmit jitter parameters, see Transmit Jitter in the Technical Notes.

With Transient Analysis, configure the Eye Source to enable the application of Spread Spectrum Clocking. (Please note that QuickEye and VerifEye do not use Spread Spectrum clocking.)

- Click the **Value** field of the **SpreadSpectrum** property to open the Spread Spectrum Data window.



- Click the **Enable SSC** check box to activate the spread spectrum clock parameter fields.
- Select the **Modulation profile**: **triangular** (the default) or **sinusoidal**.
- Select the **Spread type**: **down** (the default), **center**, or **up**.
- Select the **Spread rate (%)**, a percentage of the main clock frequency (which is calculated on the UI or BPS value in the Eye Source Bits tab). The Spread rate must be non-zero to enable Spread Spectrum clocking. The default is 0.5%.
- Set the **Modulation frequency** to a non-zero value to enable Spread Spectrum clocking. The default is 33kHz.

Coding Tab

The Coding tab contains parameters to set up bit encoding and select NRZ or PAM4 modulation.

- The **do_encoding** parameter controls 8b10b, 64b66b, 128b130b, or 128b132b encoding of the transmitted bit stream. The default is no encoding (None). Select 8b10b encoding, 64b66b encoding, 128b130b encoding, or 128b132b encoding on the drop-down menu. Please note that the **do_encoding** option is valid only when **modulation=NRZ**. **do_encoding** is ignored when **modulation=PAM4**.
- The **modulation** to be used. The default **NRZ** uses two voltage levels (binary data), **PAM4** uses four voltage levels to represent two-bit symbols. See **Coding**.
- In the window, the **coding** parameter is labeled **coding_PAM4_only** as a reminder that it is ignored for NRZ transmissions. The **coding** parameter defines how digital bits are sent as analog waveforms in PAM-4 transmissions. In PAM-4, two bits ("00"/"01"/ "10"/"11") are transmitted at a time; in Gray coding, "00" is transmitted as "vlow," "01" as "vlow + (vhigh-vlow)/3," "11" as (vhigh - (vhigh-vlow)/3), "10" as "vhigh". In Linear coding, these voltage levels (smallest to largest) are used to transmit bits "00," "01,"10," and "11," respectively. The **coding** parameter has no effect in NRZ transmissions. In NRZ, one bit ('0' or '1') is transmitted at a time; '0' is transmitted as "vlow," and 1 as "vhigh."

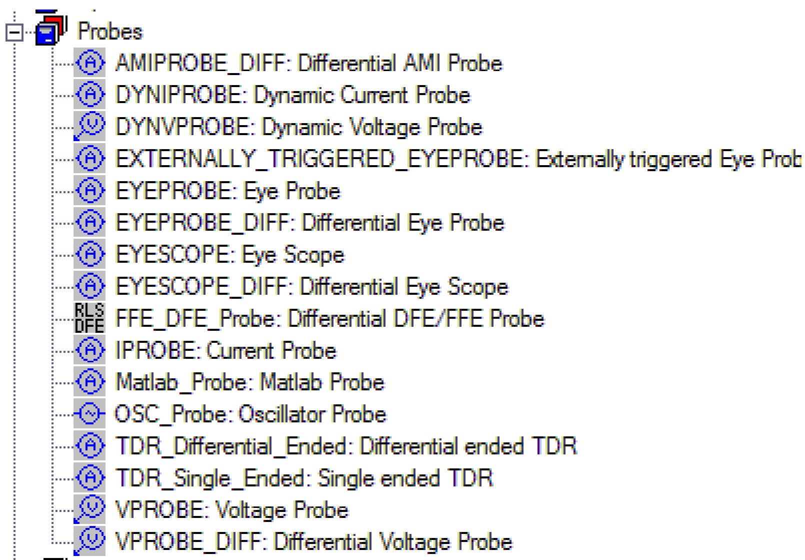
Equalization Tab

The Equalization tab allows you to set up Feed-Forward Equalization in the Eye Source. To apply Feed-Forward Equalization, click **FFE_data** which opens the **FFE Data** window.

Add an Eye Probe to a Schematic

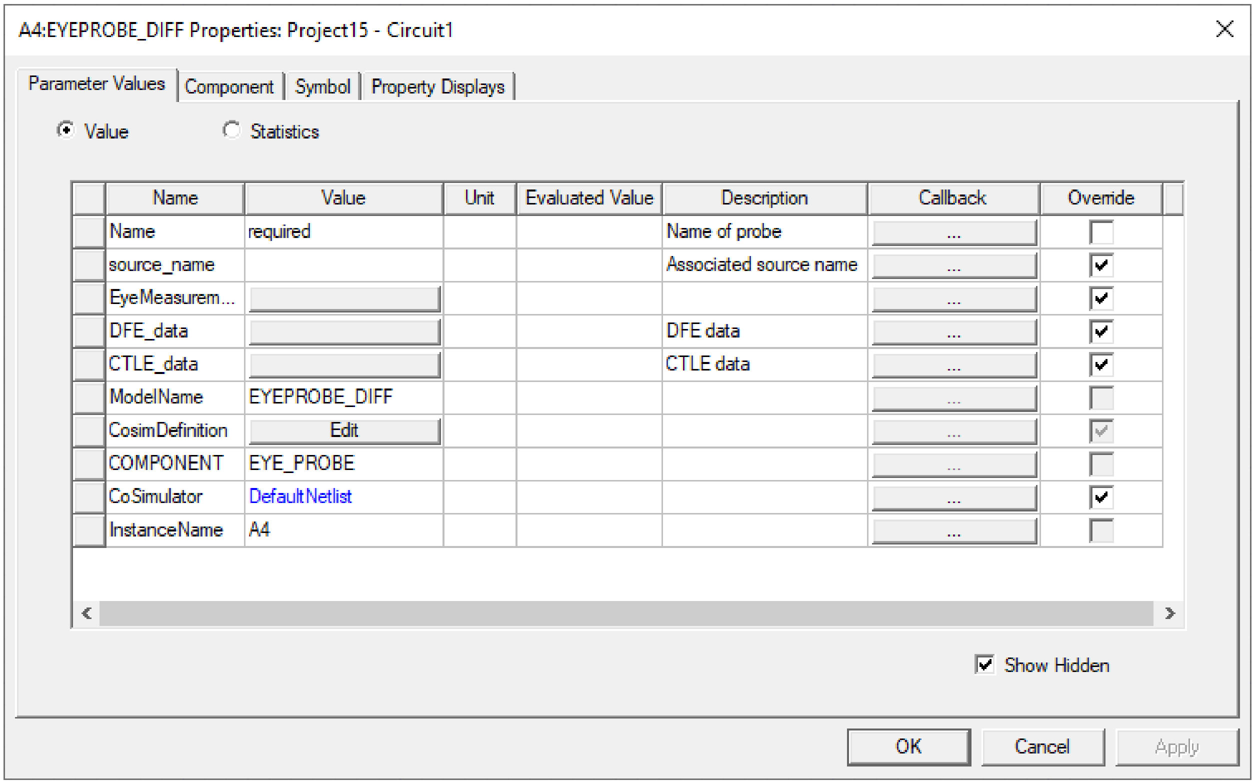
An Eye Probe is required when the QuickEye, VerifEye, or Transient analysis uses an Eye Source as the transmit device. All QuickEye and some Transient analyses use one or more Eye Sources and Eye Probes. Each Eye Probe receives data from a specific Eye Source, allows for Decision-feedback and Continuous-time equalization at the receiver end, and preserves all the transient analysis data generated during the analysis.

The Eye Probe may be selected on the Probes list on the Components tab:



Both single-ended and differential versions are available.

Click the Eye Probe to open its parameter list:

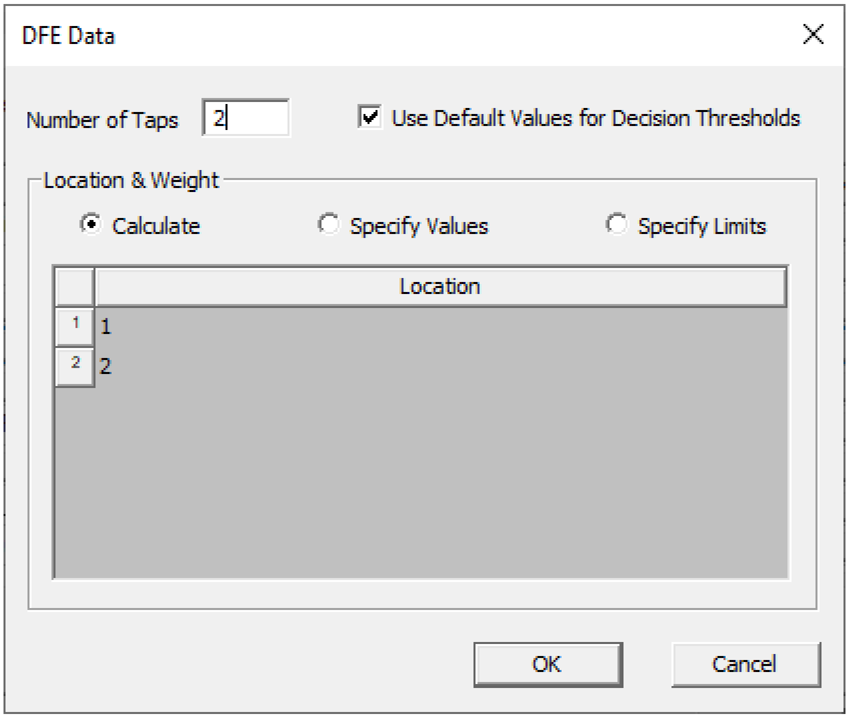


The **Parameter Values** tab is the default.

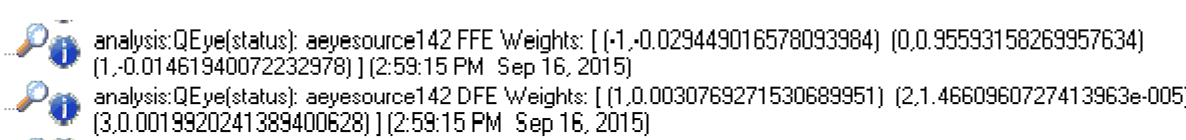
- Specify a name for the Eye Probe.
- Select the Eye Source for this probe on the **source_name** drop-down menu. A channel can have multiple sources, but only the source actually at the transmitter should be the reference source.

The **source_name** field is automatically filled in with the name of the Eye Source when the Source and Probe are paired on the GUI. See Pairing an Eye Source and an Eye Probe in the Schematic.

- To apply decision-feedback equalization to the received bits, click **DFE_data** to open the window:

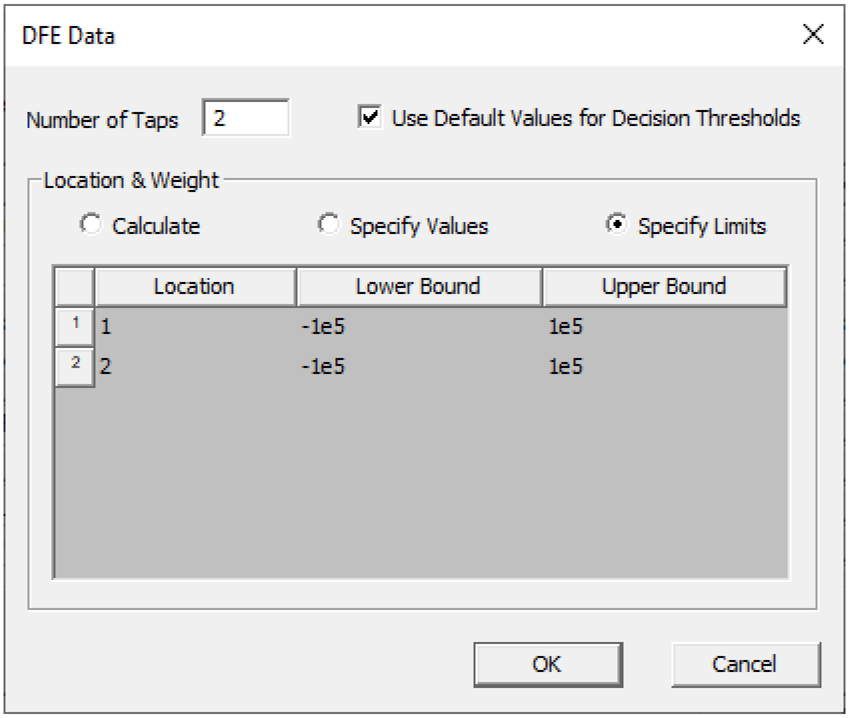


- To enable Decision-Feedback Equalization at the probe, set the **Number of taps** in the **DFE** field to a positive, non-zero value. The default is zero taps (no DFE). When one or more taps have been specified, a list of tap numbers and locations appears.
- DFE applies its tap weights to the decision threshold for each bit to correct for inter-symbol interference (ISI) in the transition from previous bits to the current bit. A DFE with M taps is like a buffer of M bits. Tap 1 is applied to the earliest bit to arrive in the DFE buffer. Tap 2 is applied to the next later bit, et cetera.
- The value of each bit is decoded, and the difference between the actual voltage and the ideal voltage is calculated. Tap weights compensate for the difference. In QuickEye, the weights are applied to the step response.
- By default, the Circuit solver calculates the weights to apply to the DFE taps. The tap weights calculated by QuickEye are displayed in the **Message Manager** window after the Quick Eye analysis has completed.

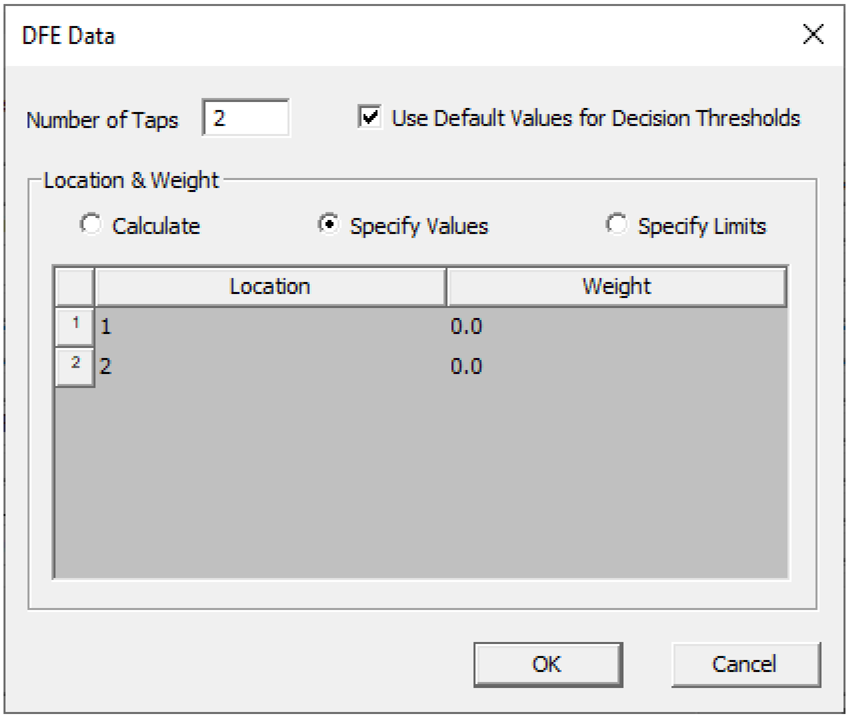


- To automatically calculate the tap weights, choose **Specify Limits** and set the limits. The tap weights are calculated in the bounds you specify. The default value for bounds are +/-

1e5 for taps. There are no requirements on the limits.



- To specify your own set of tap weights, select **Specify Values** and click the weights to specify them.

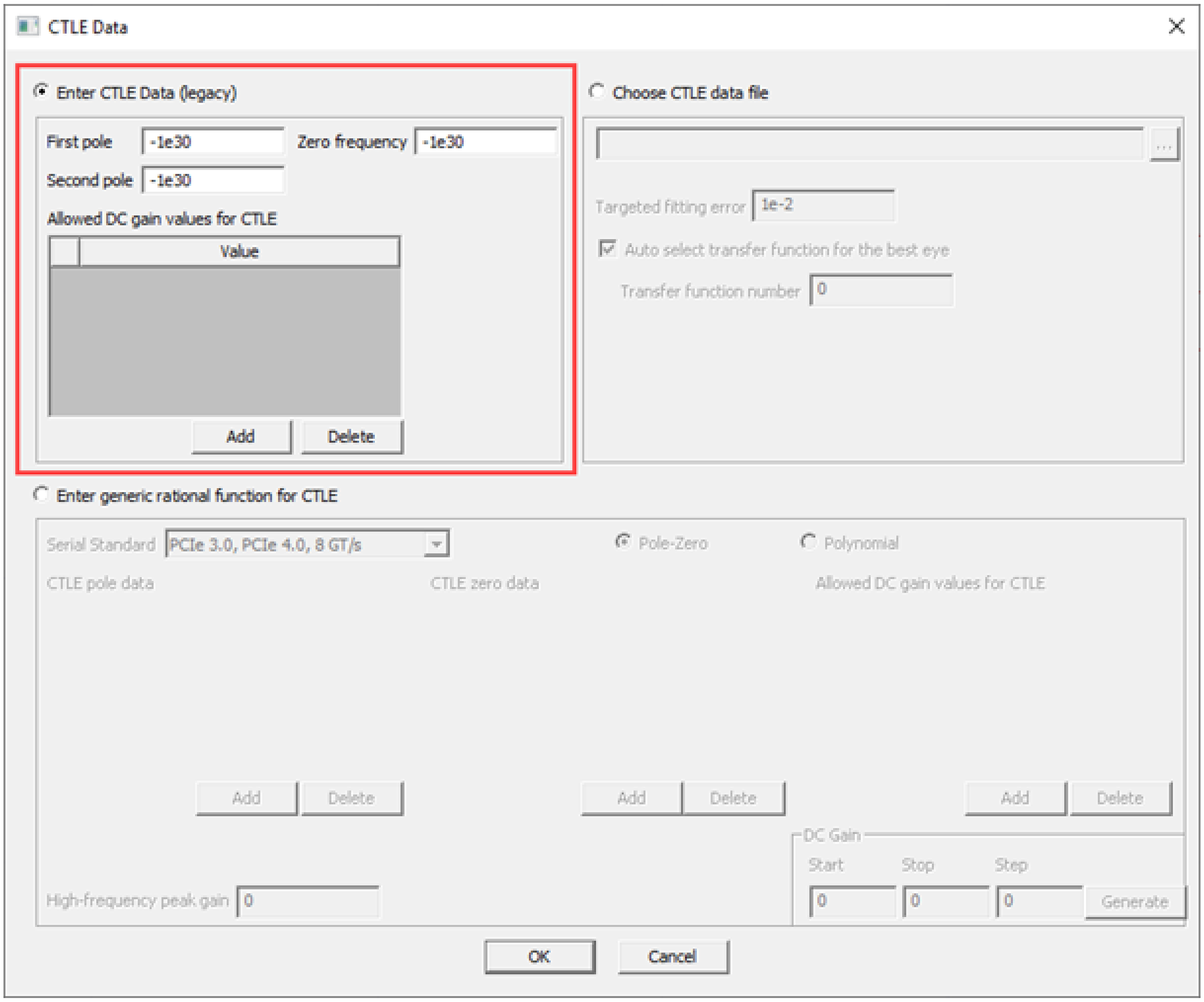


- To decode the received bits, mark **Use Default Values for Decision Thresholds**. For NRZ transmissions, you enter a single voltage threshold. For PAM4, you enter three voltage thresholds, one for each eye. The default is to have Nexxim calculate the thresholds.
- For more information on DFE, see Decision-Feedback Equalization.
- Click **OK** to close the **DFE_data** window and return to the Properties window.

Click **CTLE_data** to open the **CTLE Data** window. (See Continuous Time Linear Equalization in the QE/VE Technical Notes for details).

- Select **Enter CTLE Data (legacy)**, **Choose CTLE data file**, or **Enter generic rational function for CTLE**.

If you select **Enter CTLE Data**, only the CTLE data group box is activated:

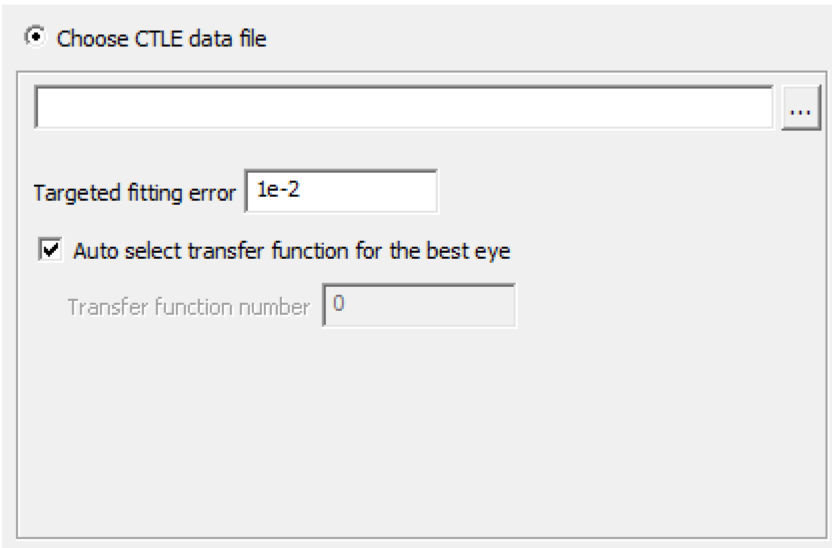


- Specify the first pole frequency in Hz (positive). The first pole frequency typically is the lower bound of the high passband.
- Optionally, specify a second pole frequency. The second pole frequency typically is the higher bound of the passband.
- Optionally, specify a zero frequency, typically where the response curve begins to rise.

> **Note:**
> 1. The default pole and zero frequency values of -1e30 are equivalent to "None."
> 2. If the zero frequency is specified, it must be greater than 0.0.
> 3. See Continuous Time Linear Equalization in the QE/VE Technical Notes for an illustration of the two-pole, one zero filter.

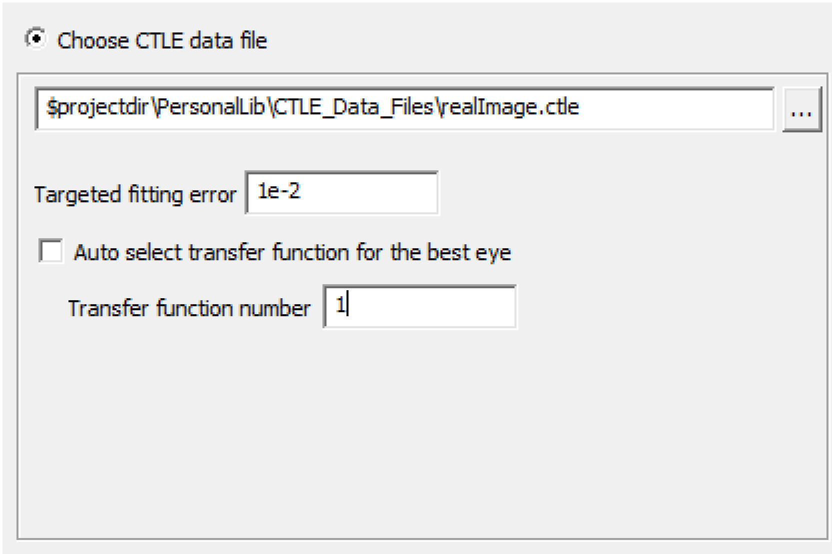
- Optionally, click **Add Row** and **Delete Row** to specify one or more allowable values for the DC gain factor in dB for the CTLE calculation. Nexxim varies the gain to find the optimum equalization (maximum eye height). If no values are provided, the Circuit solver calculates the optimum CTLE gain (between -infinity dB to 0 dB).

If you select **Choose CTLE Data File**, only the CTLE data file group box is activated. Here is that portion of the window:



- Use **Browse** to access the **File Open** window. Browse to the target directory and select the file with the CTLE transfer function data (**.ctle** extension). Nexxim fits a rational function to the data.
- Optionally, Click the **Targeted fitting error** field to specify a tolerance to apply to the CTLE rational function calculation using the transfer function data.
- By default, **Auto select transfer function for the best eye** is enabled. Nexxim tries all the transfer functions on the file, and uses the one that yields the maximum eye height. Optionally, uncheck the **Auto select** group box, then enter the number of a transfer function among the multiple transfer functions in the data file.

In the following figure, transfer function 1 is specified.

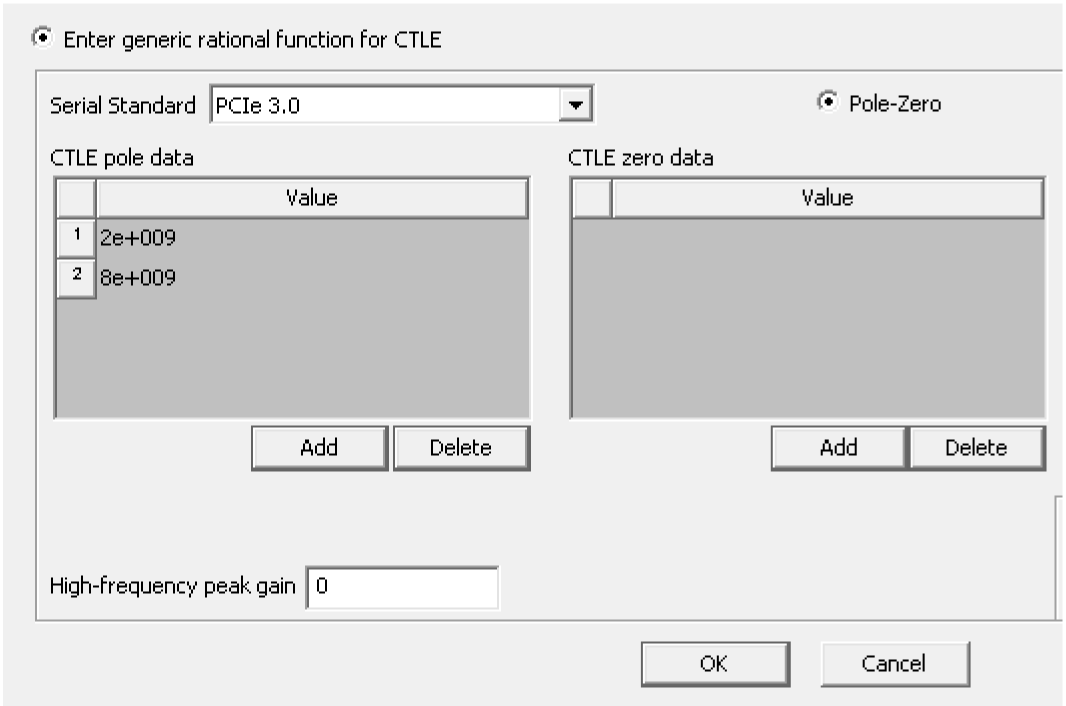


- Values for the transfer function number run from 0 (the **Auto select** default) to the number of transfer functions in the file. The first transfer function is number 1, the second is number 2, et cetera. If the number given for this parameter is less than 0 or greater than the number of transfer functions in the file, an error occurs.

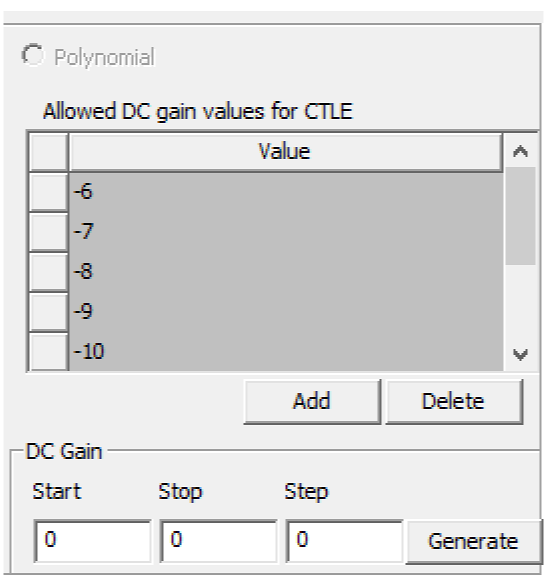
When you select **Enter generic rational function for CTLE**, only that group box is activated.

There are two formats for specifying a generic rational function, chosen by selecting **Pole-Zero** or **Polynomial**. By default, the **Pole-Zero** format is selected. With this format, the rational function is specified by a set of pole frequencies, a set of zero frequencies, the high-frequency peak gain, and a list of allowable DC gain values.

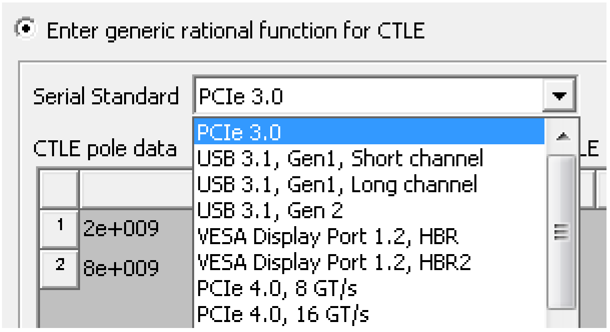
Here are the entry fields for the pole and zero frequencies and the High frequency peak gain:



Here is the entry field for the DC gain values:



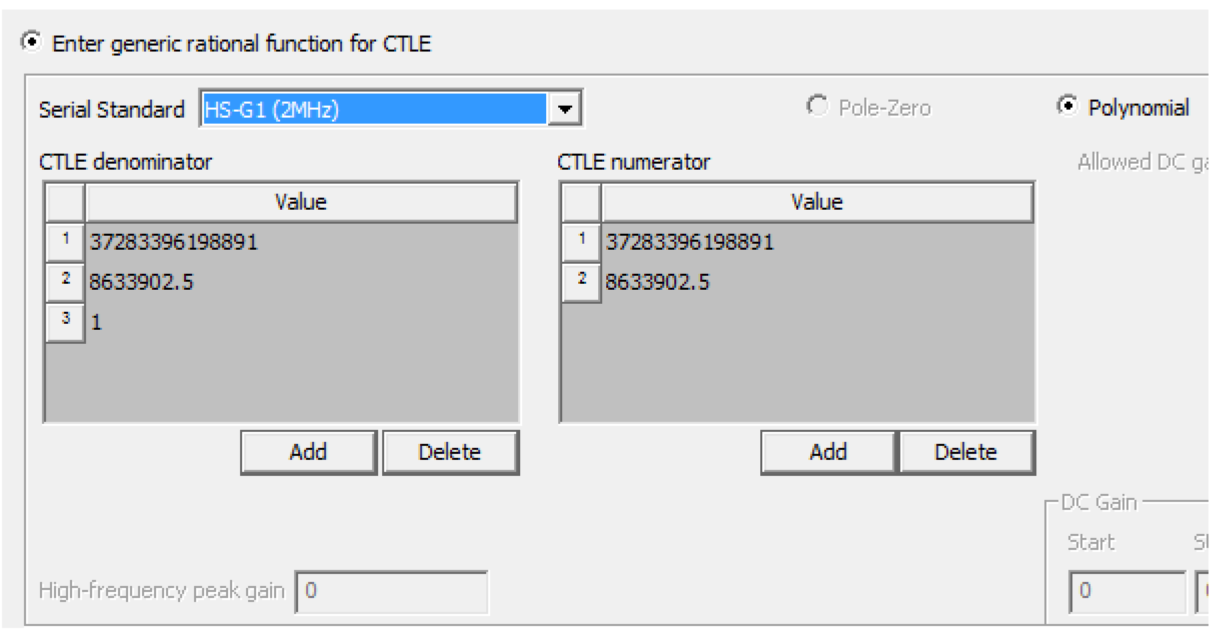
The **Serial Standard** field has a drop-down menu with a selection of preset standard CTLEs. The PCIe, USB, and VESA preset CTLEs use only the **Pole-Zero** format.



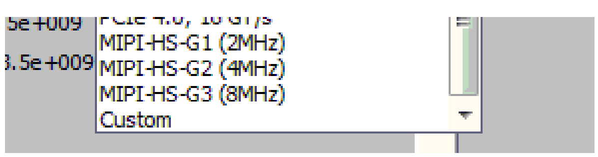
> **Note:** This illustration may not include later additions to the preset standards.

Alternatively, select **Polynomial** format. In this format, the rational function is specified by giving the coefficients for the terms in the numerator and denominator of the polynomial fraction. The **High-frequency peak gain** and **Allowed DC gain** fields are inactivated in the **Polynomial** format.

Here are the entry fields for the **Polynomial** format coefficients:



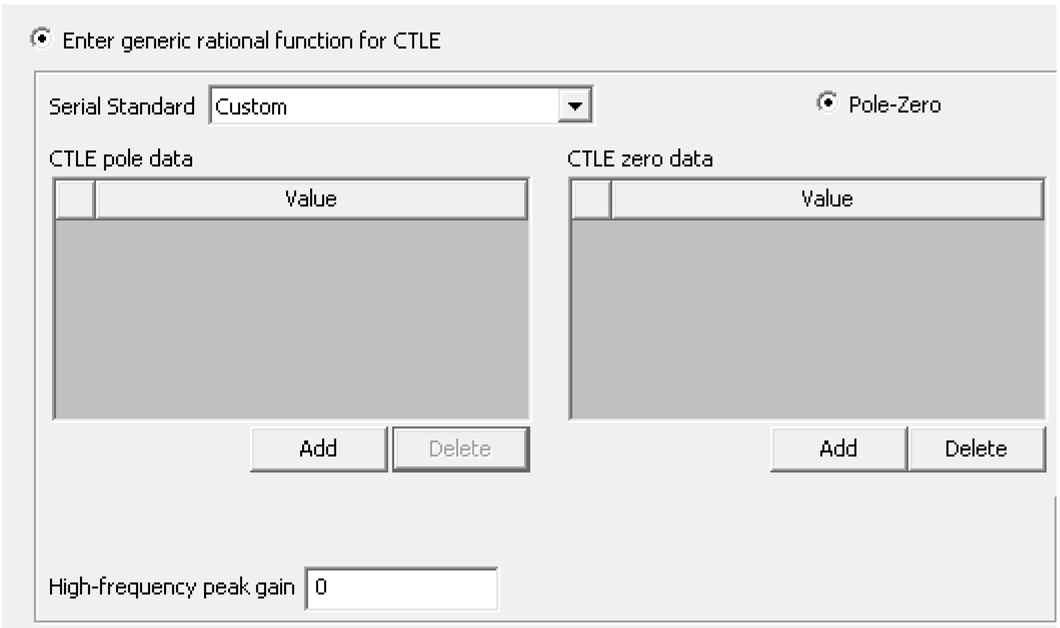
The MIPI-HS-G1, MIPI-HS-G2, and MIPI-HS-G3 CTLE presets use only the **Polynomial** format.



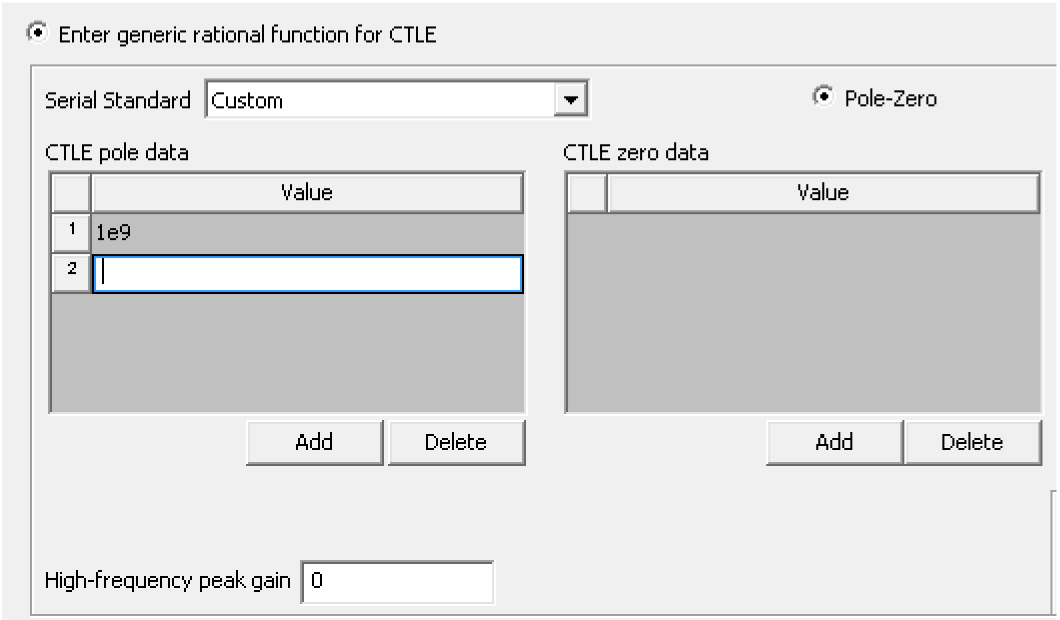
> **Note:** This illustration may not include later additions to the preset standards.

To create your own rational function, select **Custom** on the menu. The **Custom** entry allows you to specify a generic rational transfer function in either **Pole-Zero** format or **Polynomial** format.

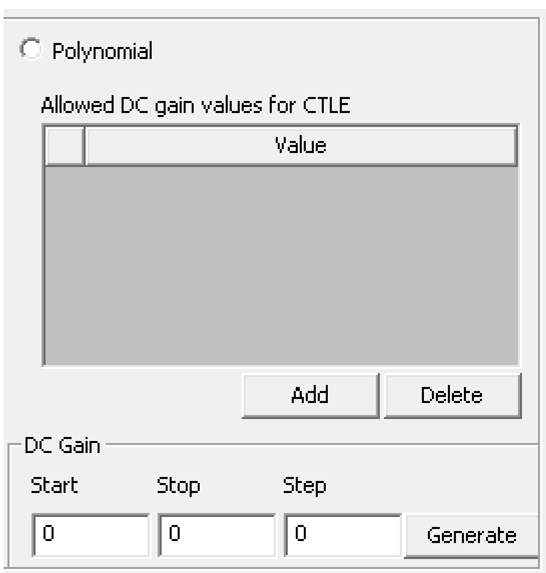
Here are the **CTLE pole data**, **CTLE zero data**, and **High-frequency gain** entry fields for the **Custom** entry in **Pole-Zero** format:



Click **Add** to add a row in the pole or zero data list. Click in the **Value** field to enter the pole or zero frequency in Hz. Click **Return** to generate a new row. Click **Delete** in either field to delete unwanted rows. Set the **High-frequency peak gain** to a appropriate value in dB.

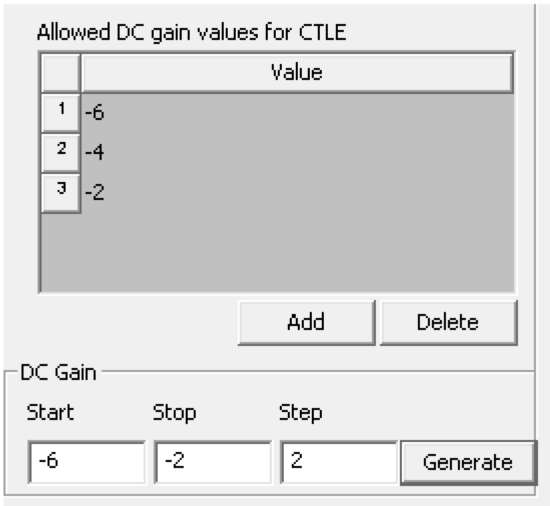


Use the **Allowed DC gain** group box to specify a range of DC gain values in dB.

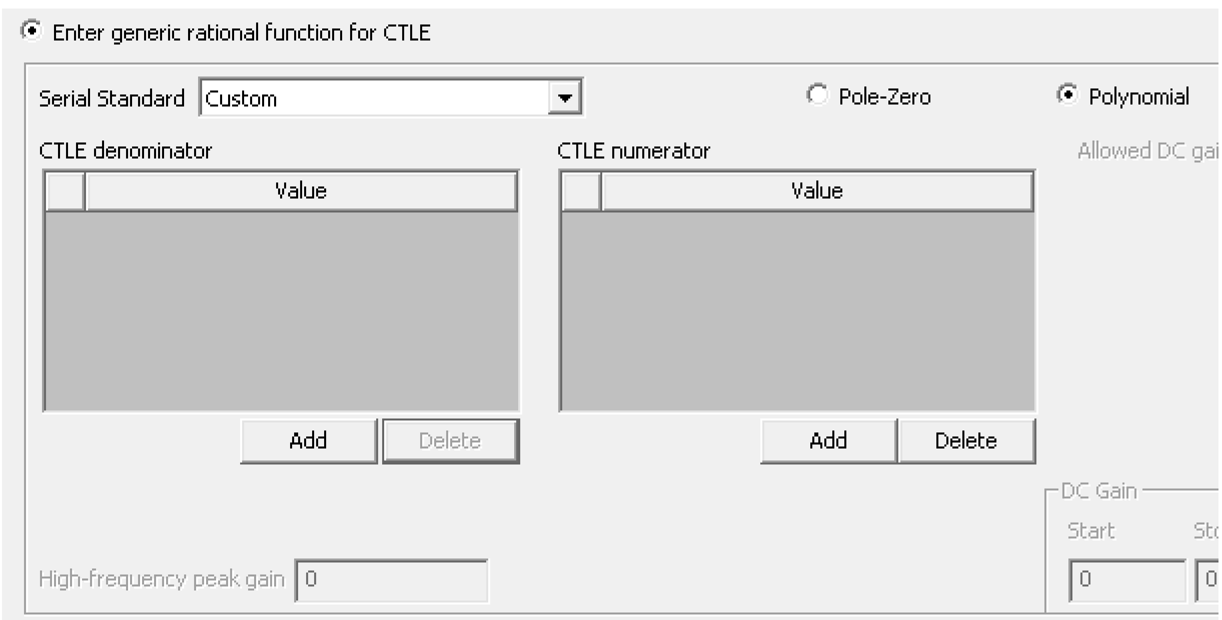


Click **Add** and **Delete** to specify one or more allowable values for the DC gain factor in dB for the CTLE calculation. Nexxim varies the gain to find the optimum equalization (maximum eye height). If no values are provided, the Circuit solver calculates the optimum CTLE gain (between -infinity dB to 0 dB).

To have Nexxim generate a sequence of DC values, enter the **Start**, **Stop**, and **Step** values in the **DC Gain** fields and click **Generate**. In this window, Start=-6, Stop=-2. and Step=2:



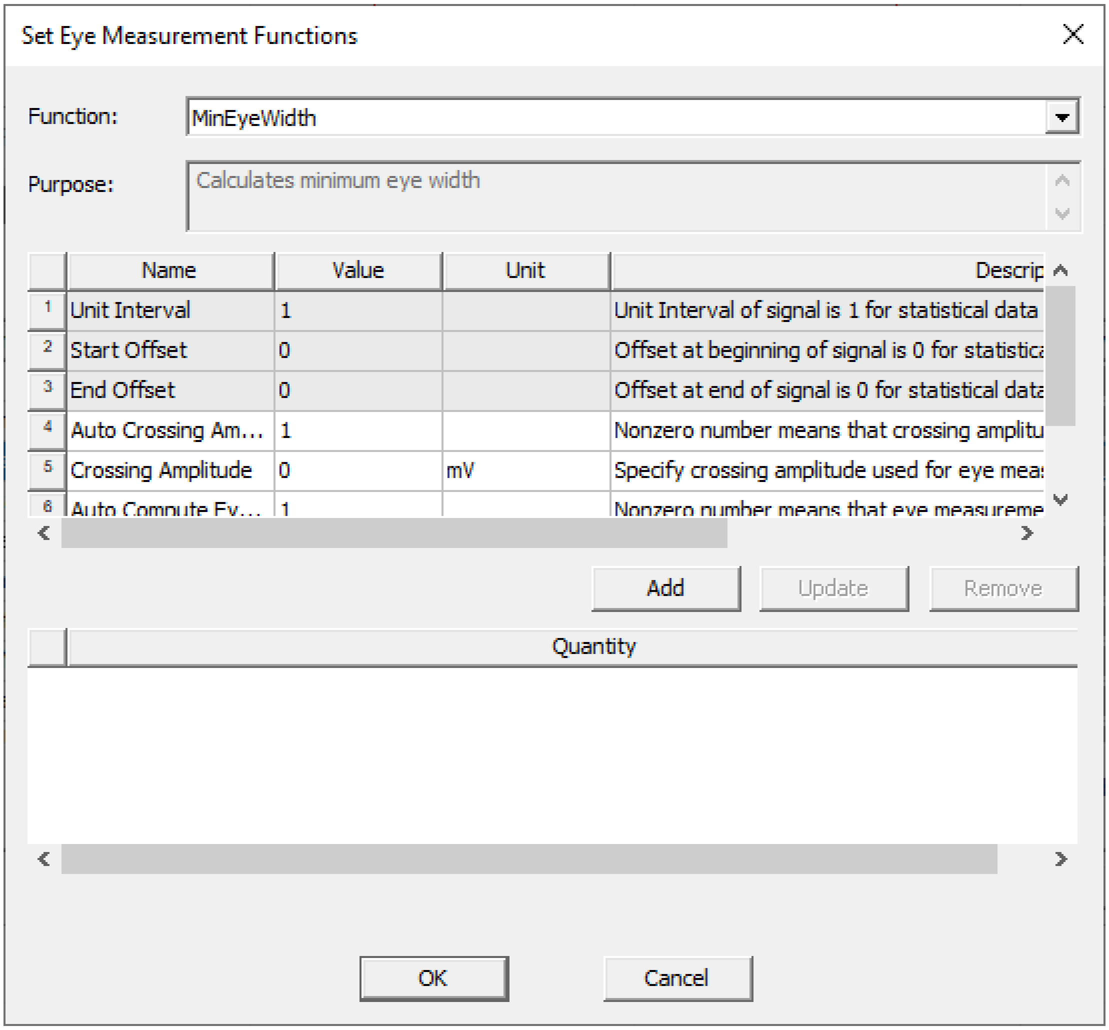
Alternatively, a **Custom** rational function CTLE can Click the **Polynomial** format.



Click **Add** to add a row in the CTLE denominator or CTLE numerator list. Click in the **Value** field to enter the (unitless) coefficient. Click **Return** to generate a new row. Click **Delete** in either field to delete unwanted rows. Click **OK** to close the **CTLE Data** window.

For more information see Continuous Time Linear Equalization.

Click **EyeMeasurementFunctions** to open the **Set Eye Measurement Functions** window.



- In the **Function** field, select **MinEyeWidth** or **MinEyeHeight**.
- Set the **Unit Interval** parameter to the Unit Interval (UI) of the Eye diagram.
- Set the **Start Offset** parameter if an initial time offset is appropriate.
- Set the **End Offset** parameter if trailing time offset is appropriate.
- Leave **Auto Crossing Amplitude** set to 1 (or any non-zero number) to have the solver compute the crossing amplitude by averaging the histograms for the high and low voltages. The crossing amplitude sets the voltage at which the minimum eye width is calculated.
- When **Auto Crossing Amplitude** is 0, the solver uses the voltage set in the **Crossing Amplitude** parameter as the crossing voltage amplitude for the minimum eye width calculation.
- Leave **Auto Compute Eye Measurement Point** set to 1 (or any non-zero number) to have the solver compute the eye measurement point by averaging the times of the

crossing points. The eye measurement point sets the time at which the minimum eye height is calculated.

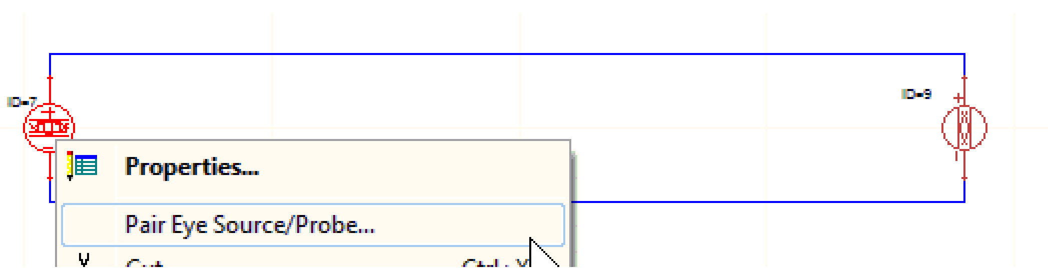
- When **Auto Compute Eye Measurement Point** is 0, the solver uses the voltage set in the **Eye Measurement Point** parameter as the time for the minimum eye height calculation.
- Click **Add** to add the measurement to the analysis. Add any number of measurements. The window above has added a **MinEyeWidth** measurement using the default settings.
- Click **OK** to close the **Set Eye Measurement Functions** window.
- For more information, see Eye Measurements in the Reports topic.

Click **OK** to close the Properties window of the probe.

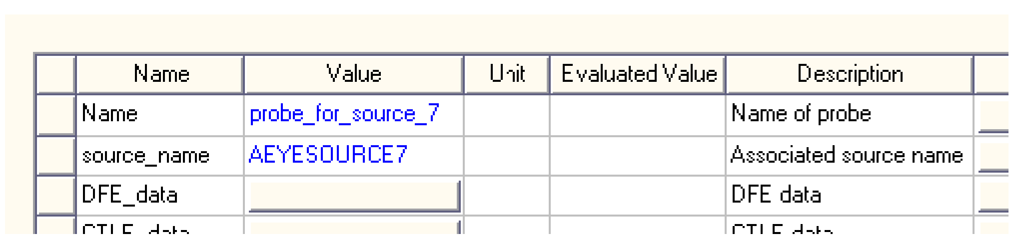
Pairing an Eye Source and an Eye Probe in the Schematic Editor

A VerifEye or QuickEye circuit may contain multiple Eye Sources and Eye Probes. Each Eye Probe must be paired with a corresponding Eye Source. The Eye Probe **source_name** parameter has a drop-down that lists all the Eye Sources in the design.

Instead of using the drop-down menu, pair a selected source and probe directly in the schematic. Use **Ctrl**+click to select the Eye Source and Eye Probe that you want to pair up. With the cursor on either component, right-click and select **Pair Eye Source/Probe ...** on the menu.

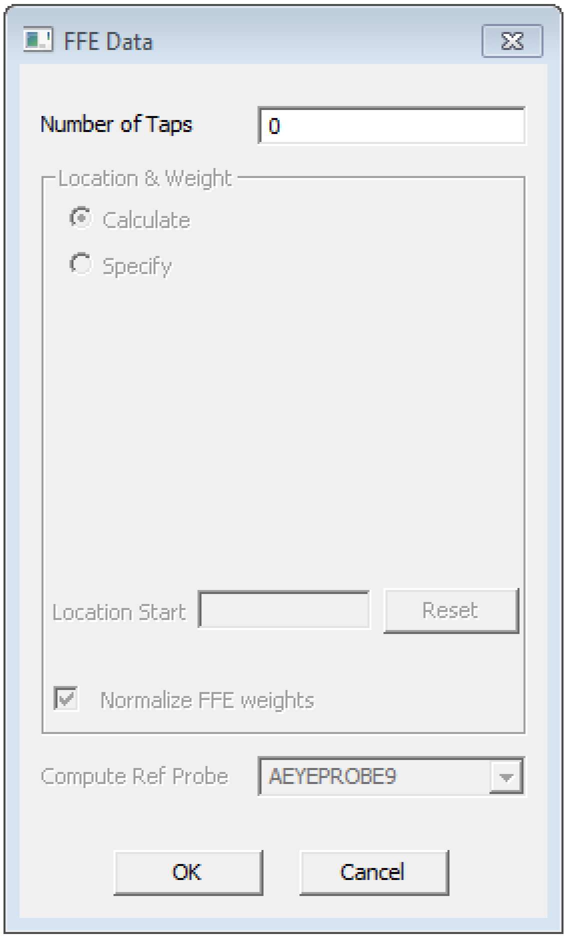


The **source_name** parameter in the selected Eye Probe is set to the name of the selected Eye Source.



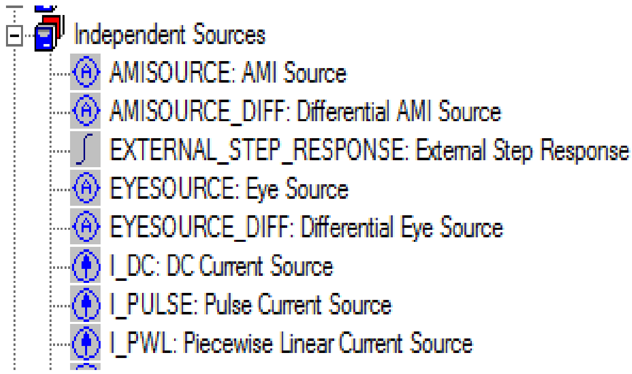
Name	Value	Unit	Evaluated Value	Description
Name	probe_for_source_7			Name of probe
source_name	AEYESOURCE7			Associated source name
DFE_data				DFE data

The **Compute ref probe** field in the FFE setup window for the selected Eye Source is set to the name of the selected Eye probe. The pairing is made whether or not FFE is enabled in the Eye Source.



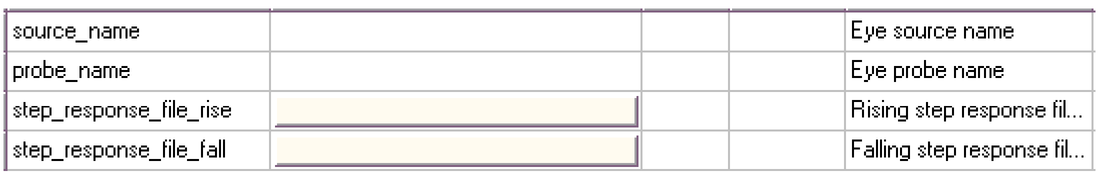
Add an External Step Response to a QuickEye Analysis

Instead of having the solver compute the step response, supply the step response via a data file, using the External Step Response component. First, place Eye Sources and Probes in the circuit. Select the External Step Response on the Independent Sources list on the Components tab:



The External Step Response component does not physically connect to the circuit.

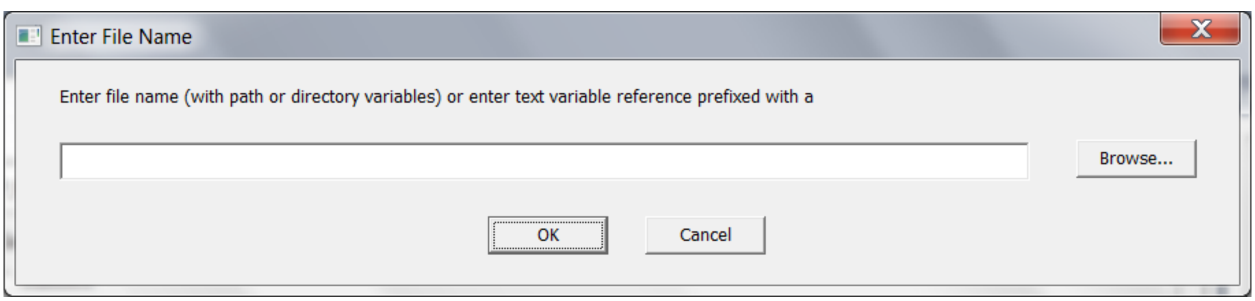
Click the component to open the Properties list:



source_name				Eye source name
probe_name				Eye probe name
step_response_file_rise				Rising step response fil...
step_response_file_fall				Falling step response fil...

Select the source name and the probe name. The source and probe names appear automatically when they are present in the circuit. Each step response source applies to one source and one probe.

Specify the files that contain the rising and falling step responses by clicking the buttons in the **Value** field. The same window applies to both files:



If only one file is supplied, the rise and fall are treated as symmetrical.

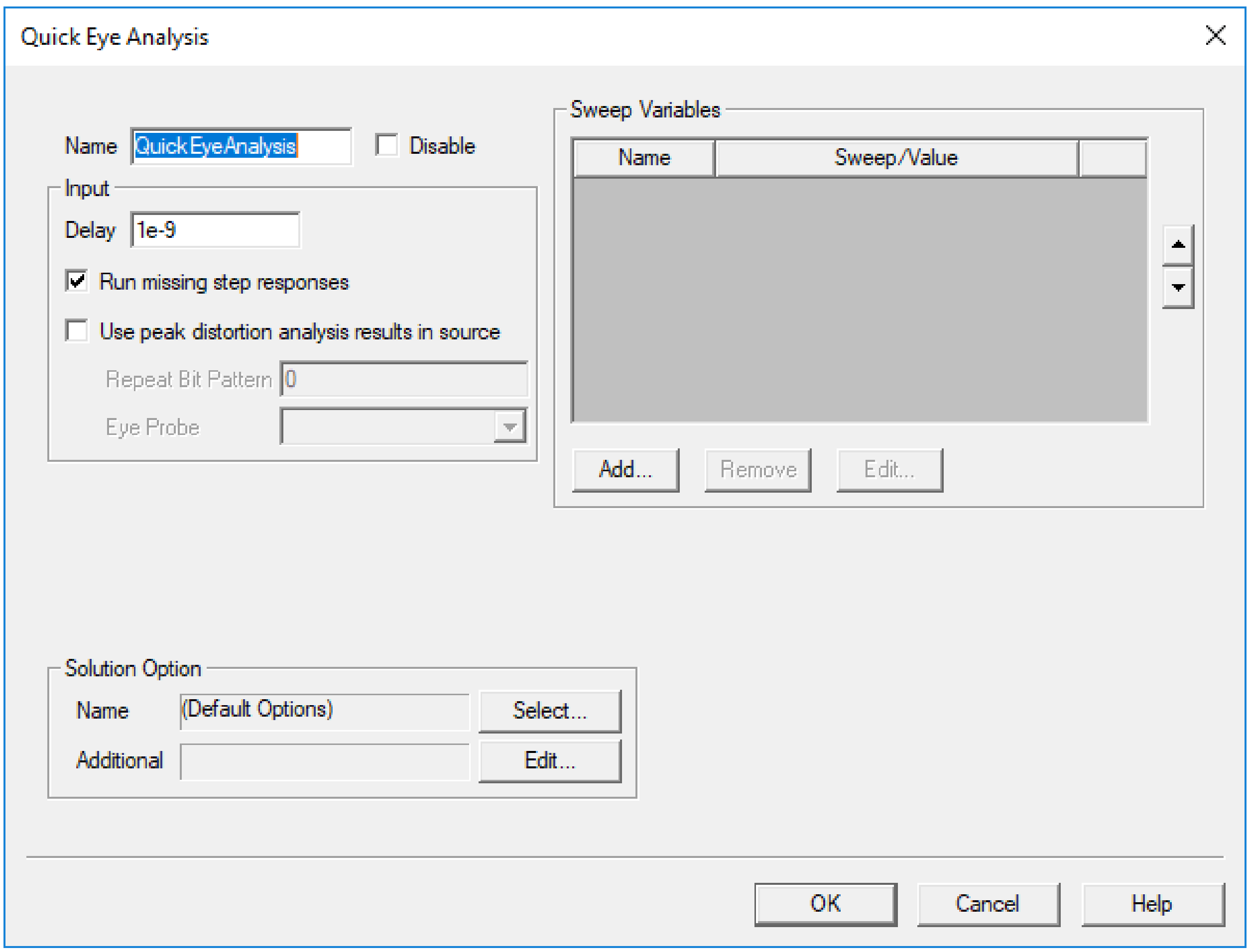
> **Note:** If no step response file is specified (or found), the default is for analysis to fail. Setting the VE/QE analysis parameter **run_missing_step_response=1** runs transient analysis to generate the missing step response.

The step response data files have a two-column format: *time voltage*. Linear interpolation is used between time points.

Set Up the Quick Eye Analysis

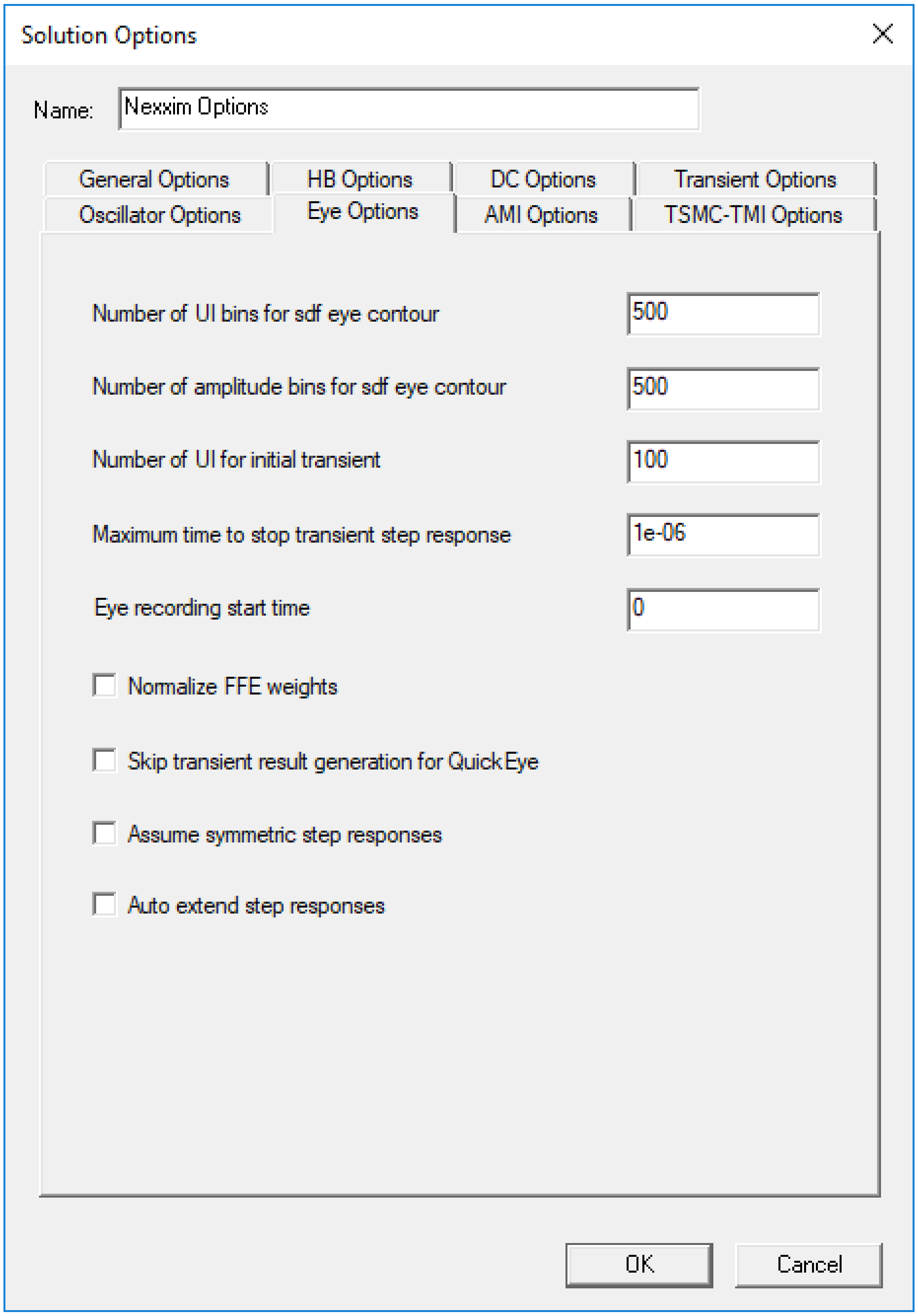
From the **Schematic Editor**, perform the following steps to set up and run a Quick Eye analysis and view the results.

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Quick Eye Analysis** to open the **Quick Eye Analysis** window.



2. Type an **Analysis Name** (or accept the default name, for example “QuickEyeAnalysis”).
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use.
4. Set the **Delay** parameter to control the delay in applying the step response.
5. **Run missing step response** allows simulation to proceed when an External Step Response component is present in the design but no step response data file can be found.

6. Click the **Use peak distortion analysis results in source** check box to enable the worst-case bit pattern to be output on the source. Set the **Repeat Bit Pattern** field to the appropriate number of repeats. When the **Use peak distortion** check box is checked, the worst-case bits from Peak Distortion Analysis is output from all Eye Sources. Any bit patterns set in the sources are ignored. See Peak Distortion Analysis for further information.
 - Specify the number of times to repeat the worst-case bit pattern.
 - Select the Eye Probe that is to be the target for all Eye Sources.
7. To add a sweep, locate the **Sweep Variable** area in the **Quick Eye Analysis** window.
 - In the **Sweep Variable** area, click **Add**, and the **Add/Edit Sweep** window opens.
 - In the **Variable** list, select **Temp** or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value, Linear step, Linear count, Decade count,** or **Octave count**.
 - Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
 - For details on sweeps, see Variable Sweep.
 - Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the**Quick Eye Analysis** window.
8. Optionally, click **Edit** to open the **Output Quantities** window.
 - Click the check boxes on the **OutputQuantities** window to add net voltages and branch currents to the data generated by the analysis.
 - Click **OK** to add the selected output values and return to the **QuickEyeAnalysis** window.
9. Optionally, use the fields in the **Solution Options** group box to add **VerifEye/QuickEye** options.
 - Click **Select** on the **Name** field to open the **Select Solution Options** window.
 - To create a new option set, click **New** to open the **Solution Options** window.



- Click the **Name** field to name the new option set.
- Click the **Eye Options** group box fields and check boxes to set any appropriate options.

Number of UI bins sets the number of UI histogram bins to use for generating the 3D and contour data. The default is 500. Option: **eye.ui_bins**.

Number of amplitude bins sets the number of amplitude histogram bins to use for generating the 3D and contour data. The default is 500. Option: **eye.ampl_bins**.

Number of UI sets the number of unit intervals (UI) over which the initial transient step response should run after any channel or source delay. The default is 100. Option: **eye.num_initial_ui**.

Maximum time sets the maximum stop time for the transient step response calculation. The default is 1µs. Option: **eye.resp_tmax**.

NOTE: QuickEye uses the minimum of (**num_initial_ui** x UI) and (**resp_tmax**) to set the transient final time for the step response calculation.

Normalize FFE weights sets the solver to normalize FFE weights (sum of absolute values of all weights equals one). This option applies only to weights supplied by the user. The Circuit solver always normalizes the taps weights that it calculates. The default for user-supplied weights is no normalization. Option: **eye.normalize_ffe_weights**.

Skip transient result turns off QuickEye storage of the eye diagram and the transient-like result. By default, QuickEye stores the eye diagram and the transient-like result. Option: **eye.qe_only_cmf**.

Assume symmetric step responses sets QE/VE to use the rise time as the fall time. The default is to use different rise and fall times. Option: **eye.sym_step_resp**.

Auto extend step responses extends the step response beyond the simulation time. By default, the step response is extended only to the end of simulation time. Option: **eye.auto_extend_step_resp**.

- For more information, see VerifEye and Quick Eye Analysis Options.
- Click **OK** to return to the **VerifEye (Statistical Eye)Analysis** window.

Run the Quick Eye Analysis and Display Results

1. Run the simulation:
  - From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
  - If the analysis is not successful, check the **Message Manager** window for an explanation, and then take corrective action.
2. Display results:
  - From the menu bar, click **Circuit**, then click **Results** >**Create Standard Report**. (Alternatively, select **Create Standard Report** on the **Results** icon in the **Project Manager** window.)

For more information, see QuickEye Analysis Outputs.

For information on reporting the Peak Distortion Analysis (PDA) data, see Peak Distortion Analysis.

Running Quick Eye Analysis on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist should contain the information for running the envelope analysis. No Solution Setup is required.

QuickEye Source Netlist Syntax

The netlist syntax for the QuickEye Source is:

```
AEYESOURCExxxxn1 [n2] COMPONENT=EYE_SOURCE
 + VLOW=valVHIGH=val TRISE=val TFALL=val
+ UI=valBPS=val
+ BITSPERSYMBOL=1|2 CODING=0|1
+ STEP_RESP_NUM_UI=val
+ DCD=valDCD_TIME=val
+ TXRJ=[val ...]TXPJ=[val ...]TXUJ=[val ...]
 + TXCJ='filename'
 + FFE_LOCS=[locations]
 + FFE_WEIGHTS=[weights]
 + NORMALIZE_FFE= val
+ FFE_COMPUTE_PROBE='probe_name'
+ RESISTANCE=val
+ BITLIST=#bitlist
+ BITFILE='file-reference'
 + RANDOM_BIT_COUNT=val REPEAT_COUNT=val RANDOM_SEED=val
+ PRBS_NO=val
+ PRBS_SEED=val
+ PRBS_INVERT=val
+ PRBS_BITLENGTH=val
+ DO_ENCODING=val
+ HOLD_LAST_BIT=val
+ DELAY=val
```

n1 is the node connected to the single-ended source. *n2* is the second node on a differential source. One of the entries **BITLIST**, **BITFILE**, **RANDOM_BIT_COUNT**, or **PRBS_NO** must be specified. The entry **COMPONENT=EYE_SOURCE** is required.

> **Note:**
>
> The Eye sources are internally grounded. All terminals should be connected to signal lines, not directly to ground.

See *Eye Source* in the Components help for details on these properties.

QuickEye Probe Netlist Syntax

The netlist syntax for the QuickEye Probe is:

```
AEYEPROBExxxxn1 [n2] COMPONENT=EYE_PROBE
 + SOURCE_NAME='source_name'
 + DFE_TAPS=val
+ DFE_LOCS=[locations]
 + DFE_WEIGHTS=[weights]
+ DFE_WEIGHT_lbS=[lower bounds]
+ DFE_WEIGHT_ubS=[upper bounds]
+ DECISION_HIGH=val
+ DECISION_LOW=val
+ THRESHOLDS=[val (val...) ]
+ CTLE_FIRST_POLE=val
+ CTLE_SECOND_POLE=val
+ CTLE_ALLOWED_GAIN=[val ...]
 + CTLE_FILE="file_reference"
 + CTLE_RELTOL=val
 + TF_NUM=val
 + CTLE_POLES=[ val ... ]
 + CTLE_ZEROS=[ val ... ]
 + CTLE_AC_GAIN=val
 + CTLE_ALLOWED_GAIN=[ val ... ]
 + CTLE_NUMERATOR=[ val ... ]
 + CTLE_DENOMINATOR=[ val ... ]
```

n1 is the node connected to the single-ended source. *n2* is the second node on a differential source. The entry **COMPONENT=EYE_PROBE** is required.

See *Eye Probe* in the Components help for details on these properties.

QuickEye External Step Response Netlist Syntax

The netlist syntax for the Eye External Step Response is:

```
AxxxxCOMPONENT=EXTERNAL_STEP_RESPONSE
 + SOURCE_NAME='eye_source '
 + PROBE_NAME='eye_probe '
 + STEP_RESPONSE_FILE_RISE='file_reference '
 + STEP_RESPONSE_FILE_FALL='file_reference '
```

The entry **COMPONENT=EXTERNAL_STEP_RESPONSE** is required.

See External Files for details on references to external files.

See *Eye External Step Response* in the Components help for details on these properties.

Quick Eye Analysis Netlist Format

Quick Eye analysis is invoked on the netlist using the **.EYE** statement:

```
.EYE QUICK=1 DELAY=val
TRISE=val TFALL=val LOW=val HIGH=val
BPS=val|UI=val FFE_TAP=val FFE_WEIGHTS=[val val...]
NORMALIZE_FFE_WEIGHTS=val DFE_TAPS=val DFE_LOCS=[val val... ]
DFE_WEIGHTS=[val val...]
(or DFE_WEIGHT_LBS=[val val ...] DFE_WEIGHT_UBS=[val val ...])
DECISION_THRESHOLD=val DECISION_HIGH=val DECISION_LOW=val
PD_COUNT=val PROBE=eyeprobe
RUN_MISSING_STEP_RESPONSE=val
```

The entry **QUICK=1** enables the Quick Eye analysis.

> **Note:** If **DFE** is automatically calculated, then neither **DEF_WEIGHTS** nor **DFE_WEIGHT_LBS/DFE_WEIGHT_UBS** show. If **DFE** is calculated with limits, **DEF_WEIGHTS** does not appear. If **DFE** is specified, neither **DFE_WEIGHT_LBS** nor **DFE_WEIGHT_UBS** appear. See Add an Eye Probe to a Schematic for DFE options.

By default, the worst-case bit pattern is calculated by Peak Distortion Analysis (PDA). The worst-case bit pattern can be displayed in a report, and can also be sent to the Eye sources for simulation with QuickEye. The **PD_COUNT** parameter controls the sending of the worst-case bit pattern.

- When **PD_COUNT** is -1, the worst case bit pattern is not sent to the Eye sources. This is the default behavior.
- When **PD_COUNT** is 0 (zero), one iteration of the worst-case bit pattern is sent to all Eye sources in the design, overriding any bit patterns set in the Eye Sources.
- When **PD_COUNT** is greater than zero, the solver repeats the PDA pattern. For example, PD_COUNT=1 gets one repeat (two iterations of the pattern), PD_COUNT=2 gets two repeats (three iterations), et cetera.

The PROBE parameter controls the PDA calculation when PD_COUNT is 0 or greater than zero. When PD_COUNT is -1, PDA worst-case bit patterns are calculated for all pairs of Eye Sources and Eye Probes, but no bits are sent to the analysis. When PD_COUNT is zero or greater, the worst-case bit patterns are calculated from all sources to the Eye Probe given by the PROBE parameter, and those bit patterns are sent from the sources.

- The **DELAY** parameter sets a time delay during which the effects of transitions are ignored. After the delay, the step response is applied.

- Setting **RUN_MISSING_STEP_RESPONSE=1** runs transient analysis to compute the step response when an External Step Response data file reference is missing or cannot be found. Without this option, missing step response data causes analysis to fail.

The other parameters are global versions of the Eye Source and Eye Probe parameters. The global version of a parameter is used only when the Eye Source or Eye Probe does not have that parameter set locally.

Run Quick Eye Analysis and Display Results

1. Run the simulation:
   - From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
   - If the analysis is not successful, check the **Message Manager** window for an explanation, and then take corrective action.
2. Display results:
   - From the menu bar, click **Circuit**, then click **Results** >**Create Standard Report**. (Alternatively, select **Create Standard Report** on the **Results** icon in the **Project Manager** window.)

For more information, see QuickEye Analysis Outputs.

For information on reporting the Peak Distortion Analysis (PDA) data, see Peak Distortion Analysis.

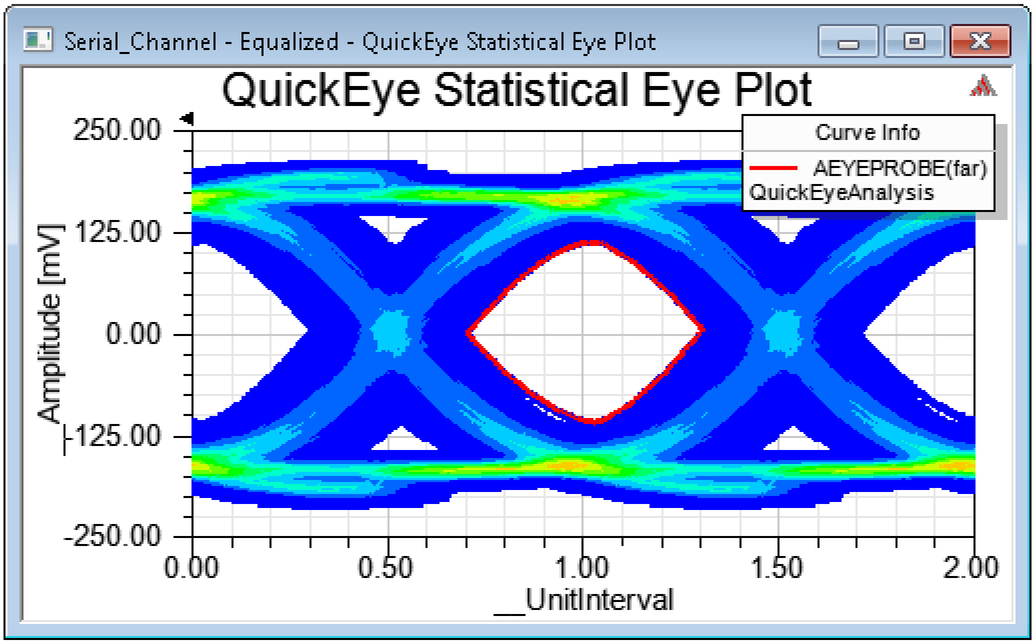
Display Quick Eye Analysis Outputs

To open the results of a Quick Eye analysis, click on the Results icon in the **Project Manager** window and select one of the **Create Report** options. There are many ways to plot the data; this topic gives examples of a Statistical (Sampled) Eye Diagram, a 2D Bathtub plot, standard and statistical Eye diagrams with PAM4 modulation, and a 2D Bathtub plot with PAM4 modulation.

Sampled Eye Diagram

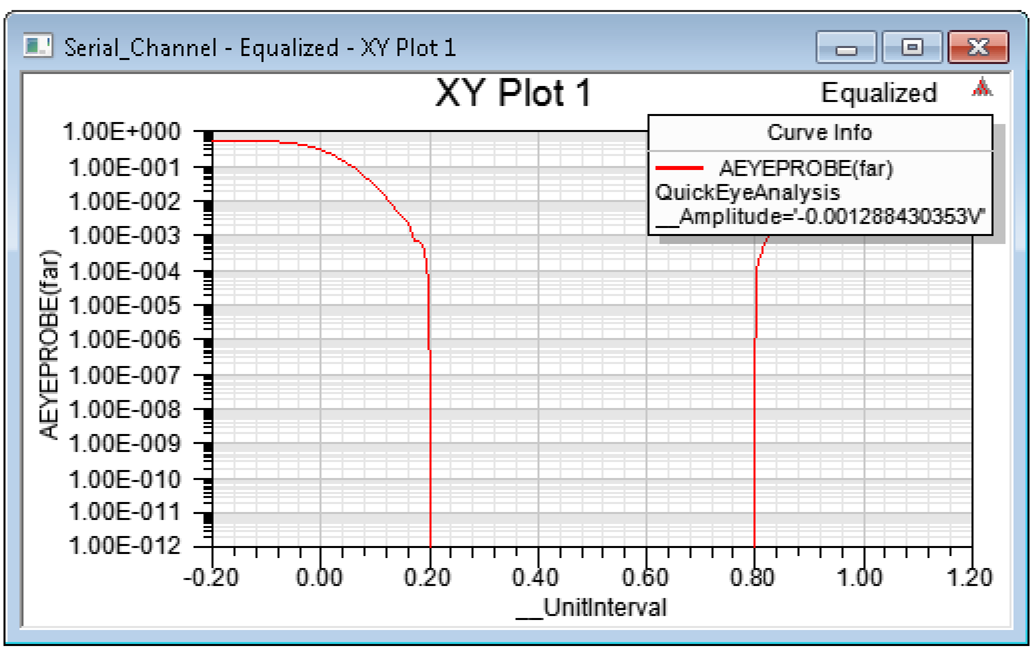
The sampled or statistical eye report generates the eye diagram but does not retain the transient data for further post-processing. The sampled eye diagram plot requires an Eye Scope be present at the appropriate measurement point.

To generate a sampled eye diagram, select **Create Statistical Eye Report>Statistical Eye Plot**. Select your Quick Eye Analysis as the **Solution** ; **UI** as the Domain; **Eye** as the Category; the appropriate Eye Probe as the Quantity Type; and **<none>** as the Function.



2D Bathtub BER Plots

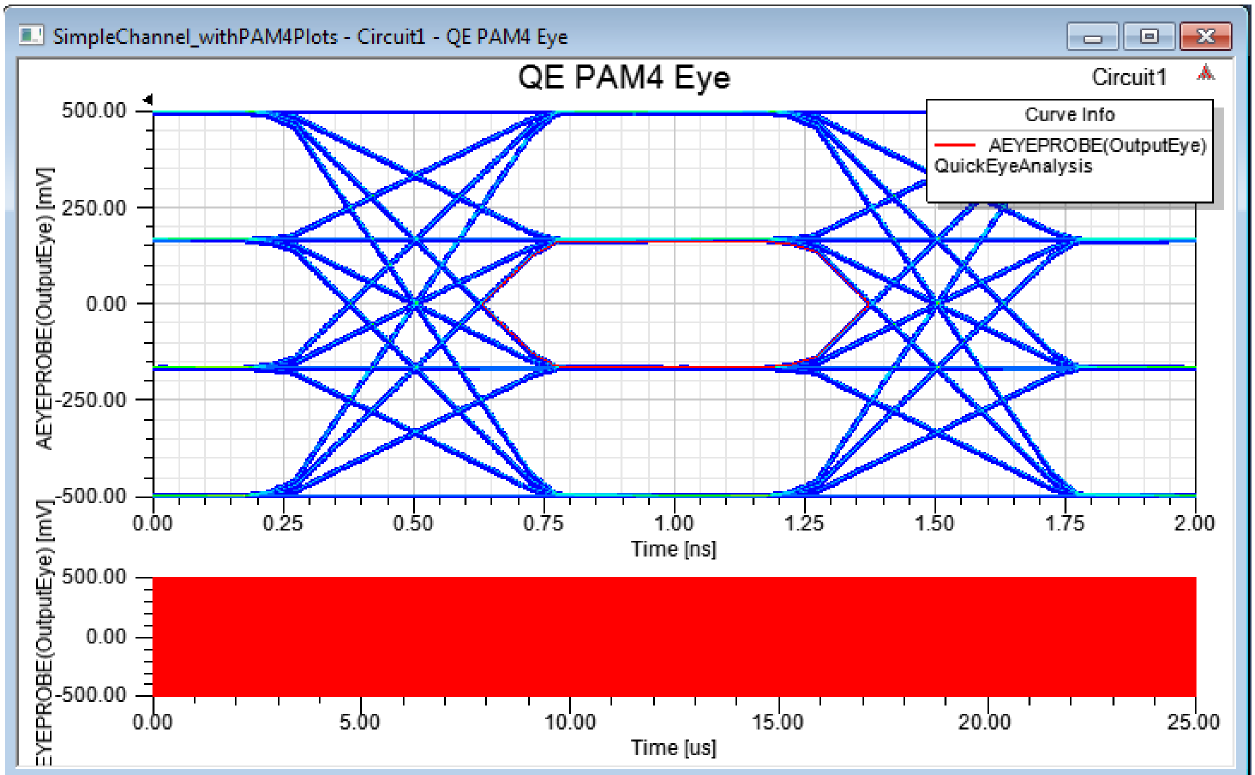
To view the data as a 2D bathtub curve, select **Create Standard Report>Rectangular Plot**. From the Report window, select **UI** as the Domain and **Bathtub** as the Category. Click **New Report**.



This graph shows the BER at various locations in the unit interval. By default, the amplitude is set at the midpoint of its value range. The BER values are on the Y-axis using logarithmic scaling and a minimum of 1e-16.

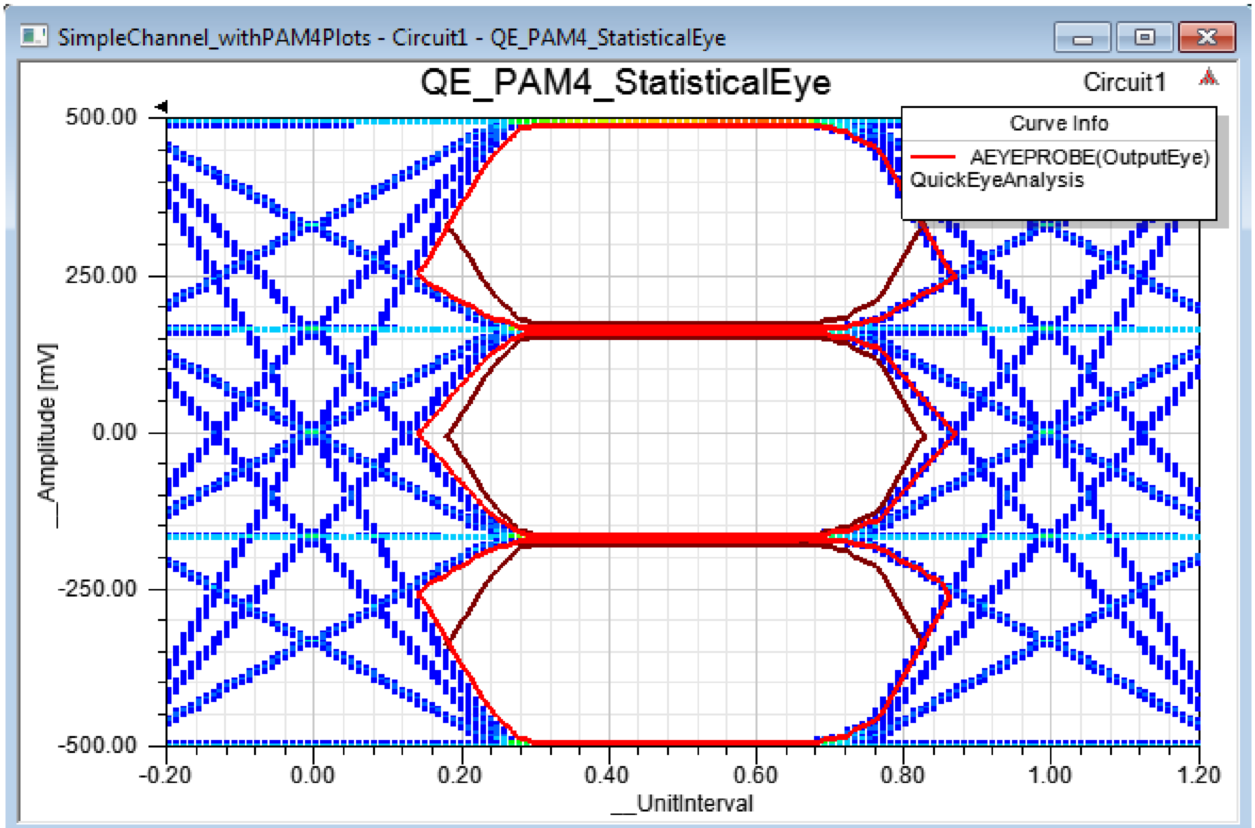
2D PAM4 Eye Diagrams

To enable PAM4 modulation, set the **modulation** parameter in the Eye source to **PAM4**. Select **Gray** or **Linear** for the **coding** parameter as applicable. When analysis completes, select **Create Eye Diagram > Rectangular Plot**.



The plot shows four voltage levels, each one corresponding to a transition between two bits. This example uses the default Gray coding: the lowest level represents 00, the next higher level represents 01, the next higher level represents 11, and the highest level represents 10. The four voltage levels form three eyes in the diagram.

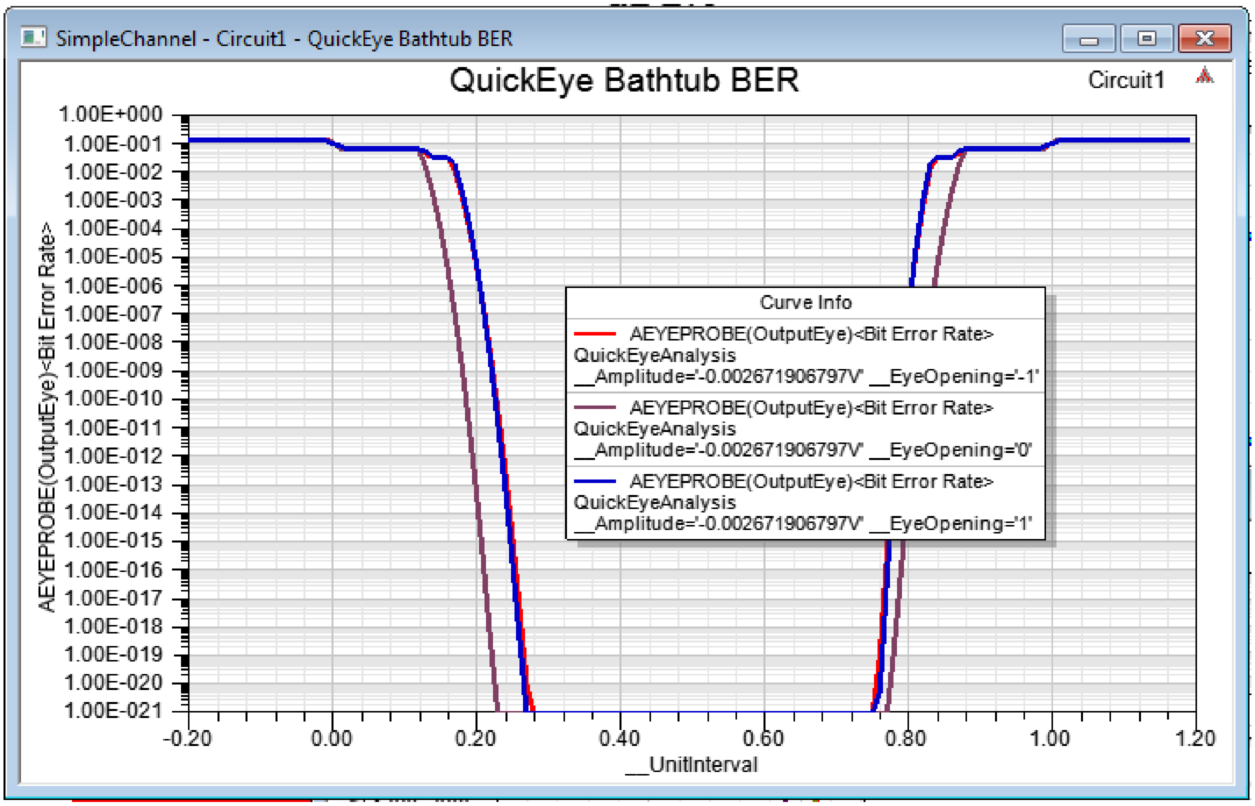
Alternatively, select **Create Statistical Eye Diagram Report**.



The red lines show the eye openings. The dark lines show the intersection of all three eye openings.

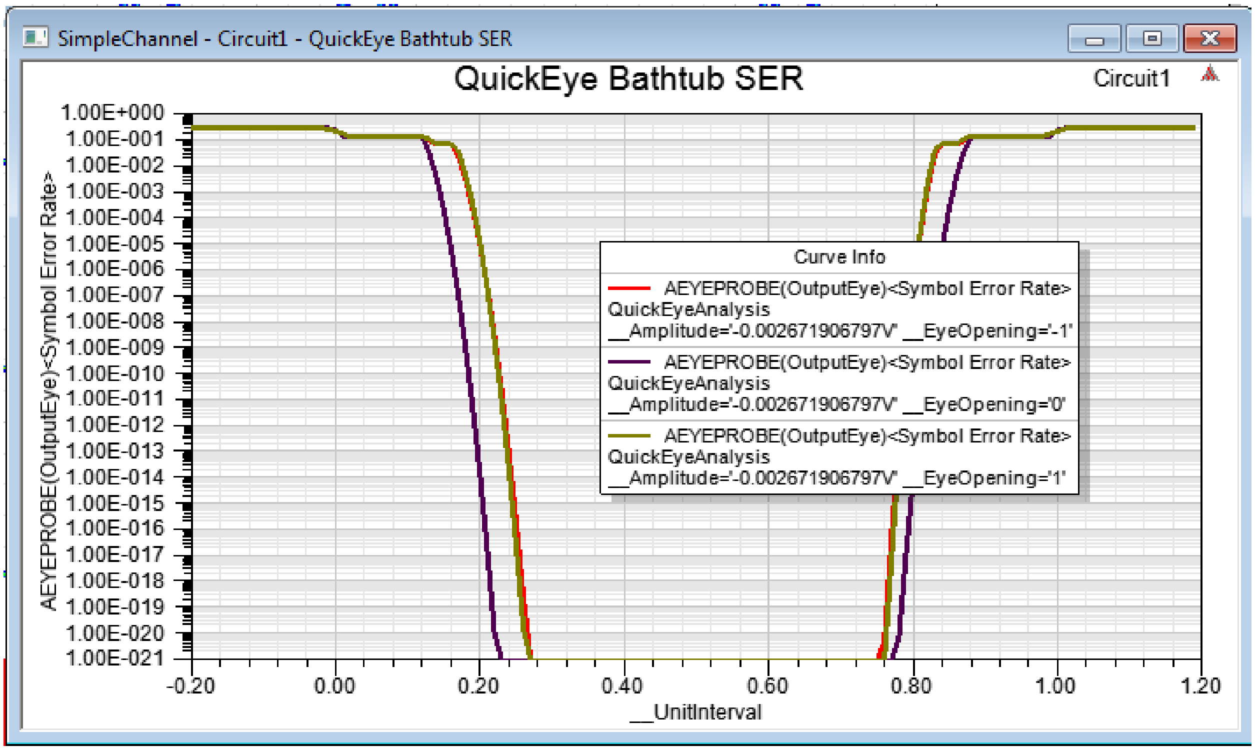
2D PAM4 Bathtub Plots

To enable PAM4 modulation, set the **modulation** parameter in the Eye source to **PAM4**. Select **Gray** or **Linear** for the **coding** parameter as applicable. When analysis completes, select **CreateStandard Report> Rectangular Plot**. From the Report setup, click the Families tab and edit the _EyeOpening family. The entries are -101 (combined error rate), -1 (upper eye error rate), 0 (middle eye error rate) and 1 (lower eye error rate). From the Report window, select **UI** as the Domain and **Bathtub** as the Category. Select the **<Bit Error Rate>** trace to see the bit error rate.



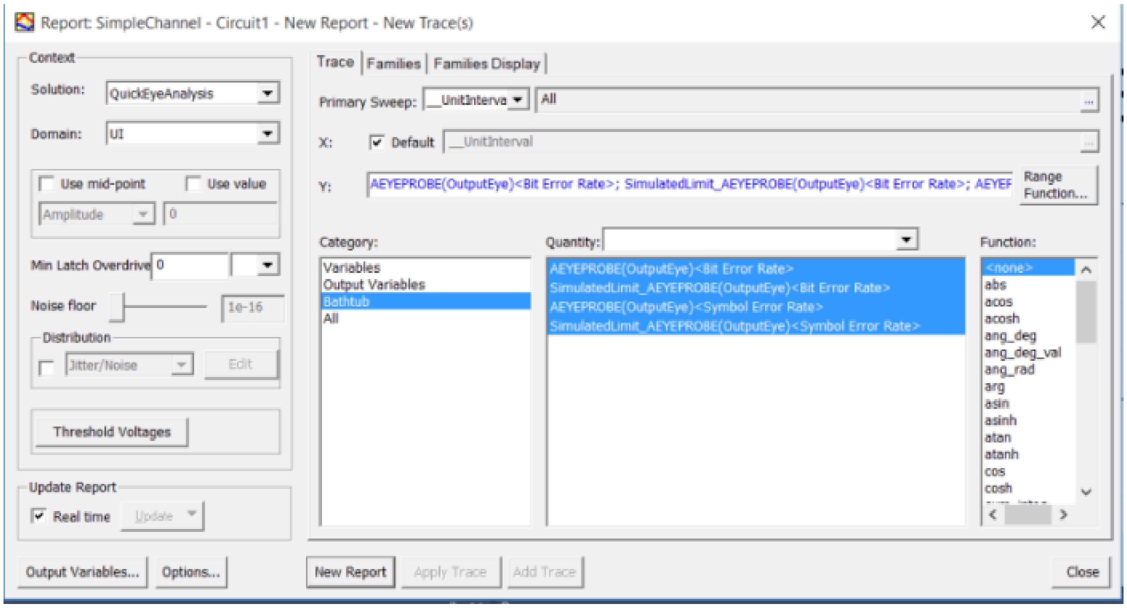
The example plot shows the separate bit error rates for the three PAM4 eyes (-1, 0, and 1 selected).

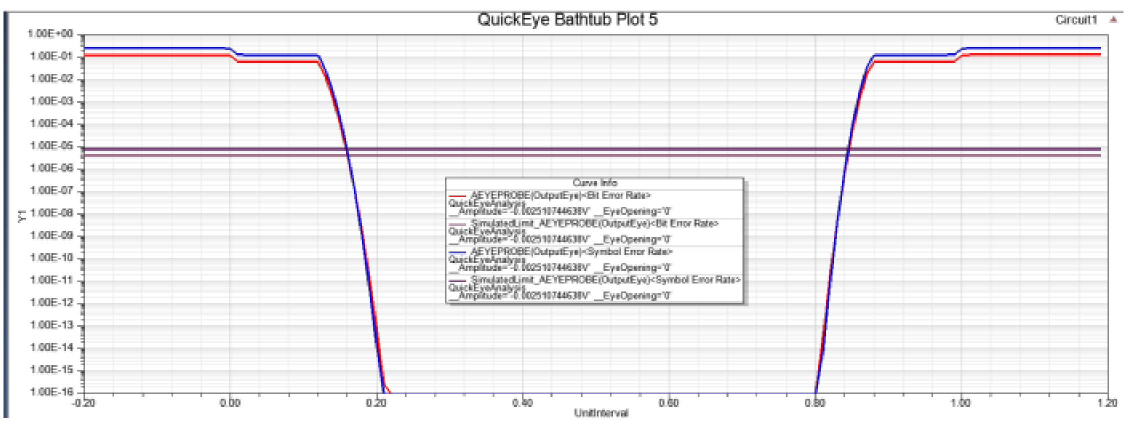
Thebathtub curvebathtub curve for a PAM4 simulation can show the symbol error rate instead of the bit error rate. With PAM4, a symbol is two bits. The report setup is the same, but now select the **<Symbol Error Rate>** trace.



For bathtub curves of QE, distinguish simulated/extrapolated regions using a partitioning horizontal line via the ‘SimulatedLimit’ option that is in available the ‘Quantity’ column. Such processing allows for the robust extrapolation of bathtub curve arising out of QE and AMI analyses.

- BER/SER level of partitioning = 1/bit or symbol_length simulated
- symbol_length = bit_length for NRZ, or, 0.5*bit_length for PAM4





VerifEye and Quick Eye Analysis Options

The options apply to both VerifEye and Quick Eye analyses.

For more information see the VerifEye and Quick Eye Analysis Options Reference.

All QuickEye and VerifEye analysis options use the prefix **eye**.

See Analysis-Specific Options for details on setting or changing these options.

See Resolutions of Eye Diagrams for details on setting amplitude and UI bin resolutions.

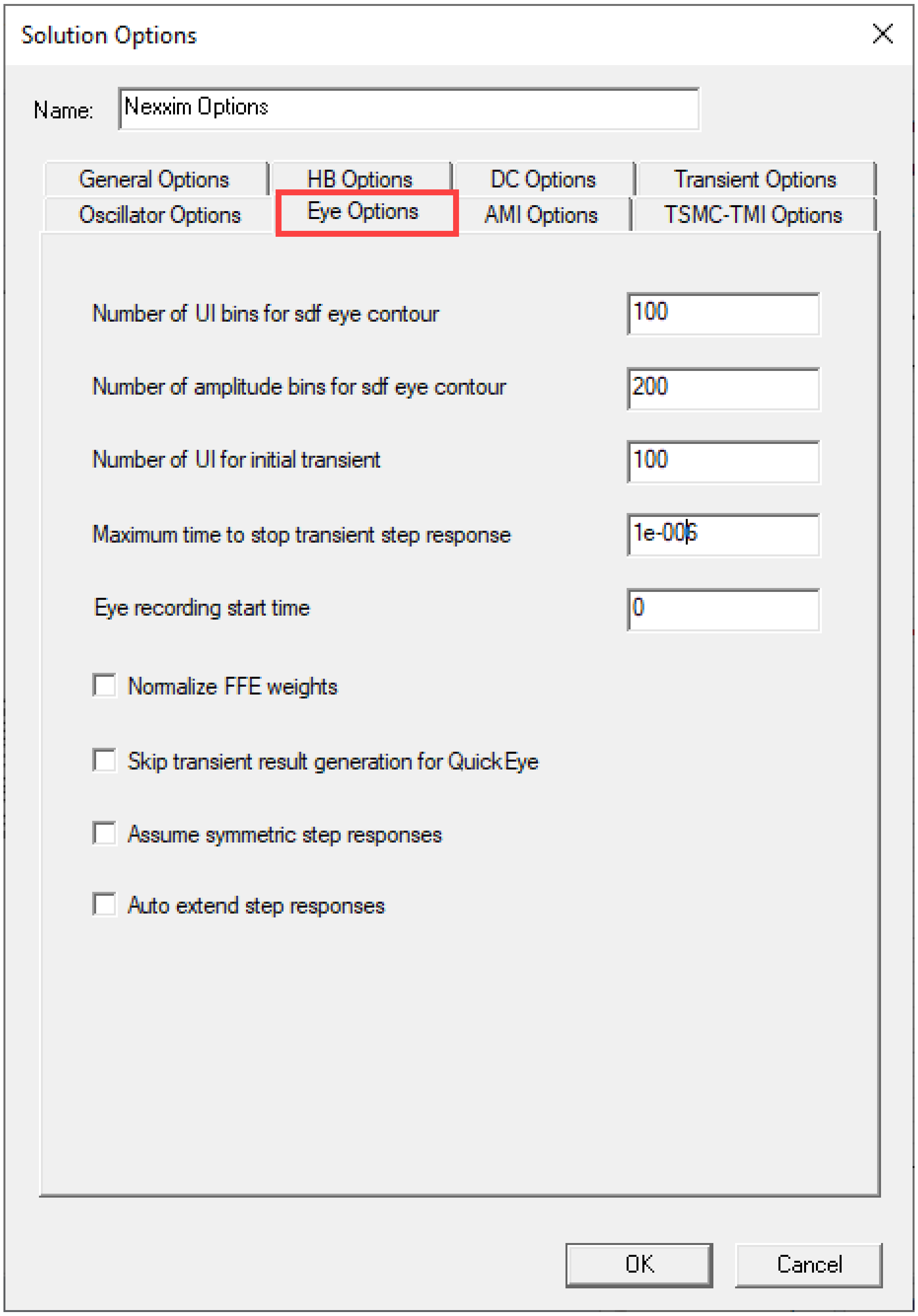
Resolutions of Eye Diagrams

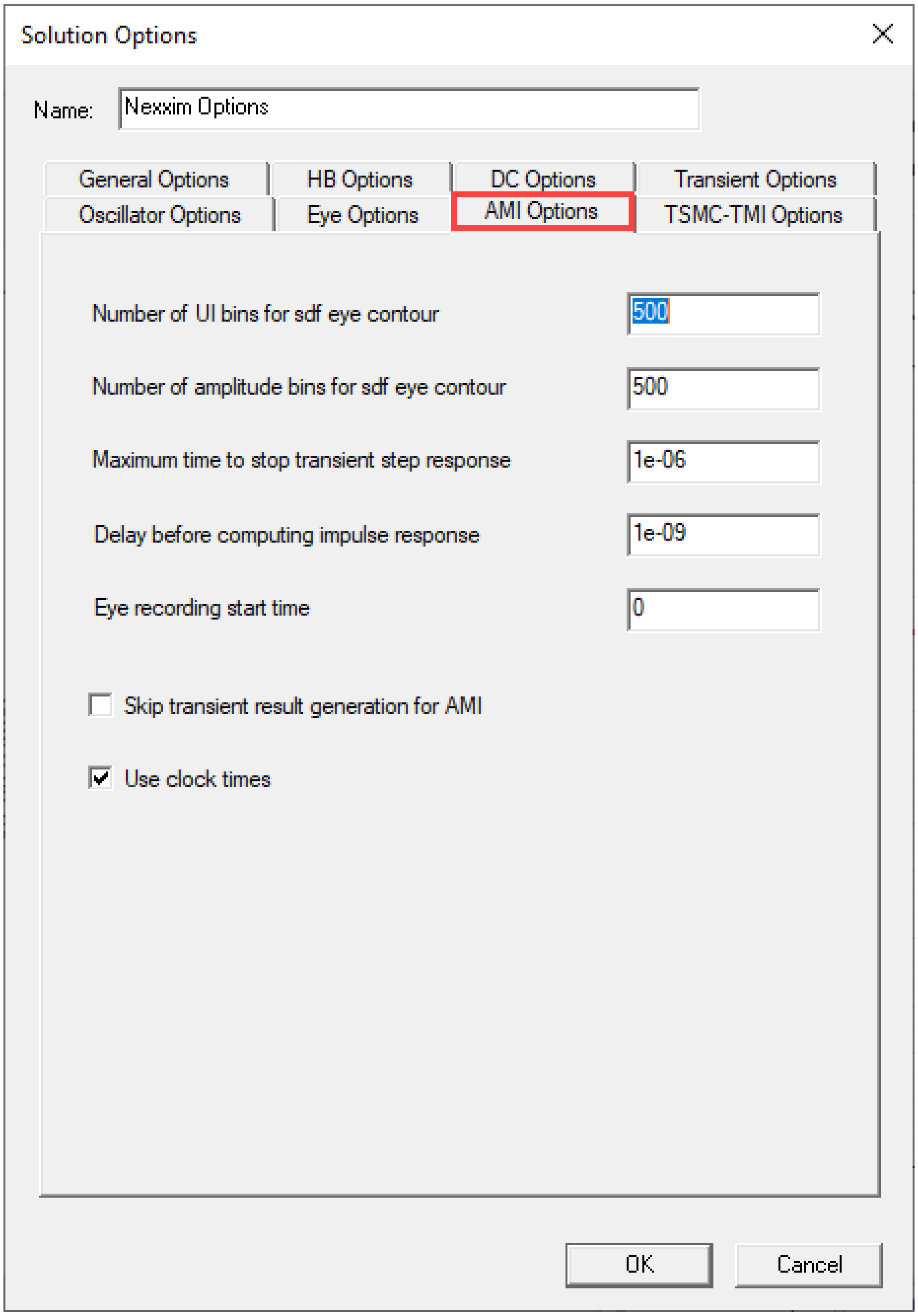
In a schematic, define sets of analysis-specific options with the **Add Solution Options** selection. You can then select one of your defined option sets when you create or edit a solution setup. Using this mechanism, use the same option set in multiple analysis setups, or use different option sets in the same solution setup by editing the setup.

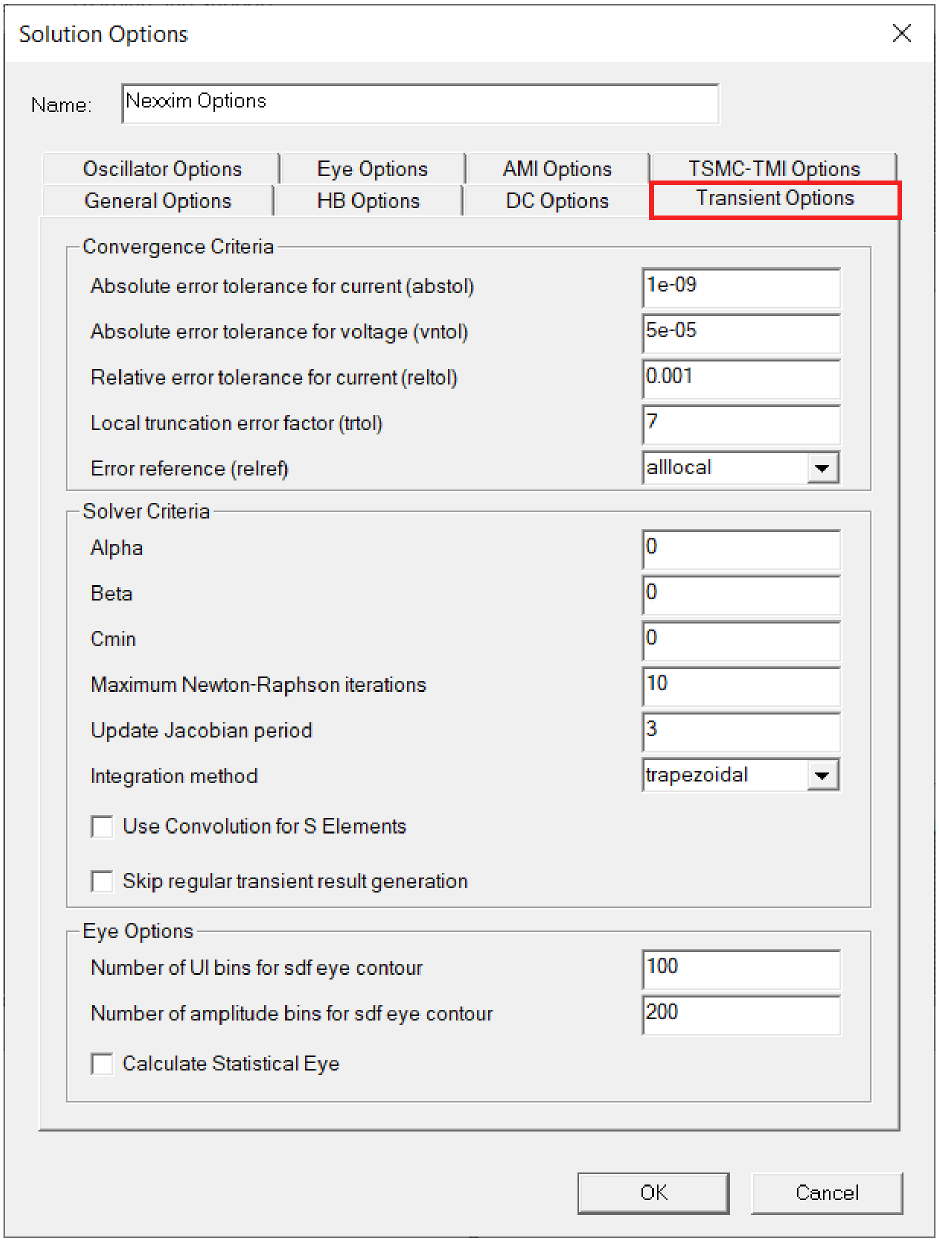
There are multiple ways to create an option set:

- From the **Circuit** menu, select **Add Solution Options ...**.
- From the **Project Manager** window, expand the **Project Tree**. Then right-click **Analysis** and select **Add Solution Options ...** to open the **Solution Setup** window.
- Click **Select** in the **Solution Options** group box in a **Solution Setup** window.

Any of these actions opens the **Solution Options** window. Click the **Eye Options**, **AMI Options**, and **Transient Options** tabs to view the following options.

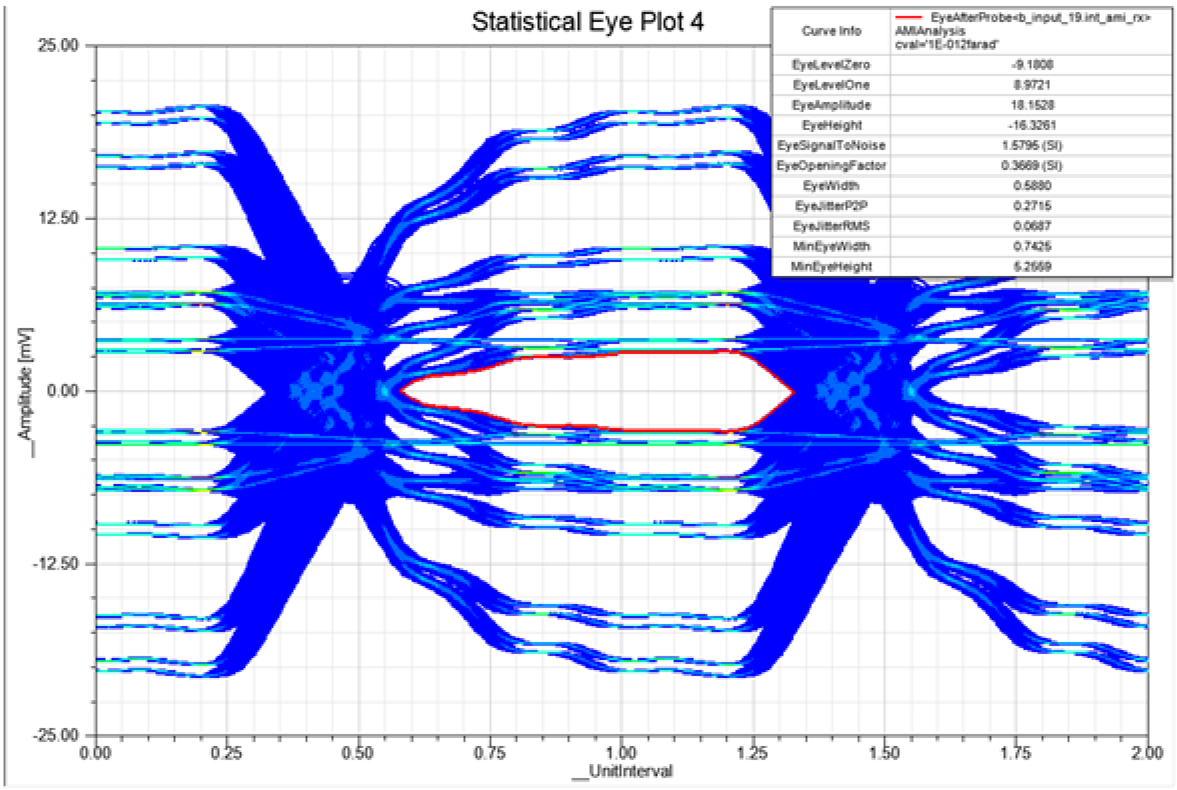

Solution Options
Name:
Nexxim Options
General Options
HB Options
DC Options
Transient Options
Oscillator Options
Eye Options
AMI Options
TSMC-TMI Options
Number of UI bins for sdf eye contour
100
Number of amplitude bins for sdf eye contour
200
Number of UI for initial transient
100
Maximum time to stop transient step response
1e-006
Eye recording start time
0
Normalize FFE weights
Skip transient result generation for QuickEye
Assume symmetric step responses
Auto extend step responses
OK
Cancel

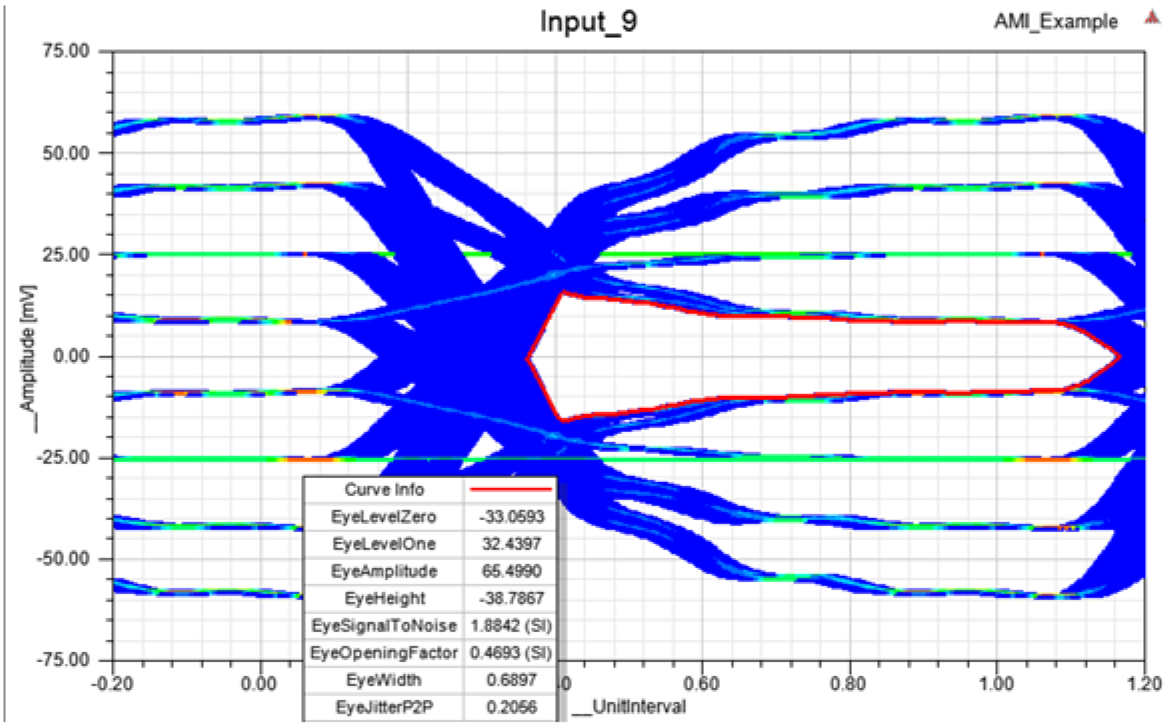

Solution Options
Name:
Nexxim Options
General Options
HB Options
DC Options
Transient Options
Oscillator Options
Eye Options
AMI Options
TSMC-TMI Options
Number of UI bins for sdf eye contour
500
Number of amplitude bins for sdf eye contour
500
Maximum time to stop transient step response
1e-06
Delay before computing impulse response
1e-09
Eye recording start time
0
Skip transient result generation for AMI
Use clock times
OK
Cancel



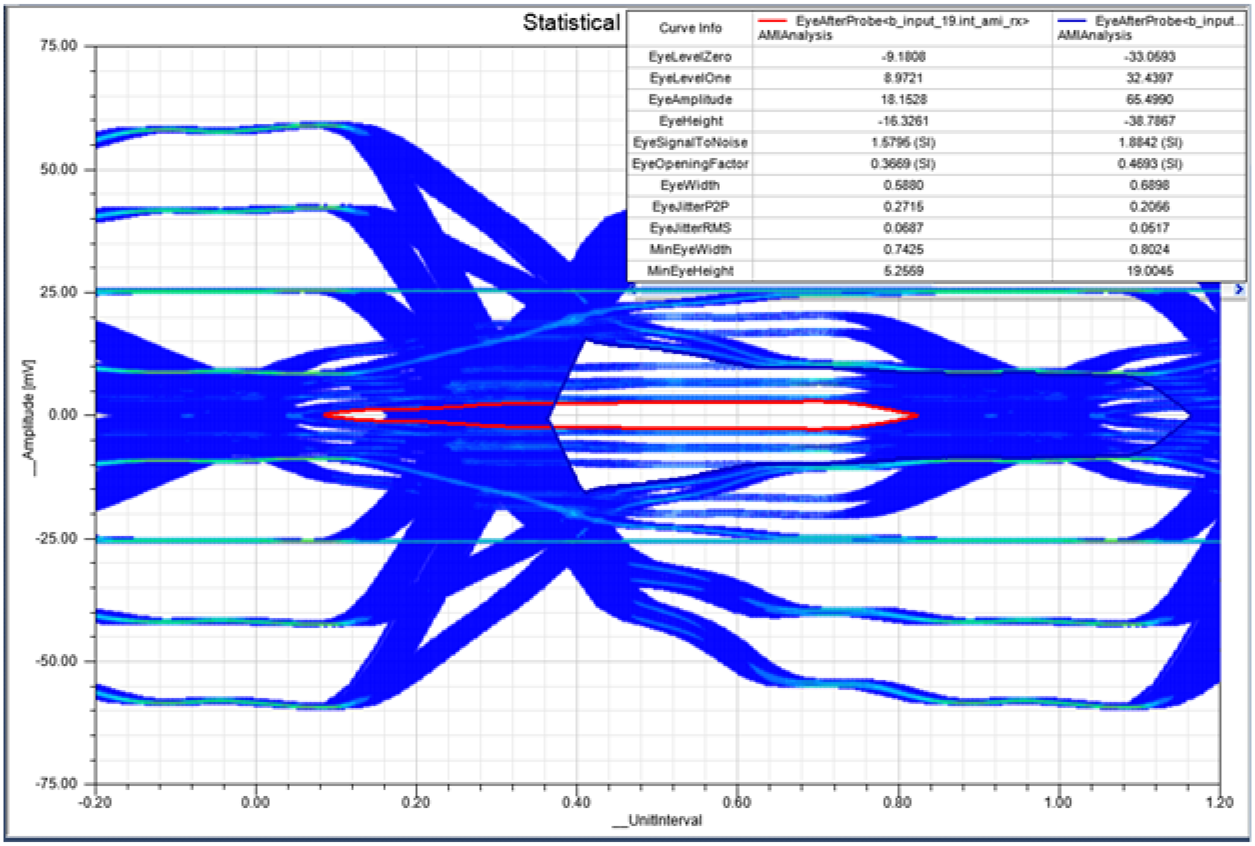
The **Solution Options** window controls the UI and amplitude resolutions of the generated sampled/statistical eye diagrams. And this amplitude resolution in turn controls the fineness of the eye diagram plot and measurements. The limits on these bin counts are set to 1500 which allows a finer resolution (at the possible expense of analysis time). Any eye diagram generated at a resolution higher than 1500 is sub-sampled to fit into a 1500 resolution which could potentially introduce visual and measurement artifacts.

Additionally, when combining multiple traces into a statistical eye diagram with different amplitude or UI bin resolutions, the plot tries to **up-sample** as many traces as needed to ensure that no trace needs to be sub-sampled. In such a case, the allowed max resolution of the composite eye is increased to 3000. For example, if the two traces (whose individual statistical eye diagrams plots are displayed in the following figures) are combined into one statistical plot.





- Assume both are set to have Amplitude resolutions of 500. Until the specified UI or Amplitude resolution in the Options is maintained at less than 1500, the trace is displayed faithfully.
- "**Statistical Eye Plot 4** " has a range of 50mV and **Input_9** has an Amplitude range of 150mV. When these two are added to one plot, the composite eye diagram which superimposes both the eyes now has to fit the 50mV amplitude range of one trace into the combined 150mV Amplitude range: To avoid sub-sampling artifacts, this means that the 50mV range has to occupy at least 500 central bins in the composite eye. This means that the 150mV trace needs to be up/super-sampled to 1500 Amplitude bins (UI bins are upsampled also). In this case of mixed traces with different traces each having different Amplitude ranges, AnsysEDT super-samples the sampled eyes such that none of the constituent eyes are sub-sampled and this limit is increased on the individual eye limit of 1500 to 3000.



These limits are in place to prevent uncontrolled memory usage and users are not expected to be concerned with them. However, if these limits need to be removed (accepting the consequences for memory swell), set the **ANSOFT_DISABLE_PMF_BIN_SIZE_LIMITS=1** environment variable.

For more information see Analysis-Specific Options.

Copying a VerifEye or Quick Eye Analysis

Once a VerifEye analysis has been set up in the schematic editor, save a copy of the setup as a Quick Eye analysis, so you can compare the results of the two types of analysis with the same setup.

1. Right-click the VerifEye analysis in the **Project Manager** window, and select **Copy as QuickEye** on the menu. The copy is made on the clipboard.
2. Select the project that is to contain the copy of the analysis. Right-click the **Analysis** icon in the Project window, and select **Paste** on the menu. The copy is pasted into the list of analyses, with the default name “QuickEyeAnalysis.” Rename the analysis as appropriate.

Once a Quick Eye analysis has been set up in the schematic editor, save a copy of the setup as a VerifEye analysis, so you can compare the results of the two types of analysis with the same setup.

1. Right-click the Quick Eye analysis in the **Project Manager** window, and select **Copy as VerifEye** on the menu. The copy is made on the clipboard.
2. Select the project that is to contain the copy of the analysis. Right-click the **Analysis** icon in the Project window, and select **Paste** on the menu. The copy is pasted into the list of analyses, with the default name “VerifEyeAnalysis.” Rename the analysis as appropriate.

QuickEye, VerifEye, and Bathtub Extrapolation References

[1] Anthony Sanders, Mike Resso, John D. Ambrosia, *Channel Compliance Testing Using Novel Statistical Eye Methodology*, DesignCon 2004.

[2] M Tsuk, D. Dvorscak, C.S. Ong, J. White, “An Electrical-Level Superposed-Edge Approach to Statistical Serial Link Simulation,” ICCAD ’09, November 2-5, 2009, San Jose California, USA. Copyright ©2009 ACM 978-1-60558-800-1/09/11.

[3] Li, Mike Peng, *Jitter, Noise, and Signal Integrity at High-Speed*. Prentice-Hall, Copyright ©2008.

[4] *Jitter analysis: The dualDirac model, RJ/DJ, and Q-scale*. White Paper, Agilent Technologies, Inc., Copyright ©2004.

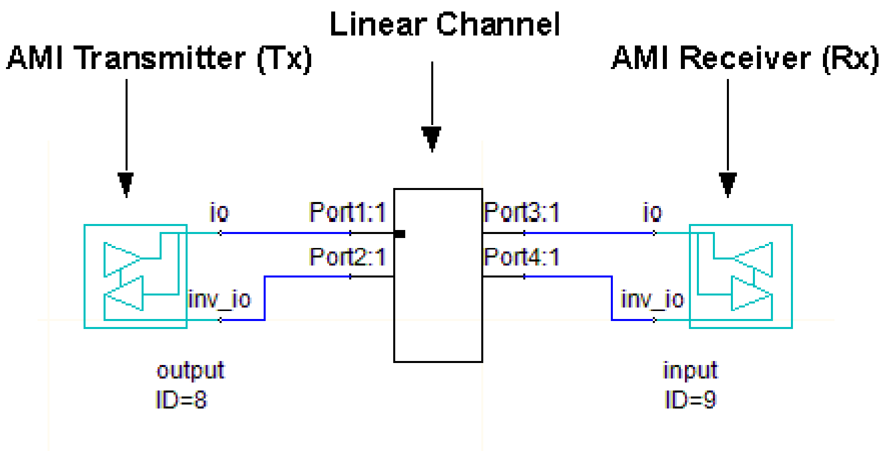
[5] Eric Evans, *ADN2817/ADN2818 BER Monitor and Sample Phase Adjust User Guide*, One Technology Way, Copyright ©2015.

AMI Analysis

The IBIS Algorithmic Modeling Interface (AMI) allows time-domain simulation of a linear-channel Circuit design using customer-supplied models for the transmitter and receiver. Reports include plots of step responses from VerifEye, eye diagrams like those available for QuickEye, and bit-error-rate (BER) contours.

AMI Analysis with IBIS Buffer Models

The Algorithmic Modeling Interface (AMI) simulates user-specified transmitter/receiver models, furnished as IBIS models and C-interface compiled AMI libraries, over a linear channel. The AMI Specification is published in the IBIS Specification (See AMI Analysis References).



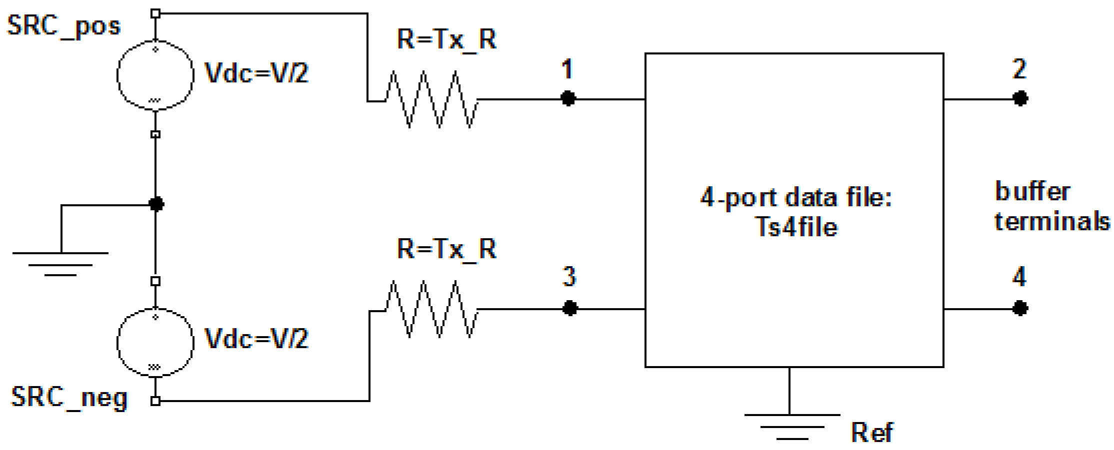
The first step is to either import a single AMI transmitter/receiver component or separate AMI transmitter and receiver components.

Continue to Importing AMI Components with IBIS Buffer Models.

The AMI Transmitter Model

The AMI transmitter consists of an Analog output model and three Algorithmic processing functions, Tx **AMI_Init**, Tx **AMI_GetWave**, and Tx **AMI_Close** that model equalization behavior and are provided in a .dll or .so. The IBIS specification allows the transmitter to omit the TX **AMI_GetWave** function.

The Analog output model can be specified using traditional IBIS methods (including linearized I/V tables and voltage ramps), or with 4-port frequency-dependent data from a Touchstone-format file containing the transmitter behavior.



When the transmitter model uses 4-port data, the reserved parameters **Ts4file** (name of Touchstone file), **TX_V** (transmitter voltage swing), and **Tx_R** (series resistances) control the transmitter operation.

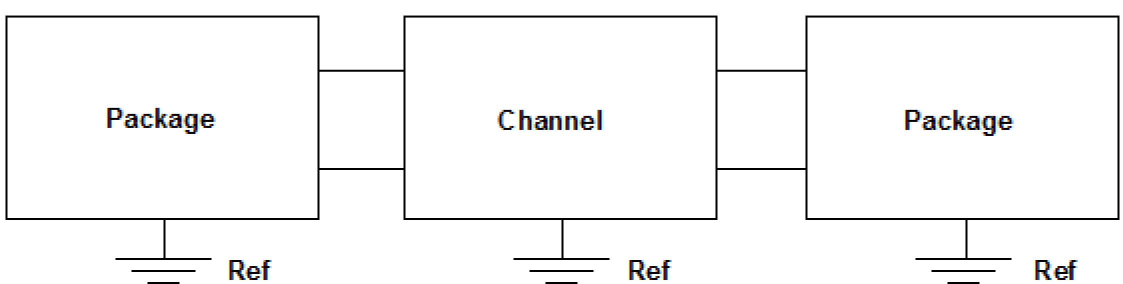
- The voltages of the voltage sources correspond to **V=Tx_V** for logic level 1 and **V=-Tx_V** for logic level 0.
- The series resistors have resistance **R=Tx_R**.

See "AMI Ts4 Analog Buffer Model Parameters " on page 11-150"AMI Ts4 Analog Buffer Model Parameters " on page 11-150 for details on these parameters.

The Channel Impulse Response

AMI Analysis assumes a linear channel. The channel is characterized by its impulse response function.

When 4-port Touchstone files are referenced for the AMI transmitter and receiver, the channel is assumed to include the package behavior in the impulse response:

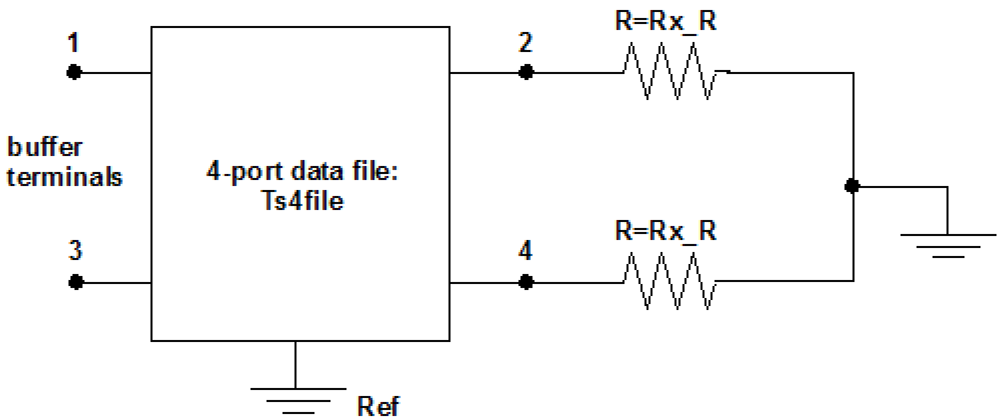


The AMI Receiver Model

The AMI receiver consists of an Analog input model and three Algorithmic processing functions that model equalization behavior and are provided in a .dll or .so: Rx **AMI_Init**, Rx **AMI_GetWave**, and Rx **AMI_Close**. The IBIS specification allows the receiver to omit the Rx **AMI_GetWave** function, unless it is part of an AMI retimer link (See AMI Repeaters).

The RX **AMI_GetWave** function in the receiver model can generate clock times internally, and return the clock ticks as parameters to the solver for post-processing. For AMI channel simulation, the results of clock tick sampling can be observed in the Statistical Eye Plot for the EyeAfterProbe quantity. For an AMI Retimer link, the clock times are used to generate a new digital waveform (See AMI Repeaters).

The Analog Receiver model can be specified using traditional IBIS methods or with 4-port frequency-dependent data from a Touchstone-format file containing the receiver behavior.



When the receiver model uses 4-port data, the reserved parameters **Ts4file** (name of Touchstone file), and **Rx_R** (terminating resistances) control the Receiver analog section operation. The terminating resistors have resistance **R=Rx_R**. See "AMI Ts4 Analog Buffer Model Parameters " on page 11-150 for details on these parameters.

Importing AMI Components with IBIS Buffer Models

This section contains the following topics:

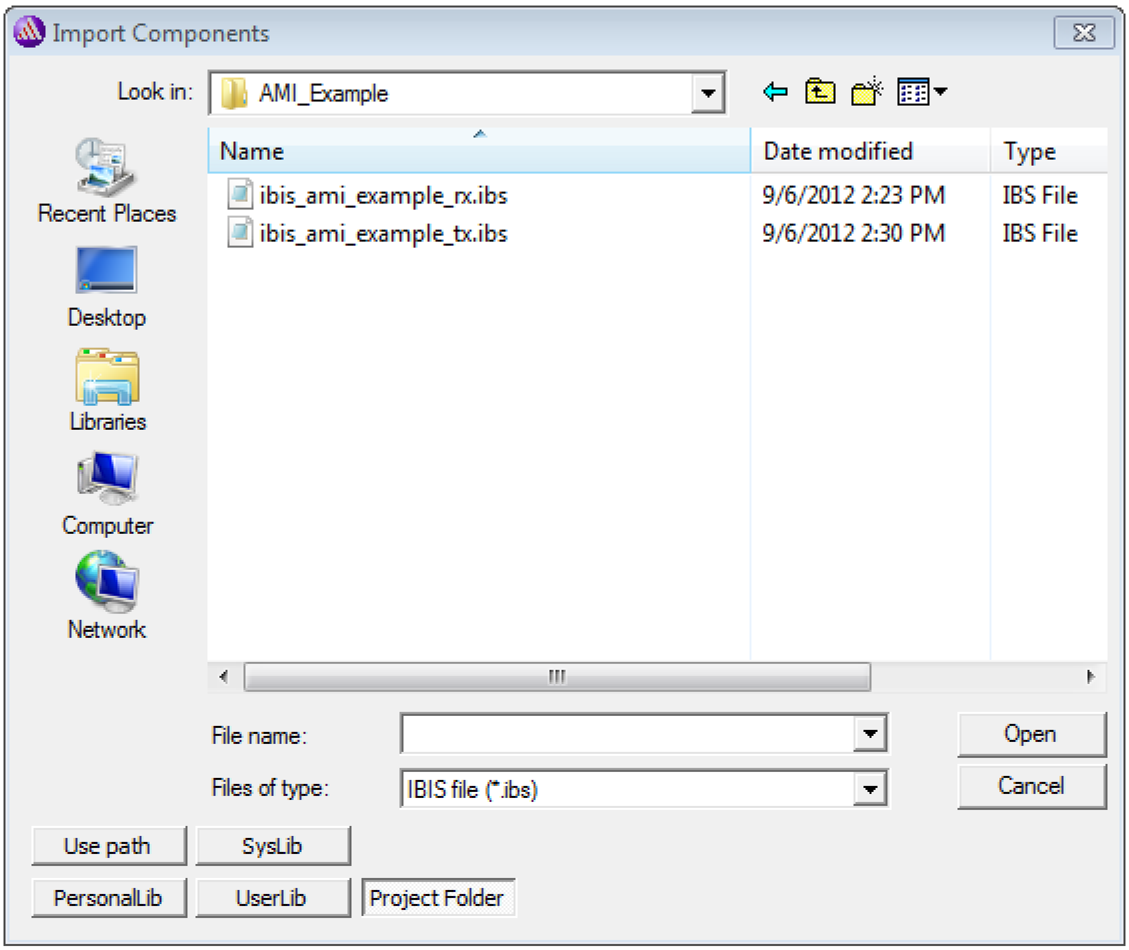
- Importing the AMI Transmitter
- Importing the AMI Receiver

- Importing an AMI Transmitter/Receiver

Importing the AMI Transmitter

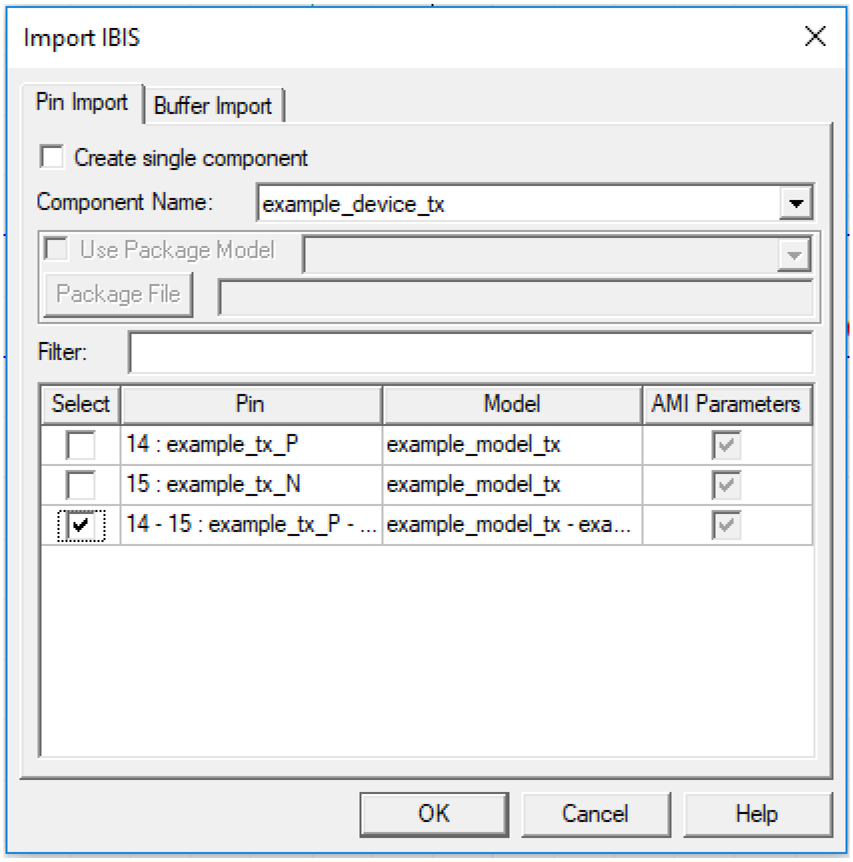
To import an IBIS AMI transmitter model from an IBIS library file (.ibs):

1. View the **Components Manager**. Open the **Symbols** group box. Click **Import Models**.
2. From the group box of icons, select the **IBIS** icon. This opens the **Import Components** window:



3. Specify the location of the IBIS file by clicking **Use Path**, **PersonalLib**, **UserLib**, or **SysLib**.
4. Locate and select the IBIS file that references the AMI transmitter.

5. Click **Open** to open the **Import IBIS** window.



6. The **Pin Import** tab displays the pins defined for the component in the IBIS file, the model or model selector associated with each pin, and whether the model is an AMI model. Single pins are listed first, then differential pin pairs. On this tab:

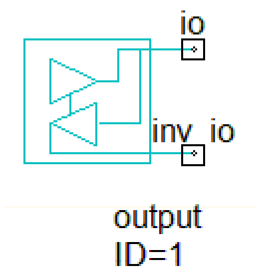
   a. To import the transmitter as a single component with multiple pins wherein each pin retains its own model and buffer type, check **Create single component**. See Import IBIS Single Component or Pin or Buffer Element for details.

   b. The **Component Name** field lists all the IBIS components contained in the selected IBIS library. Scroll the list and select the name of the component to be imported.

   c. Click the check box to select the pin or differential pair of pins to import.

   > **Note:** If an attempt is made to import a single pin that is part of a differential pin pair, a warning appears in the Message Manager. The import can still proceed.

7. Click **OK** to import the transmitter model. An instance of the component is attached to the cursor so you can drag and drop it into the schematic.

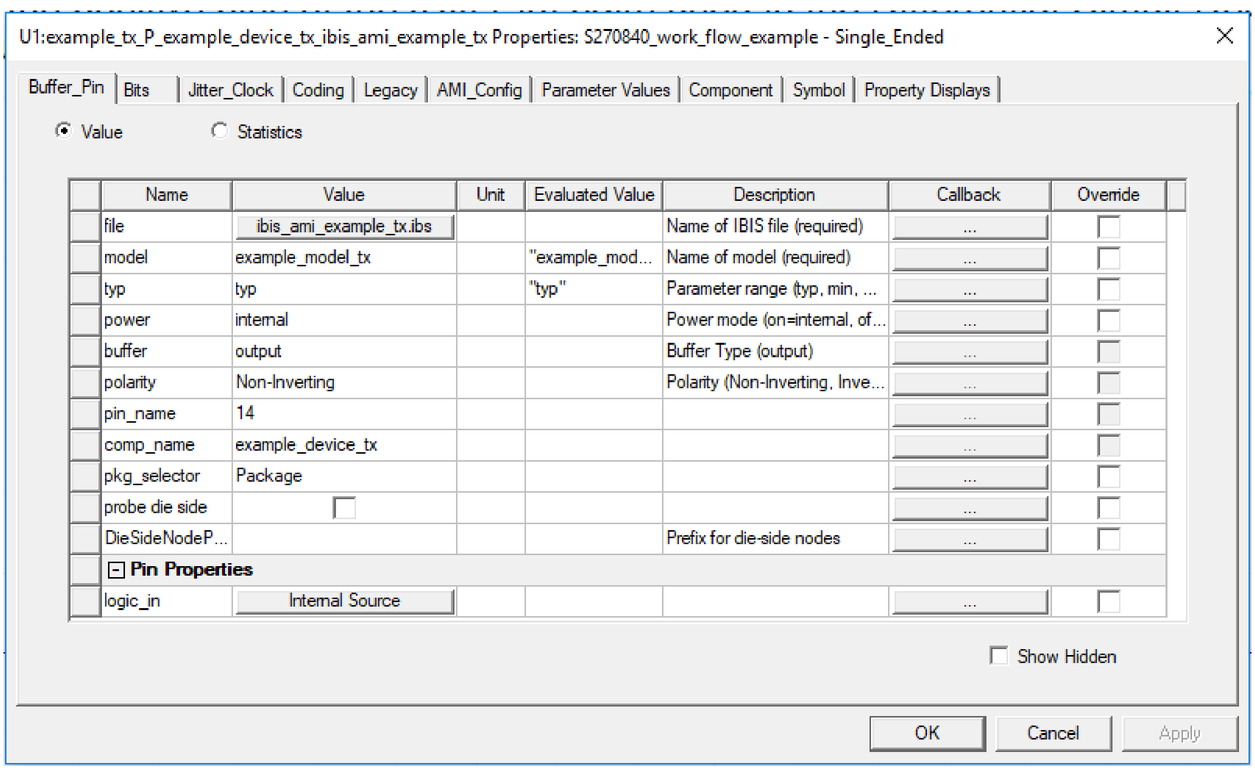


The next step is to edit the properties on the transmitter component.

Set AMI Transmitter Properties

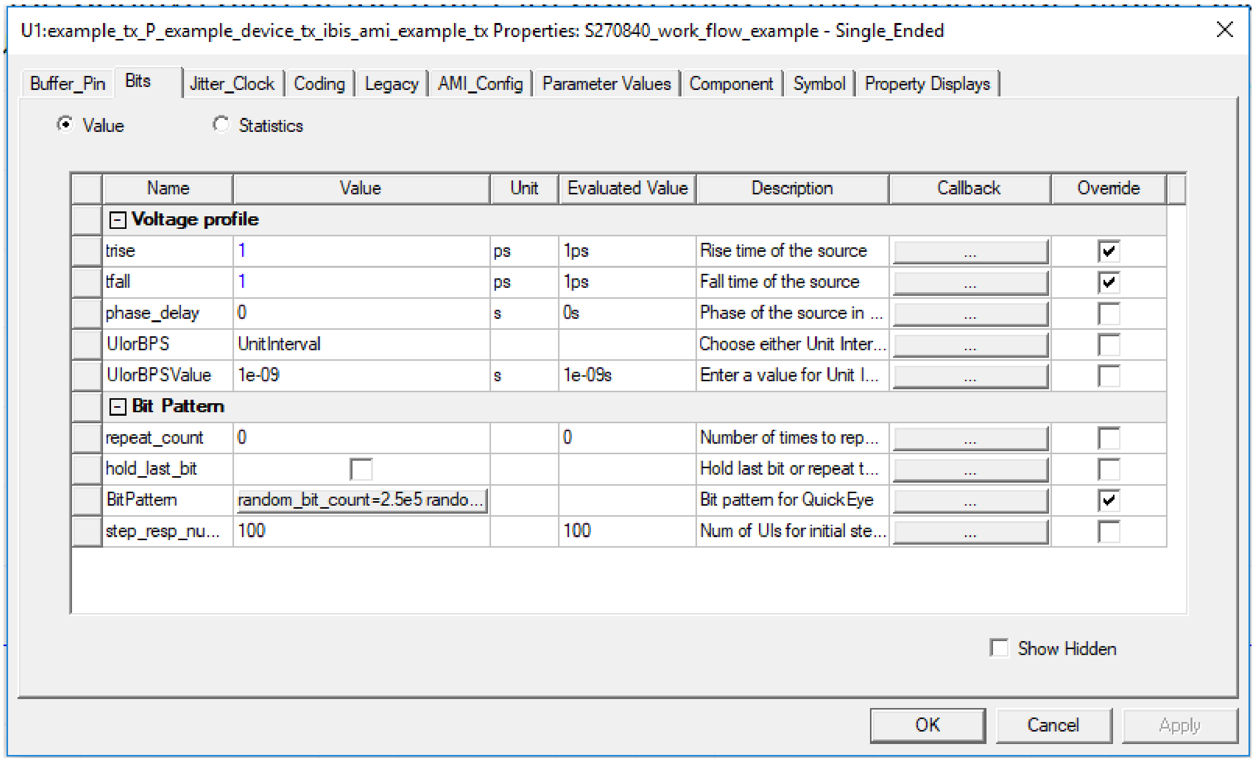
To view or set the properties on the AMI transmitter:

1. Select the component and click **Edit Properties** on the menu. Here are the basic buffer properties for an imported single-ended AMI transmitter example:



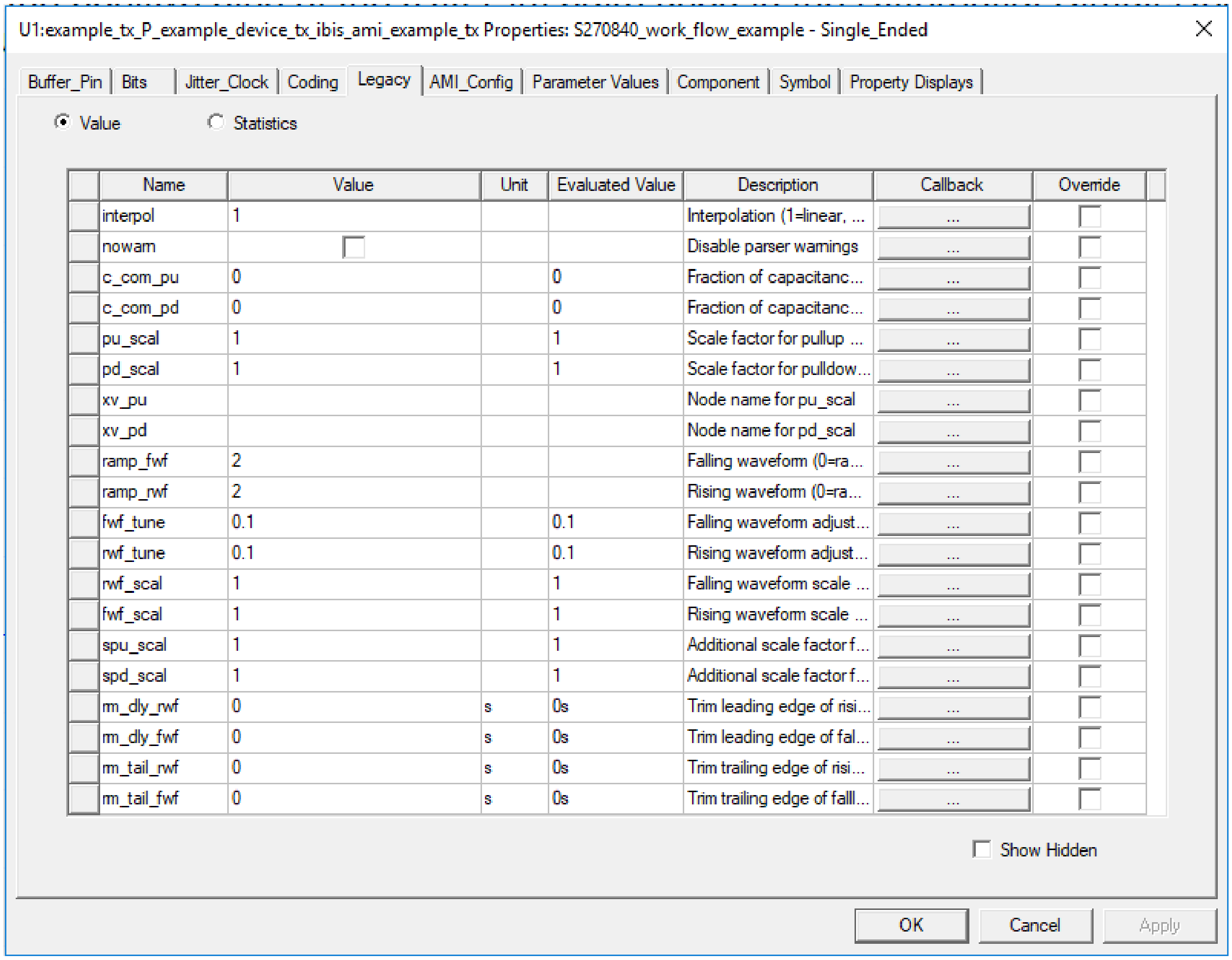
An AMI transmitter model is imported as an **output or I/O** buffer type with internal power source and other basic parameters set as shown. See IBIS Library Support for additional information.

2. In the Pin properties for the AMI transmitter, the **logic_in** pin is set to **Internal Source**. With this setting, Nexxim displays the Eye Source parameters for the transmitter to use when generating the waveform on the **Bits**, **Jitter_Clock**, and **Coding** tabs.



See IBIS Library Support or Add a QuickEye Source to a Schematic for more information on the eye source properties.

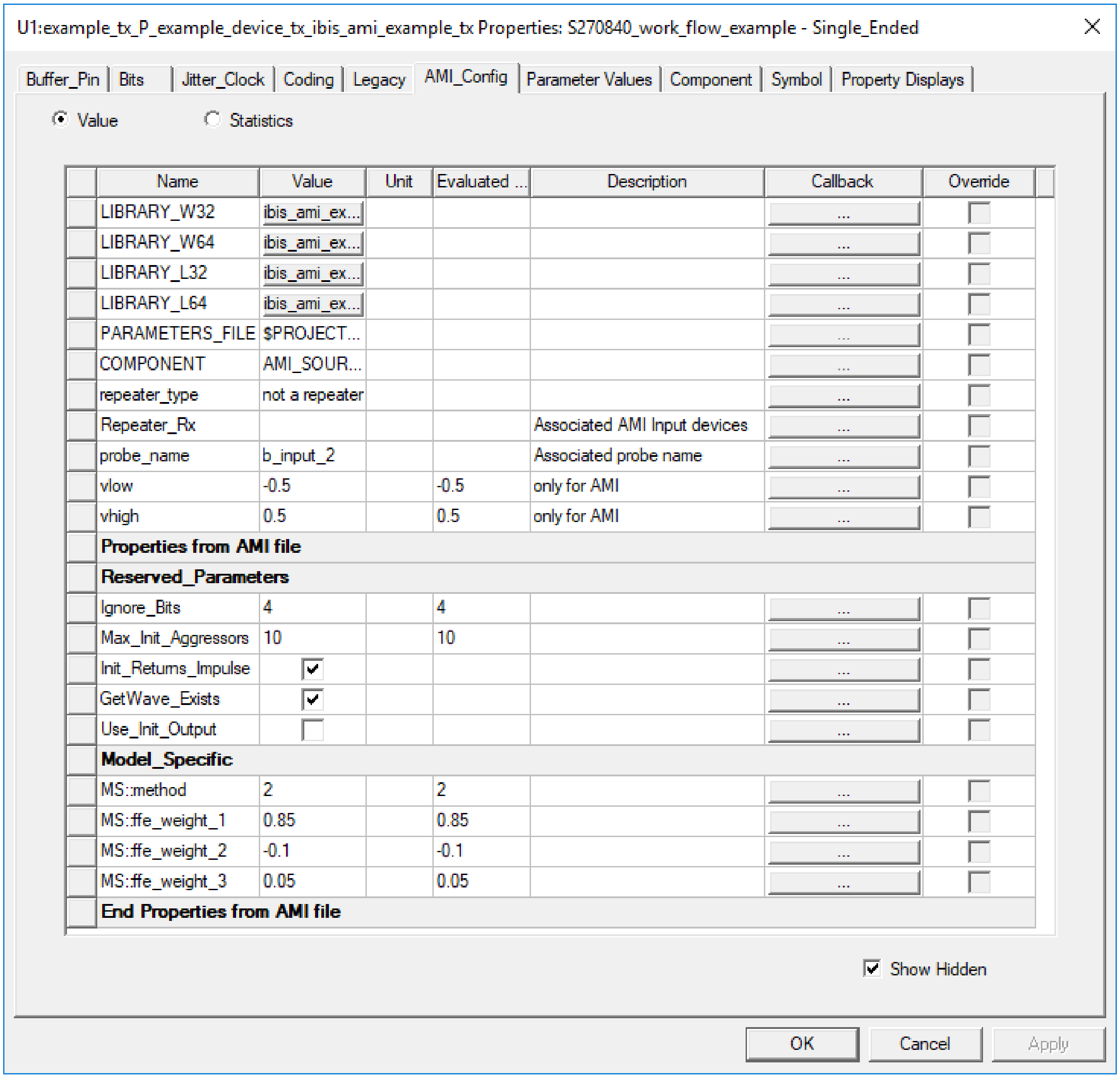
3. The AMI transmitter properties also include the waveform properties on the **Legacy** tab. These are rarely used for AMI simulations.



- See IBIS Library Support for more information on the waveform properties.

4. When the IBIS Pin Import selection references an Algorithmic Model (AMI), the properties include the AMI file references and any AMI parameters that are made user-visible in the AMI file. At import time, the AMI files are parsed and validated. Parameters from the AMI file are added to the component property list.

Here is a listing of the properties for an AMI Transmitter output model:



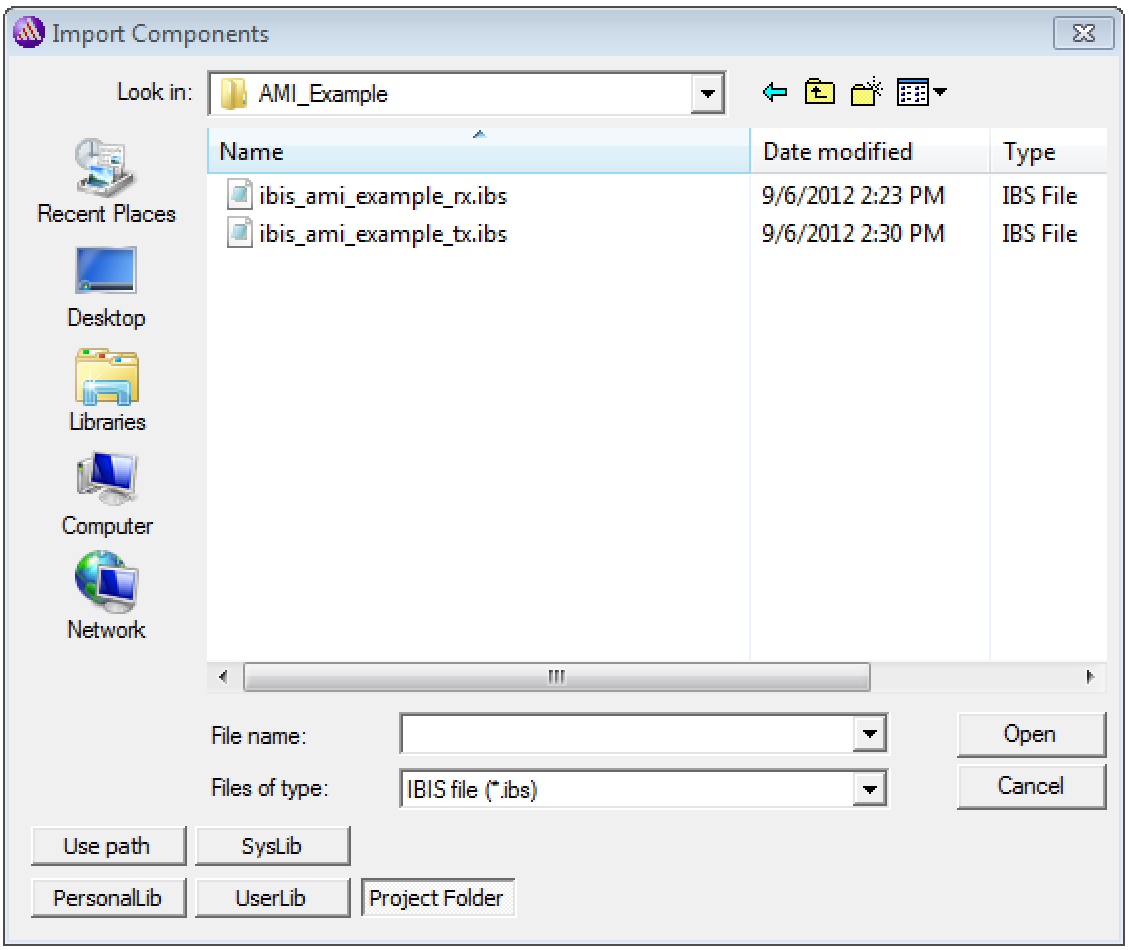
- The **LIBRARY** properties show the executable files on the .ibs library file.
- The **COMPONENT** property has been set to AMI_SOURCE automatically.
- See AMI Repeater Properties for details on the **repeater_type** and **Repeater_Rx** properties.
- The **vlow** and **vhigh** properties have been set to -0.5V and +0.5V, respectively, per the IBIS AMI specification.
- The Transmitter AMI properties list includes a property **probe_name** which lists the available AMI Receivers in the design. Then choose a receiver on the drop-down menu.

- The **probe_name** field is automatically filled in with the name of the AMI receiver when the AMI Source and AMI Probe are paired on the GUI. See Pairing an AMI Source and an AMI Probe in the Schematic.
- The **AMI_Config** tab includes listings of the parameters on the AMI parameters file. The reserved and model specific AMI parameters are broken out in separate groups

Importing the AMI Receiver

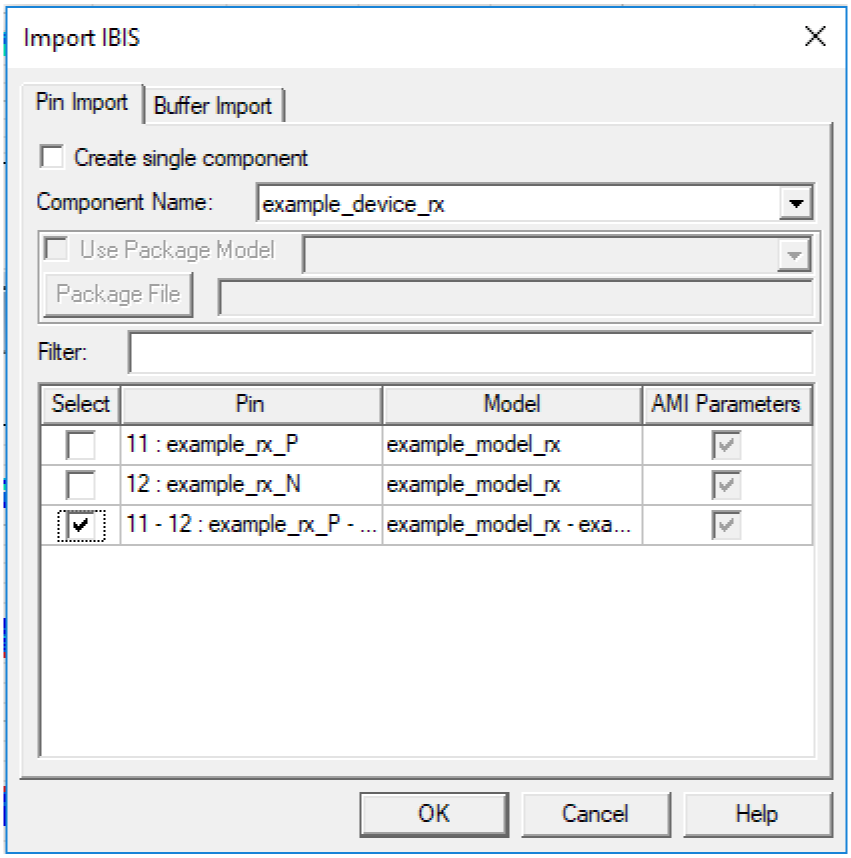
To import an IBIS AMI receiver model from an IBIS library file (.ibs):

1. Click **Import Models** on the **Symbols** tab of the **Component Libraries** group box. Select the **IBIS** icon. This opens the **Import Components** window:



2. Locate and select the IBIS file that references the AMI receiver.

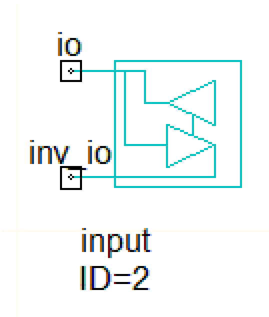
3. Click **Open** to open the **Import IBIS** window.



4. The **Pin Import** tab displays the pins defined for the component in the IBIS file, the model or model selector associated with each pin, and whether the model is an AMI model. Single pins are listed first, then differential pin pairs. On this tab:
   a. To import the receiver as a single component with multiple pins wherein each pin retains its own model and buffer type, check **Create single component**. See Import IBIS Single Component or Pin or Buffer Element for details.
   b. The **Component Name** field lists all the IBIS components contained in the selected IBIS library. Scroll the list and select the name of the component to be imported.
   c. Click the **Select** check boxes to designate the differential pair of pins to import.

   > **Note:** If an attempt is made to import a single pin that is part of a differential pin pair, a warning appears in the Message Manager. The import can still proceed.

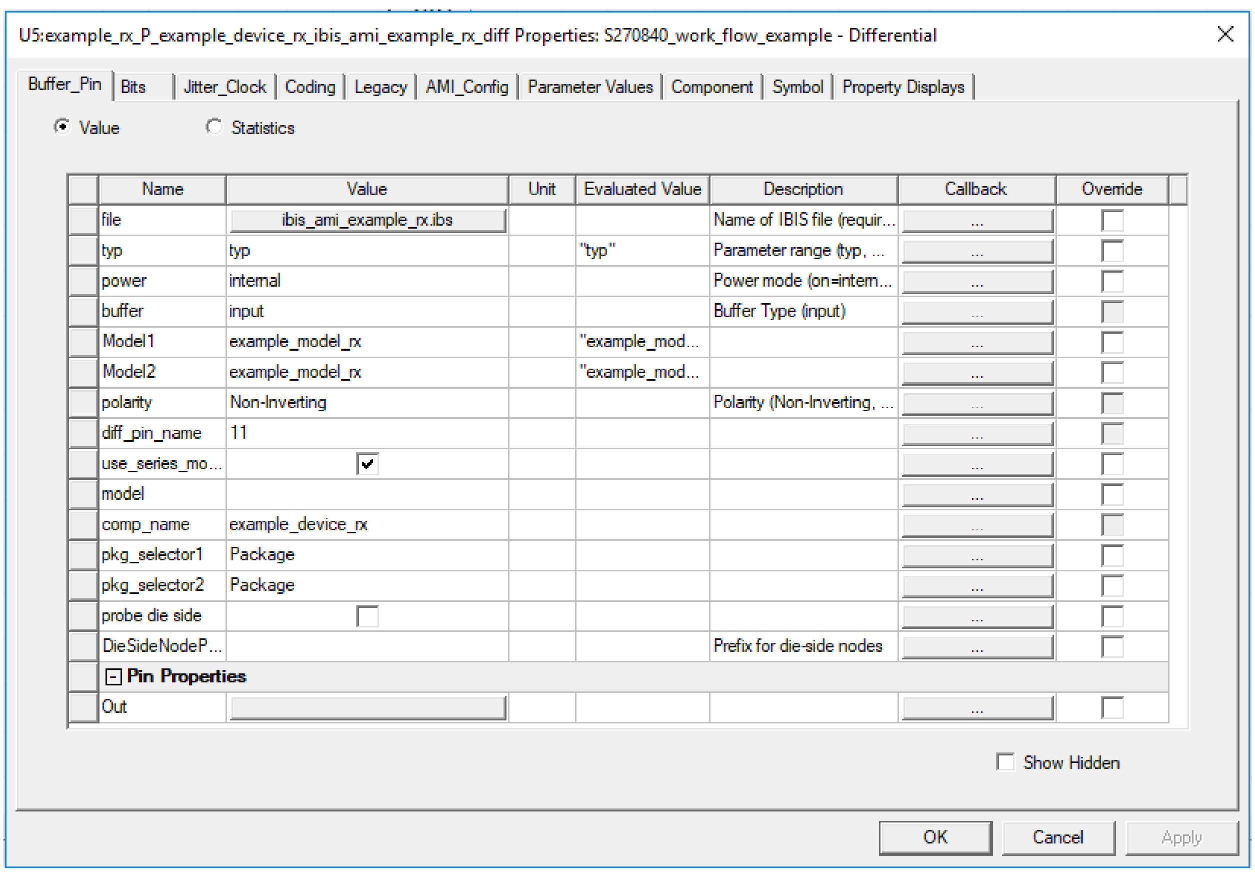
5. Click **OK** to import the transmitter model. An instance of the receiver is attached to the cursor. Drag and drop it into the schematic.



The next step is to edit the properties on the receiver component.

Setting AMI Receiver Properties

To view or set the properties on the AMI receiver, select the component and click **Edit Properties** on the menu. Here are the basic buffer properties for an imported differential pair AMI receiver example:



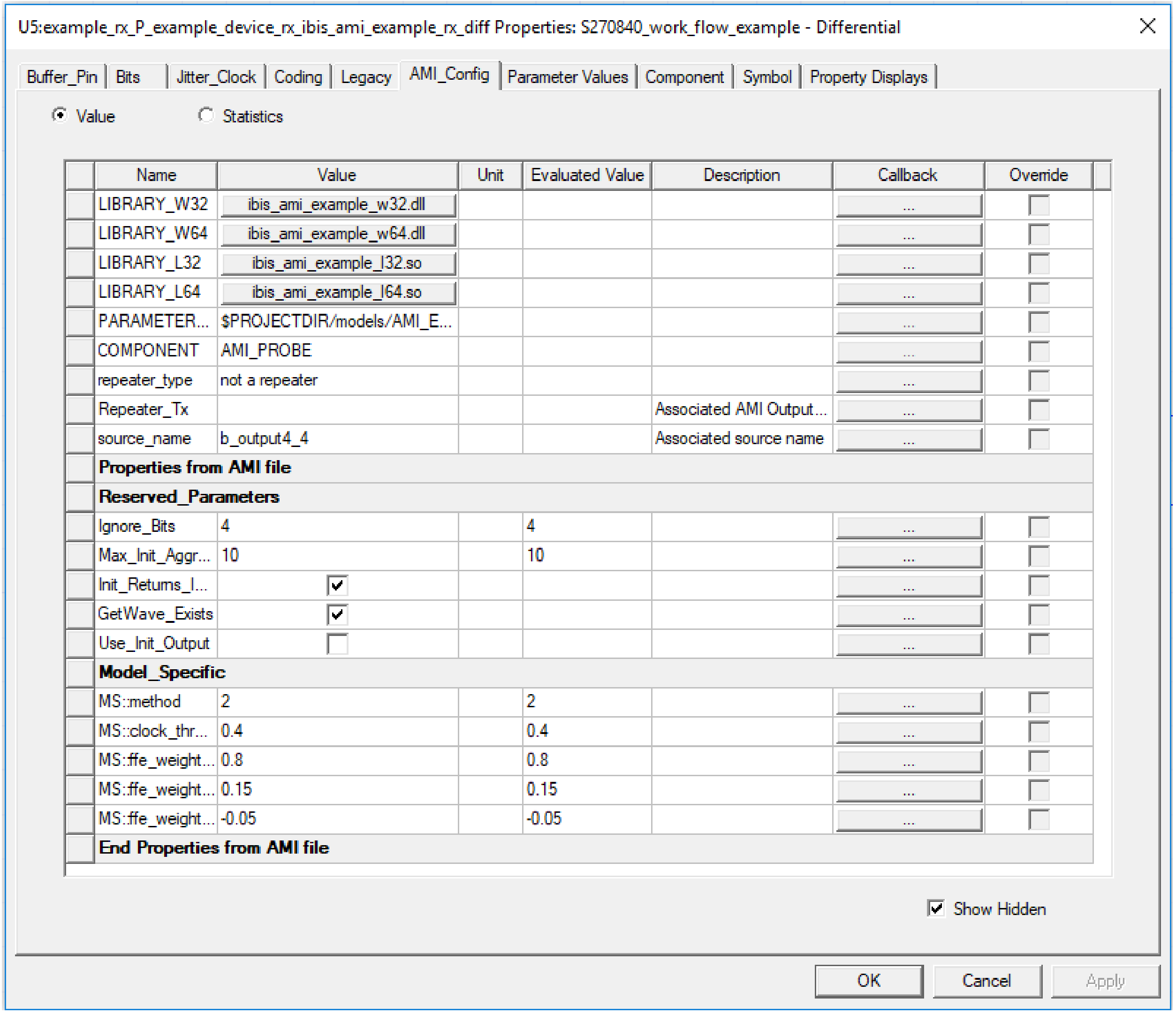
An AMI receiver model is imported as an **input or I/O** buffer type with internal power source and other basic parameters set as shown. See IBIS Library Support for additional information.

- The **Out** property under the Pin Properties row, is set using a net name, selected from a drop-down menu populated automatically on the completed circuit. See IBIS Library Support for additional information. The **Out** pin is not relevant to AMI analysis.
- The **diff_pin_name** property shows the **Pin Import** selection.
- See IBIS Library Support for more information on the **use_series_model**, **package_selector**, and **probe die side** properties.

This is the AMI Parameters Tab information:

- The **LIBRARY** properties show the executable files on the [Model] in the .ibs file.
- The **COMPONENT** property has been set to AMI_PROBE automatically.
- See AMI Repeater Properties for details on the **repeater_type** and **Repeater_Tx** properties.
- The AMI Receiver properties list includes a property **source_name** which lists the available AMI Transmitters in the design. Then choose a transmitter on the drop-down menu.
- The **source_name** field is automatically filled in with the name of the AMI Transmitter when the AMI Transmitter and AMI Receiver are paired on the GUI. See Pairing an AMI Source and an AMI Probe in the Schematic.

Finally, the AMI_Config tab includes listings of the parameters on the AMI parameters file. The reserved and model specific AMI parameters are broken out in separate groups.



If your system includes AMI Repeaters or Retimers, the steps for setting them up are in AMI Repeater Properties.

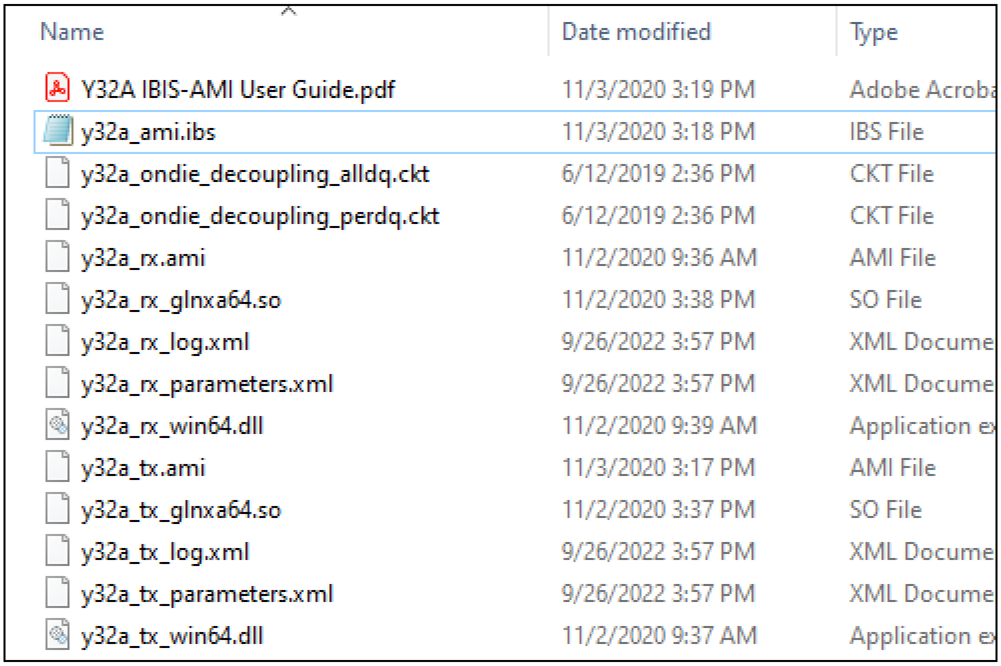
You complete the circuit by adding a channel and wiring the connections. If the circuit is complete, you are ready to run the AMI Analysis.

Importing a Bi-directional AMI Transmitter/Receiver

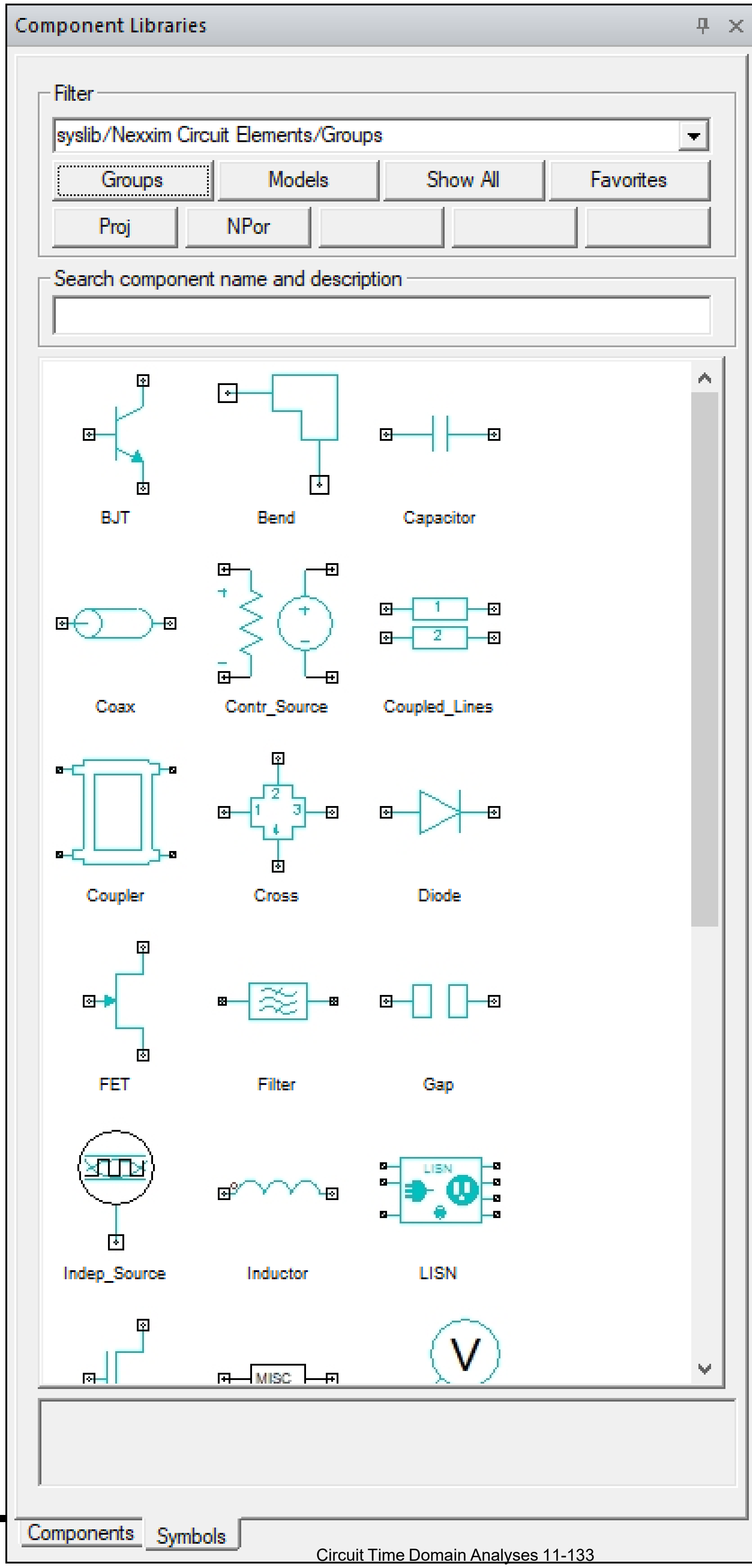
Complete the following steps to import an IBIS bi-directional AMI transmitter/receiver model from an IBIS library file (*.ibs*).

1. Import an *.ibs* file by doing one of the following:

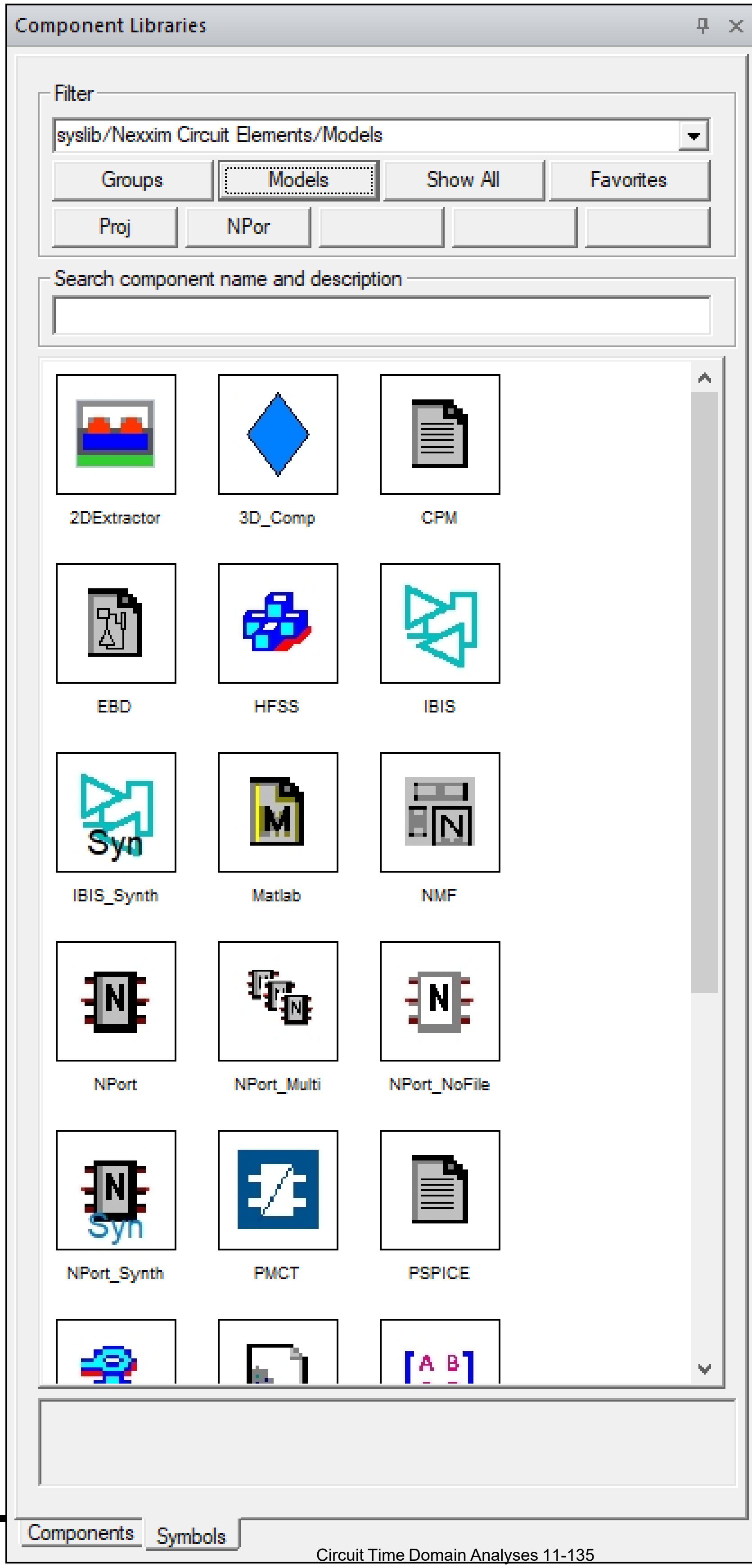
   - Navigate to the appropriate *.ibs* file from an explorer window. Then **drag+drop** the file into the **Electronic Desktop Schematic Editor** to open the **Import IBIS** window.



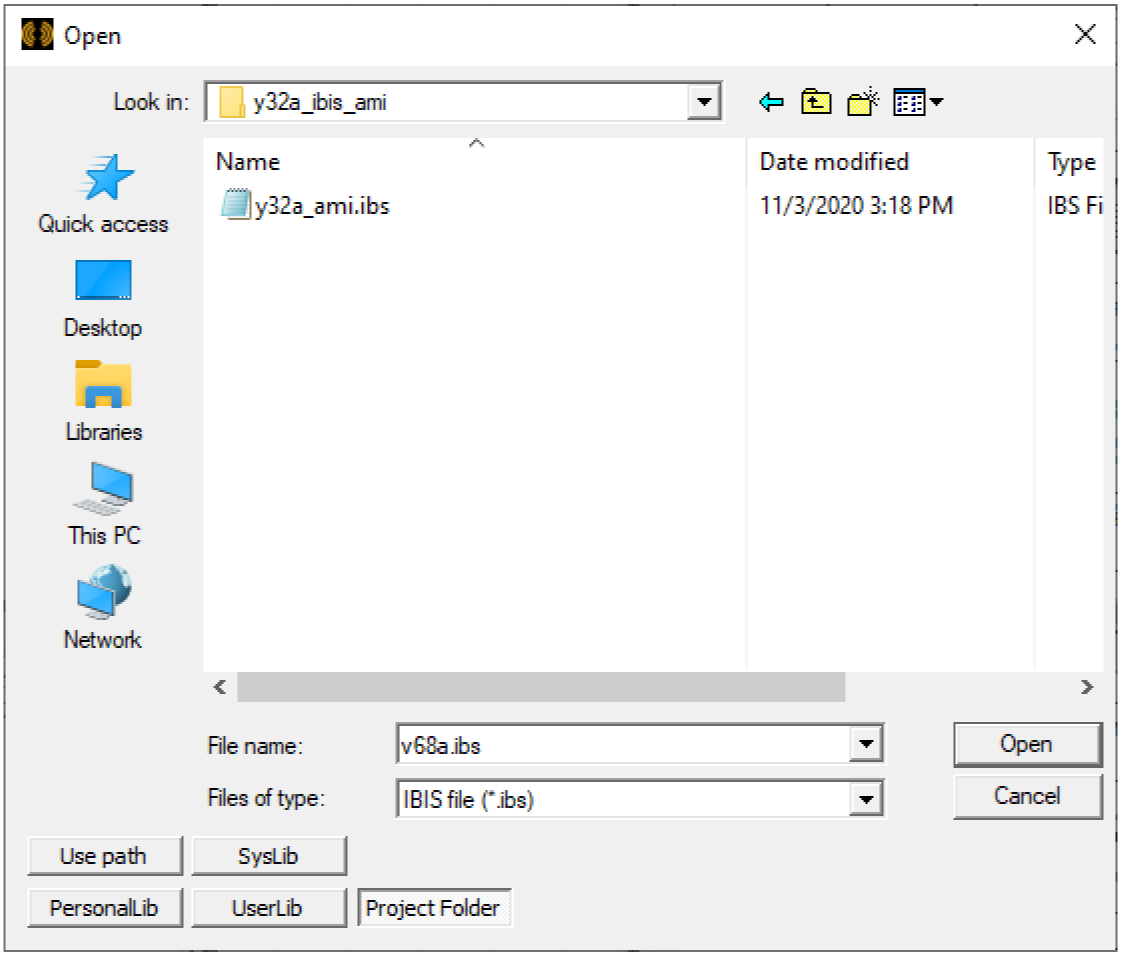
- Navigate to the **Component Libraries** window > **Symbols** tab.


Component Libraries
Filter
syslib/Nexxim Circuit Elements/Groups
Groups
Models
Show All
Favorites
Proj
NPor
Search component name and description
BJT
Bend
Capacitor
Coax
Contr_Source
Coupled_Lines
Coupler
Cross
Diode
FET
Filter
Gap
Indep_Source
Inductor
LISN
Components
Symbols

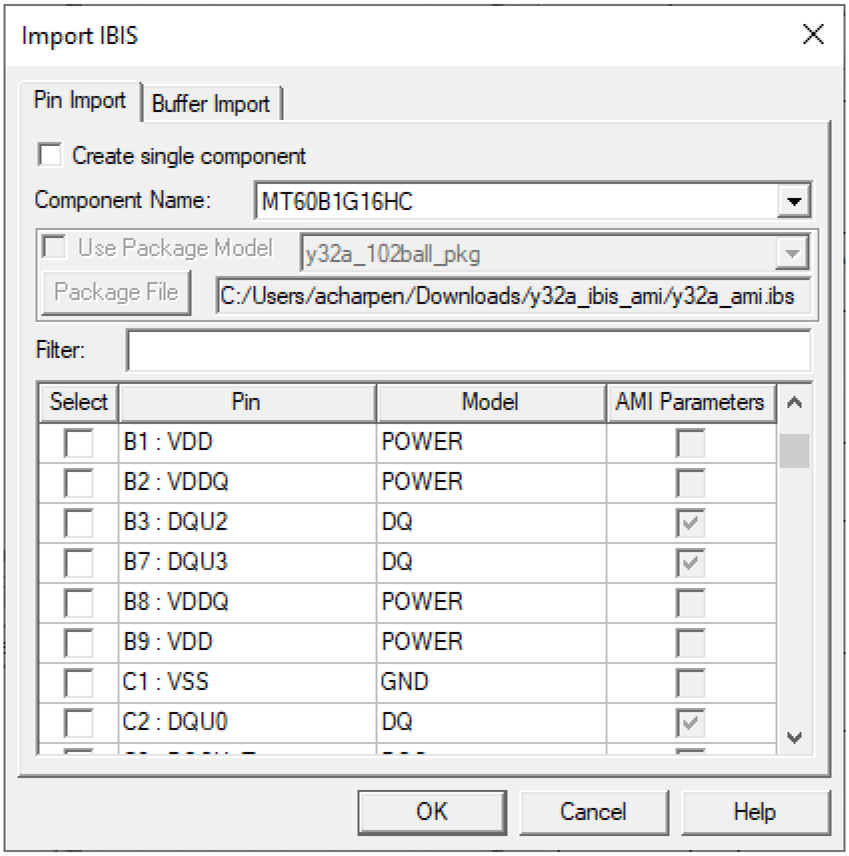
From the **Filter** group box, click **Models**.


Component Libraries
Filter
syslib/Nexxim Circuit Elements/Models
Groups
Models
Show All
Favorites
Proj
NPor
Search component name and description
2DExtractor
3D_Comp
CPM
EBD
HFSS
IBIS
Syn
IBIS_Synth
Matlab
NMF
NPort
NPort_Multi
NPort_NoFile
Syn
NPort_Synth
PMCT
PSPICE
Components
Symbols

Click **IBIS** to open an explorer window. Then navigate to the appropriate *.ibs* file. Select the file and click **Open** to open the file in the **Import IBIS** window.



2. The **Import IBIS** > **Pin Import** tab displays the component's pins identified in the IBIS file, the model or model selector associated with each pin, and whether the model is an AMI model. Single pins are listed first, then differential pin pairs.



From the **Pin Import** tab, do the following:

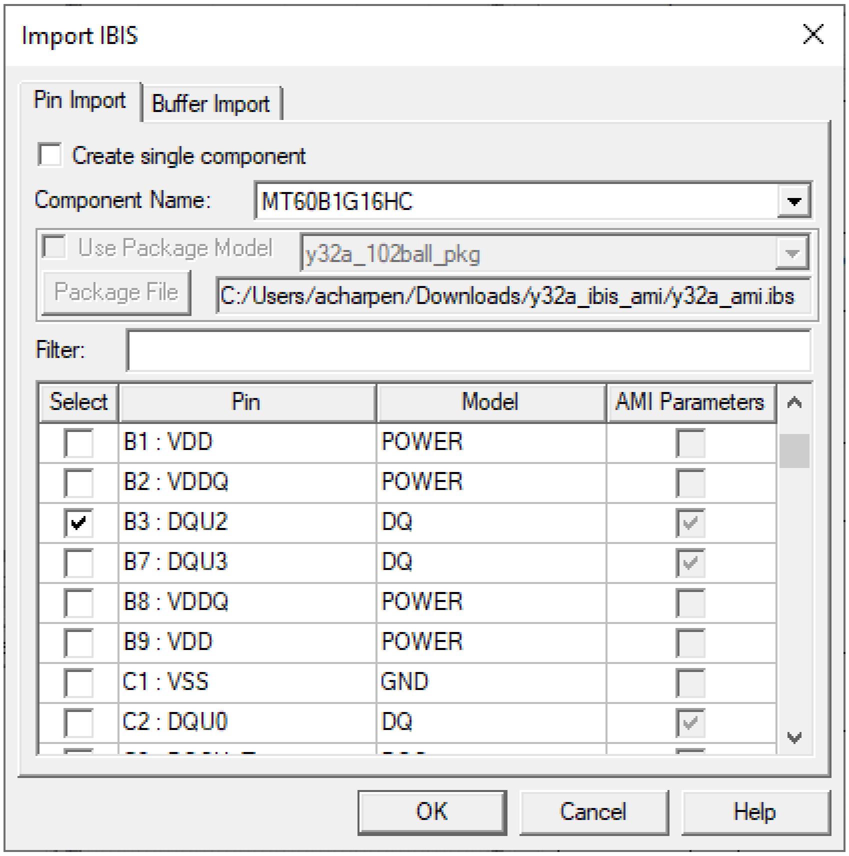
a. To import the *.ibs* file as a single component with multiple pins wherein each pin retains its own model and buffer type, check **Create single component**. See Import IBIS Single Component or Pin or Buffer Element for details.

b. The **Component Name** drop-down menu lists all the IBIS components contained in the *.ibs* file (i.e., the IBIS library). Select the appropriate component.

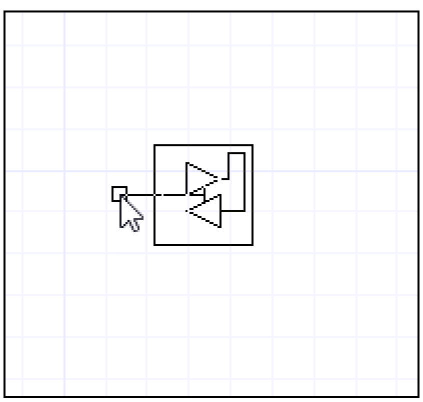
c. Check boxes in the **Select** column to choose which pins to import (i.e., check the box adjacent to the component(s) with the appropriate AMI parameters).

> **Note:** If an attempt is made to import a single pin that is part of a differential pin pair, a warning appears in the **Message Manager**. The import can still proceed.

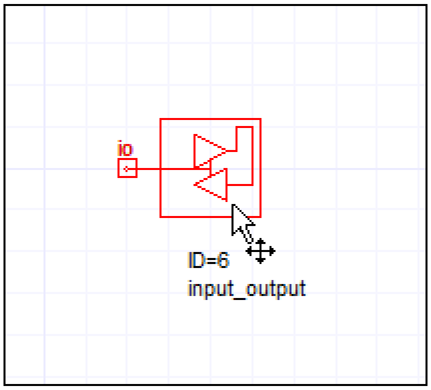
d. Click **OK** to close the **Import IBIS** window.



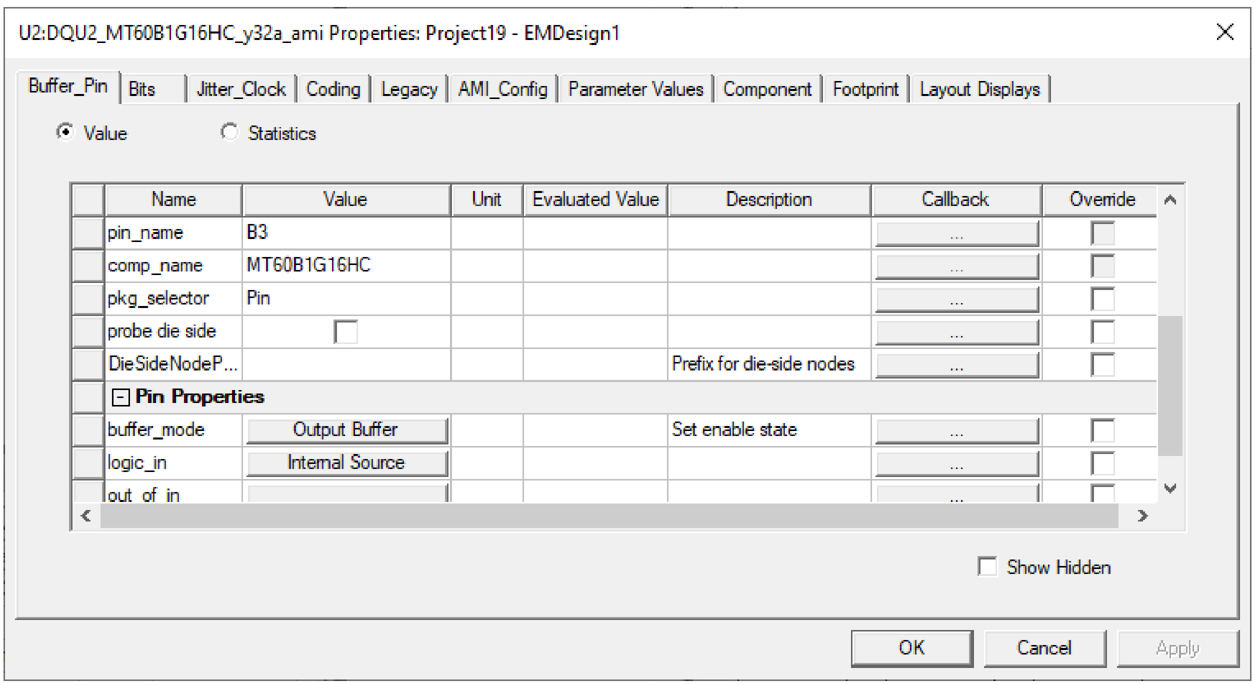
3. An instance of the component is now pinned to the cursor.



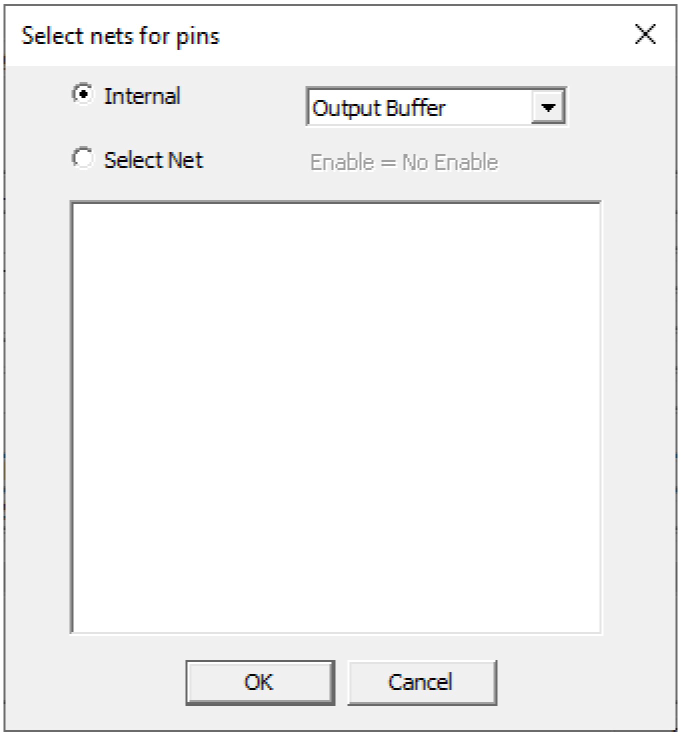
4. **Drag+click** an appropriate place in the **Schematic Editor** to drop the component.



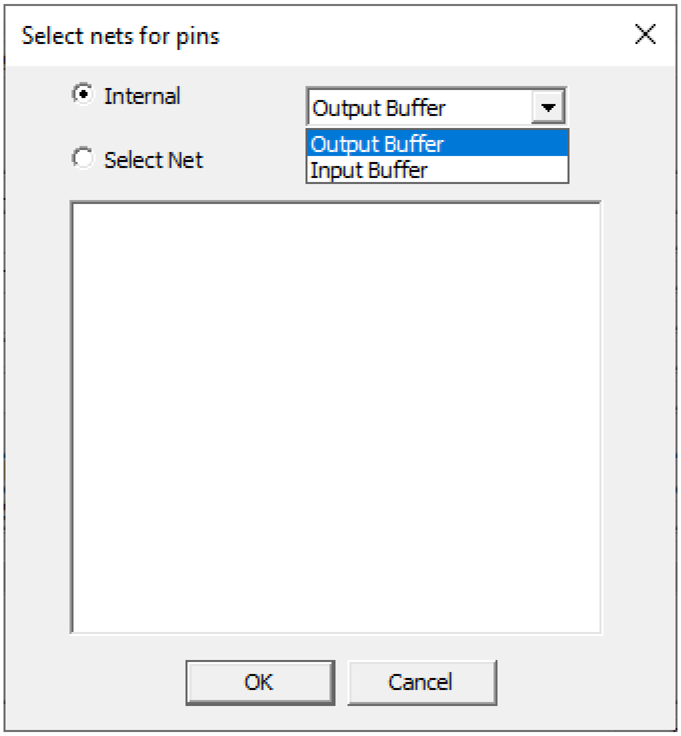
5. Double-click the component to open the **Properties** window. Then scroll to the **buffer_ mode** row.



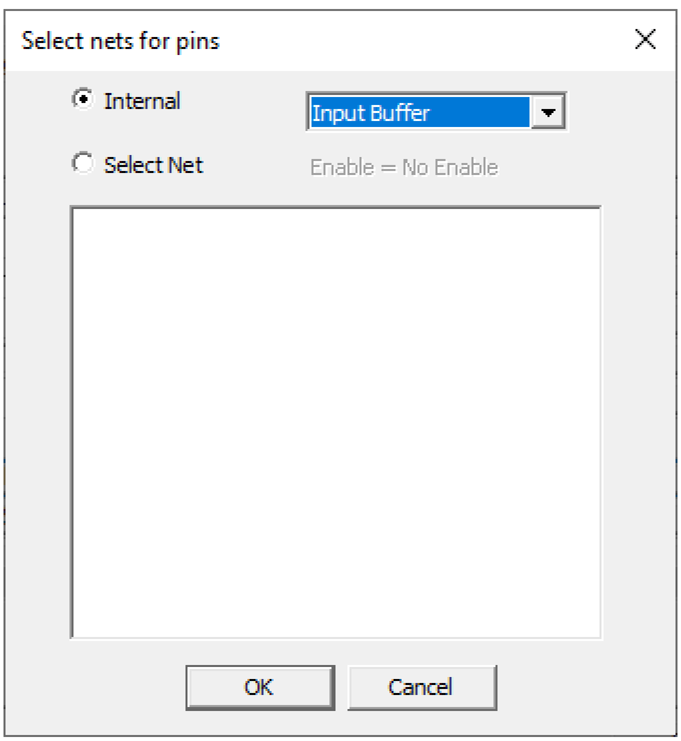
6. Click the button in the **buffer_mode Value** field to open the **Select nets for pins** window.



7. Select the appropriate buffer from the **Internal** drop-down menu (e.g., **Output Buffer** or **Input Buffer**).

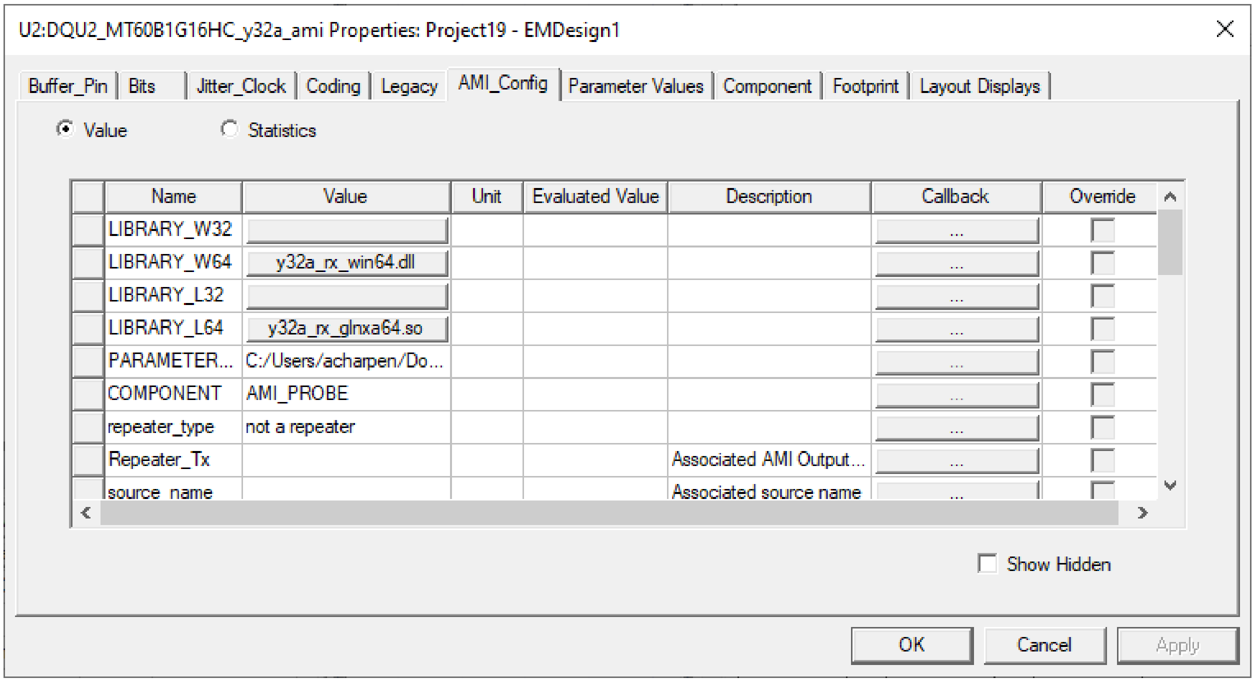


8. Click **OK** to close the **Select nets for pins** window.

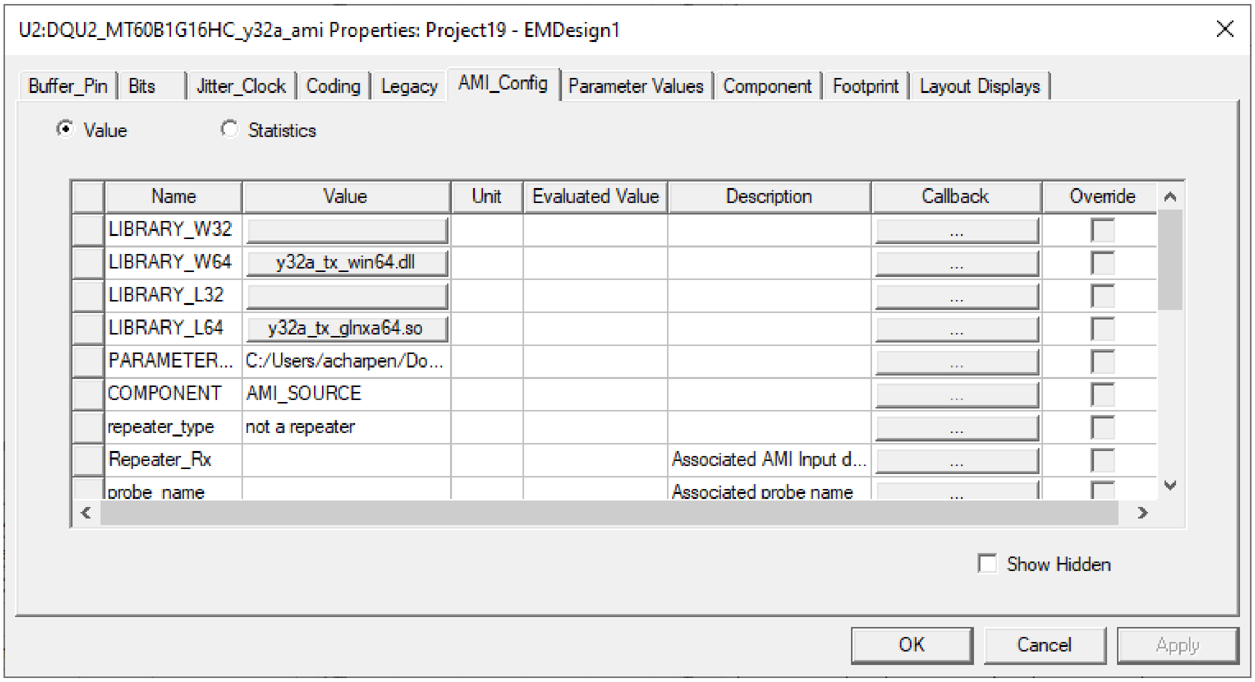


9. Navigate to the **AMI_Config** tab. The appropriate library files will have immediately populated the **LIBRARY Value** fields after making a selection in **Steps 7-8**.

   The following example shows the fields populated with receiver files (e.g., **y32a_rx_win64.dll** and **y32a_rx_glnxa64.so**) after **Input Buffer** is chosen.



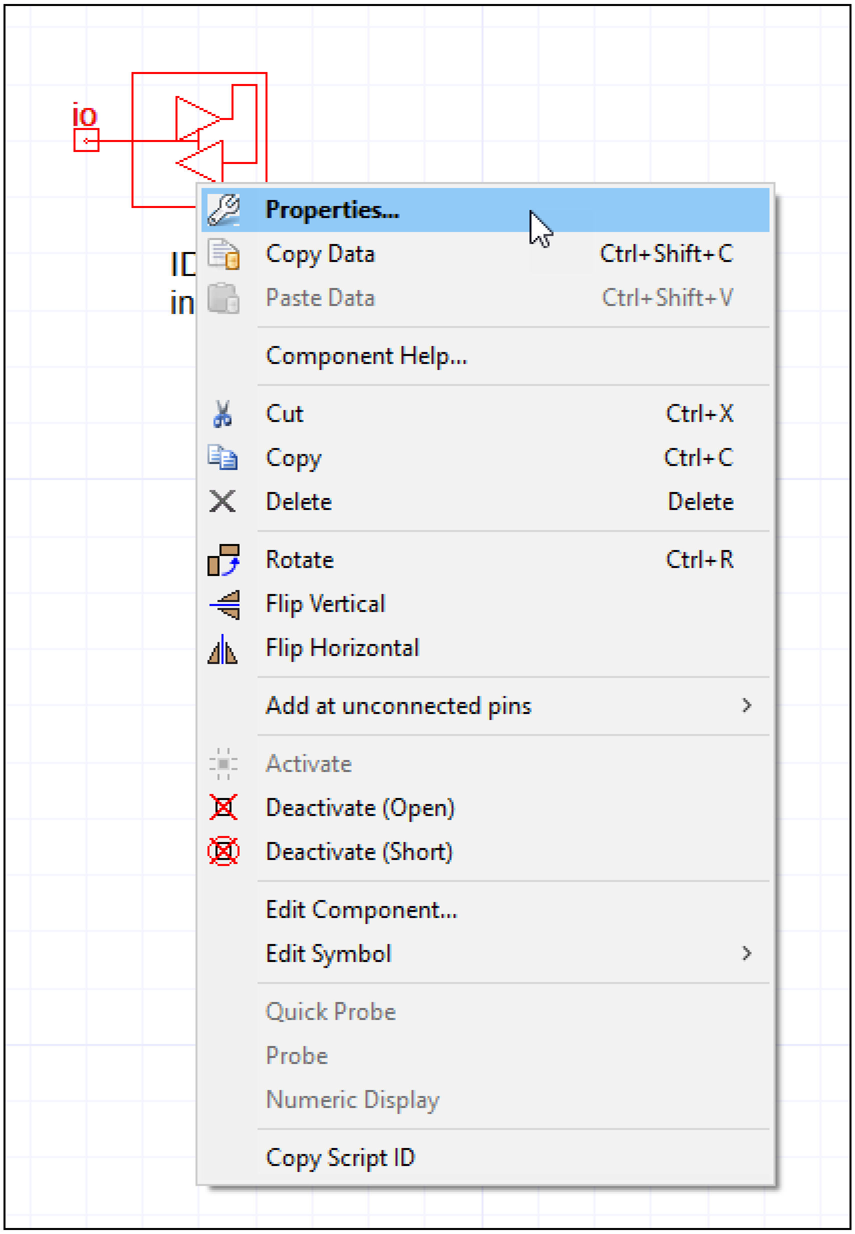
   The following example shows the fields populated with the appropriate transmitter files (e.g., **y32a_tx_win64.dll** and **y32a_tx_glnxa64.so**) after **Output Buffer** is chosen.



10. Complete the circuit by adding a channel and wiring the connections. Refer to the following subsection for more information about the **Properties** window. When the circuit is complete, continue to Set Up and Run AMI Analysis.

Setting AMI Transmitter/Receiver Properties

Right-click the component and select **Properties...** to open the **Properties** window.



> **Note:** For bidirectional Tx/Rx AMI models, changing the model from the **Properties** dialog requires clicking **OK** and then reopening the **Properties** dialog to populate all of the model's property fields.

An AMI transmitter/receiver model is imported as an **I/O** buffer type with an internal power source and basic parameters. Refer to IBIS Library Support for more information about the parameters.

The **Buffer_Pin** tab has parameters that are not exclusive to AMI models, such as the following examples

- The **out_of_in** property under the **Pin Properties** row, is set using a net name, selected from a drop-down menu populated automatically. Refer to IBIS Library Support for additional information.
- The **file** property shows the **Pin Import** selection (i.e., **y32a_ami.ibs**).
- See IBIS Library Support for more information on the **model**, **pkg_selector**, and **probe die side** properties.

The **AMI_Config** contains AMI-specific parameters, such as the following examples:

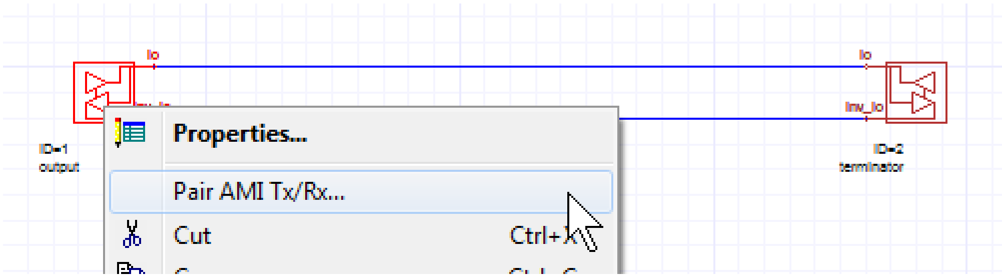
- The **LIBRARY** properties show the executable files extracted from the *.ibs* file.
- The **COMPONENT** property has been set to **AMI_SOURCE** automatically.
- Refer to AMI Repeater Properties for information about the **repeater_type** and **Repeater_Tx** parameters.
- The *.ibs* file may include other parameters that populate underneath the **Properties from AMI file** row. The parameters may even distribute into subrows (e.g., **Reserved_ Parameters**).

If the design includes AMI Repeaters or Retimers, refer to AMI Repeater Properties.

Pairing an AMI Transmitter and an AMI Receiver in the Schematic

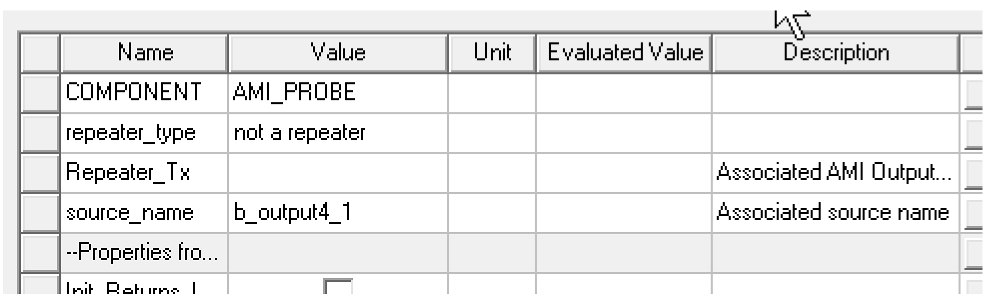
An AMI circuit may contain multiple AMI Transmitters and AMI Receivers. Each AMI Transmitter must be paired with a corresponding AMI Receiver, and vice versa. The **probe_name** parameter in the AMI Transmitter has a drop-down menu that lists all the AMI Receivers in the design. The AMI Receiver **source_name** parameter has a drop-down that lists all the AMI Transmitters in the design.

Instead of using the drop-down menus, pair a selected Transmitter and Receiver directly in the schematic. Use **Ctrl**+click to select the AMI Transmitter and AMI Receiver that you want to pair up. With the cursor on either component, right-click and select **Pair AMI Tx/Rx ...** on the menu.



The **probe_name** parameter in the selected AMI Transmitter is set to the name of the selected AMI Receiver.

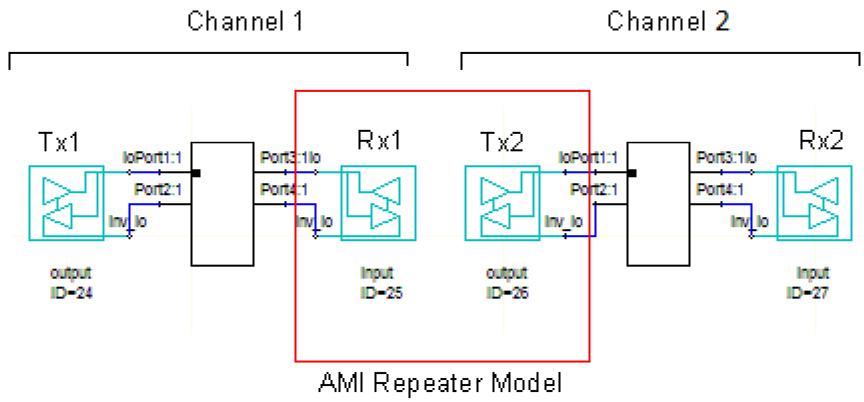
The **source_name** parameter in the selected AMI Receiver is set to the name of the selected AMI Transmitter.



AMI Repeater Properties

A repeater is a device added to a channel to compensate for channel loss. The AMI model of a repeater links two AMI channels together. The AMI Repeater model simulates a repeater connection between the Receiver on the upstream channel and the transmitter on the downstream channel. See AMI Repeaters in the AMI Analysis Technical Notes for details.

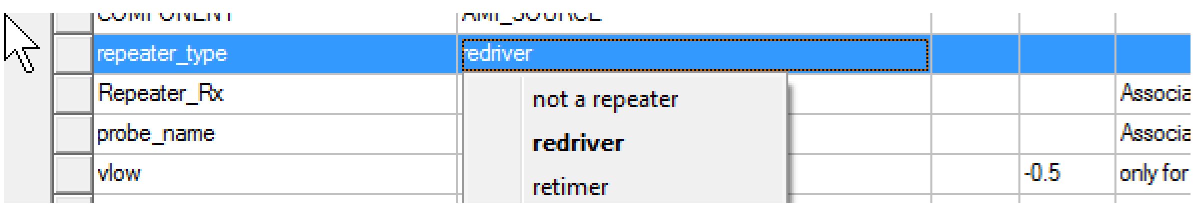
Here is an example:



In this example, upstream channel 1 consists of transmitter Tx1 and receiver Rx1, while downstream channel 2 consists of transmitter Tx2 and receiver Rx2. Each of these transmitters and receivers is imported as a standalone component. Each component has its own fully defined differential pin pair and model in an IBIS file. The AMI Repeater model consists of a two "halves," the receiver half (Rx1) and the transmitter half (Tx2). AMI Repeater properties in the components connect the output of Rx1 to the input of Tx2.

Here is the procedure for setting up the AMI Repeater, assuming the two channels are already defined.

1. Select the transmitter Tx2 from the downstream channel or the receiver Rx1 on the upstream channel. This example starts with the transmitter. Edit the Properties. Scroll down to the **repeater_type** property. Here are the AMI Repeater properties for the Tx2 component in the example above.



2. The default for **repeater_type** is **not a repeater**. To create an AMI repeater model, set the **repeater_type** property on Tx2 to **redriver** or **retimer**.

> **Note:**
>
> When the **repeater_type** property is set to **redriver** or **retimer**, the Internal Source properties for the AMI transmitter component are not displayed in the Properties window.

3. The **Repeater_Rx** field is populated with a list of available AMI receivers (probes). Select the probe corresponding to receiver Rx1 on the upstream channel.
4. Verify that the Tx2 **probe_name** property points to receiver Rx2 at the output of the downstream channel.
5. Click **OK** to close the Properties window.

The GUI automatically sets the AMI Repeater properties on the upstream receiver Rx1 to match those on downstream transmitter Tx2. You can verify the property settings in the receiver. You can also set the **Rx_Receiver_Sensitivity** parameter if it is present.

6. Select the receiver Rx1 on the upstream channel. Edit the Properties. Scroll down to the **repeater_type** property. Here are the properties for the Rx1 component in the example above.

COMPONENT	AMI_PROBE			
repeater_type	redriver			
Repeater_Tx	b_output4_26			Associated AMI Output devices
source_name	b_output4_24			Associated source name

7. Verify that the **repeater_type** property has been set to the **repeater_type** in the Repeater Tx2, **redriver** in this example.
8. Verify that the **Repeater_Tx** property in Rx1 has been set to the netlist name of the Tx2 component.
9. If the AMI file for the receiver contains an **Rx_Receiver_Sensitivity** property, that property is displayed among the **Properties on the AMI File**. Click the **Value** field to change the sensitivity. The **Rx_Receiver_Sensitivity** parameter sets the voltage thresholds (plus and minus) for sampling the equalized signal.

> **Note:**
>
> If you start with the upstream receiver Rx1, set the **repeater_type** property to **redriver** or **retimer**, then select the downstream **Repeater_Tx** from the list. The corresponding properties in that transmitter is automatically set. You can verify the settings.

10. Click **OK** to close the Properties window.
11. See AMI Repeaters in the AMI Technical Notes for details.

This completes the IBIS import process. If the circuit is complete, you are ready to run the AMI Analysis.

AMI Reserved Parameters

This topic discusses some of the AMI Reserved parameters. The definitions of Reserved parameters are given in the IBIS Specification (See AMI Analysis References).

AMI DLL_ID Parameter

The `DLL_ID` parameter was created so the EDA tool can provide the AMI executable (.dll or .so) with a string value that is guaranteed to be unique. The AMI executable can then use this value, for example, as part of filenames it uses when it creates any temporary or permanent output files.

At import time, if the AMI Reserved parameter DLL_ID is found, the parameter is displayed in the **AMI_Config** tab of the Parameters dialog, and whatever value was read from the AMI file is displayed as the initial value. Often the AMI file specifies a value of “placeholder”, as the actual value is to be provided by the EDA tool at simulation time.

In **Electronics Desktop**, the user may modify the value of DLL_ID in the **AMI_Config** tab. The value specified will be used as the prefix of the final value generated by Circuit and passed to the AMI executable. This may make it easier for the user to correlate any output files with the AMI device that created them. The final value of DLL_ID passed to the AMI executable has the following form:

```
<prefix provided by user>_<device_name>_<process ID of Circuit
Solver>
```

This name is guaranteed to be unique to the instance of the AMI model.

For example, if the user specifies `my_name` as the value of the DLL_ID parameter in the **AMI_Config** tab, and the name of the AMI device is `b_output4_1`, and the process ID of the instance of the Circuit solver is 784, then the value of DLL_ID passed to the AMI executable would be:

```
my_name_b_output4_1_784
```

By defining the value of DLL_ID (“my_name” in this example), the user can control the way any output files created by instances of the AMI model are grouped alphabetically in a file browser.

AMI DLL_Path Parameter

The **DLL_Path** parameter is provided so the EDA tool can tell the AMI executable (.dll or .so) the path to the directory where the executable resides. The AMI executable can then locate any needed support files relative to its own directory.

At import time, if the Reserved parameter "DLL_Path" is found, the parameter is displayed in the Parameters window, and whatever value was read on the AMI file is replaced with a period (.).

The value shown in the display is the one that is passed to the AMI executable; you should not change the value in the window.

The **Electronics Desktop** uses the ami_child_process when it has to launch an AMI .dll. The ami_child_process process always starts by changing its working directory to the directory containing the .dll. Therefore, the value set in the **DLL_Path** parameter is always “.” to represent the current working directory of the AMI executable.

AMI Ts4 Analog Buffer Model Parameters

Instead of using the legacy IBIS constructs for modeling analog portions of the AMI transmitter and receiver, 4-port Touchstone files may be specified for the AMI transmitter's analog output network and for the AMI receiver's analog input network.

The reserved parameters in this topic control the operation when a Touchstone file is used. For reference diagrams, see "The AMI Transmitter Model " on page 25-317and "The AMI Receiver Model " on page 25-319.

Ts4file

The **Ts4file** parameter specifies the name of the 4-port Touchstone-format file for either the transmitter or receiver. The file may contain a path; if no path is specified, the file is assumed to be in the same directory as the .ami file. There is no default.

Tx_V

The **Tx_V** parameter defines the voltage swing of the stimulus input to the transmitter circuit. The voltages of the voltage sources correspond to **V=Tx_V** for logic level 1 and **V=-Tx_V** for logic level 0.

Tx_R

The **Tx_R** parameter specifies the resistance of the series resistors in the transmitter circuit.

Rx_R

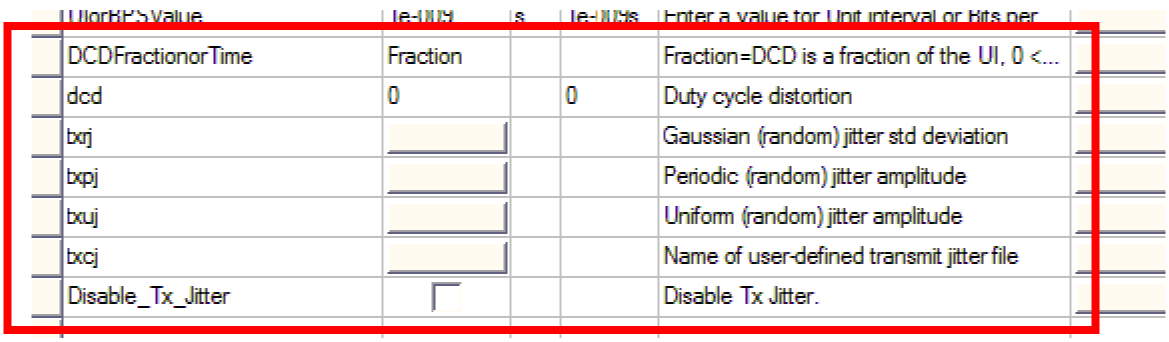
The **Rx_R** parameter specifies the resistance of the terminating resistors in the receiver circuit.

AMI Transmit Jitter Parameters

Nexxim analysis of the AMI transmitter provides two ways to specify transmit jitter: with the Nexxim jitter properties or with jitter parameters read on the AMI file.

Both methods are controlled by the **Disable_Tx_Jitter** parameter of the AMI transmitter. When **Disable_Tx_Jitter** is checked, both Nexxim and AMI transmit jitter calculations are deactivated. See the following example for the location of the **Disable_Tx_Jitter** control.

Nexxim Transmit Jitter Properties. Nexxim jitter properties are added to the AMI transmitter component properties. You can specify one or more transmit jitter distributions with the **txrj** (Gaussian), **txpj** (Periodic), and **txuj** (Uniform) properties, and duty-cycle distortion with the **dcd** (Duty Cycle Distortion) property. These properties are highlighted in the red box in the following example. The Nexxim transmit jitter and DCD properties are identical to the properties from QuickEye and VerifEye.



> **Note:**
> 1. The Nexxim **txcj** (Custom or user-defined) jitter property is not supported for AMI analyses.
> 2. The AMI analysis uses one value for each Nexxim property, taking only the first value if a list of values is given.

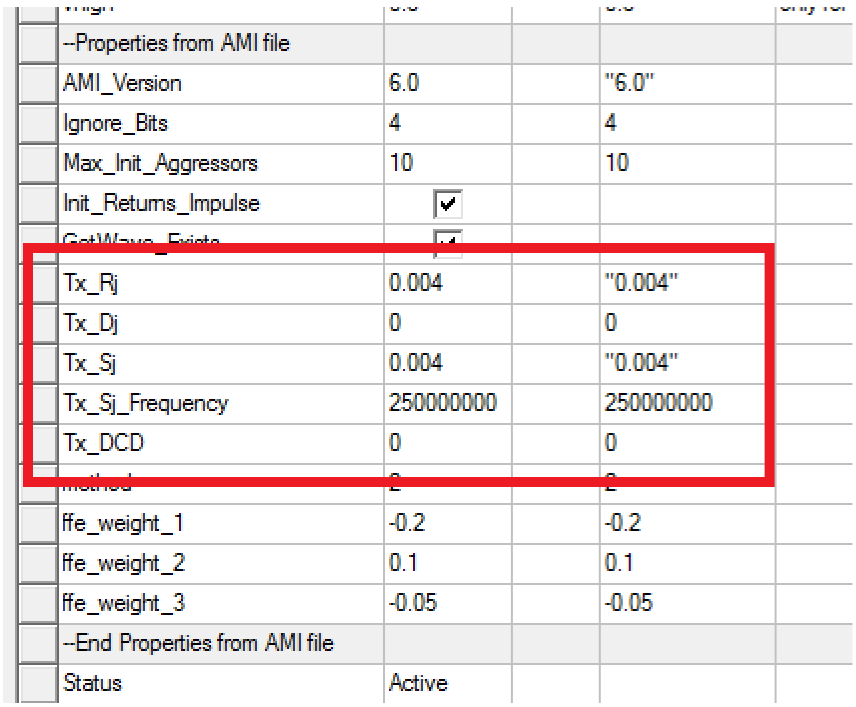
- Duty cycle distortion: Click the **DCDFractionorTime** dropdown menu to select **Fraction** (the default) or **Time**. For **Fraction**, the DCD value is a decimal fraction of the UI between 0.0 and 1.0. For **Time**, the DCD value is a time between 0.0 and UI in seconds. A time unit such as **ps** (picosecond) may be used to set the DCD time. The default for both **Fraction** and **Time** is 0. See also Duty Cycle Distortion in the QE/VE Technical Notes.

For **txrj**, **txpj**, or **txuj**, click the **Value** field and specify one value for the AMI analysis. All three types of jitter may be added, generating multiple jitter sources.

- For random transmit jitter (**txrj**) the value is the standard deviation for the Gaussian distribution. The default is 0 seconds.
- For Periodic random transmit jitter (**txpj**) the value is the amplitude. Default: 0 seconds.
- For Uniform random transmit jitter (**txuj**) the value is the amplitude. Default is 0 seconds.

> **Note:** For more information on the QE/VE transmit jitter parameters, see Transmit Jitter in the Quick Eye and VerifEye Technical Notes.

Transmit Jitter Parameters in the AMI File. If the AMI file contains transmit jitter parameters, they are imported into the transmitter component and displayed with the AMI Properties (red box in the following example).



--Properties from AMI file			
AMI_Version	6.0	"6.0"	
Ignore_Bits	4	4	
Max_Init_Aggressors	10	10	
Init_Returns_Impulse	☑		
GetWave_Exists	☑		
Tx_Rj	0.004	"0.004"	
Tx_Dj	0	0	
Tx_Sj	0.004	"0.004"	
Tx_Sj_Frequency	250000000	250000000	
Tx_DCD	0	0	
method	2	2	
ffe_weight_1	-0.2	-0.2	
ffe_weight_2	0.1	0.1	
ffe_weight_3	-0.05	-0.05	
--End Properties from AMI file			
Status	Active		

When the AMI model does not include transmit jitter parameters, use the appropriate Nexxim jitter properties to provide the equivalent jitter or duty cycle distortion for most AMI transmit jitter parameters. The AMI transmit jitter parameters **Tx_Rj** and **Tx_Sj** map directly to the **txrj** and **txpj** Nexxim properties, while the AMI **Tx_DCD** and **Tx_Dj** parameters map to the Nexxim **dcd** and **txuj** properties with minor adjustments to obtain the same result, as shown in the following table.

Jitter Type	Mapping to Nexxim Jitter Properties from AMI Parameters
Gaussian or Random	**txrj** (time) = **Tx_Rj** (time or fractional UI)

Jitter Type	Mapping to Nexxim Jitter Properties from AMI Parameters
Periodic or Sinusoidal	**txpj** (time) = **Tx_Sj** (time or fractional UI)
Duty Cycle Distortion	**dcd** (fractional UI) = 2 x **Tx_DCD** (time or fractional UI)
Uniform or Deterministic	**txuj** (time) = 2 x **TX_Dj** (time or fractional UI)

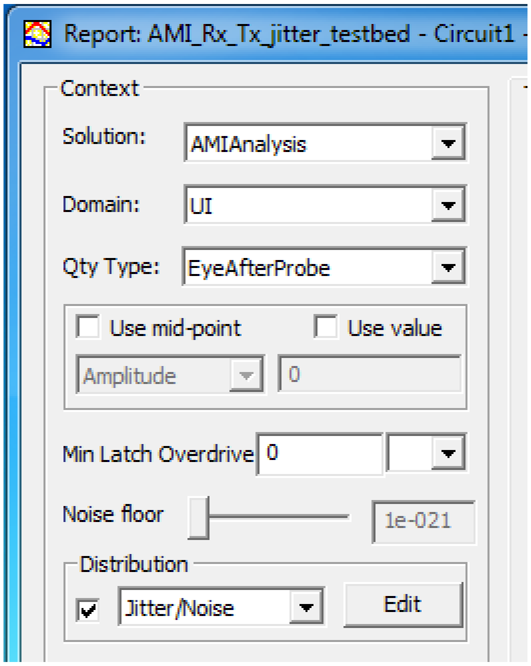
Note:

1. The mappings for duty cycle distortion and uniform jitter incorporate a factor of 2, because the AMI parameters specify half the peak-to-peak values, while the Nexxim properties specify the full values.
2. Nexxim can apply the AMI **Tx_Sj_Frequency** parameter only in AMI Time Domain simulation (AMI analysis). The AMI **Tx_Sj_Frequency** parameter is ignored for Nexxim AMI Statistical simulation (VerifEye analysis). The **Tx_Sj_ Frequency** parameter does not map to any Nexxim transmit jitter parameter. Nexxim periodic jitter **txpj** relies only on the amplitude of the periodic noise source (**Tx_Sj**).
3. When a Nexxim jitter property has been set and the AMI file contains the corresponding AMI jitter parameter, the Nexxim property setting is applied, the AMI parameter is ignored, and a warning is displayed.

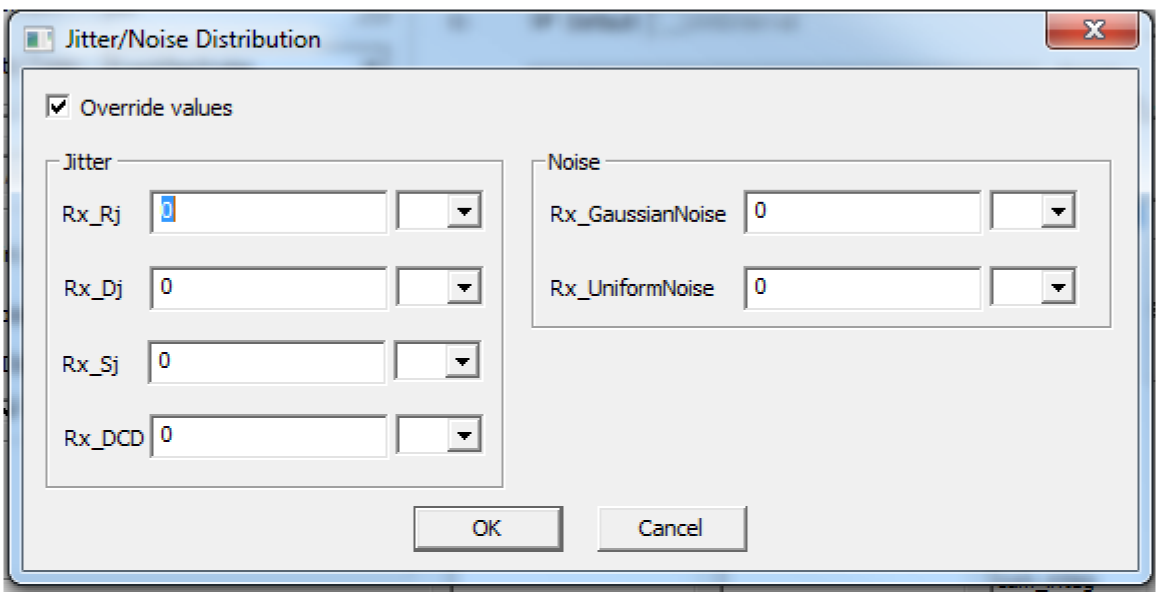
AMI Receive Jitter Parameters

Receive jitter and noise are specified using the Receive Jitter/Noise **Distribution** group box on the Report setup window. For AMI analysis simulations (including statistical analysis using VerifEye with AMI), the values of the parameters may be provided by the simulation. Values may come on the .ami file at import time (and potentially be modified by the user in the **Properties** window for the Rx device), or may be returned by the AMI model after the call to AMI_Init(). In either case, the values provided by the simulation are used to initialize the Jitter/Noise window.

Receive Jitter/Noise Distributions. Receive jitter and noise can be added or modified in the Reports window. For AMI, the **Distribution** group box appears when you select EyeAfterProbe for the Bathtub plot:



Click **Edit** to open the Jitter/Noise Distribution window.



The parameter names and definitions are as defined in the IBIS specification.

- For random receive jitter (**Rx_Rj**) the value is the standard deviation of the Gaussian distribution in time units. The default is 0 seconds (overridden when the AMI file contains a value for **Rx_Rj**).

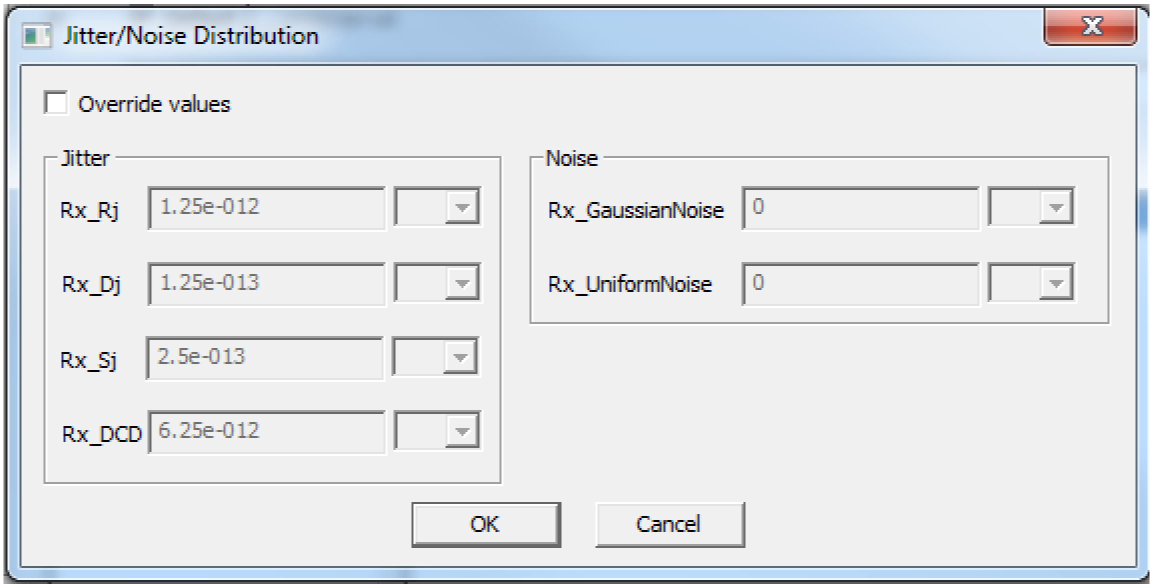
- For uniform receive jitter (**Rx_Dj**) the value is one-half the width of the jitter distribution in time units. Default is 0 seconds (overridden when the AMI file contains a value for **Rx_Dj**).

- For sinusoidal jitter (**Rx_Sj**) the value is the amplitude of the distribution in time units. Default is 0 seconds (overridden when the AMI file contains a value for **Rx_Sj**).
- For clock duty cycle distortion (**Rx_DCD**) the value is one-half the peak-to-peak variation of the distortion in time units. Default is 0 seconds (overridden when the AMI file contains a value for **Rx_DCD**).
- For random receiver noise, (**Rx_GaussianNoise**) the value is the standard deviation of the Gaussian distribution in voltage units. The default is 0 volts (overridden when the AMI file contains a value for **Rx_GaussianNoise**).
- For uniform receiver noise (**Rx_UniformNoise**) the value is one-half the width of the noise distribution in voltage units. Default is 0 volts (overridden when the AMI file contains a value for **Rx_UniformNoise**).

Receive Jitter Parameters in the AMI File. If the AMI file contains receive jitter parameters, they are imported into the receiver component and displayed with the AMI Properties. Note that the quantities in this Properties listing may be in terms of fractions of the UI (as in the following example), or in absolute times.

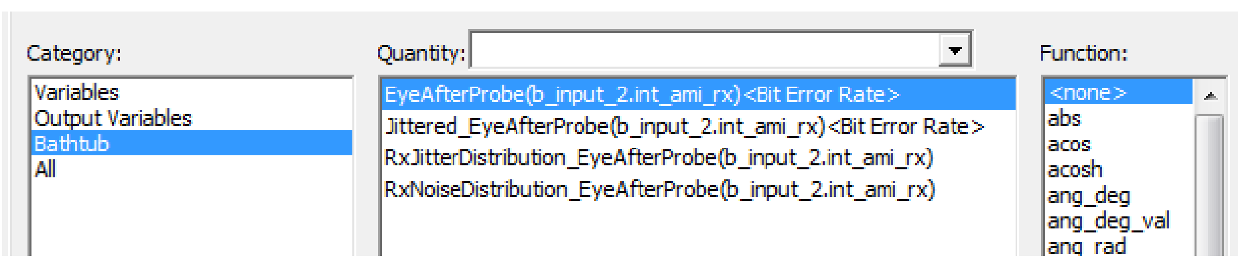
Rx_Receiver_Sensitivity	0.03		0.03	
Rx_Rj	0.01		0.01	
Rx_Dj	0.001		0.001	
Rx_Sj	0.002		0.002	
Rx_DCD	0.05		0.05	
Rx_GaussianNoise	0		0	
Rx_UniformNoise	0		0	

If there are receive jitter parameters in the AMI file, they is copied into the Receive Jitter window during simulation. When the AMI receive jitter parameters are copied into the Jitter/Noise Distribution window, they are converted to absolute times if necessary. The window in the following diagram illustrates the fractional UI values on the example above converted to absolute times in seconds.



When you create a bathtub plot of the EyeAfterProbe quantity, any values provided by the AMI simulation are used to populate the Jitter/Noise window if the user chooses to add the Jitter/Noise distributions. Modify these values in the window. When the AMI model does not include receive jitter parameters, enter the jitter parameters manually.

When the Jitter/Noise distribution box is checked for a bathtub plot of the EyeAfterProbe, four traces are available for plotting:



The quantities are the bathtub without jitter applied, the bathtub with noise and jitter applied, and the distributions that represent the Jitter and Noise parameters.

AMI Modulation Parameters

Modulation

The **Modulation** parameter specifies the modulation scheme to be used for analysis using AMI_GetWave and for post-processing the simulation results. The choices are "NRZ" and "PAM4". When the **Modulation** parameter is not included in the .ami file, the default is "NRZ". If a list of both values is given in the AMI file, the transmitter component property display offers a drop-down menu to select the value.

When **Modulation** is "NRZ", Nexxim sets the input stimulus using -0.5V to represent a logic 0 and 0.5V to represent a logic 1.

When **Modulation** is "PAM4", Nexxim sets the input stimulus using voltage levels of -0.5V, -0.166V, 0.166V and 0.5V, which are mapped to PAM4 two-bit values or symbols (00, 01, 10 and 11) using the **PAM4_Mapping** parameter.

PAM4_Mapping

The **PAM4_Mapping** parameter declares a four-character string that maps the four voltage levels to the PAM4 symbols.

The **PAM4_Mapping** parameter is ignored when the **Modulation** parameter has value "NRZ".

The values of the characters in the **PAM4_Mapping** string are:

"0" = symbol 00

"1" = symbol 01

"2" = symbol 10

"3" = symbol 11.

The positions in the string (1st, 2nd, 3rd, 4th) correspond to signal voltage levels, beginning with the most negative voltage and becoming incrementally more positive.

The **PAM4_Mapping** parameter must contain four characters and each of the four characters "0", "1", "2" and "3" must occur once.

For example, a **PAM4_Mapping** value string of "0132" (the Grey code default) sets up the following mapping:

• The most negative voltage (-0.5v) represents binary 00

• The next higher voltage should be interpreted as binary 01

• The next higher voltage should be interpreted as binary 11

• The most positive voltage should be interpreted as binary 10

A **PAM4_Mapping** value string of "0123" sets up the following mapping:

• The most negative voltage should be interpreted as binary 00

• The next higher voltage should be interpreted as binary 01

• The next higher voltage should be interpreted as binary 10

• The most positive voltage should be interpreted as binary 11

If **Modulation** is set to "PAM4" and **PAM4_Mapping** is not present, the default mapping is the Gray code value "0132".

PAM4_UpperThreshold

PAM4_CenterThreshold

PAM4_LowerThreshold

Nexxim uses the three threshold voltages in conjunction with the **RX_Receiver_Sensitivity** parameter to identify the PAM4 voltage levels at the Receiver:

• A voltage lower than (**PAM4_LowerThreshold** – **Rx_Receiver_Sensitivity**) is the lowest voltage level (by default -0.5V).

• A voltage lower than (**PAM4_CenterThreshold** – **Rx_Receiver_Sensitivity**) but greater than (**PAM4_LowerThreshold** + **Rx_Receiver_Sensitivity**) is the second lowest voltage level (default -0.166V).

• A voltage lower than (**PAM4_UpperThreshold** – **Rx_Receiver_Sensitivity**) but greater than (**PAM4_CenterThreshold** + **Rx_Receiver_Sensitivity**) is the second highest voltage level (default +0.166V).

• A voltage greater than (**PAM4_UpperThreshold** + **Rx_Receiver_Sensivity**) is the highest voltage level (default +0.5V).

A voltage that is not in one of these ranges is treated as a symbol error.

If the **Modulation** parameter lists "PAM4" either as a Value or as a List selection, **PAM4_UpperThreshold** and **PAM4_LowerThreshold** are required for Rx AMI parameter definition files.

If **PAM4 _CenterThreshold** is not declared, the value of **PAM4_CenterThreshold** defaults to 0.0 volts.

The **PAM4 Threshold** parameters are ignored when the **Modulation** parameter has value "NRZ".

PAM4_UpperEyeOffset (Value Float)

PAM4_CenterEyeOffset (Value Float)

PAM4_LowerEyeOffset (Value Float)

These three parameters specify sampling clock offsets for Upper, Center and Lower PAM4 eyes.

During PAM4 analysis, receiver sampling data is referenced to a nominal eye centered between consecutive edge transition times. The PAM4 Upper, Center and Lower eyes may have a time shift with respect to the nominal eye. Each of the three Offset parameters defines a sampling offset on the nominal eye of one of the PAM4 eyes.

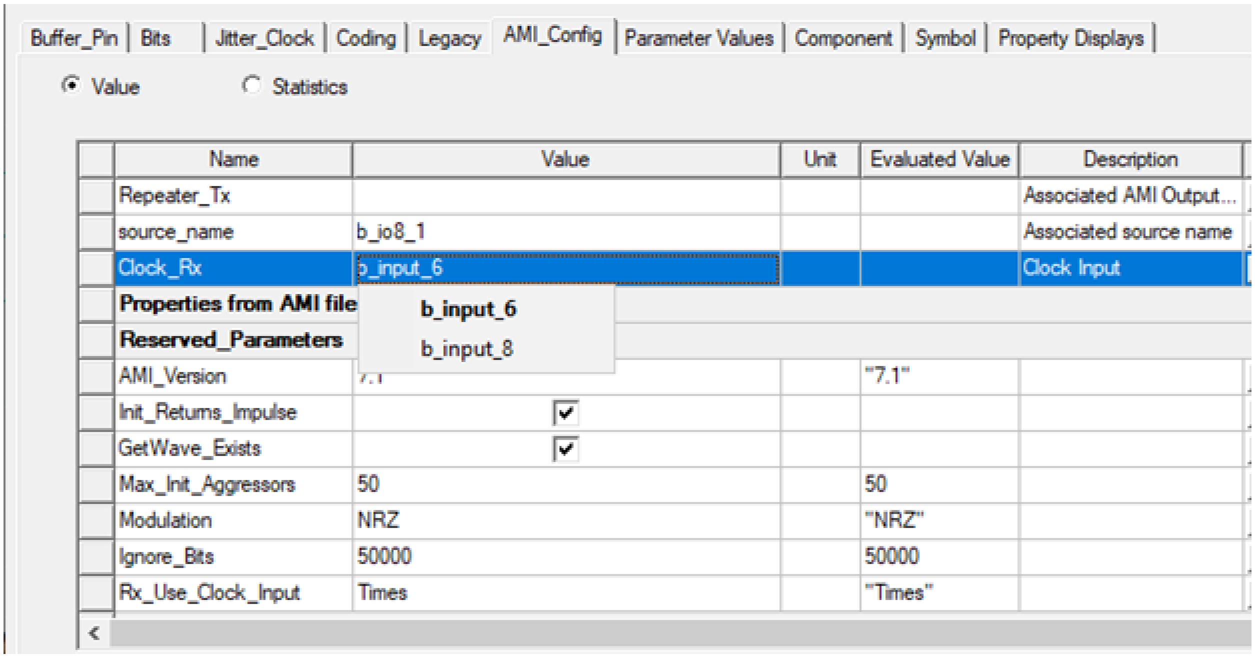
When the offset is a positive value, sampling occurs after the sample time for the nominal eye. When the offset is a negative value, sampling occurs before the sample time for the nominal eye.

If the **Modulation** parameter is set to "PAM4" and any offset values are not present, the default value is 0.0 for each offset parameter not declared. The Offset parameters are ignored when the **Modulation** parameter is not present or is set to "NRZ".

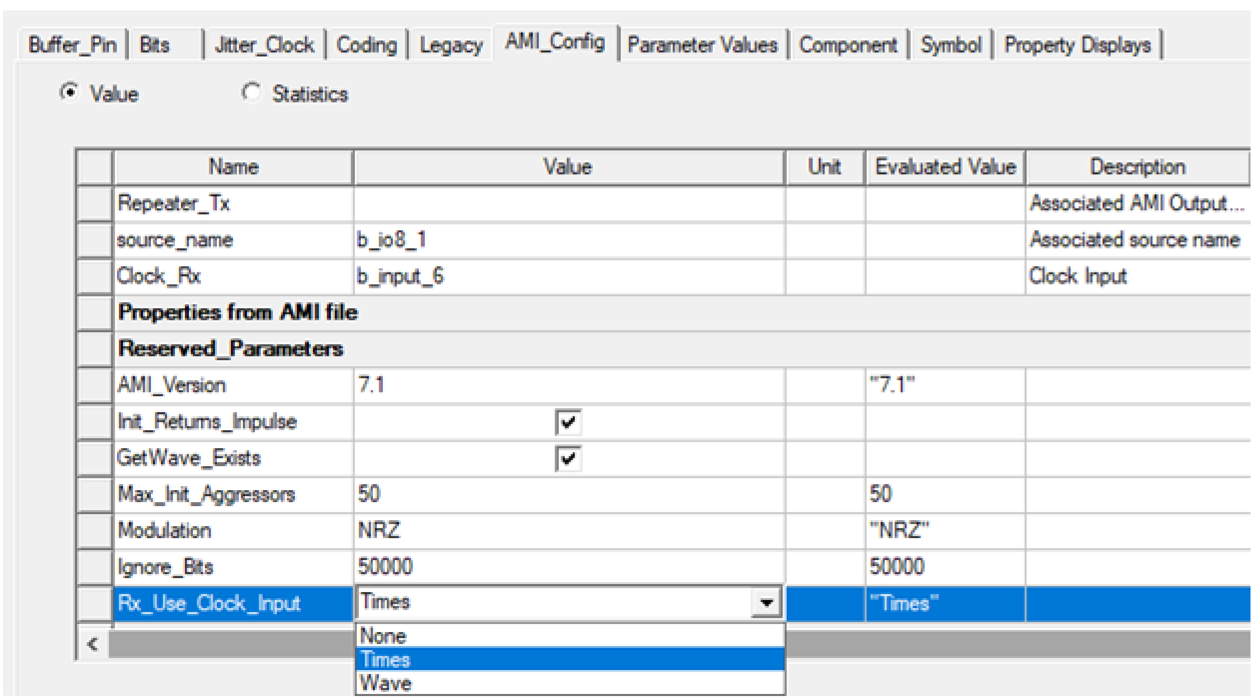
AMI Rx_Use_Clock_Input Parameter

The **Rx_Use_Clock_Input** parameter was introduced in version 7.1 of the IBIS specification to support clock forwarded architectures such as DDR. If a data (DQ) Rx model requires a clock input from a clock (DQS) Rx model, then it specifies the **Rx_Use_Clock_Input** parameter.

At import time, if the AMI Reserved parameter **Rx_Use_Clock_Input** is found, the parameter is displayed in the **AMI_Config** tab of the Parameters dialog. In addition, a new property **Clock_ Rx** is automatically added. It provides a drop-down box with a list of all the AMI Rxs in the Circuit Design. The drop-down box allows the user to choose the Rx model that will provide the clock to this data Rx model:



The allowed values of Rx_Use_Clock_Input are **None**, **Times**, or **Wave**. The model may specify one of these values, or it may specify that multiple settings are supported. If multiple values are supported, a drop-down box allows the user to specify which setting to use:



- The value **None**, specifies that the AMI data model does not take any clock information as an input. This is traditional AMI behavior, and it is the same as if **Rx_Use_Clock_Input** does not exist in the model.
- The value **Times** specifies that the data Rx model expects the clock times output by the clock Rx model to be provided as an input.
- The value **Wave** specifies that the data Rx model expects the entire sampled waveform returned by the clock model to be provided as an input.

In the case of either **Times** or **Wave**, the Circuit solver ensures that the clock model is called prior to the calling the data model so that the outputs of the clock model are available to the data model. The Circuit solver passes the appropriate information to the data model according to the **Times** or **Wave** setting.

> **Note: Eye Diagrams and Clock Times for the Rx Data Model**
> Even though **Rx_Use_Clock_Input** allows a data Rx model to specify an external clock source, the Eye_After_Probe eye diagram for the data Rx model works as it does for any other AMI Rx. The IBIS specification states that (IBIS 7.2, pg. 230):
>
> *A receiver model that specifies the **Rx_Use_Clock_Input** parameter must return valid clock times.*
>
> Therefore, a data Rx model will always return a final set of clock times, which Circuit uses to create the **Eye_After_Probe** in the same way it would for a conventional AMI Rx that returned clock times.

Set Up and Run AMI Analysis

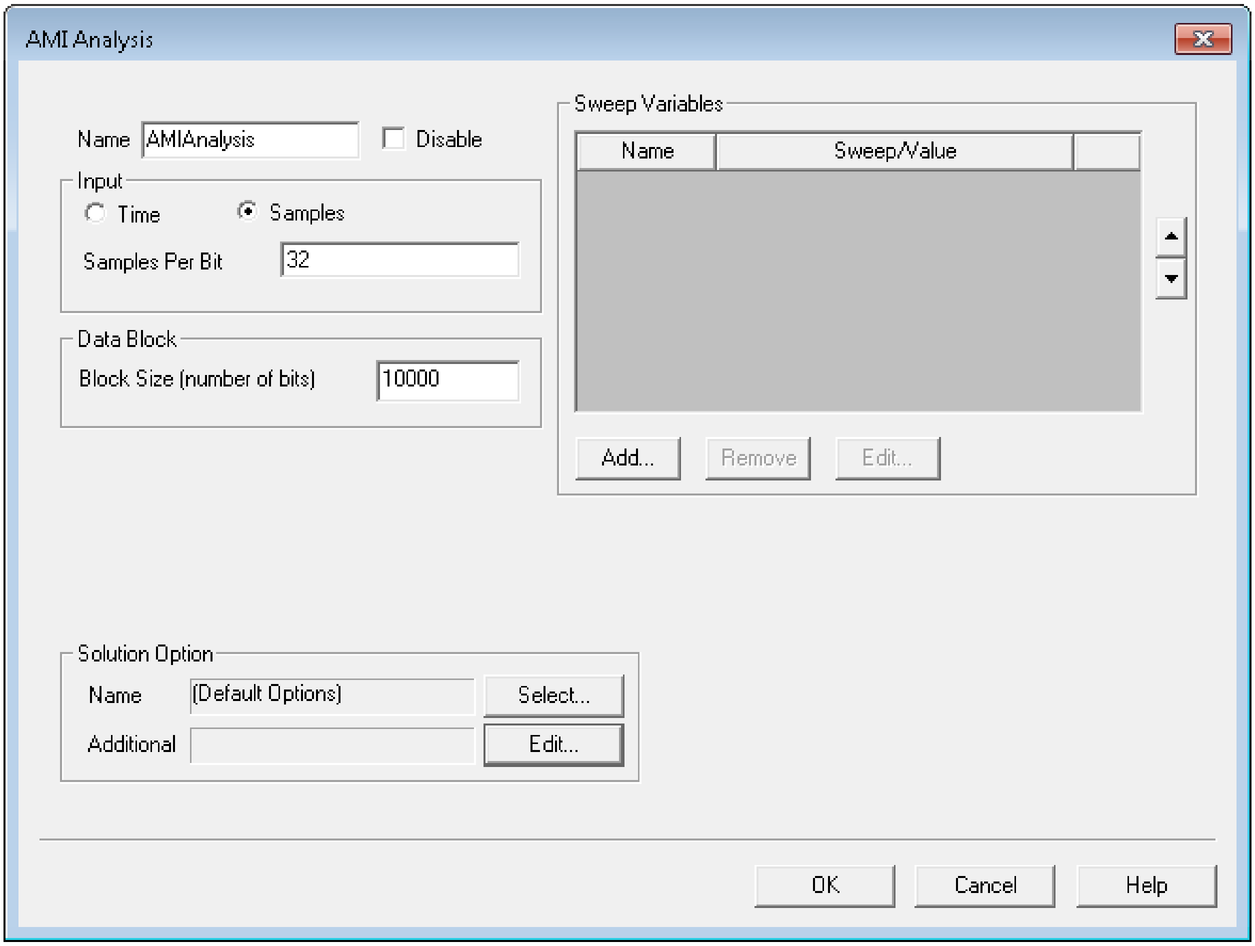
AMI analysis is equivalent to the Time Domain simulation described in the IBIS AMI specification. To run AMI analysis, the design must include one or more IBIS AMI transmitters (or AMI Sources) and corresponding IBIS AMI receivers (or AMI Probes). Transmitter and receiver parameters are set up in the sources and probes (See "Importing AMI Components with IBIS Buffer Models" on page 11-119).

Other parameters and options can be specified in the AMI Analysis setup window, as described in the following topics.

Set Up the AMI Analysis

From the **Schematic Editor**, perform the following steps to set up, run, and view the results of an AMI analysis using the Circuit solver.

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **AMI Analysis** to open the **AMI Analysis** window.

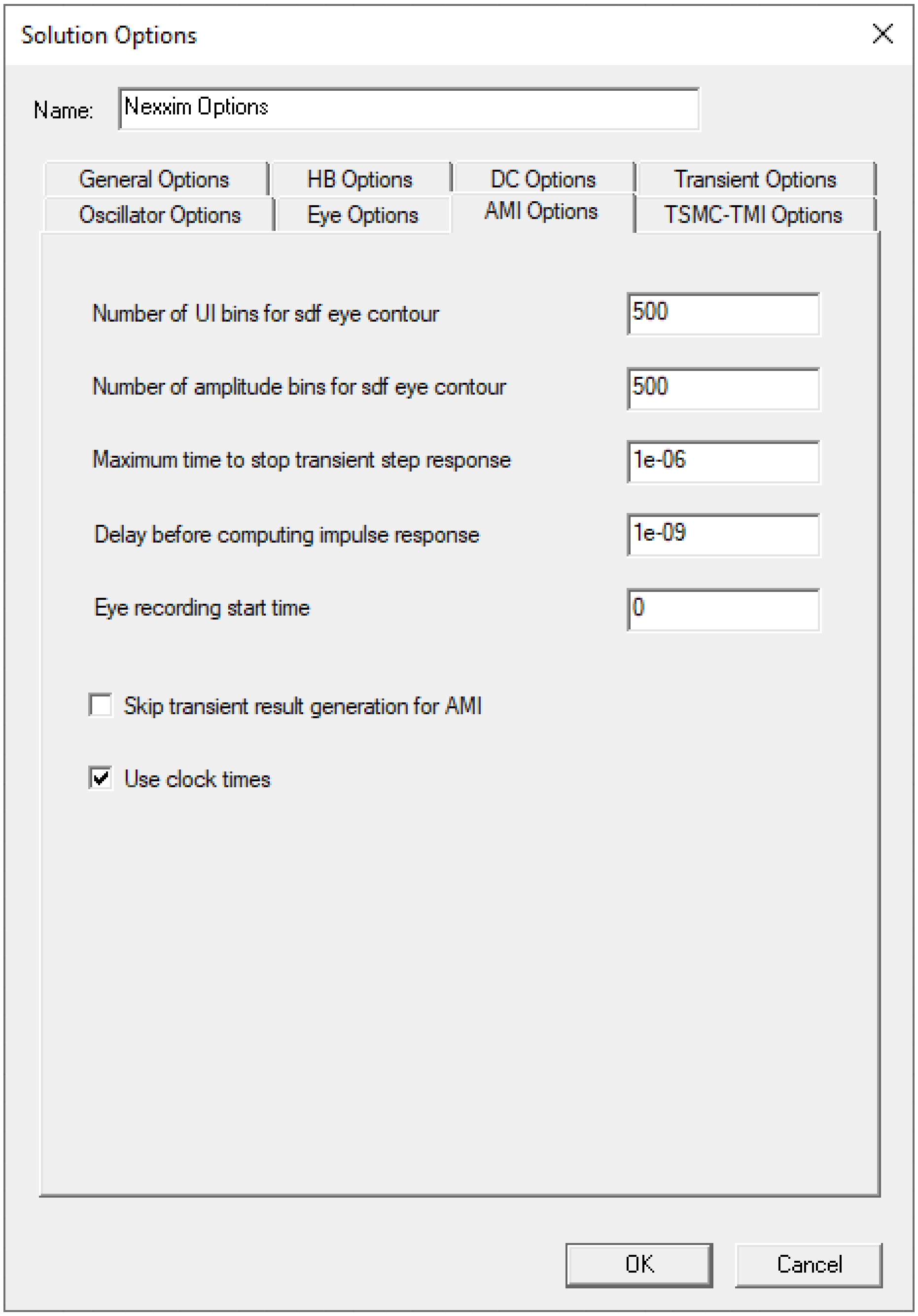


2. Type an **Analysis Name** (or accept the default name, "AMIAnalysis").
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use.
4. AMI simulation uses a fixed timestep for sampling the impulse response and any processed waveforms. Click **Time** or **Samples** to define the interval at which the impulse response and waveforms are sampled.

- **Samples** allows you to specify the number of samples per bit (UI). The default is 32 samples per bit. Most AMI models and simulations are set up using **Samples** per bit rather than the absolute **Time** step.
- **Time** allows you to set the timestep in seconds. The timestep must be greater than 0, and must be a fraction of the unit interval (UI). The default timestep is 5e-10 seconds.

5. The **Block Size** field sets the number of bits per AMI data block. Default is 10,000 bits per block.

6. Optionally, use the fields in the **Solution Options** group box to add AMI options.
- Click **Select** on the **Name** field to open the **Select Solution Options** window.
- Click **New** to create a new option set and open the **Solution Options** window.



- Click the **Name** field to name the new option set.

- Click the **AMI Options** tab fields and check boxes to set any appropriate options.
- Click **OK** to return to the **AMIAnalysis** window.

For more information, see AMI Analysis Options

Run the AMI Analysis

1. From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
2. The **Message Manager** window signals success or failure.

The next step is to view the outputs as reports; see AMI Analysis Outputs.

For additional details on creating and modifying reports, see Generating Reports and Post-processing.

AMI Analysis Options

Options that apply to AMI analyses are listed in the AMI Analysis Options Reference. All options use the prefix **ami** and all transient analysis options are valid with AMI Analysis.

AMI Analysis Outputs

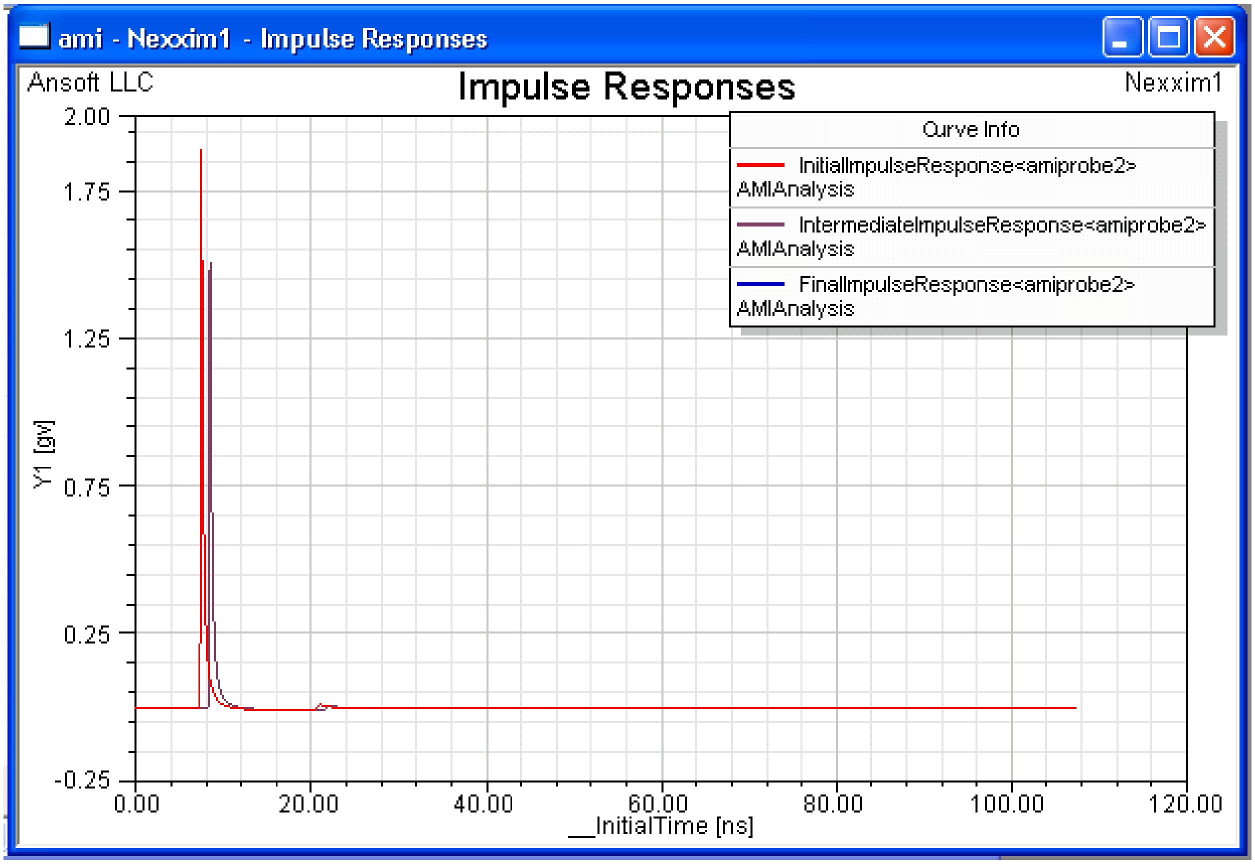
To open the results of an AMI analysis, click the Results icon in the **Project Manager** window and select one of the following **Create Report** options.

AMI Impulse Response versus Time

To plot the impulse responses in the time domain:

1. Select **Create Standard Report>Rectangular Plot**.
2. Select your AMI Analysis as the **Solution** ; **Initial Response** as the Domain; **Voltage** as the Category; **InitialImpulseResponse**, **IntermediateImpulseResponse**, or **FinalImpulseResponse** as the Quantity; and **<none>** as the Function.

The following chart shows all three impulse responses.



- The Initial Impulse Response is the one generated by the Circuit solver.
- The Intermediate Impulse Response reflects any modifications made by the AMI transmitter **AMI_Init** function at initialization.
- The Final Impulse Response reflects any further modifications made by the AMI receiver **AMI_Init** function at initialization.
- In the chart above, the Intermediate and Final responses are identical.

Note:

When the AMI transmitter model contains an **AMI_GetWave** function, the Intermediate Impulse Response in the graph is not the one that is applied unless the transmitter's **AMI_Init** and **AMI_GetWave** functions perform identical modifications.

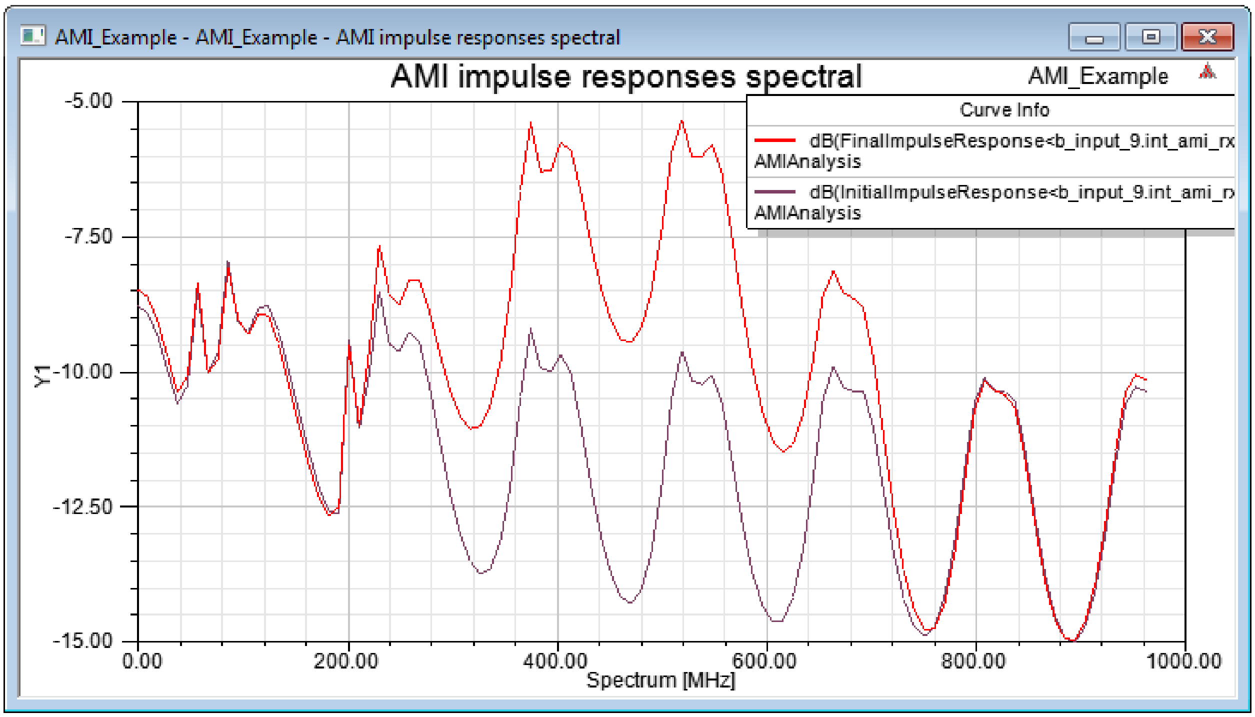
When the AMI receiver model contains an **AMI_GetWave** function, the Final Impulse Response in the graph is not the one that is applied unless the receiver's **AMI_Init** and **AMI_GetWave** functions perform identical modifications.

AMI Impulse Response in Spectral Domain

To plot the impulse responses:

1. Select **Create Standard Report>Rectangular Plot**.
2. Select your AMI Analysis as the **Solution** ;
   Spectral as the Domain;
   Voltage as the Category;
   InitialImpulseResponse, **IntermediateImpulseResponse**, or **FinalImpulseResponse** as the Quantity;
   and **dB** as the Function.

The following shows the initial and final impulse responses.



- The Initial Impulse Response is the one generated by the Circuit solver.
- The Final Impulse Response reflects any modifications made by the AMI transmitter and receiver at initialization.

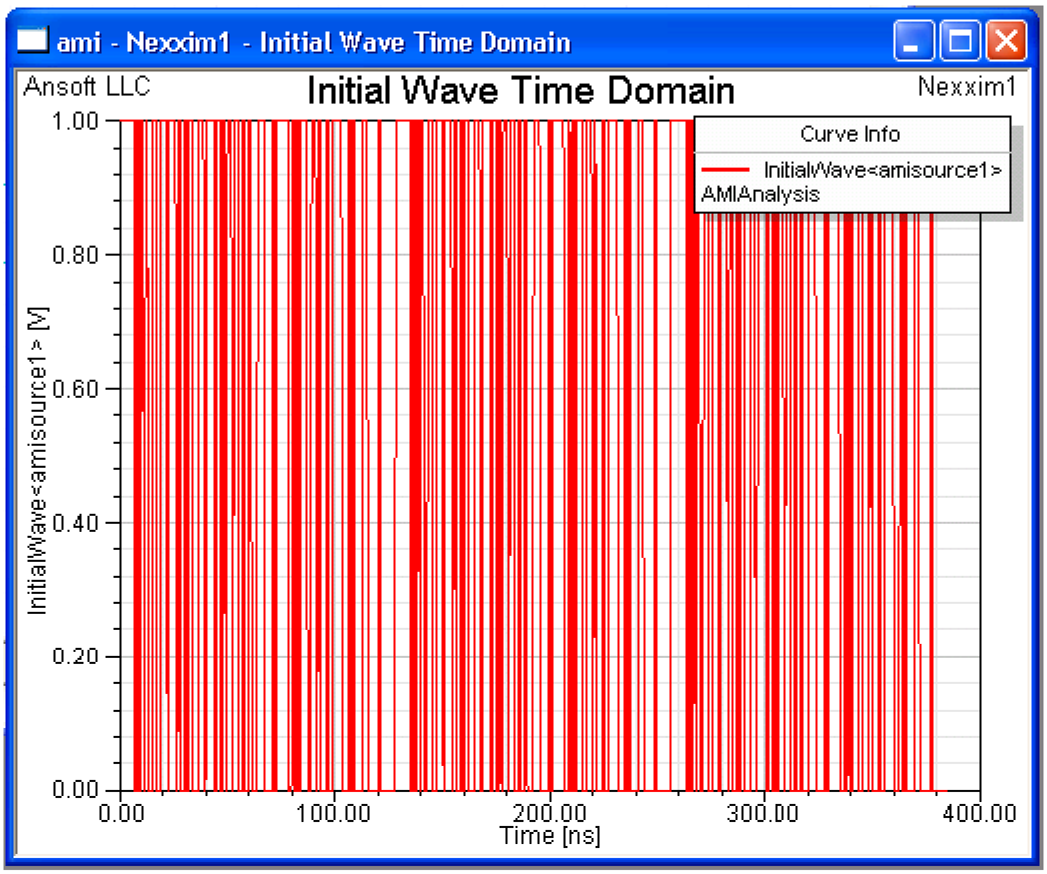
For more information, see Plotting Spectral Domain Data in the Reports topic. (Although that topic mentions only transient solutions, the Context controls also apply to AMI analyses.)

AMI Time Domain Waveforms

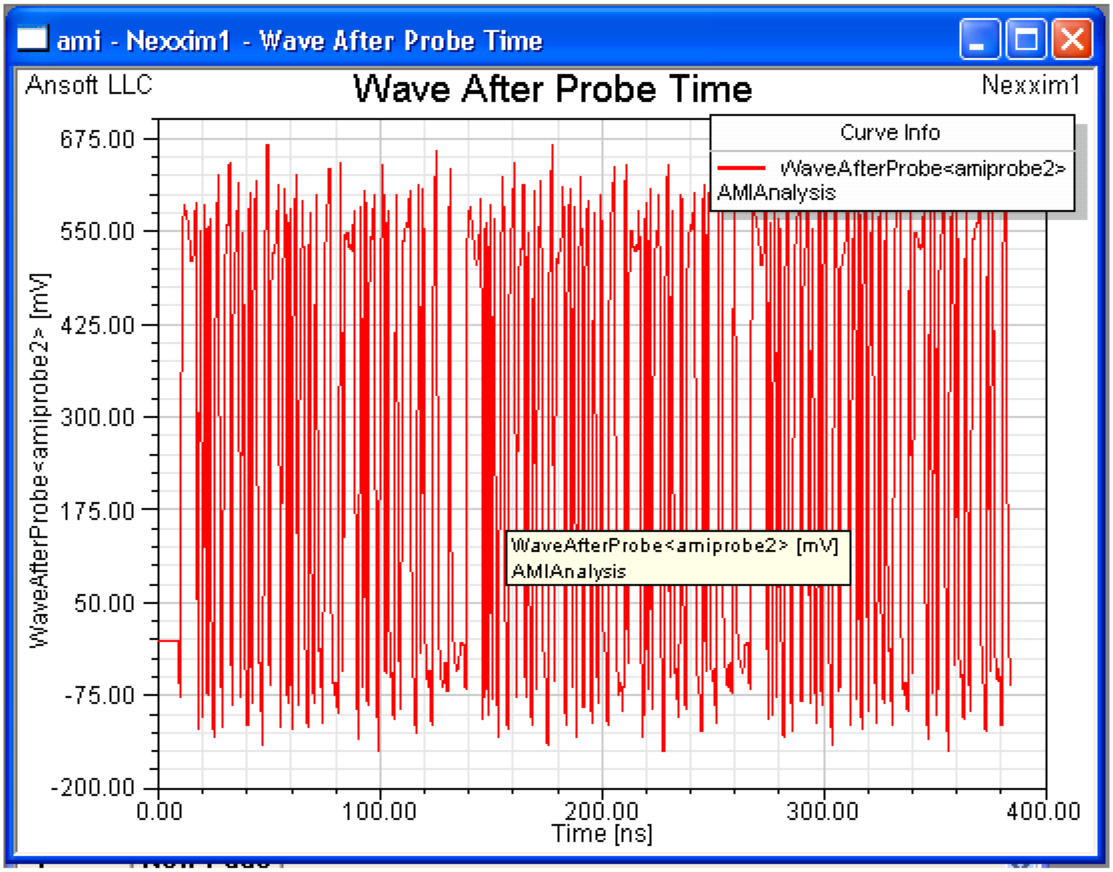
The report can show the Time Domain waveform at four points in the circuit:

- The Initial Waveform shows the Piecewise Linear data input to the transmitter, including any jitter.
- The Waveform After Source shows the output from the transmitter.
- The Waveform After Channel shows the input to the receiver.
- The Waveform After Probe shows the output from the receiver.

To plot the waveforms, select **Create Standard Report>Rectangular Plot**. Select your AMI Analysis as the **Solution** ; **Time** as the Domain; **Voltage** as the Category; **InitialWaveform**, **WaveformAfterSource**, **WaveformAfterChannel**, or **WaveformAfterProbe** as the Quantity; and **<none>** as the Function. The following chart shows the Initial Waveform.



This second chart shows the Waveform After Probe in the Time Domain:

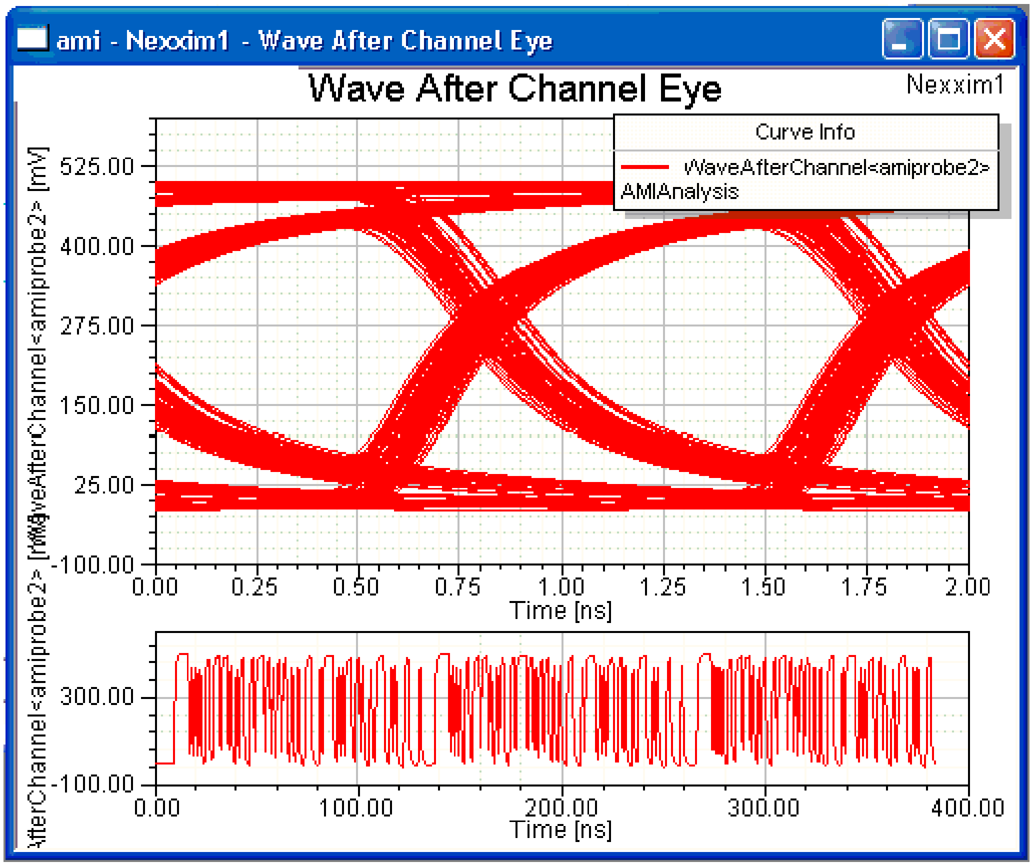


AMI Eye Diagram Waveforms

The report can show Eye diagrams of the waveform at four points in the circuit:

- The Initial Waveform shows the Piecewise Linear data input to the transmitter, including any jitter.
- The Waveform After Source shows the output from the transmitter.
- The Waveform After Channel shows the input to the receiver.
- The Waveform After Probe shows the output from the receiver.

To plot the waveforms, select **Create Eye Diagram Report>Rectangular Plot**. Select your AMI Analysis as the **Solution** ; **Time** as the Domain; **Voltage** as the Category; **InitialWaveform**, **EyeAfterSource**, **EyeAfterChannel**, or **EyeAfterProbe** as the Quantity; and **<none>** as the Function. The following chart shows the Waveform After Channel.



AMI Bathtub Charts

The AMI bathtub report can illustrate curves of the Bit Error Rate (BER) at four points in the circuit:

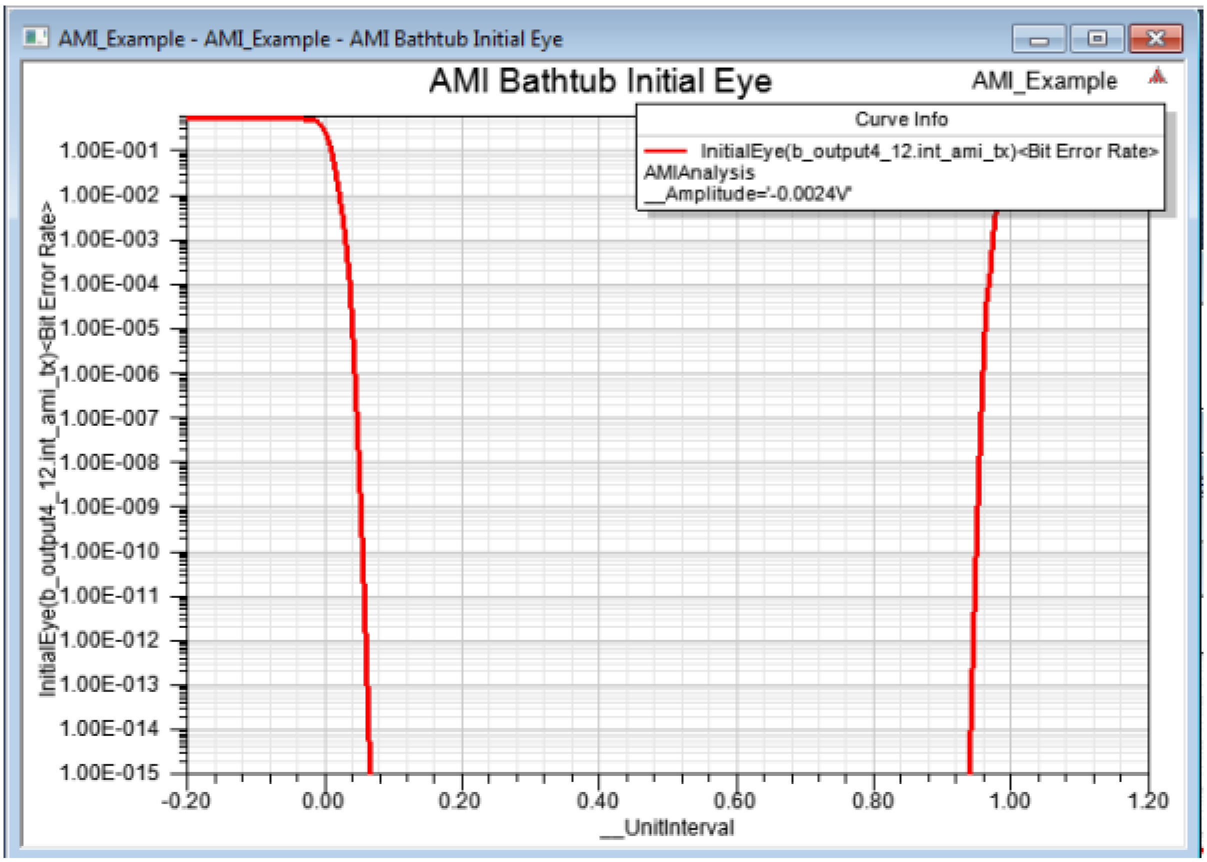
- The Initial Waveform shows the Piecewise Linear data input to the transmitter, including any jitter.
- The Waveform After Source shows the output on the transmitter.
- The Waveform After Channel shows the input to the receiver.
- The Waveform After Probe shows the output on the receiver.

> **Note:** With AMI plots, when Bathtub curves can be plotted and txrj is present, a warning message is displayed if the number of bits is less than 2.5e5.

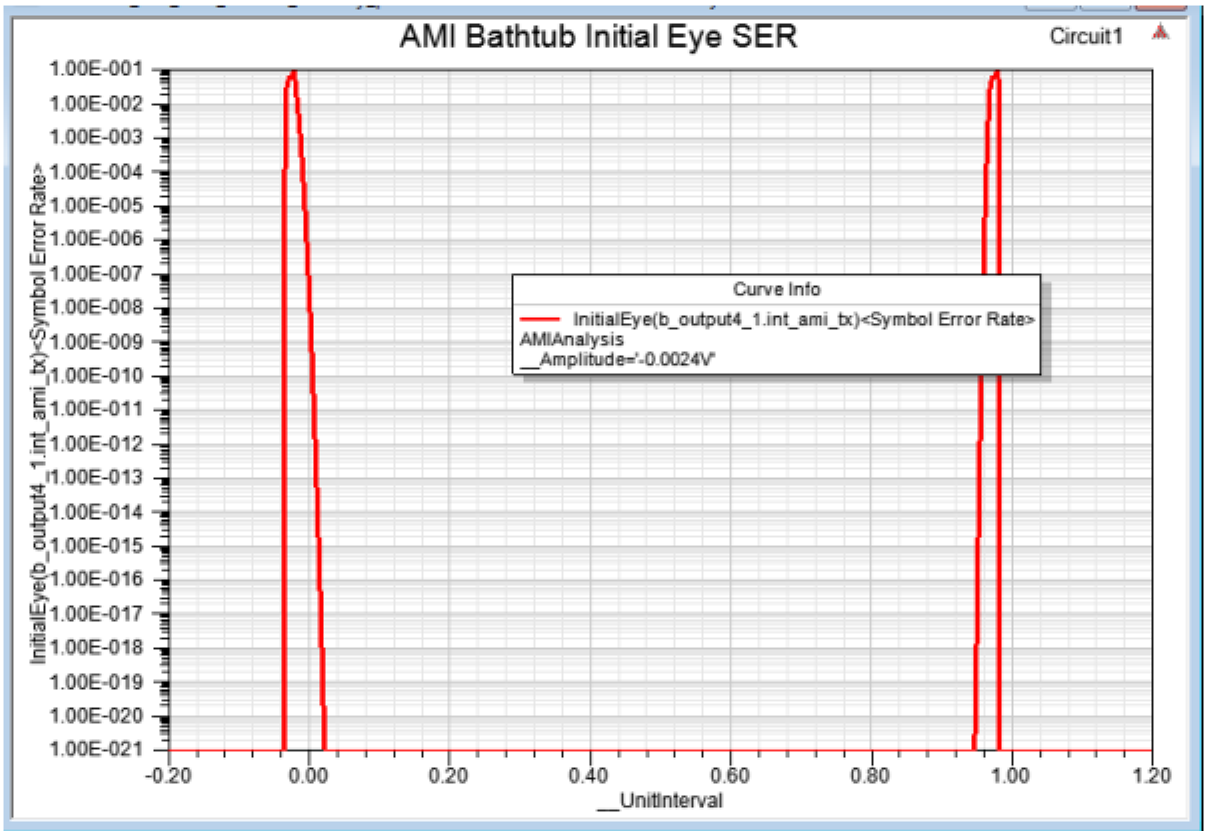
To plot the bathtub diagram:

1. Select **Create Standard Report>Rectangular Plot**.
2. Select the following:
   - **Solution**: AMI Analysis
   - **Domain**: UI
   - **Quantity**: InitialEye, EyeAfterSource, EyeAfterChannel, or EyeAfterProbe
   - **Primary Sweep**: _UnitInterval
   - **Function**: <none>

The following chart shows the BER bathtub curve for the **Initial Eye** waveform. From the Y-Axis, the **Scaling** is set to **Log**, with **Max** 0.6 and **Min** 1e-15.



3. With **modulation=PAM4**, plot the symbol error rate. PAM4 symbols are two bits. Click the same Report setup, but select the **<Symbol Error Rate>** trace.



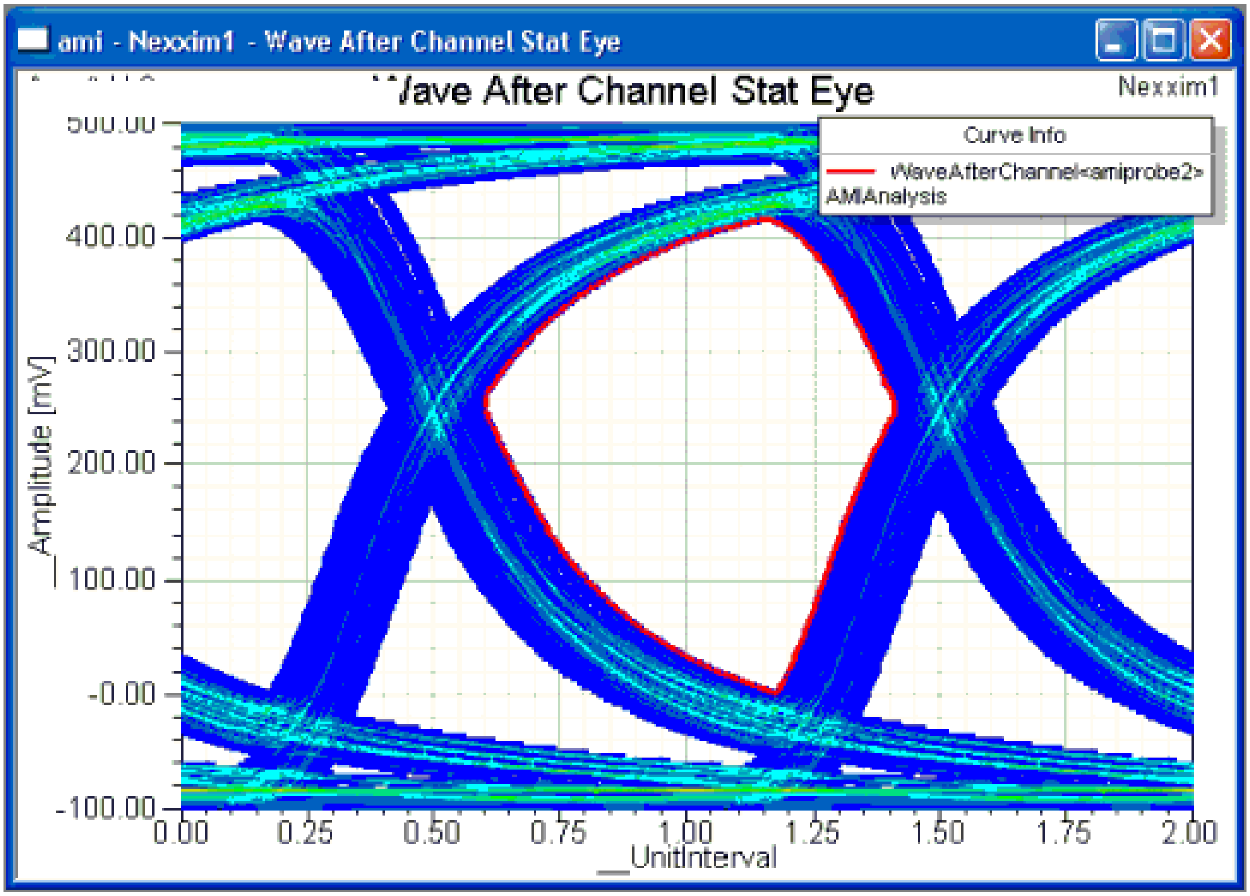
For bathtub curves of AMI, distinguish simulated and extrapolated regions with a partitioning horizontal line using the 'SimulatedLimit' option that is in the 'Quantity' column. For more information, please see Display Quick Eye Analysis Outputs.

AMI Sampled Eye Diagrams

The sampled or statistical eye report generates the eye diagram but does not retain the transient data for further post-processing. The Circuit solver can generate sampled eye diagrams of the waveform at four points in the circuit:

- The Initial Waveform shows the Piecewise Linear data input to the transmitter, including any jitter.
- The Waveform After Source shows the output from the transmitter.
- The Waveform After Channel shows the input to the receiver.
- The Waveform After Probe shows the output from the receiver.

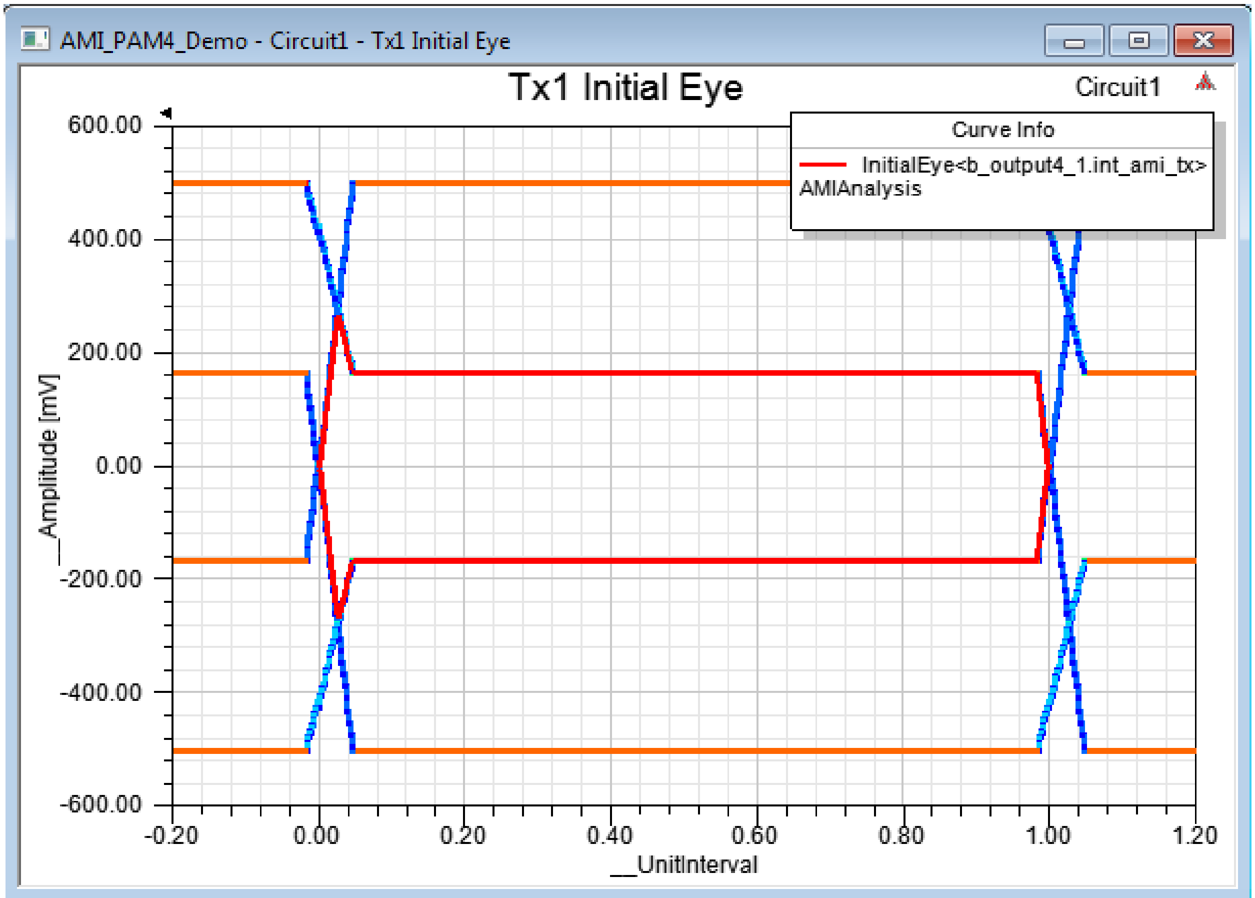
To generate a sampled eye diagram, select **Create Statistical Eye Report>Statistical Eye Plot**. Select your AMI Analysis as the **Solution** ; **UI** as the Domain; **Eye** as the Category; **InitialWaveform**, **WaveformAfterSource**, **WaveformAfterChannel**, or **WaveformAfterProbe** as the Quantity Type; and **<none>** as the Function. The following chart shows the Wave After Channel.



The receiver model can generate clock times internally, and provide the clock ticks to the simulator for post-processing. The clock times are among the parameters returned on the AMI Receiver’s GetWave call. The results of clock tick sampling can be observed only in the Statistical Eye Plot for the "EyeAfterProbe" quantity. By default, the solver uses the clock ticks on the receiver when sampling the statistical eye diagram based on the clock times values produced by the AMI receiver model (option **ami.use_clock_times=1**). If the clock ticks are not present or are deactivated (option **ami.use_clock_times=0**), the eye diagram is sampled at even UI intervals. For more information, see the AMI Analysis Options.

AMI PAM4 Eye Diagrams

PAM4 is a Pulse Amplitude Modulated (PAM) signal with four amplitude levels. An AMI eye report accurately displays the output from such a PAM-4 modulated transmission. A PAM-4 transmission is enabled with PAM-4 reserved parameters. For more information, see "AMI Reserved Parameters" on page 11-149.



This plot shows the multiple eye diagram generated with Gray coding (00, 01, 11, 10 from lowest-to-highest), using a repeating 0, 1, 2, 3 symbol pattern (00011011).

Analyzing AMI Models with VerifEye

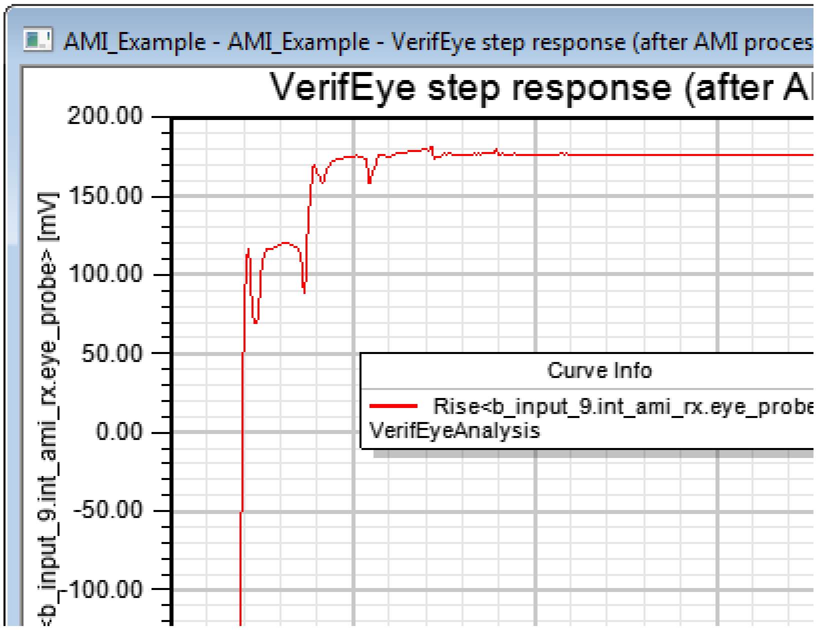
Instead of using the **AMI_GetWave()** data to create a time-domain waveform, use VerifEye to generate the impulse response on the channel parameters and apply the modifications in the **AMI_Init()** functions from both transmitter and receiver. Analysis with VerifEye is equivalent to the Statistical Simulation described in the IBIS AMI specification.

When used with AMI, VerifEye uses only the **AMI_Init()** functions. The **AMI_GetWave()** function is ignored. VerifEye is more accurate than standard AMI analysis, but computes only the linear part of the model's behavior. VerifEye analysis assumes that incoming 0 and 1 bits are independent and occur with equal probability 0.5.

The VerifEye setup and run is the same as for standard VerifEye analysis. See VerifEye and Quick Eye Analyses for further information.

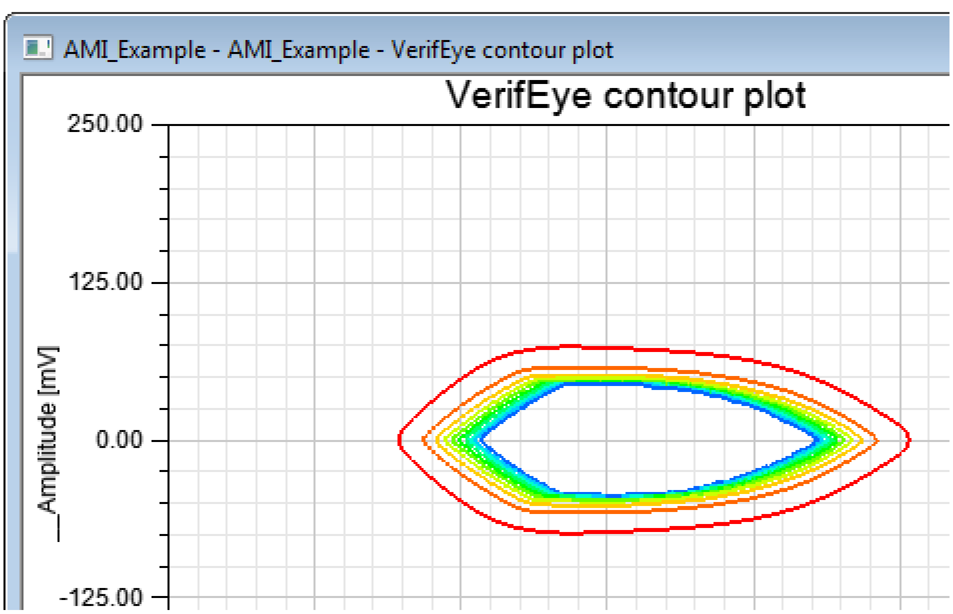
Three kinds of VerifEye reports are available: step response, contour plot, and bathtub curve.

Here is an example of a step response report:



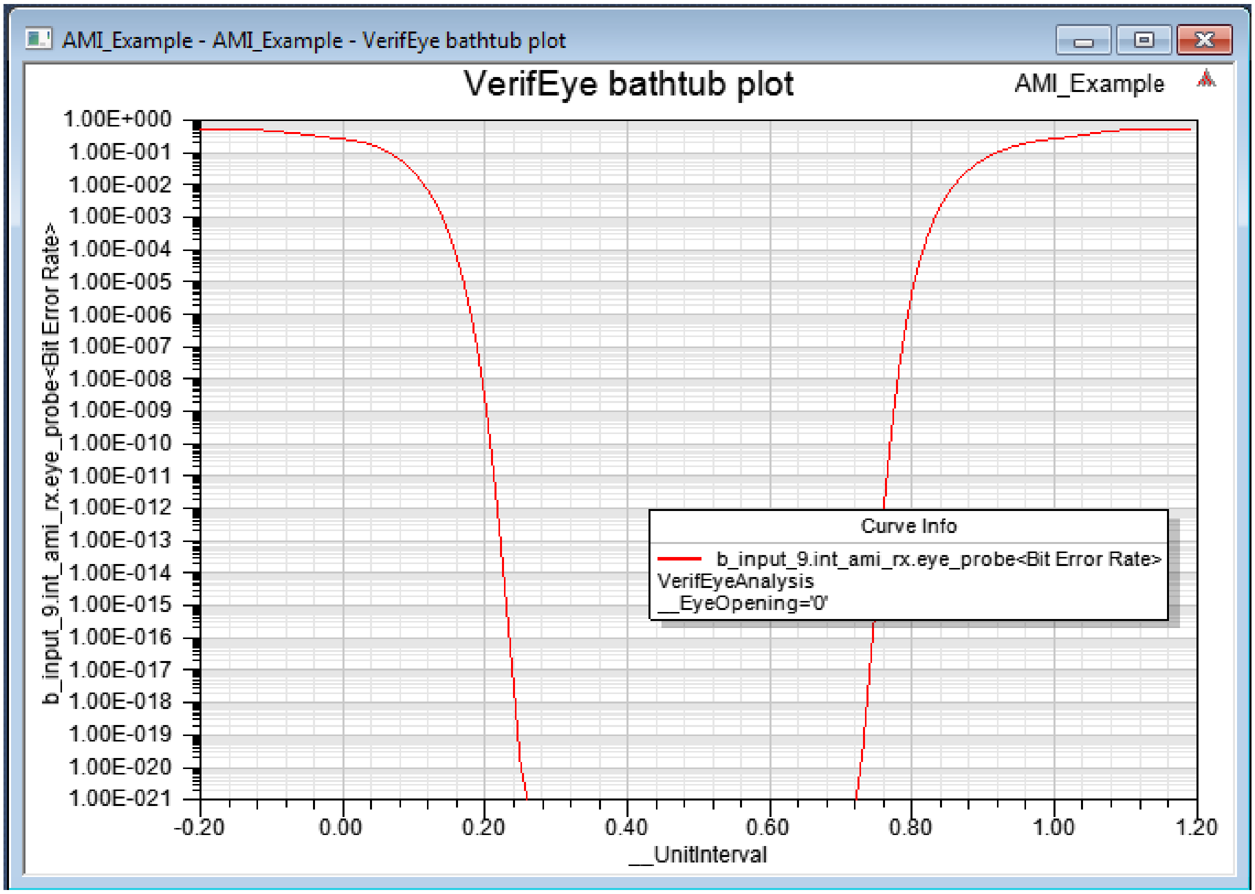
View the initial and final impulse responses in AMI analysis . With VerifEye see only the final step response (after the **AMI_Init()** functions have been applied).

Here is an example of a contour plot:



The VerifEye contour plot is always centered around 0.5 of the UI, while the AMI transient eye takes into account the actual phase shift of the signal.

Here is an example of a bathtub plot using the <Bit Error Rate> trace.:



The bathtub curve for a PAM4 simulation can show the symbol error rate instead of the bit error rate. With PAM4, a symbol is two bits. The report setup is the same, but now select the **<Symbol Error Rate>** trace.

Prototyping with MATLAB Files

Instead of C++ modules, Circuit AMI analysis can simulate from library files containing MATLAB functions for the transmitter and receiver. The parameter files are the same for both C++ and MATLAB implementations.

The MATLAB files for the transmitter and receiver have the standard **.m** extension. Each file should contain the MATLAB code for the **Init()**, **GetWave()**, and **Close()** functions described in the AMI Analysis Technical Notes.

Other than the **library** file, the setups for the source, receiver and analysis are the same for both MATLAB and C++ implementations.

AMI Analysis Technical Notes

The IBIS Algorithmic Modeling Interface simulates user-specified transmitter and receiver models, furnished as C-interface compiled libraries, over a linear channel. The AMI Specification is published in the IBIS Specification (See AMI Analysis References).

AMI Analysis References

[1] *IBIS I/O Buffer Specification*, Version 5.0, ©2008 IBIS Open Forum.
Sections 6c and 10 are the Algorithmic Modeling Interface (AMI) Specification.

[2] *IBIS I/O Buffer Specification*, Version 6.0, ©2013 IBIS Open Forum.
Section 10 is the Algorithmic Modeling Interface (AMI) Specification.

[3] *IBIS I/O Buffer Specification*, Version 7.0, ©2019 IBIS Open Forum.
Section 10 is the Algorithmic Modeling Interface (AMI) Specification.

HSPICE in Circuit Designs

Ansys HFSS Circuit projects offer direct integration with HSPICE® analysis software. This integration does not require the Ansys RSM Service. HSPICE is a trademark of Synopsys, Inc.

Supported HSPICE Integration

Ansys Desktop products offer a direct integration with HSPICE simulation software, and this integration does not require Ansys RSM Service. The following list summarizes the nature of the HSPICE integration that is supported by Ansys HFSS Desktop.

- Only transient simulations are supported
- You can simulate netlists and schematic-based designs
- Solver on Demand co-simulation is available for simulation using Ansys field solvers (HFSS, SIwave, Planar EM, etc.)
- Remote simulation is not fully supported. Direct or indirect include files might not be accessible on the remote machine, or their location might be relative to the original computer, in which case remote simulation fails.
- You can plot results directly from HSPICE output data files
- Custom HSPICE command line options can be added
- Many advanced methods are supported, including .alter
- HSPICE encrypted libraries can be included and simulated
- Many schematic Circuit library components can be used:

— You can override a standard Circuit netlist with a custom product netlist. However, no distributed components are supported.

> **Note:**
>
> To run HSPICE on Linux, set the AD_HSPICE_INSTALLDIR environment variable to the full path of the HSPICE executable. For example, on a 64-bit Linux system where the HSPICE executable is hspice64, set AD_HSPICE_INSTALLDIR to /opt/hspice-2009.03-SP1/hspice/linux/hspice64.

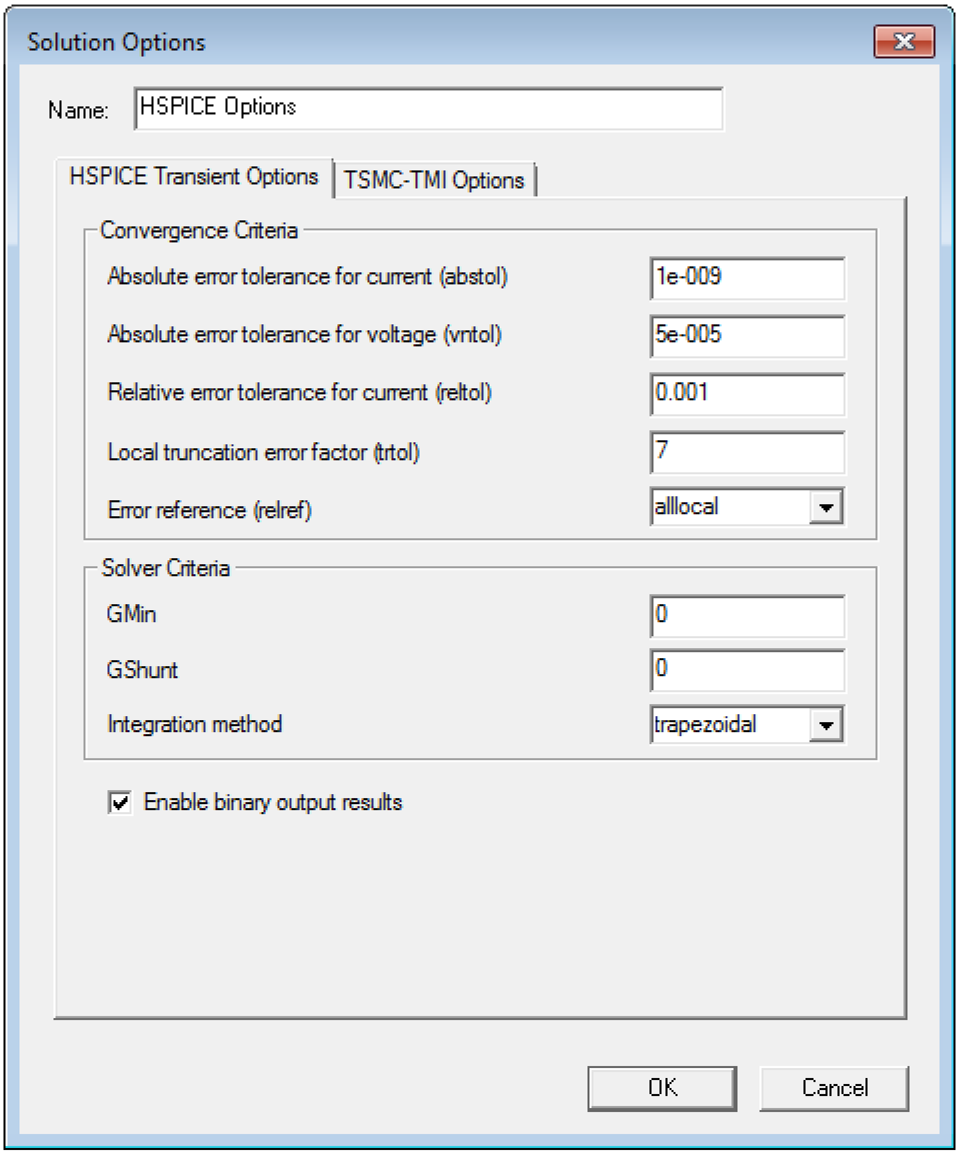
Running an HSPICE Transient Analysis

From the **Schematic Editor**, perform the following steps to set up an HSPICE transient analysis, run the analysis, and view the results.

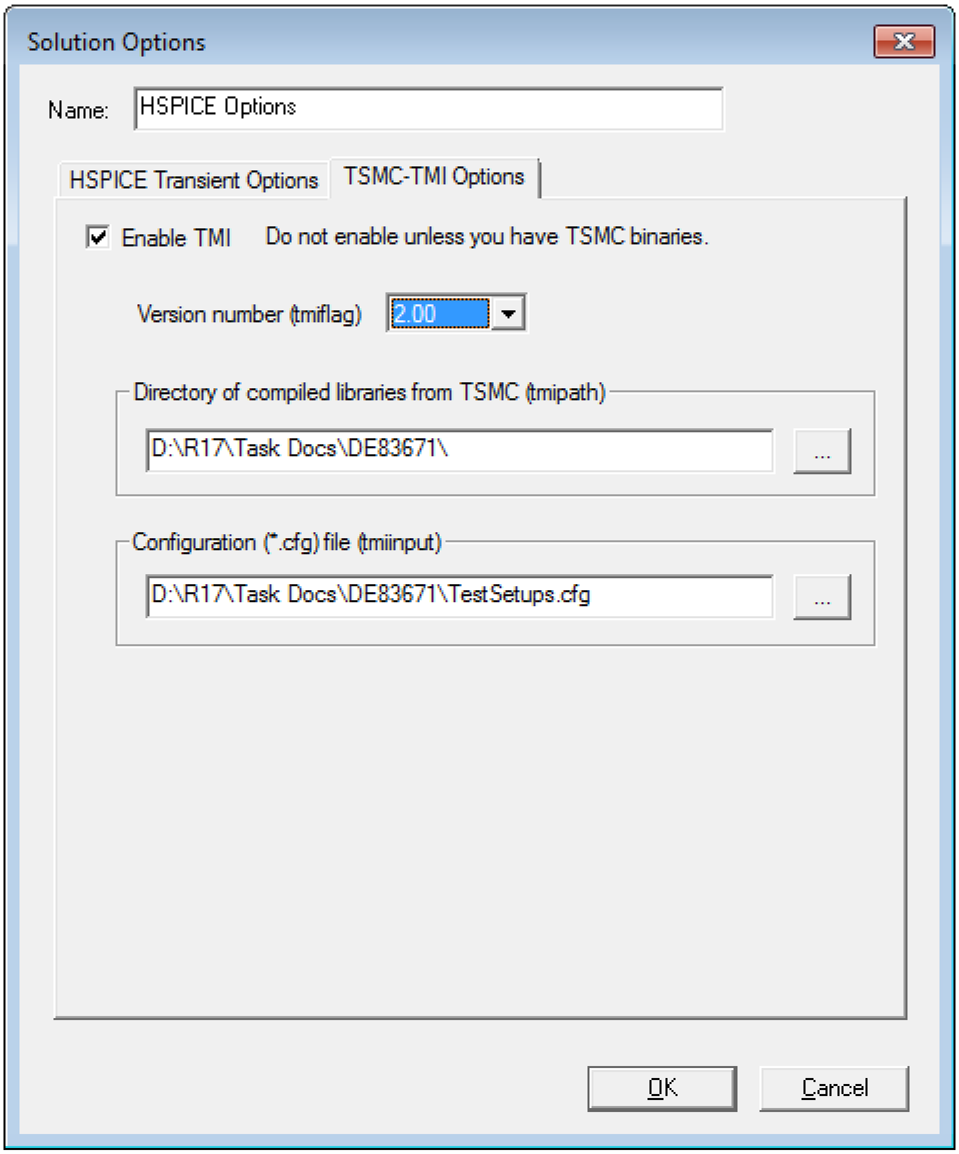
1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add HSPICE Solution Setup** > **Transient Analysis** to open the **HSPICE Transient Analysis** window.
2. Enter an **AnalysisName** or accept the default (e.g., "HSPICE Transient1").
3. For most simulations, leave the **Disable** box unselected. (Selecting this box lets you store multiple solution setups for later use. Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
4. Under **Analysis Control**, specify **Step** time and **Stop** time using the drop-down menus. Optionally, Click the **RunLevel** slider to set the **Accuracy** level.
5. To add a sweep, locate the **Sweep Variable** area in the **Transient Analysis** window.
   - From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
   - In the **Variable** group box, select **Temp** (or the name of a variable, if one has been defined for the design). Next, select one of the following: **Single value, Linear step, Linear count, Decade count,** or **Octave count**.
   - Enter the sweep values into the **Value** field (for **Single** value), or into the **Start**, **Stop**, and **Step** text fields (for **Linear, Decade,** or **Octave count**), and ensure that the appropriate units are selected for each.
   - For more information, see Variable Sweep.
   - Click **Add**, then **OK** to close the **Add /EditSweep** window and reopen the **Transient Analysis** window.
6. Optionally, use the fields in the **Solution Options** group box to select or add **Transient Analysis** options and other **HSPICE** options to the design.
   - Click **Select** on the **Name** field to open the **Select Solution Options** window.
   - If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the HSPICE **Transient**

Analysis window. The named option set appears in the **Name** field in the **Solution Options** group box.

- To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Make the appropriate changes to option values, then click **OK** to return to the **SelectSolutions** window. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the HSPICE **Transient Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.
- The **HSPICE Transient Options** tab contains basic options for controlling the convergence for transient analysis using HSPICE. See Transient Analysis Options for details on most of these options.



- The HSPICE **TSMC-TMI Options** tab contains settings that apply only to HSPICE simulation using TMI models. Your installation must include DLLs from TSMC to use these options. Click the **Enable** check box to enable the HSPICE solver to access the TSMC-TMI DLLs and activate the fields in the **TMI Options** tab. See TSMC-TMI Options for details on the TSMC-TMI analysis options.



7. Optionally, click **Edit** in the **Additional** field on the HSPICE **Analysis Setup** window to open the **Edit Additional Options** text-entry window.
   - Click the field to enter any HSPICE options exactly as they are to appear in the netlist. Do not include the keyword ".OPTIONS". The .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement. The line can contain multiple options settings. Use spaces to separate the options settings. To modify or

delete an option once it has been added, edit or delete its entry in the list. To clear all options, delete the entire line.

8. Click **OK** to return to the HSPICE **Transient Analysis** window. Options in the **Additional** field are automatically added to the netlist when the solution is performed. For more information, see the HSPICE user manuals.
9. To run the simulation, click **Circuit** > **Analyze**.
   - If the circuit is set up correctly, the analysis begins and a progress bar appears. (Note, however, that the progress bar does not move during HSPICE simulations as it does with Circuit simulations.)
   - If the analysis does not begin or does not complete successfully, check the **Message Manager** window.
10. To display results, click **Circuit** > **Create Report** (Alternatively, select **Create Report** on the **Results** icon in the **Project Manager** window window) to open the **Create Report** window.
   - Select a **Report Type** and a **Display Type**, then click **OK**. When the **Report** window appears, make the appropriate selections for context, solution, and plotting range. Next click **Add Trace**, then click **Done** to open a Report window to display a graph of the analysis results.
   - For more information, see Generating Reports and Post-processing.

Troubleshooting HSPICE Analysis

Use the following guidelines to troubleshoot HSPICE simulations.

- Check the HSPICE logs.
- Check for a locked project.
- Check for path errors:

— Verify that the path is correct to specify the HFSS installation directory.

— Verify that a full path is used to specify the project directory.

The following table lists the nature and solution of some common troubleshooting issues for running HSPICE simulations in the **Electronics Desktop**.

Issue	Solution
HSPICE does not run from within Electronics Desktop.	• Verify that HSPICE runs correctly outside of Ansys. Consult the HSPICE documentation for questions about installation, licensing and basic operation. HSPICE must work correctly on its own before it can be run inside the Ansys framework.
HSPICE runs outside of Electronics Desktop but not from	It is likely that Electronics Desktop cannot locate the HSPICE executable. On Windows and Linux platforms:

within Electronics Desktop.	• Set the AD_HSPICE_INSTALLDIR environment variable to the full path of the HSPICE executable. For example, on a 64 bit Linux system where the HSPICE executable is hspice64, set AD_HSPICE_INSTALLDIR to be: /opt/hspice-2009.03-SP1/hspice/linux/hspice64 • Note that if AD_HSPICE_INSTALLDIR is set, the value of INSTALLDIR is ignored. On Windows PCs: • Set the AD_HSPICE_INSTALLDIR environment variable to the full path of the HSPICE executable. For example, on Windows where the HSPICE executable is hspice.exe, set AD_HSPICE_INSTALLDIR to be: D:\synopsys\Hsp ice_H-2013.03-S P2\WIN64\hspice.exe • Note the value of the INSTALLDIR environment variable. Look for the HSPICE executable (hspice.exe) in a "bin" directory beneath INSTALLDIR. That is, the executable should be located in: <value of INSTALLDIR>/bin/hspice.exe • If the HSPICE executable is not at this location, adjust the value of INSTALLDIR accordingly or verify your HSPICE installation. • Note that the value of INSTALLDIR is ignored if AD_ HSPICE_INSTALLDIR is set.
HSPICE runs from within Electronics Desktop but does not give expected results.	• Check the HSPICE logs by right-clicking on the analysis setup and selecting "Browse log file." This displays the log file generated by HSPICE. • Check for a locked project. • Check for path errors to libraries and included files.

Virtual Compliance Module Toolkit

The Virtual Compliance Module (VCM) Toolkit is used to test the level of compliance of circuit transient simulations for DDR4/GDDR5 and DDRx. **Electronics Desktop** VCM testing generates reports that utilize both a Compliance Test Viewer and Embedded Waveform Viewer to help you configure:

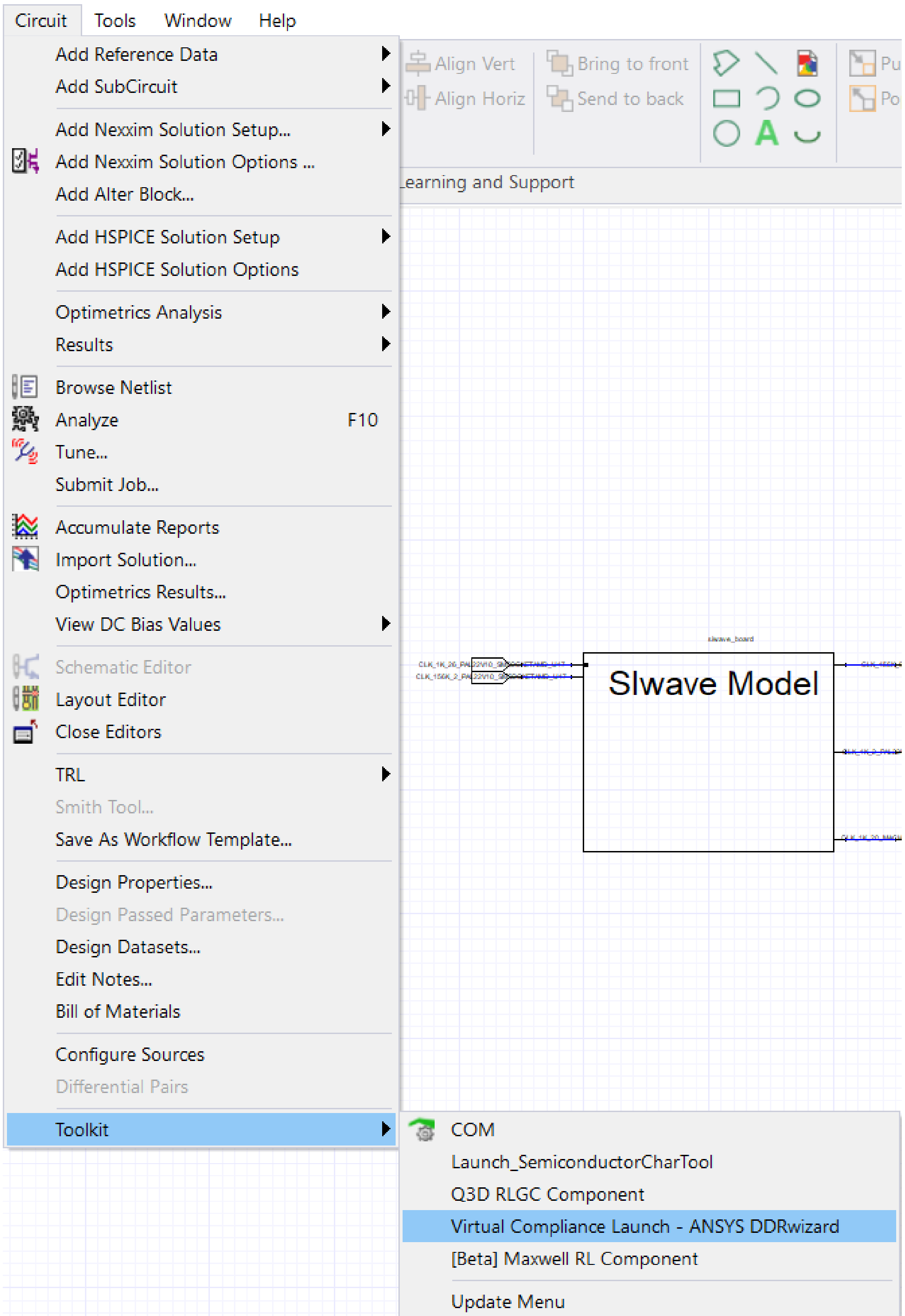
- Net mapping
- DDR4 net classification
- GDDR5 net classification
- I/O power assignment
- Net type
- Etc

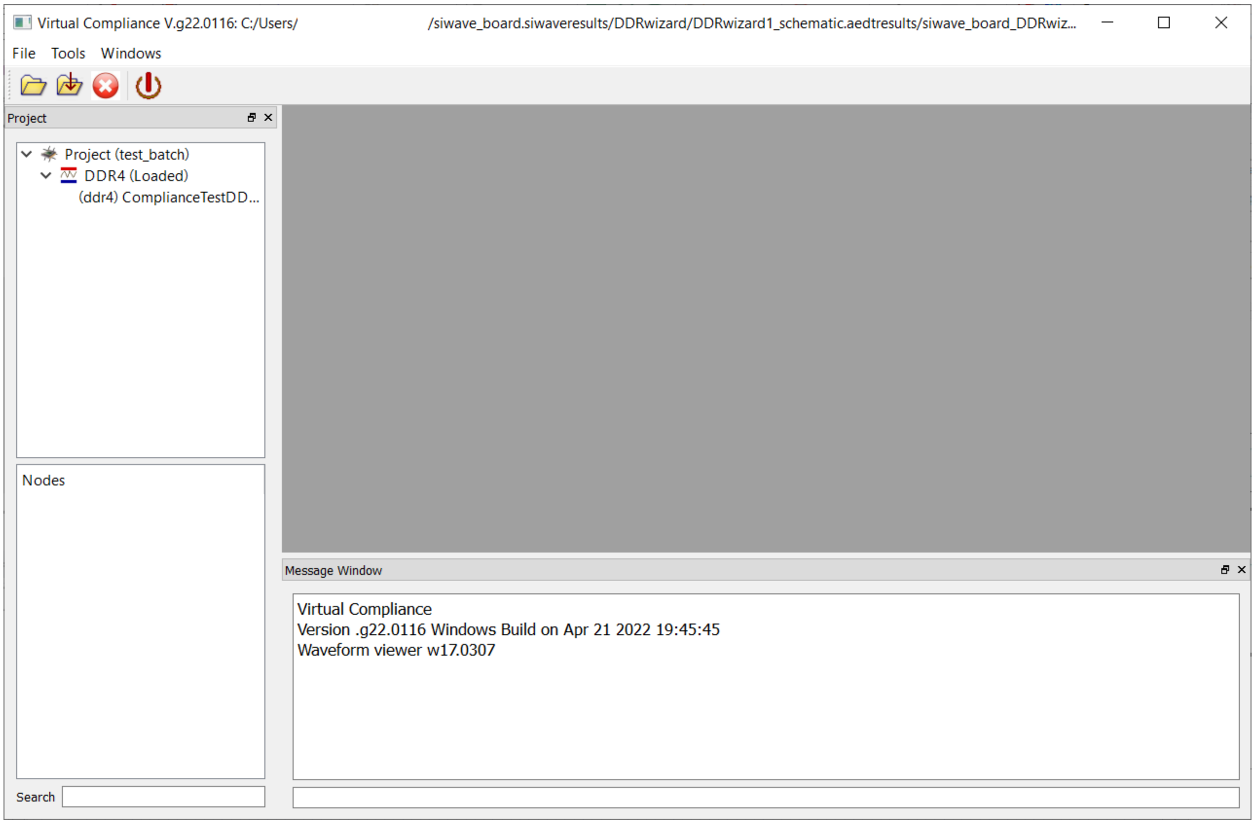
Launch Virtual Compliance Module

To launch the Virtual Compliance Module, the Circuit Schematic has to be created from Ansys SIwave's DDRwizard. For setup instructions, refer to SIWave help:

SIWave > Running Simulations > Cosimulation with Ansys Electronics Desktop > Cosimulation using DDR Wizard

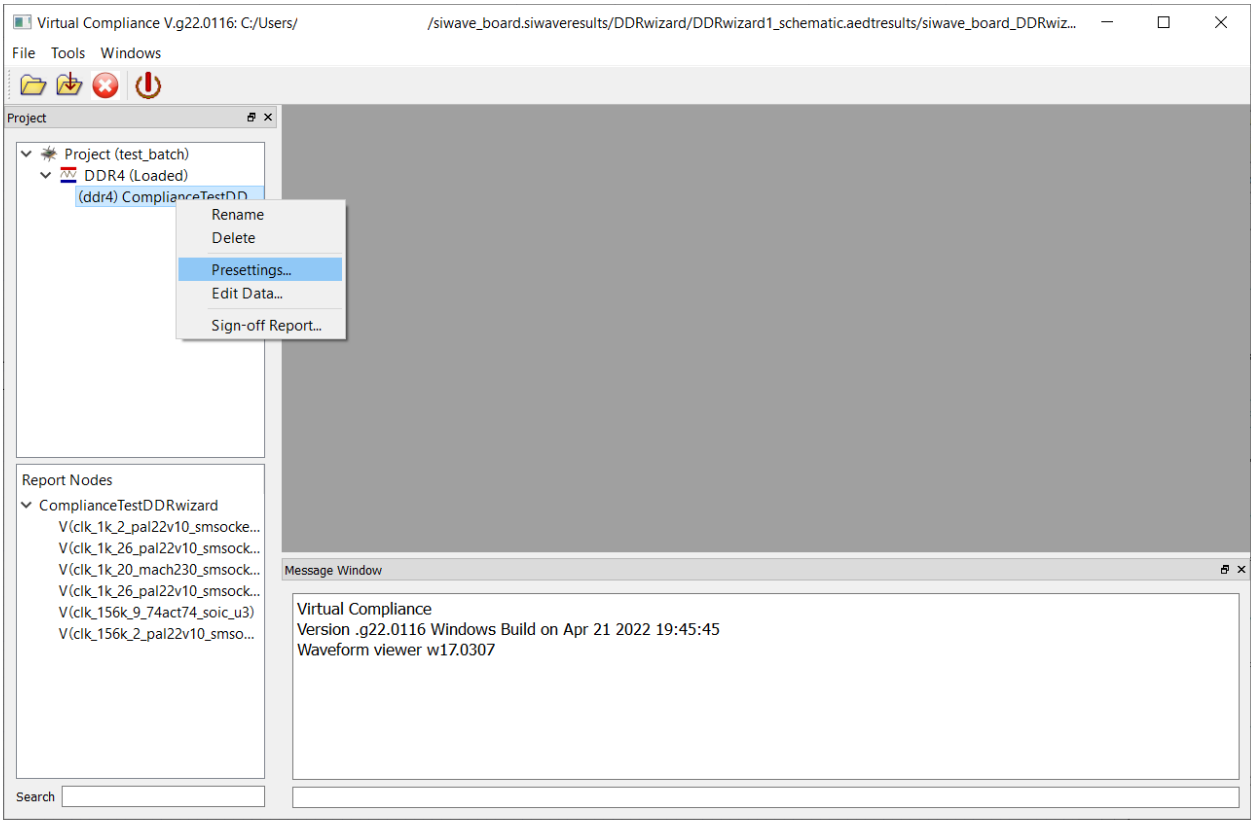
Running the DDRWizard from SIWave will automatically open the schematic in Circuit. To launch the Virtual Compliance Tool, select **Circuit > Toolkit > Virtual Compliance Launch - Ansys DDR Wizard**.

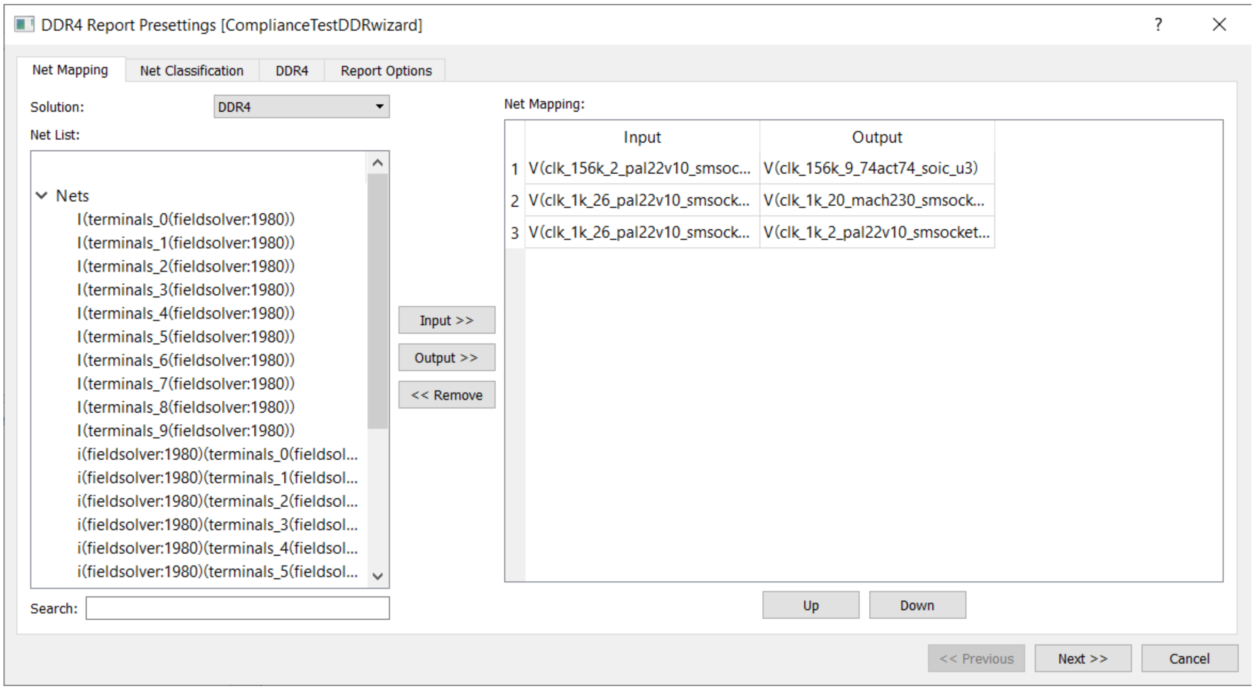

Circuit
Tools
Window
Help
Add Reference Data
Add SubCircuit
Add Nexxim Solution Setup...
Add Nexxim Solution Options ...
Add Alter Block...
Add HSPICE Solution Setup
Add HSPICE Solution Options
Optimetrics Analysis
Results
Browse Netlist
Analyze F10
Tune...
Submit Job...
Accumulate Reports
Import Solution...
Optimetrics Results...
View DC Bias Values
Schematic Editor
Layout Editor
Close Editors
TRL
Smith Tool...
Save As Workflow Template...
Design Properties...
Design Passed Parameters...
Design Datasets...
Edit Notes...
Bill of Materials
Configure Sources
Differential Pairs
Toolkit
Align Vert
Align Horiz
Bring to front
Send to back
Learning and Support
Slwave Model
COM
Launch_SemiconductorCharTool
Q3D RLGC Component
Virtual Compliance Launch - ANSYS DDRwizard
[Beta] Maxwell RL Component
Update Menu



To launch the presettings window:

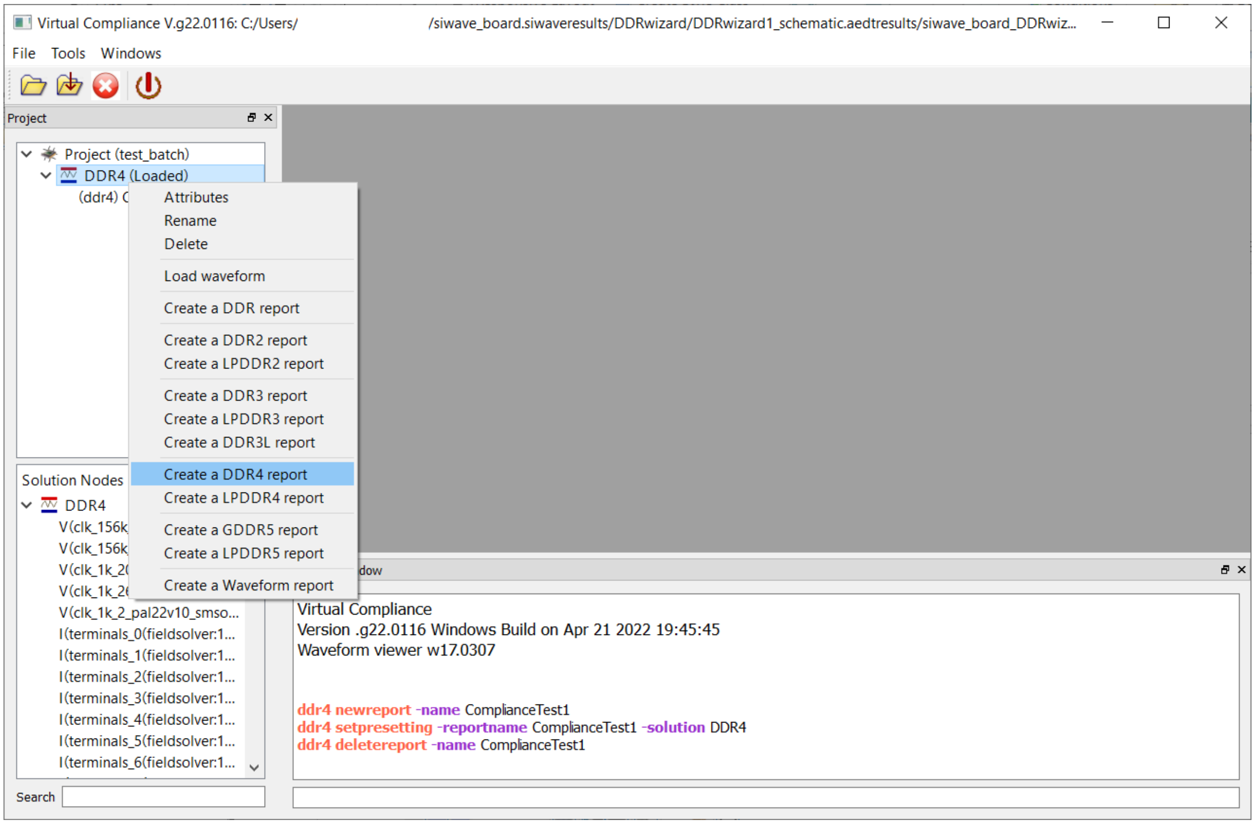
1. Right -click the DDR Compliance Test.

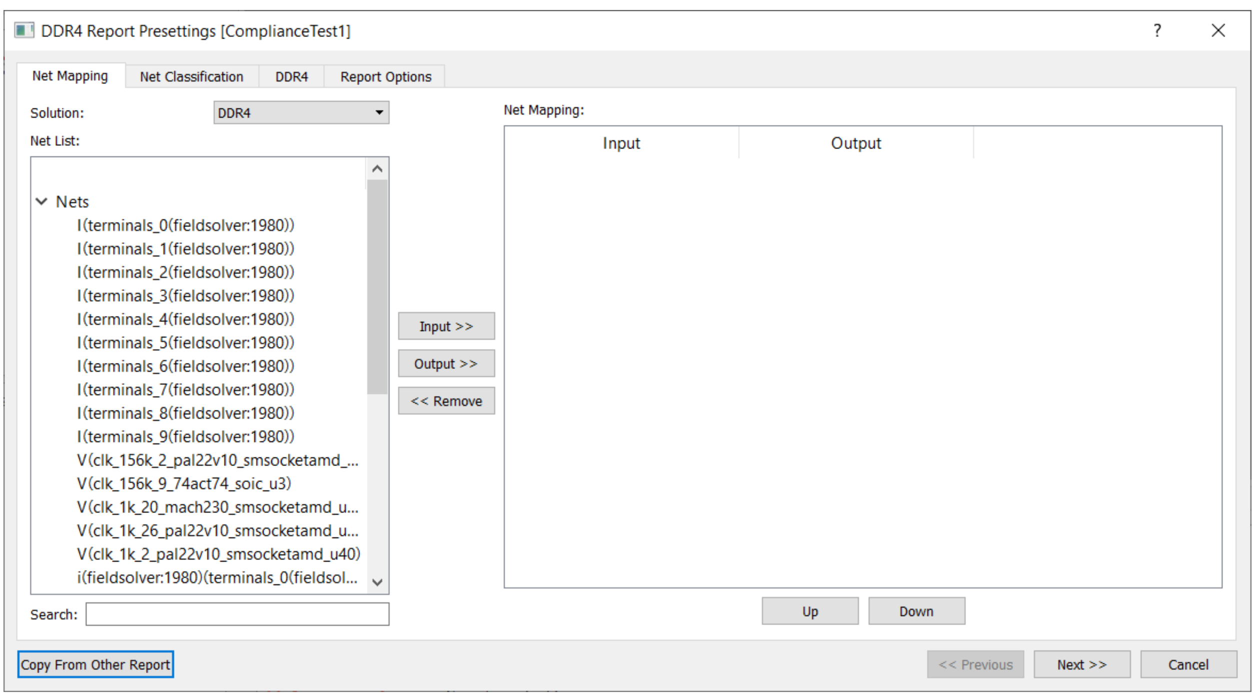




To launch the Report Presettings window:

1. For a DDR design, right-click **DDR** , then click **Create DDR report**.

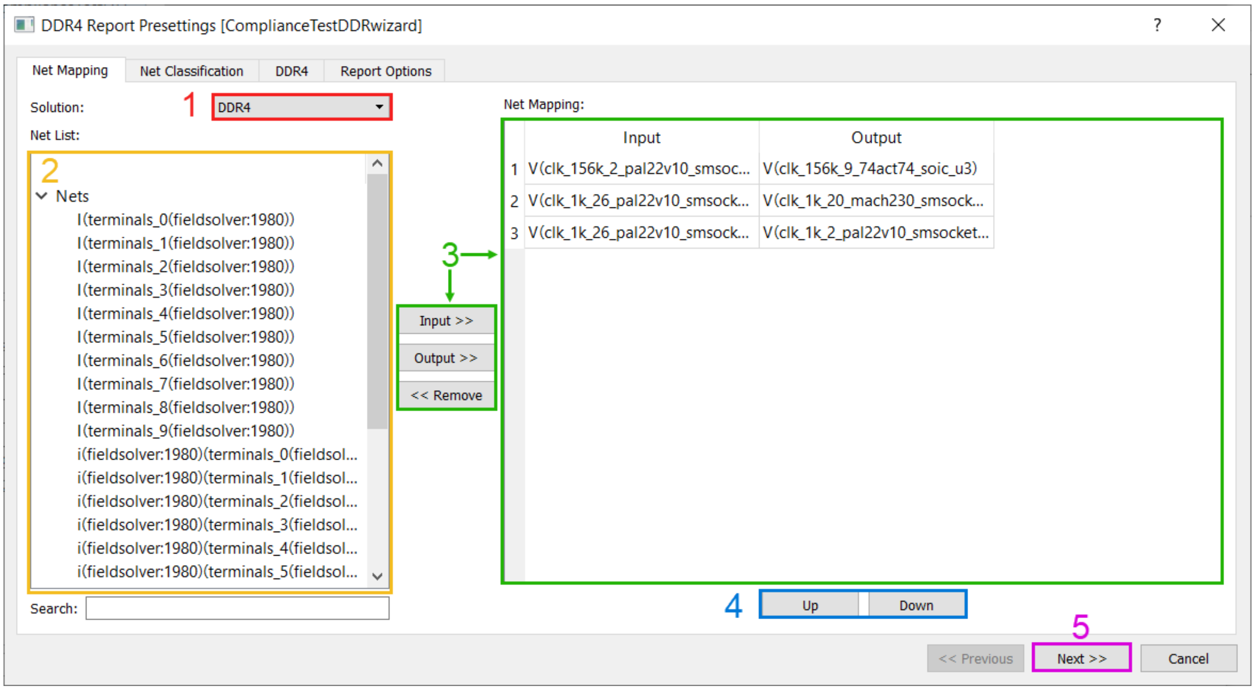

Virtual Compliance V.g22.0116: C:/Users/ /siwave_board.siwaveresults/DDRwizard/DDRwizard1_schematic.aedtresults/siwave_board_DDRwiz...
File Tools Windows
Project
Project (test_batch)
DDR4 (Loaded)
Attributes
Rename
Delete
Load waveform
Create a DDR report
Create a DDR2 report
Create a LPDDR2 report
Create a DDR3 report
Create a LPDDR3 report
Create a DDR3L report
Create a DDR4 report
Create a LPDDR4 report
Create a GDDR5 report
Create a LPDDR5 report
Create a Waveform report
Solution Nodes
DDR4
Virtual Compliance
Version .g22.0116 Windows Build on Apr 21 2022 19:45:45
Waveform viewer w17.0307
ddr4 newreport -name ComplianceTest1
ddr4 setpresetting -reportname ComplianceTest1 -solution DDR4
ddr4 deletereport -name ComplianceTest1
Search


DDR4 Report Presettings [ComplianceTest1]
Net Mapping
Net Classification
DDR4
Report Options
Solution:
DDR4
Net List:
Nets
I(terminals_0(fieldsolver:1980))
I(terminals_1(fieldsolver:1980))
I(terminals_2(fieldsolver:1980))
I(terminals_3(fieldsolver:1980))
I(terminals_4(fieldsolver:1980))
I(terminals_5(fieldsolver:1980))
I(terminals_6(fieldsolver:1980))
I(terminals_7(fieldsolver:1980))
I(terminals_8(fieldsolver:1980))
I(terminals_9(fieldsolver:1980))
V(clk_156k_2_pal22v10_smsocketamd_...
V(clk_156k_9_74act74_soic_u3)
V(clk_1k_20_mach230_smsocketamd_u...
V(clk_1k_26_pal22v10_smsocketamd_u...
V(clk_1k_2_pal22v10_smsocketamd_u40)
i(fieldsolver:1980)(terminals_0(fieldsol...
Search:
Input >>
Output >>
<< Remove
Net Mapping:
Input
Output
Up
Down
Copy From Other Report
<< Previous
Next >>
Cancel

Virtual Compliance for DDR4 and GDDR5

The following topics describe how to conduct compliance testing for DDR4 and GDRR5.

Net Mapping (DDR4 and GDDR5)



1. List all the analysis setup/solution, each setup/solution loads corresponding waveform information.

2. Net list lists all the nets which have a waveform information, the nets name can be searched by keywords.

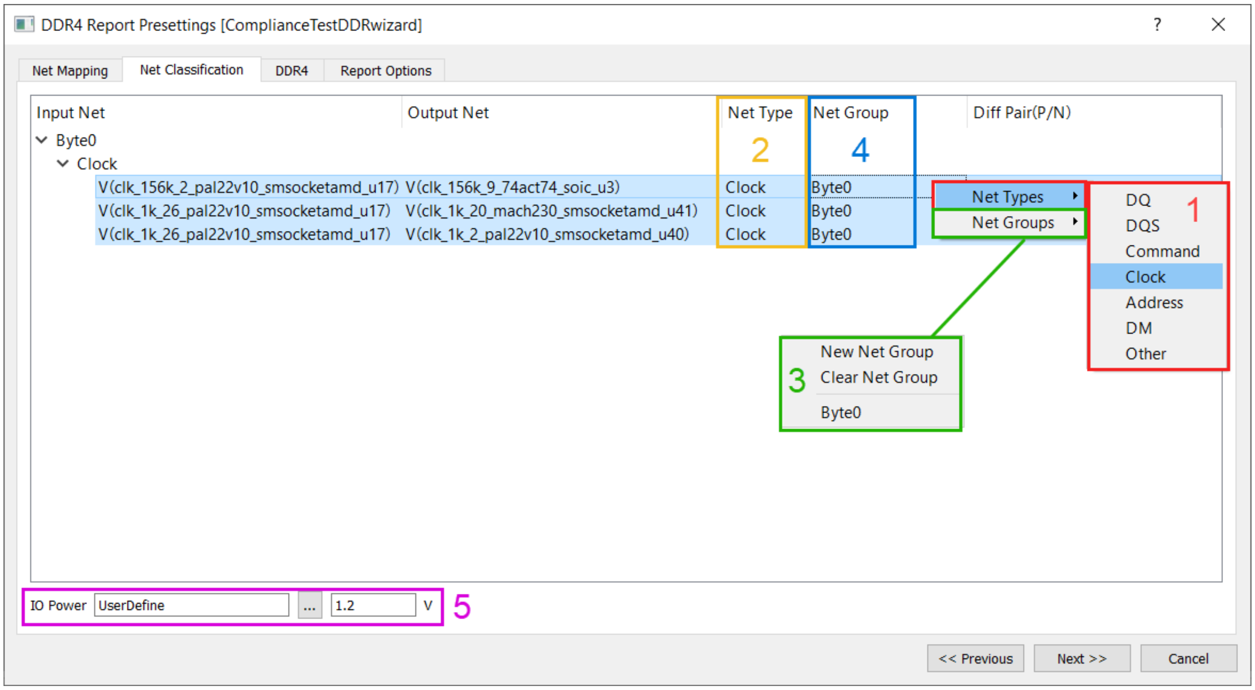
3. Select the nets into input or output list for virtual compliance testing, and make the in-out connection.

4. Click **Up** and **Down** to ensure the in-out connection correct, main usage of the in-out connection is Self-delay calculation, the self-delay table calculates the timing delay between Input and Output, if you don’t need report self-delay parameter, ignore Input nets and assign Output nets only.

5. You can navigate back and forth through the Presettings pages by clicking **<<Previous** and **Next>>**, or click **Cancel** to quit.

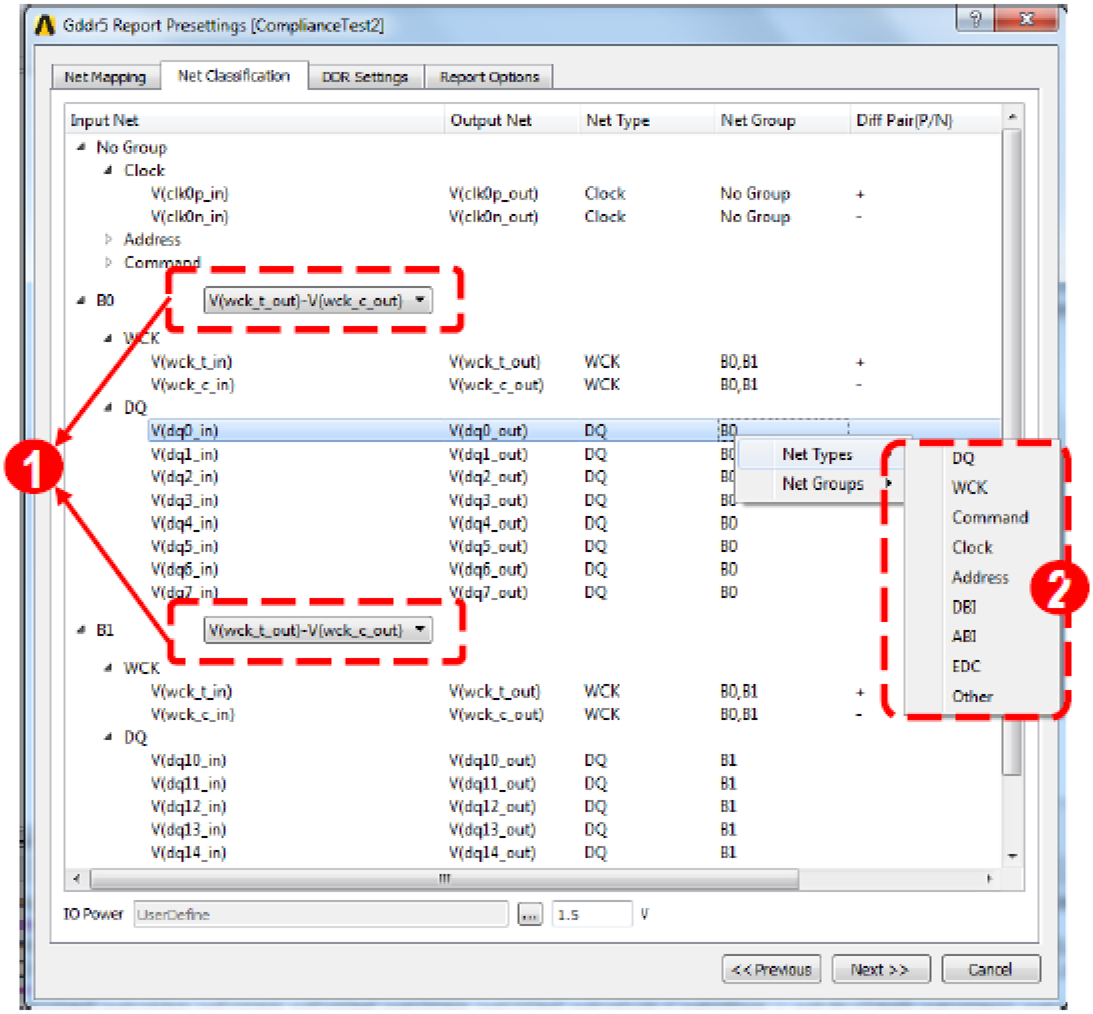
Net Classifications

The following shows how to configure **net classification for DDR4**.



1. Right-click the highlight nets to define Net Type for the chosen nets.
2. All the net type defined nets shows their net type in the list, otherwise keep it as Others.
3. Create new net groups or assign nets to a existing group.
4. Net Group column shows all the net group belongings.
5. Define the IO power value, using for logic level calculation and other parameters.

The following shows how to configure **net classification for GDDR5**.



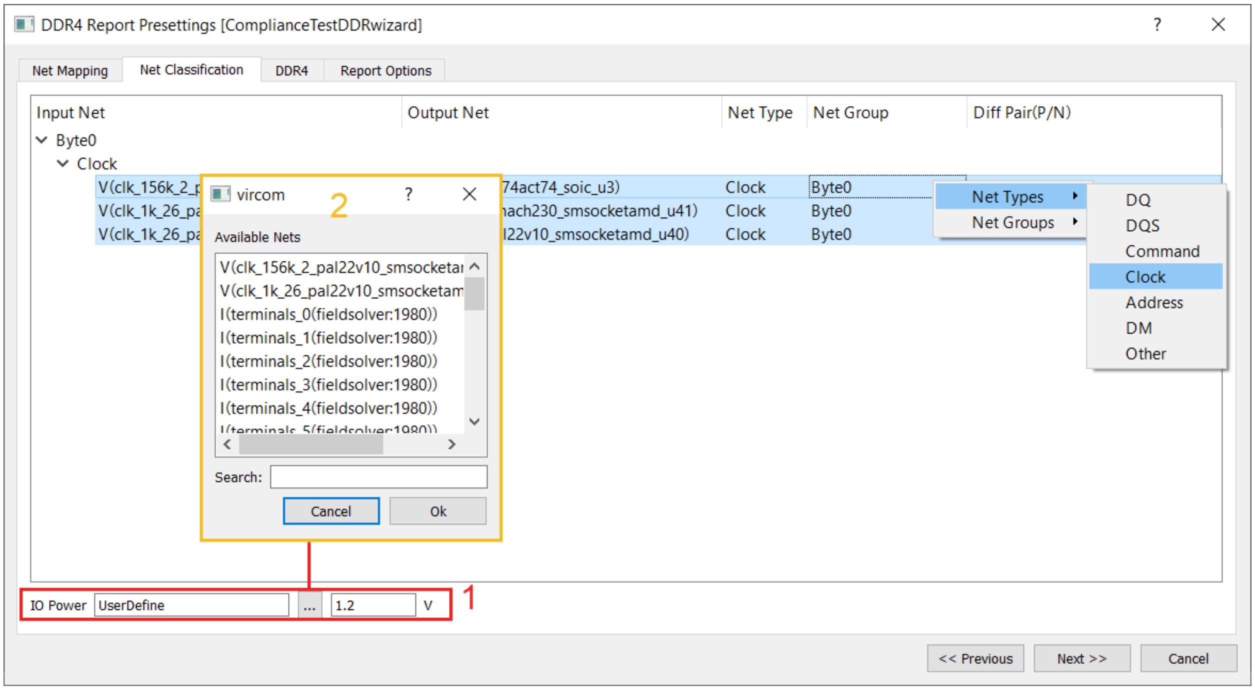
1. Support same strobe pair using in different Byte-lane group (GDDR5 feature).
2. More GDDR5 net types are supported.

Net Type

NET Type	DDR4	Note
Clock	CK, CK#	Differential Pair
DQS	DQSx, DQSx#	Differential Pair
Command	CKE, CS#, RAS#, CAS#, WE#, ODTx	Reference Voltage: Vrefc
Address	BG[3:0], BA[3:0], A[13:0]	Reference Voltage: Vrefc
DQ	DQx, DMx, DBIx	Reference Voltage: Vrefd

NET Type	GDDR5	Note
Clock	CK_t, CK_c	Differential
Forward Clock (DQS)	WCK01_t, WCK01_c & WCK23_t, WCK23_c	Differential
Command	RAS_n, CAS_n, WE_n, CS_n, and CKE_n	Reference Voltage: Vrefc
Address	BA[3:0], A[13:0], and ABI_n	Reference Voltage: Vrefc
DQ	DQx, DBIx_n, EDCx	Reference Voltage: Vrefd

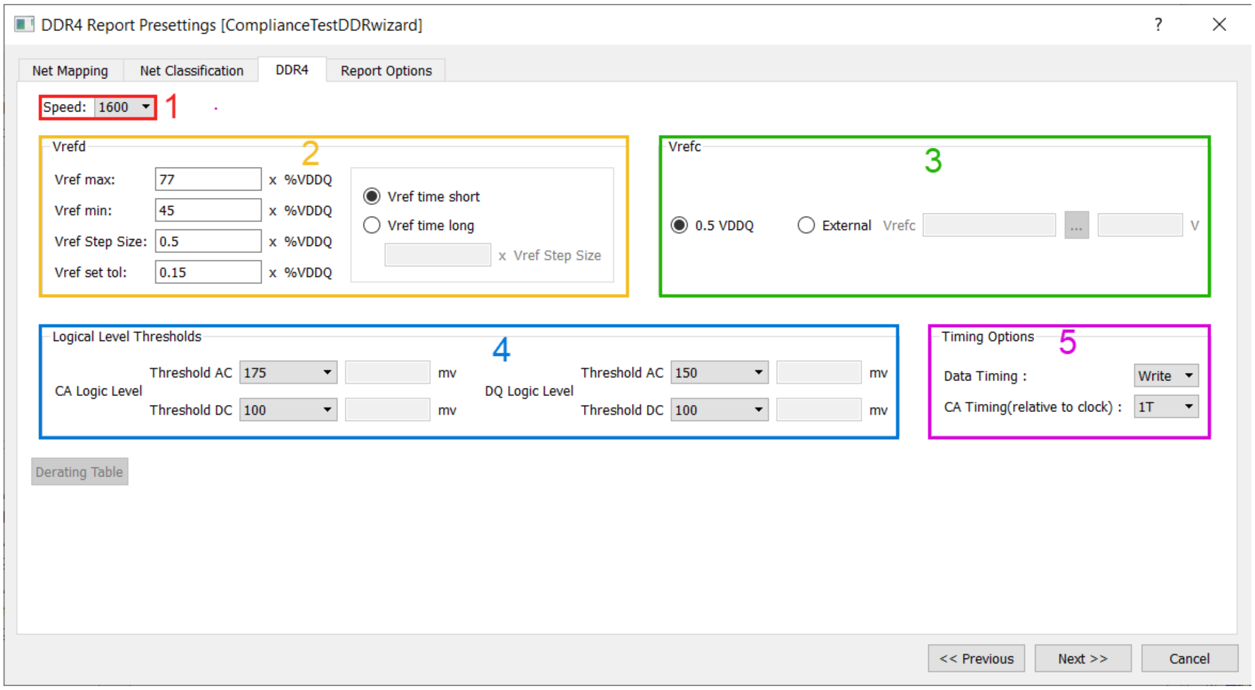
I/O Power Assignment



Both virtual I/O power input and real I/O power net assignment are supported:

1. You can input an I/O power value as a virtual I/O power.
2. Or click **[...]** to assign a net as an I/O power net. The Toolkit can automatically calculate the I/O power according to your selections.

DDR4 Related Settings

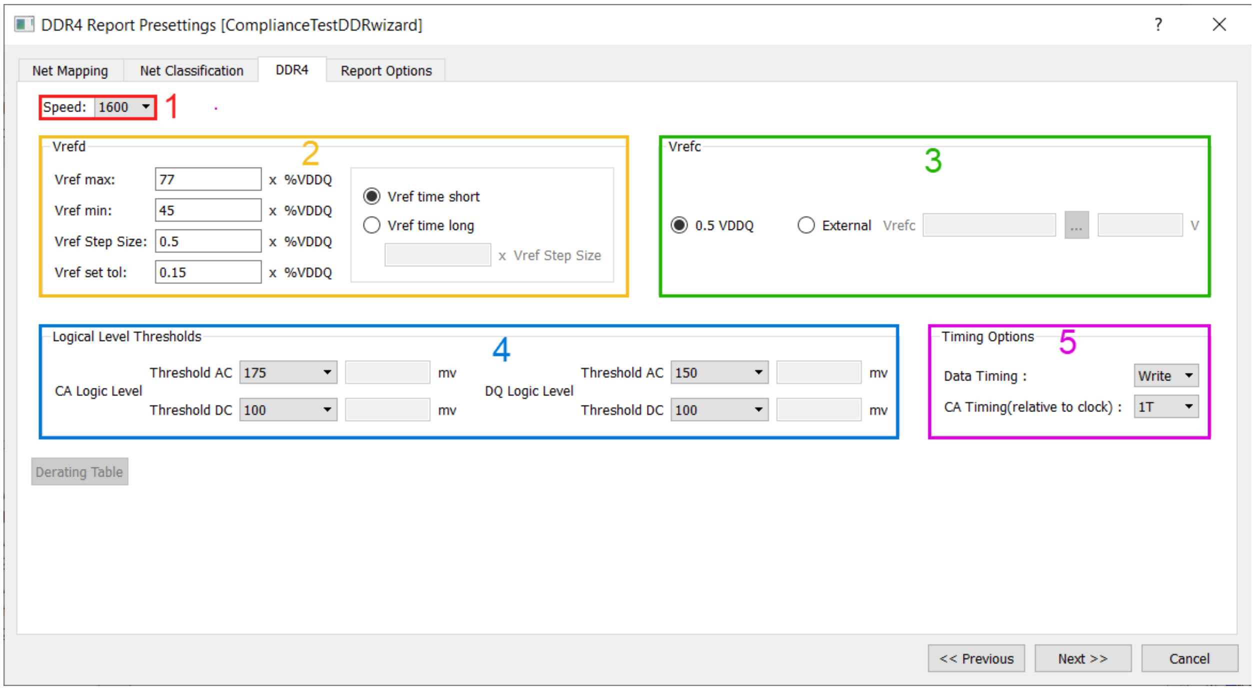


1. Support typical DDR4 speed: 1600Mbps, 1866Mbps, 2133Mbps, 2400Mbps, 2666Mbps and 3200Mbps.
2. All of these settings are Vrefd-related settings that follow **JEDEC Spec JESD79-4 “Table 36 — DQ Internal Vref Specifications** ”.
3. Vrefc settings:
   a. Vrefc = 0.5 IO power
   b. External Vrefc; assign a net as Vrefc net. The Toolkit can automatically calculate the Vrefc voltage value according to your selection.
4. dropdown menu for AC/DC Threshold selection; also supports user-defined values.
5. Write/Read options:
   a. Write Mode
   b. Read Mode

   Command, address and timing options:

a. 1T= CA signal period is 1x the clock period.
b. 2T = CA signal period is 2x the clock period.

GDDR5 Related Settings

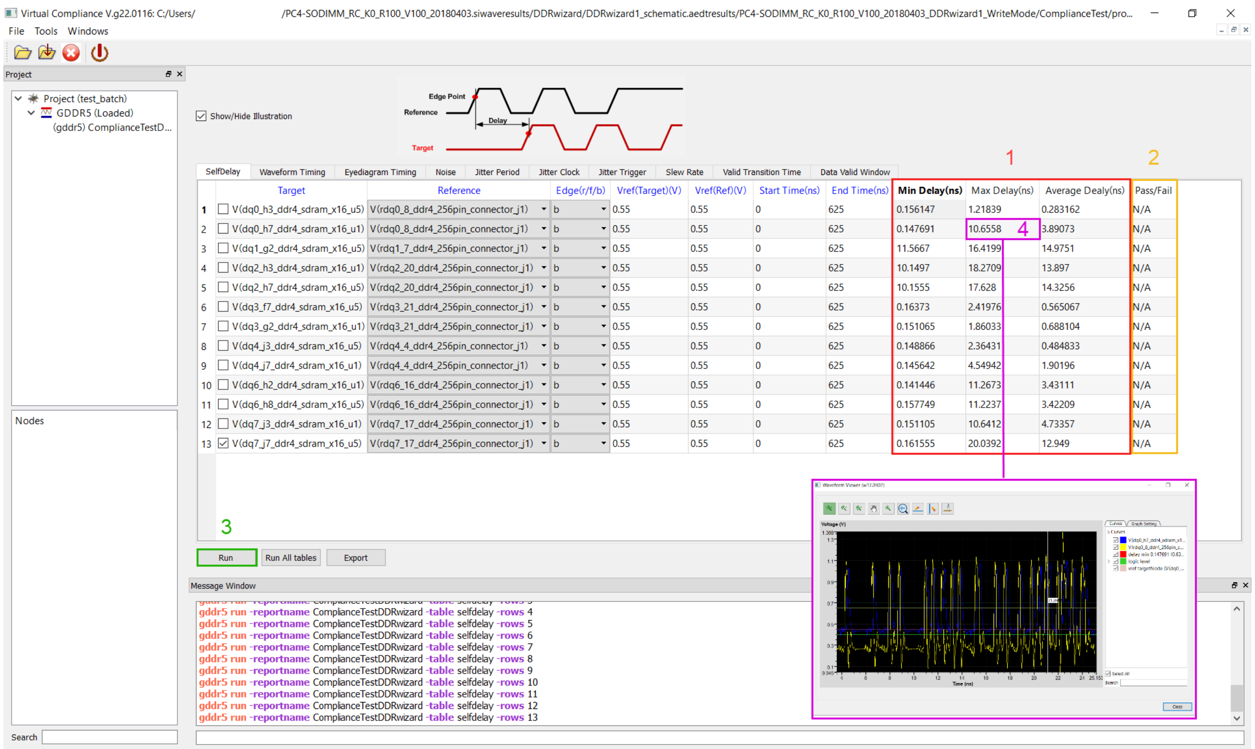


1. It support both idea IO power input and real IO power net assignment 2133Mbps, 2400Mbps, 2666Mbps and 3200Mbps.
2. All of these settings are Vrefd training related setting follow **JEDEC Spec JESD79-4 “Table 36 — DQ Internal Vref Specifications** ”.
3. Vrefc settings, 2 Options:
   a. Vrefc = 0.5 IO power
   b. External Vrefc, assign a net as Vrefc net, tools can automatic calculate the Vrefc voltage value according to your selection.
4. dropdown menu for AC/DC Threshold selection, it also support user define values.
5. Write/Read options:
   a. Write Mode
   b. Read Mode

   Command, Address and control timing options:

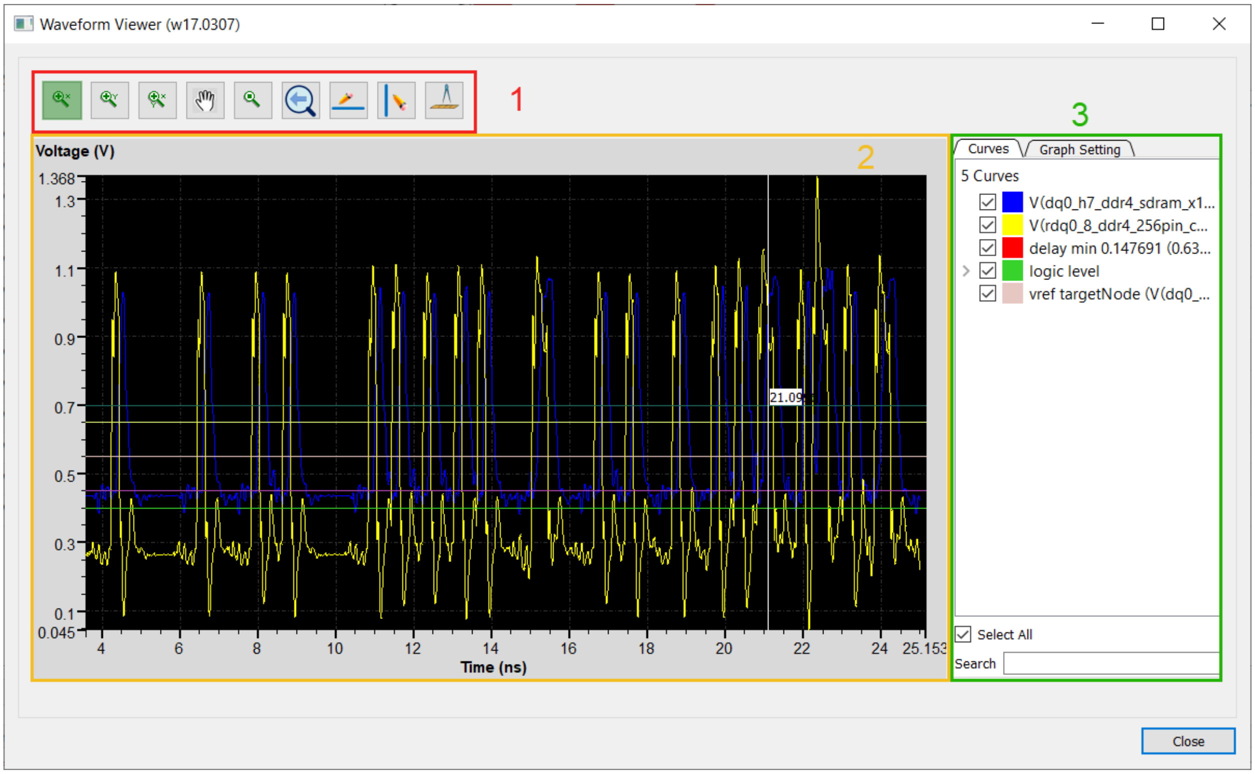
   a. 1T. Means CA signal period is 1 time clock period.
   b. 2T. Means CA signal period is twice clock period.

Compliance Test Window (GDDR5)



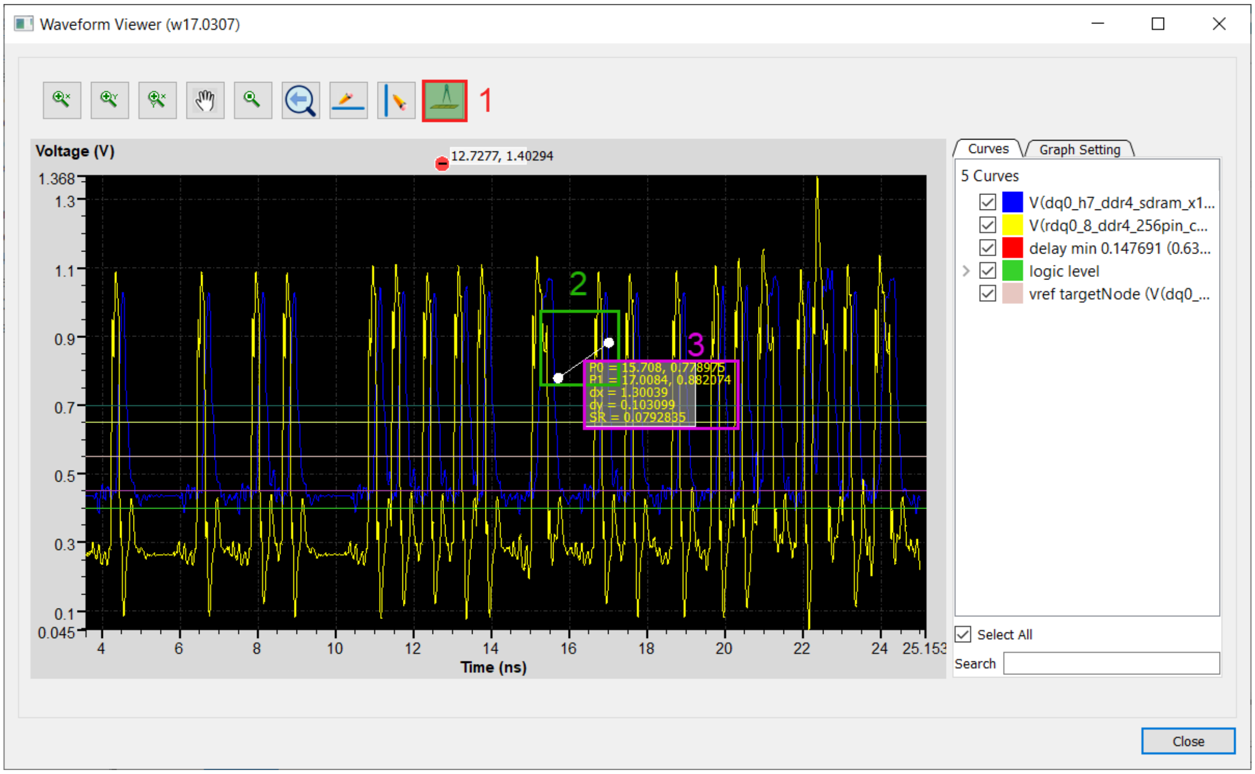
1. Report-parameter area.
2. Pass/Fail result according to JEDEC SPEC.
3. “Run” parameter calculation and Pass/Fail result.
4. Double-click the report parameter to open the embedded waveform viewer.

Embedded Waveform Viewer



1. Tool bar: zoom in, fit view, previous view and measuring tools.
2. Waveform Plot area.
3. Curve selection and Graph setting.

Measuring

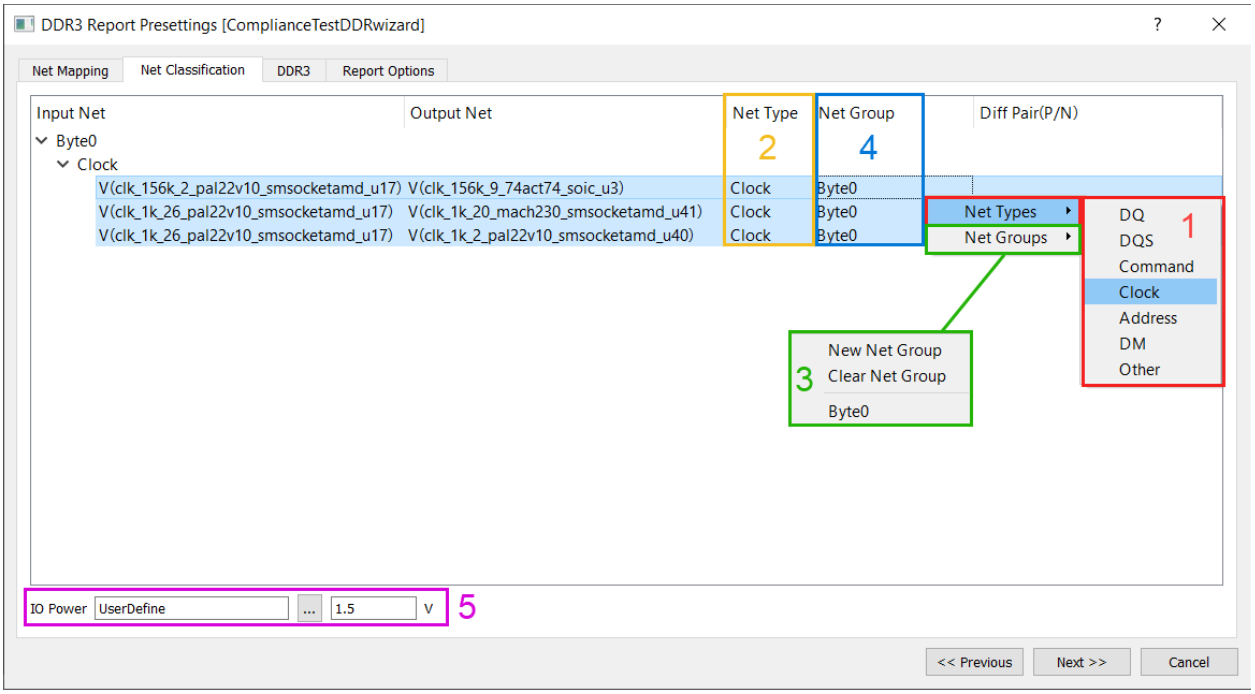


1. Click the “measure point” button.
2. Click in the display to set 2 points.
3. The delta x, delta y and slope of these 2 points is shown in the display.

Virtual Compliance for DDRx

The following topics describe how to conduct compliance testing for other DDRx.

Net Mapping (Other DDRx)

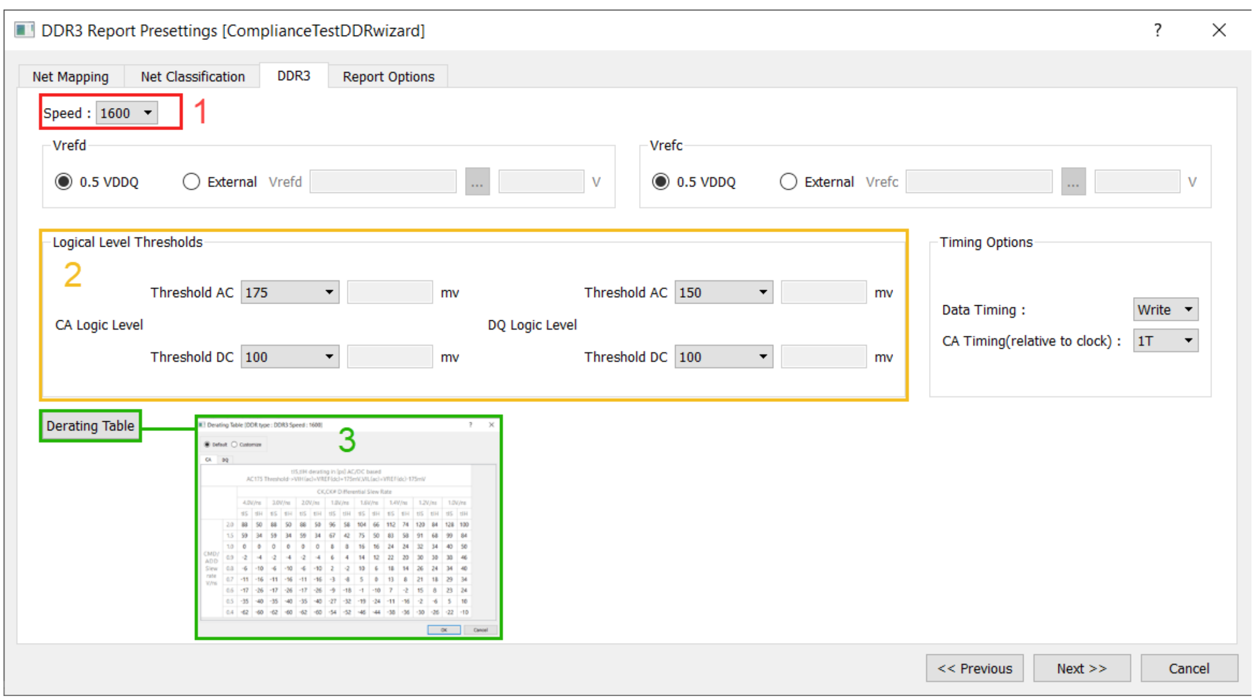


1. Right-click the highlight nets to define a Net Type for the chosen nets.
2. After the net type is defined, the net types are displayed in the list. Undefined net types are displayed as "Others".
3. You can create new net groups or assign nets to an existing group.
4. The Net Group column displays all the net group assignments.
5. Define the I/O power value, using level calculations and other parameters.

Net Type for DDRx

NET Type	LPDDR2	DDR3	Note
Clock	CK_t, CK_c	CK, CK#	Generically differential
DQS	DQSx_t, DQSx_c	DQSx, DQSx#	Generically differential
Command	CKE, CS_c	CKE, CS#, RAS#, CAS#, WE#, ODTx	
Address	CA0~CA9	BA0~BA2, A0~A12	
DQ	DQx, DMx	DQx, DMx	

DDR Related Settings



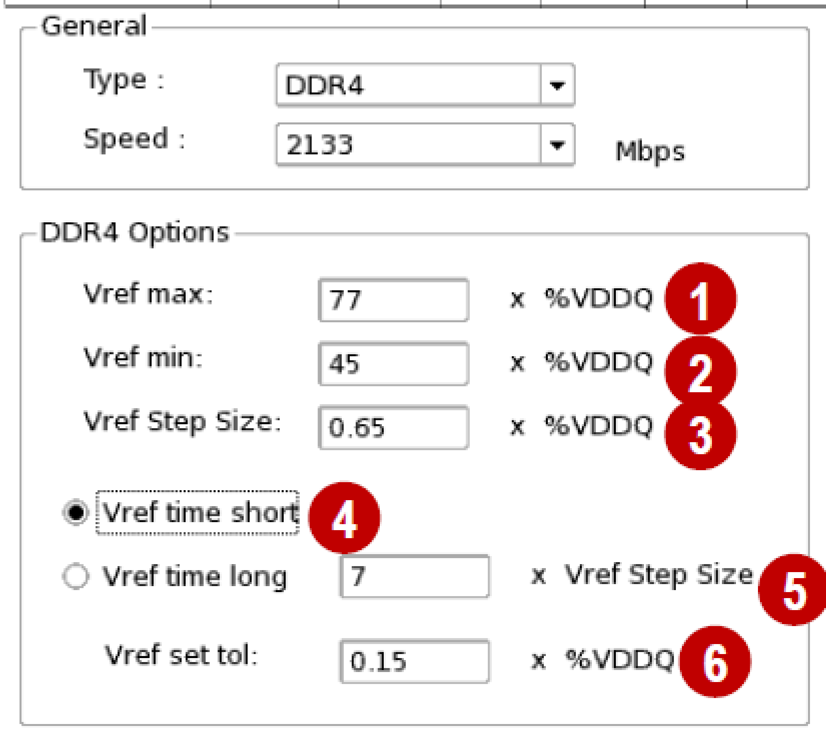
1. Speed Bins.
2. Select the CA net type AC logic level and DQ net type logic level.
3. Dereating Table button causes a window to show the default dereating value, and user can modify the default value when switch to customized dereating table.

DDR Special Options

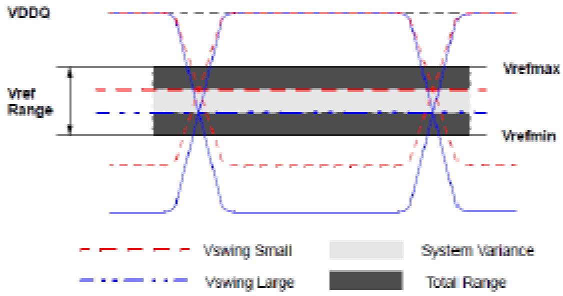
DDR4 has special options to control the Vref training.

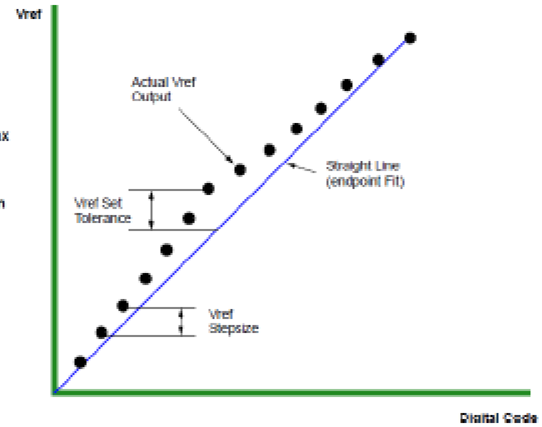
Table 36 — DQ Internal Vref Specifications

Prameter	Symbol	Min	Typ	Max	Unit	NOTE
Vref Max operating point Range1	$V_{ref\_max\_R1}$	92%	-	-	VDDQ	1, 11
Vref Min operating point Range1	$V_{ref\_min\_R1}$	-	-	60%	VDDQ	1, 11
Vref Max operating point Range2	$V_{ref\_max\_R2}$	77%	-	-	VDDQ	1, 11
Vref Min operating point Range2	$V_{ref\_min\_R2}$	-	-	45%	VDDQ	1, 11
Vref Stepsize	$V_{ref\ step}$	0.50%	0.65%	0.80%	VDDQ	2
Vref Set Tolerance	$V_{ref\_set\_tol}$	-1.625%	0.00%	1.625%	VDDQ	3,4,6
		-0.15%	0.00%	0.15%	VDDQ	3,5,7
Vref Step Time	$V_{ref\_time\text{-}Short}$	-	-	60	ns	8,12
	$V_{ref\_time\text{-}Long}$	-	-	150	ns	9,12
Vref Valid tolerance	$V_{ref\_val\_tol}$	-0.15%	0.00%	0.15%	VDDQ	10

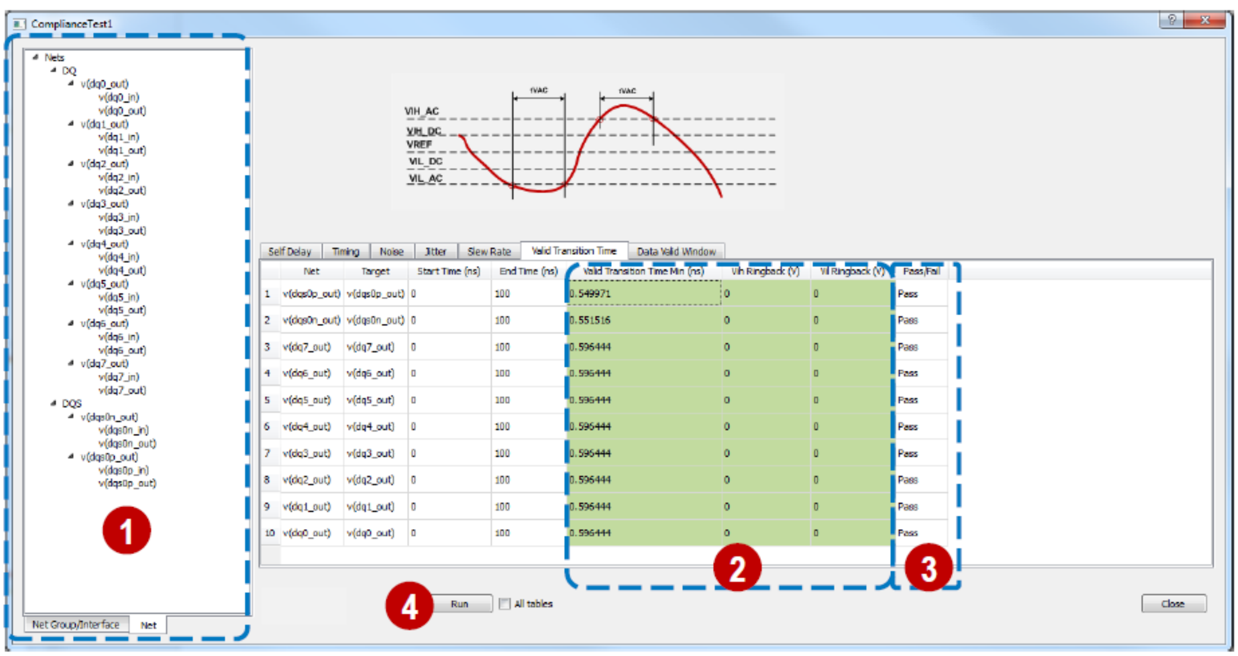


1. Define the fly-by Vref scanning upper limited.
2. Define the fly-by Vref scanning lower limited.
3. Define the fly-by Vref scanning step size.
4. Select Vref time short scans 1 step size a time.
5. Select Vref time long scans >=2 (by input) step size a time.
6. Set the tolerance of Vref scanning.





Compliance Test Window (DDRx)

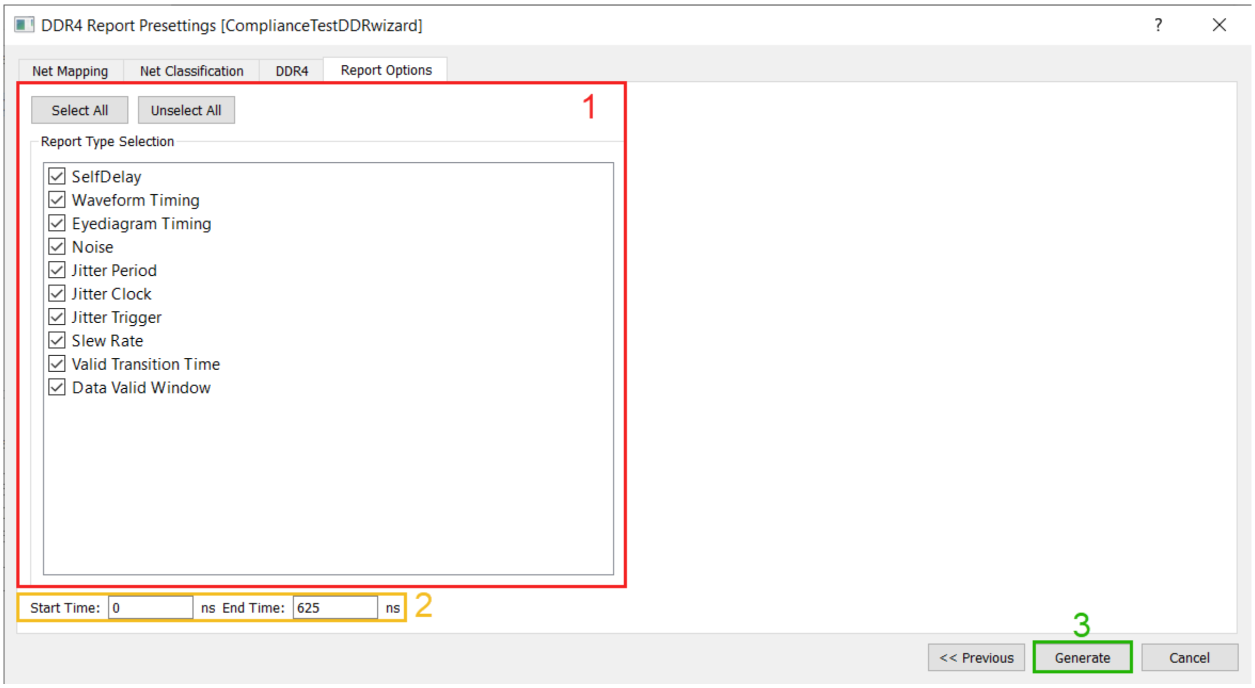


1. Net Group, Interface and Net information tree window, user can manual select target or view waveform.
2. Report parameter area.
3. Pass/Fail judgement according to JEDEC SPEC.
4. “Run” parameter calculation and pass/fail judgement.

VCM Report Tools

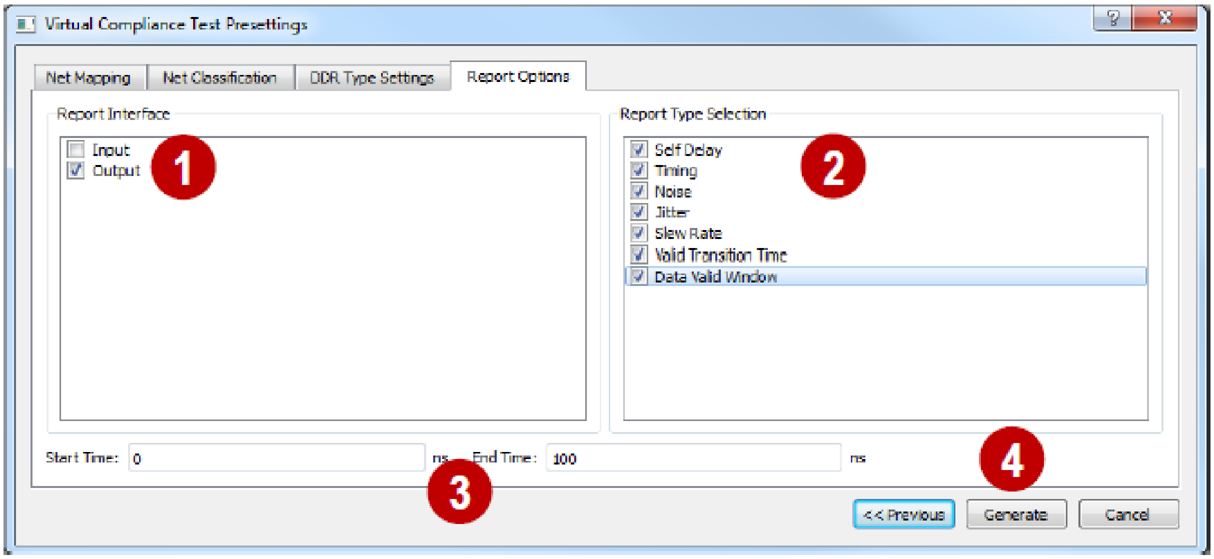
A number of reports are available in the Virtual Compliance Toolkit. The following topics describe the reports and report options that are available.

Report Options (DDR4 and GDDR5)



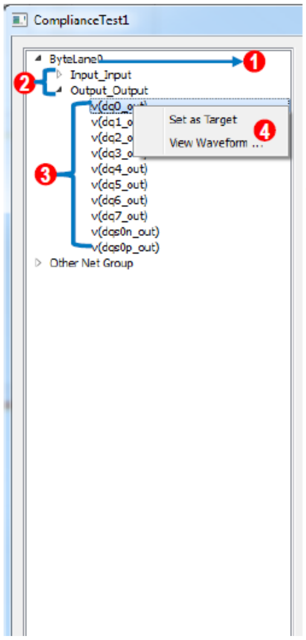
1. Report Options; the default is "Select all".
2. Input Start Time and End Time; the default is 0 ns to simulation end time.
3. After making your selections, click “Generate” to start.

Report Options DDRx

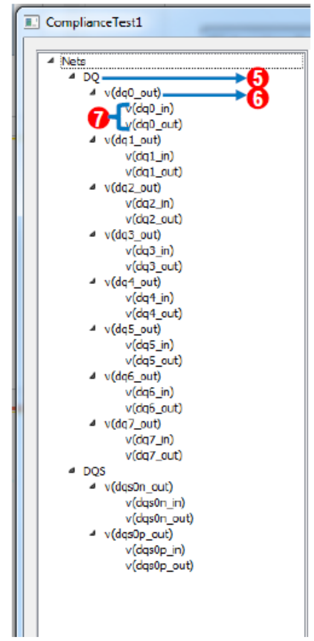


1. Choose a interface to report, (Output Interface is checked by default).
2. Report parameter types selection (all is checked by default).
3. Define the start and end time to report (simulation start/end time is chosen by default).
4. Click **Generate** to open a Compliance Test window.

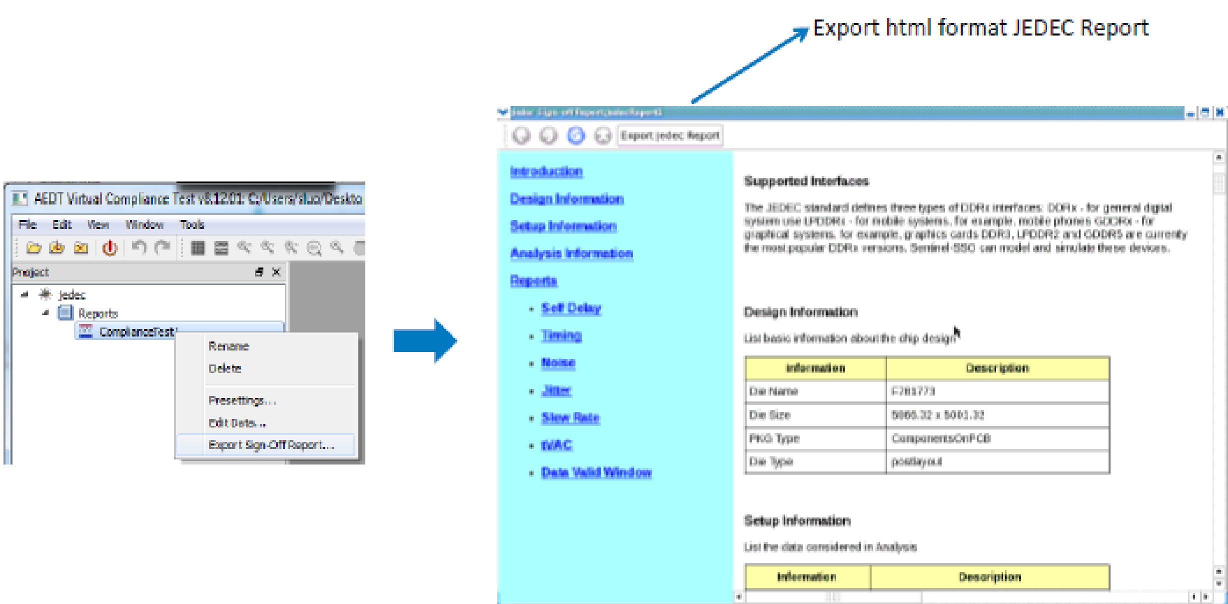
VCM Report Window



1. Net Group
2. Interface
3. Nets
4. “Set as Target” to view the parameter value; “View Waveform” to view the net waveform.
5. Net type
6. Net names
7. Nodes



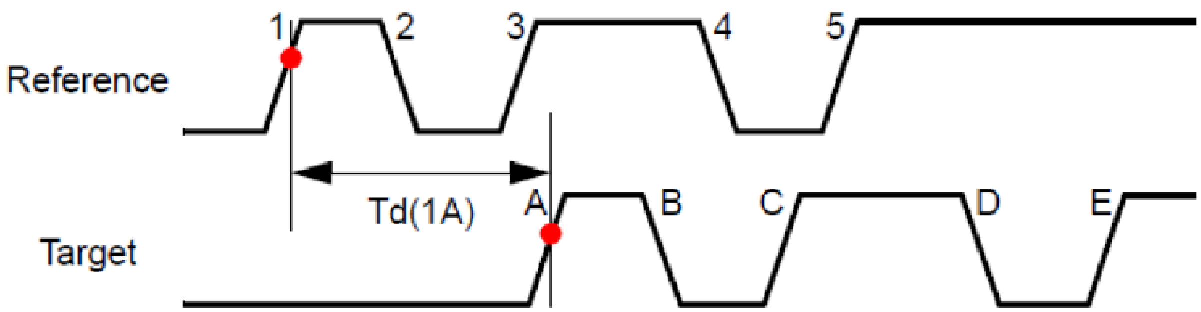
View Report



Virtual Compliance Module Examples

The following topics describe how to set up and run the Compliance Module with a number of signal and timing examples.

Self Delay



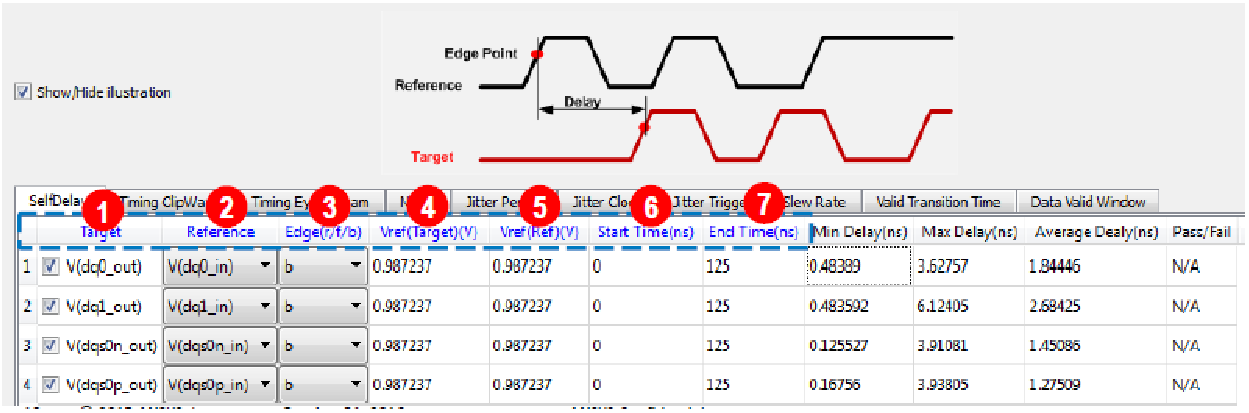
1. Reference and Target are both on the same net and are controlled by the net mapping setting.
2. Use of logic level to detect edges for both Target and Reference. For example, 5 edges of A,B,C,D, E for target and 5 edges of 1,2,3,4, 5 for reference.
3. Select the edge point. For example, the red point.
4. To calculate the delay for each edge. For example, A-1 delay1,B-2 delay2,C-3 delay3,D-4 delay4, and E-5 delay5.
5. Report the min delay and max delay. For example, to get the min delay and max delay among the values: A-1 delay1,B-2 delay2,C-3 delay3,D-4 delay4, and E-5 delay5.

Self Delay Keywords

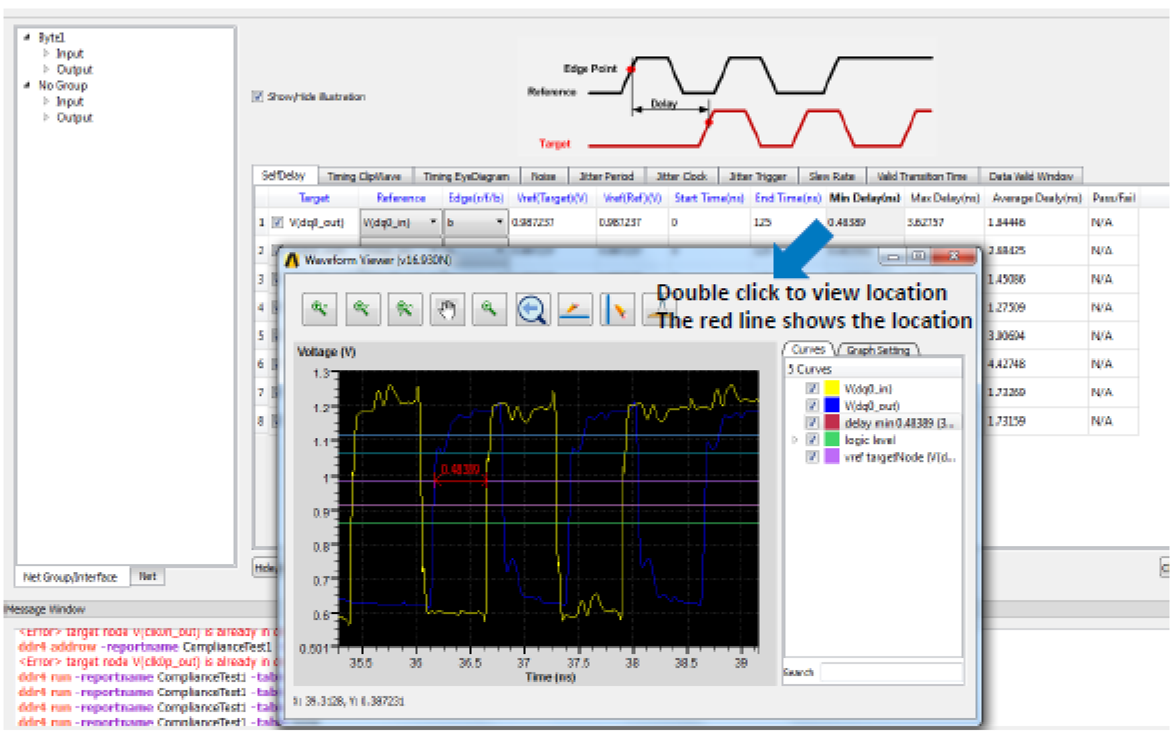
1. Target: Report target.
2. Reference: Default is the input node.
3. Edge(r/f/b): [r|f|b]. Default is both edge and user can change to rising edge or falling edge only. “r” represents rising and “f” represents falling and “b” represents both.
4. Ref_Volt(Target): Default use Vref of Target node. The user can set it manually.
5. Ref_Volt(Reference): Default is Vref of reference node. The user can set it manually.

6. Start_time: Default is 0. The user can enter any value between 0 and the amount of simulation time. Note that a negative value is not allowed.

7. End_time: Default is simulation time and exceeding it is not allowed.



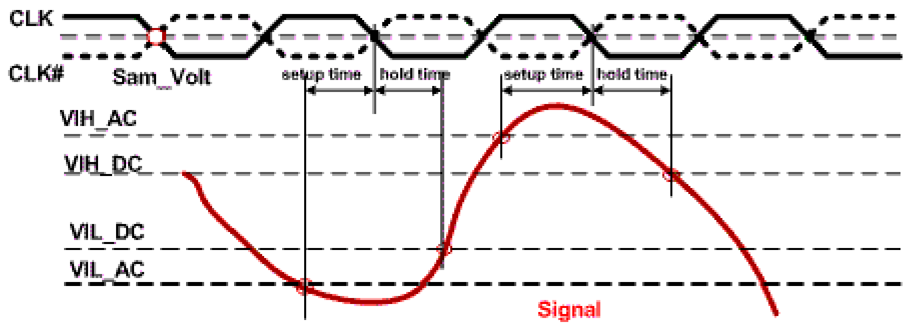
Self Delay Example



Timing Waveform

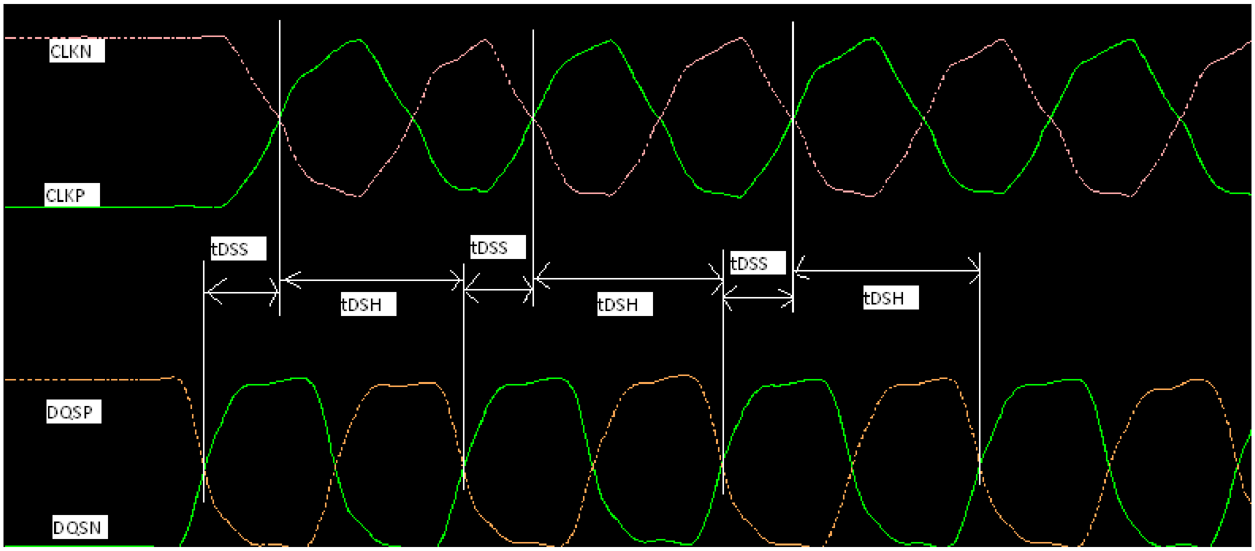
Timing: to calculate setup time & hold time

1. Detect edges on Target net.
2. Detect edges on reference net.
3. Detect cross-point on each reference edge.
4. Calculate setup time: The cross-point to the left vih_ac or vil_ac on edge of target is setup time.
5. Calculate hold time: The cross-point to the right vlh_dc or vil_dc on edge of target is hold time.
6. Report out the min & max value for both setup time & hold time.



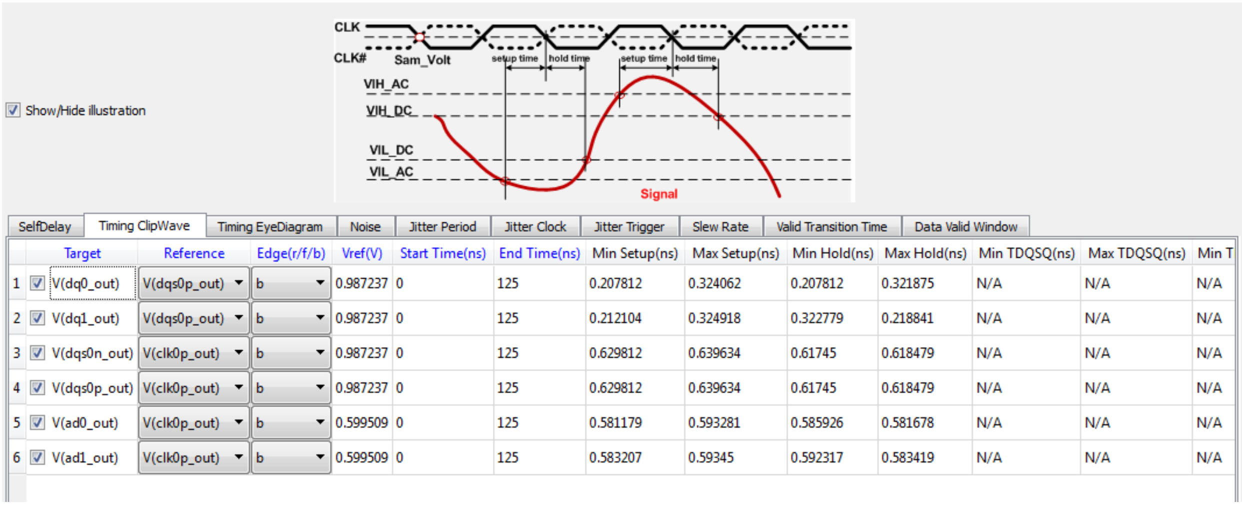
DQS setup time & hold time

1. When target is differential, calculates DQS timing automatically.
2. Detect rising edges on reference.
3. Select cross-point of reference as sample point.
4. Detect falling edges on target.
5. Calculate setup time for each sample point: the sample point to the left cross-point on falling edge is setup time.
6. Calculate hold time for each sample point: the sample point to the right cross-point on falling edge is hold time.
7. Report out min & max values for both setup time & hold time.



Timing Waveform Example

1. Target: Report target.
2. Reference: Strobes of the target, for DQ it is the DQS strobe in same net group, for Command or Address, it is Clock net.
3. Edge(r/f/b): [r|f|b]. Default is b and user can change it by list. r represents rising & f represents falling & b represents both. If r is selected, only rising edges are used to calculate delay for both target & reference. The same goes to f. If it’s DQS timing, it’s invalid.
4. Vref: reference voltage for target.
5. Start_time & End_time : Same as Self Delay.



	Target	Reference	Edge(r/f/b)	Vref(V)	Start Time(ns)	End Time(ns)	Min Setup(ns)	Max Setup(ns)	Min Hold(ns)	Max Hold(ns)	Min TDQSQ(ns)	Max TDQSQ(ns)	Min T
1	V(dq0_out)	V(dqs0p_out)	b	0.987237	0	125	0.207812	0.324062	0.207812	0.321875	N/A	N/A	N/A
2	V(dq1_out)	V(dqs0p_out)	b	0.987237	0	125	0.212104	0.324918	0.322779	0.218841	N/A	N/A	N/A
3	V(dqs0n_out)	V(clk0p_out)	b	0.987237	0	125	0.629812	0.639634	0.61745	0.618479	N/A	N/A	N/A
4	V(dqs0p_out)	V(clk0p_out)	b	0.987237	0	125	0.629812	0.639634	0.61745	0.618479	N/A	N/A	N/A
5	V(ad0_out)	V(clk0p_out)	b	0.599509	0	125	0.581179	0.593281	0.585926	0.581678	N/A	N/A	N/A
6	V(ad1_out)	V(clk0p_out)	b	0.599509	0	125	0.583207	0.59345	0.592317	0.583419	N/A	N/A	N/A

Eye Diagram Timing

1. Using eye mask to judge the violation.
2. User input: Input Data: VdiVW_Total and TdiVW_Total.
3. Tool auto-creates Target vs. Strobe (i.e., DQ vs. DQS) eye diagram.
4. Base on the Strobe cross-point, Vref, VdiVW and TdiVW to create the Rx mask and report the violations.
5. Report TdiPW, Tr, Tf and SRIN_diVW for each DQ.
6. Report Eye-width and Eye-Height @ BER=1e-16.

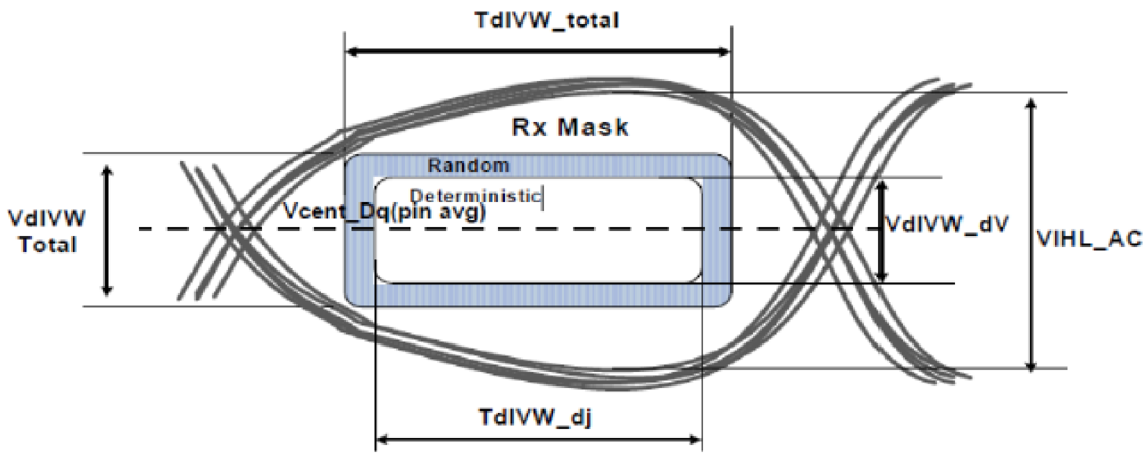
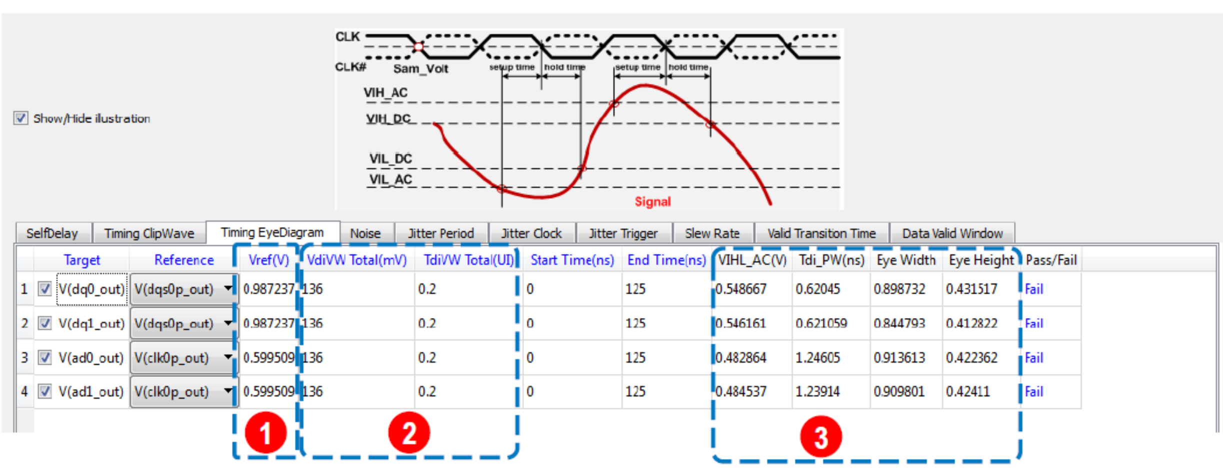


Figure 206 — DQ Receiver(Rx) compliance mask

DDR4 DQ Eye Diagram Timing Example

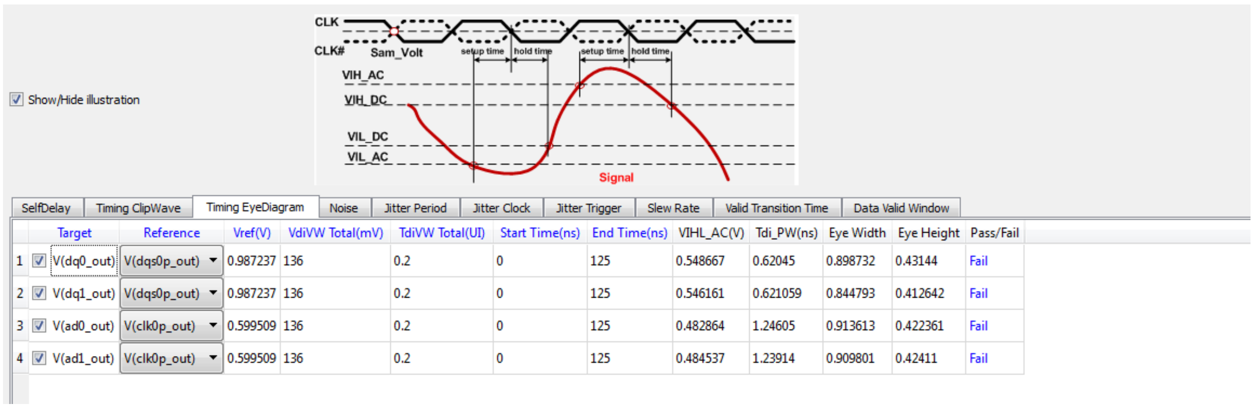
1. Classify and group nets in presetting, fly-by Vref automatically calculates by net group.
2. Input VdiVW_Total and TdiVW_Total.
3. Report the DDR4 DQ Eye-diagram related parameters.



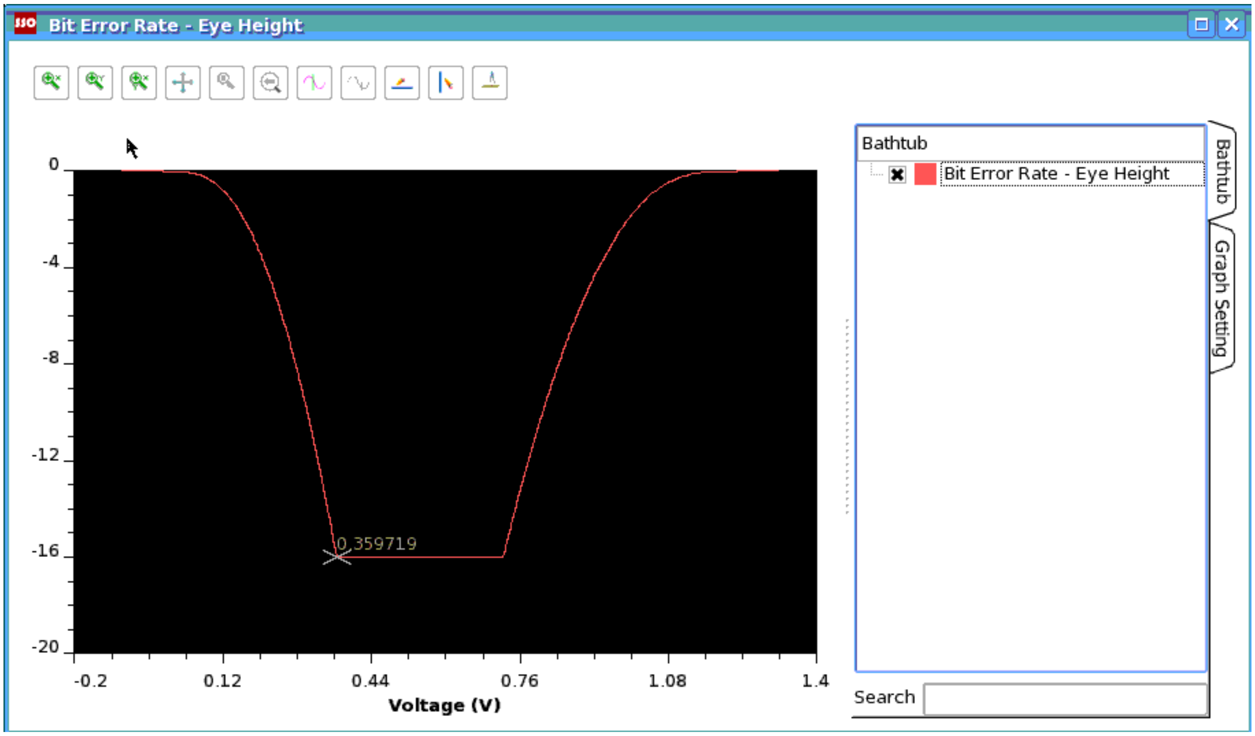
	Target	Reference	Vref(V)	VdiVW Total(mV)	TdiVW Total(UI)	Start Time(ns)	End Time(ns)	VIHL_AC(V)	Tdi_PW(ns)	Eye Width	Eye Height	Pass/Fail
1	V(dq0_out)	V(dqs0p_out)	0.987237	136	0.2	0	125	0.548667	0.62045	0.898732	0.431517	Fail
2	V(dq1_out)	V(dqs0p_out)	0.987237	136	0.2	0	125	0.546161	0.621059	0.844793	0.412822	Fail
3	V(ad0_out)	V(clk0p_out)	0.599509	136	0.2	0	125	0.482864	1.24605	0.913613	0.422362	Fail
4	V(ad1_out)	V(clk0p_out)	0.599509	136	0.2	0	125	0.484537	1.23914	0.909801	0.42411	Fail

Eye Width and Eye Height

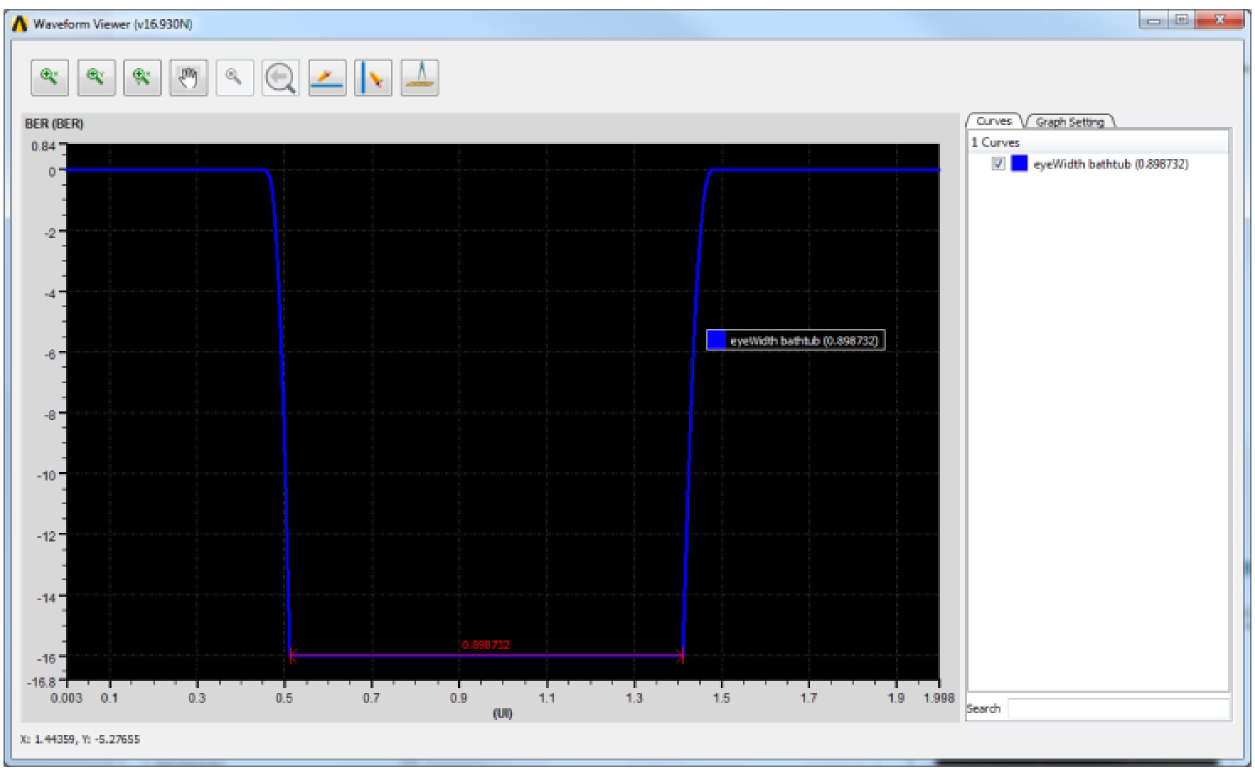
1. Select a node in Timing table and click Run.
2. Report Eye-Height/Eye-Width @BER=1e-16.
3. Double-click the value to open the Bathtub window.



		Target	Reference	Vref(V)	VdiVW Total(mV)	TdiVW Total(UI)	Start Time(ns)	End Time(ns)	VIHL_AC(V)	Tdi_PW(ns)	Eye Width	Eye Height	Pass/Fail
1	☑	V(dq0_out)	V(dqs0p_out)	0.987237	136	0.2	0	125	0.548667	0.62045	0.898732	0.43144	Fail
2	☑	V(dq1_out)	V(dqs0p_out)	0.987237	136	0.2	0	125	0.546161	0.621059	0.844793	0.412642	Fail
3	☑	V(ad0_out)	V(clk0p_out)	0.599509	136	0.2	0	125	0.482864	1.24605	0.913613	0.422361	Fail
4	☑	V(ad1_out)	V(clk0p_out)	0.599509	136	0.2	0	125	0.484537	1.23914	0.909801	0.42411	Fail



Bit Error Rate - Eye Height
Bathtub
Bit Error Rate - Eye Height
Graph Setting
0.359719
Voltage (V)
Search

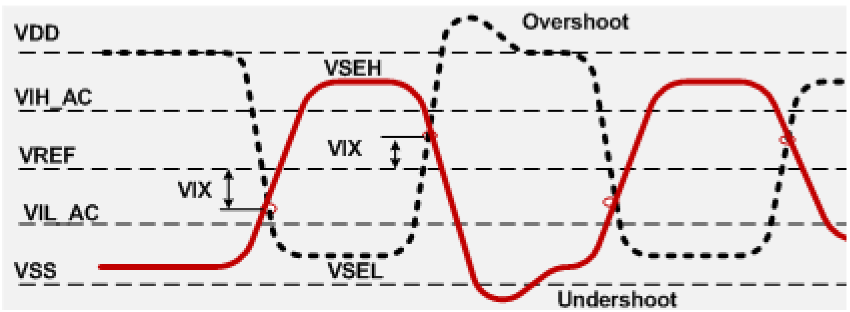


Waveform Viewer (v16.930N)
BER (BER)
eyeWidth bathtub (0.898732)
0.898732
(UI)
Curves
Graph Setting
1 Curves
Search
X: 1.44359, Y: -5.27655

Noise

Signal and Noise

1. Overshoot: Noise above IO domain power.
2. Undershoot: Nosie below 0.



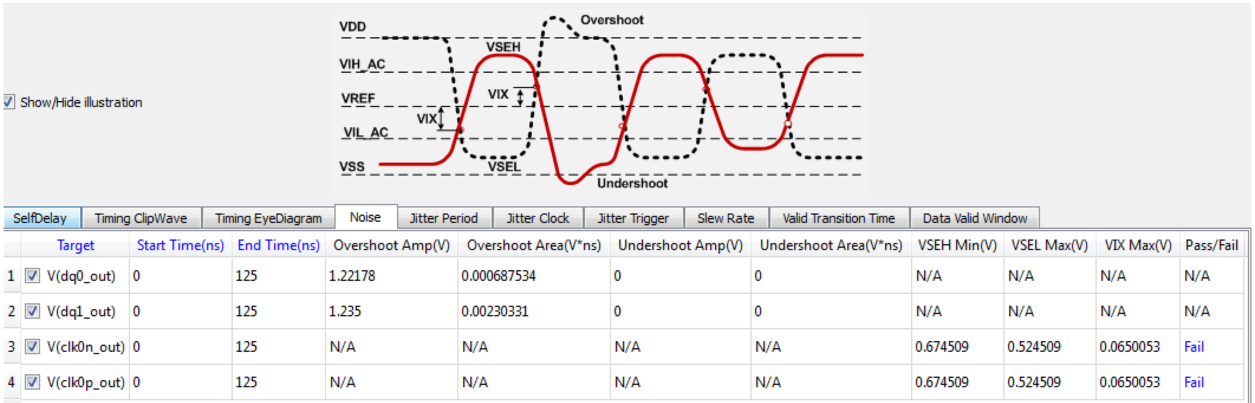
Differential Noise: VIX

1. Detect edges on target.
2. Get cross-point on edge of target.
3. Get the voltage of each cross-point.
4. To get VIX for each cross-point: |V_cross-point-VREF|= |V_cross-point-0.5*IO_Power|.
5. Report out the max value.

Noise Table Keywords

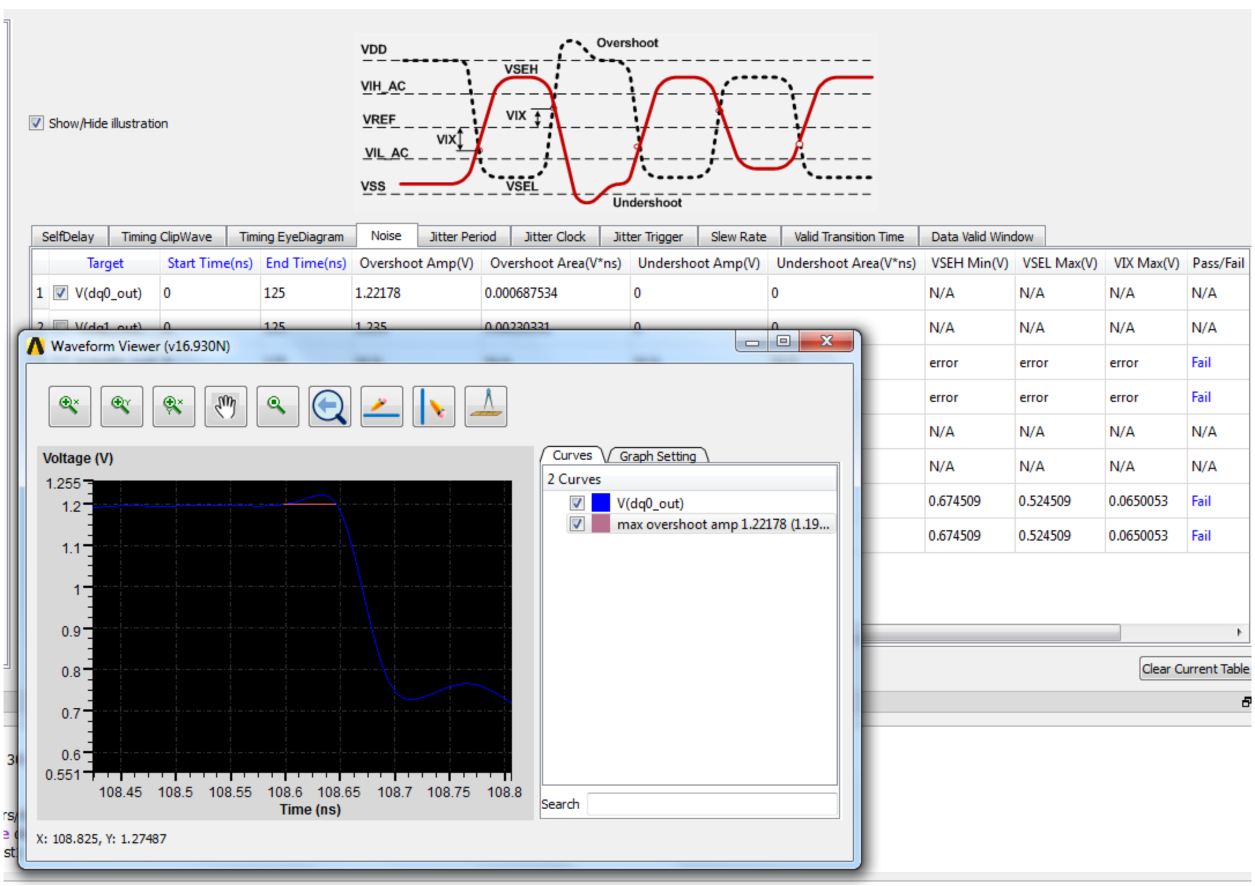
1. Target: Report target.
2. Type[Single | Differential]: single to report overshoot & undershoot. differential to report VIX&VSEL&VSEH. Differential is only valid for differential pairs.

3. Start_time & End_time : Same as selfdelay.



Noise Report Example

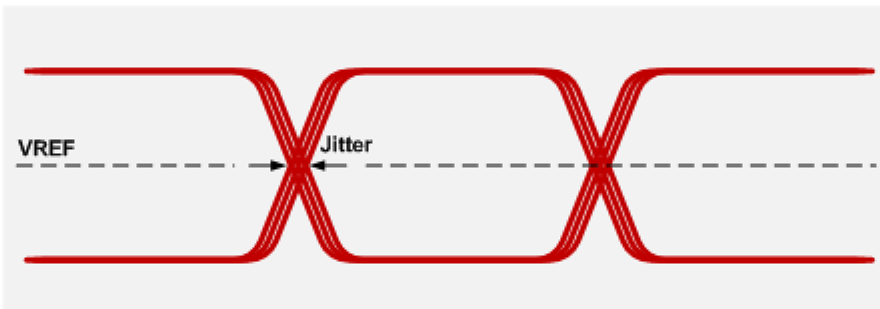
1. Select a node and click Run.
2. It reports overshoot value above IO power.
3. Double-click the value, the max overshoot shows-up.



Period Jitter

Signal jitter: to generate eye-diagram to get jitter

1. Cut-off target to many parts.
2. Overlap all the parts together to generate eye-diagram.
3. Get reference voltage VREF.
4. Report max jitter.

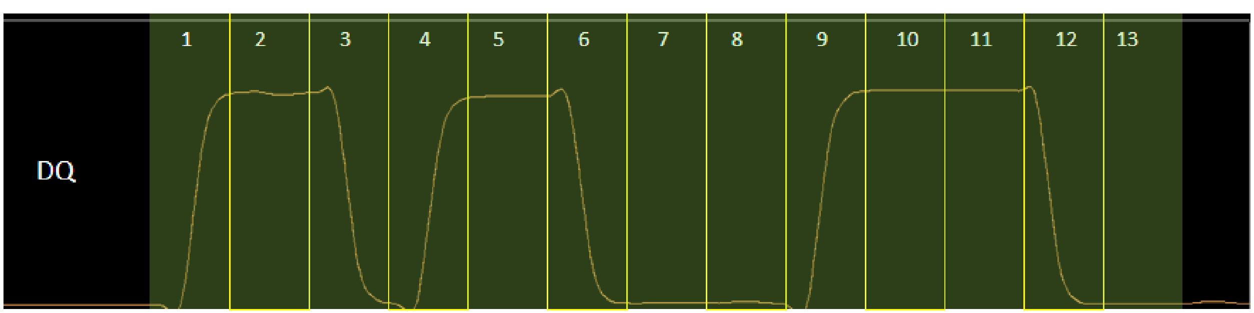


Differential Noise: VIX

1. Detect edges on target.
2. Get cross-point on edge of target.
3. Get the voltage of each cross-point.
4. To get VIX for each cross-point: |V_cross-point-VREF|= |V_cross-point-0.5*IO_Power|.
5. Report out the max value.

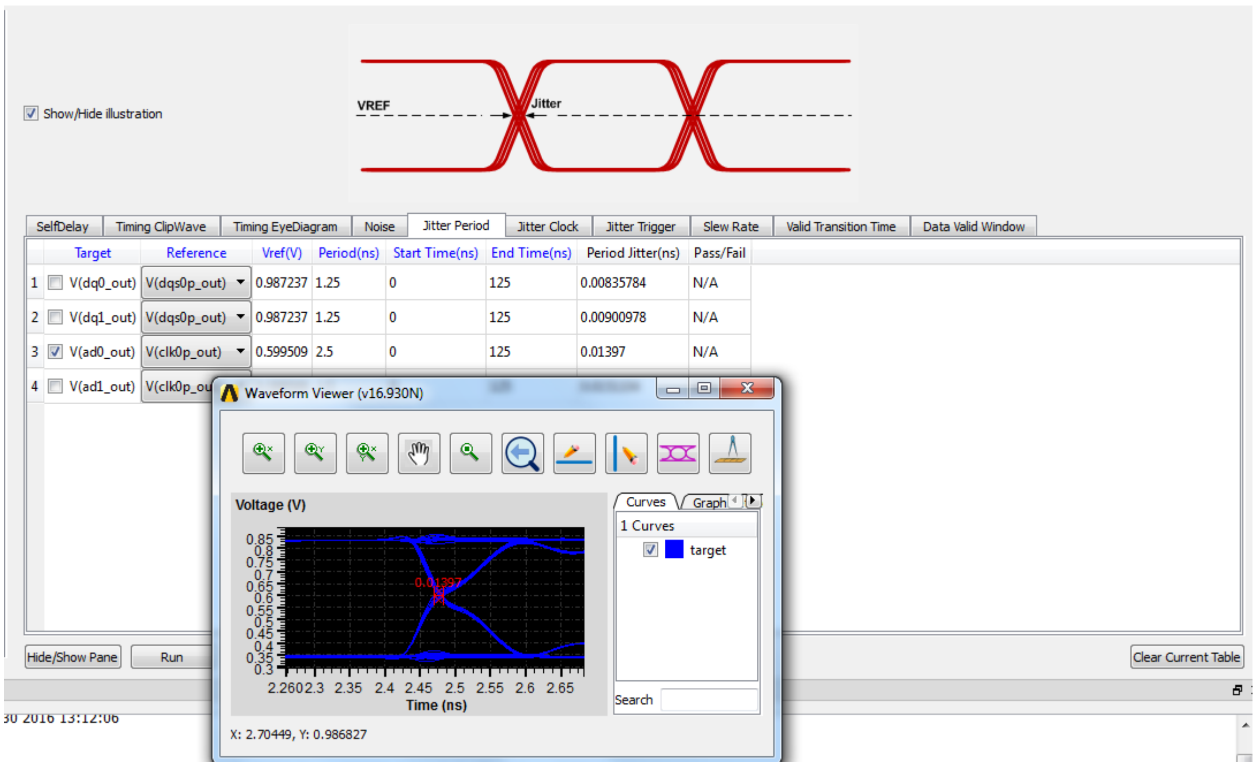
Cut-off Pattern

1. Automatically parse out period to cut off target: For example, to cut off DQ to 13 parts by period



Period Jitter Example: Eye-diagram generation

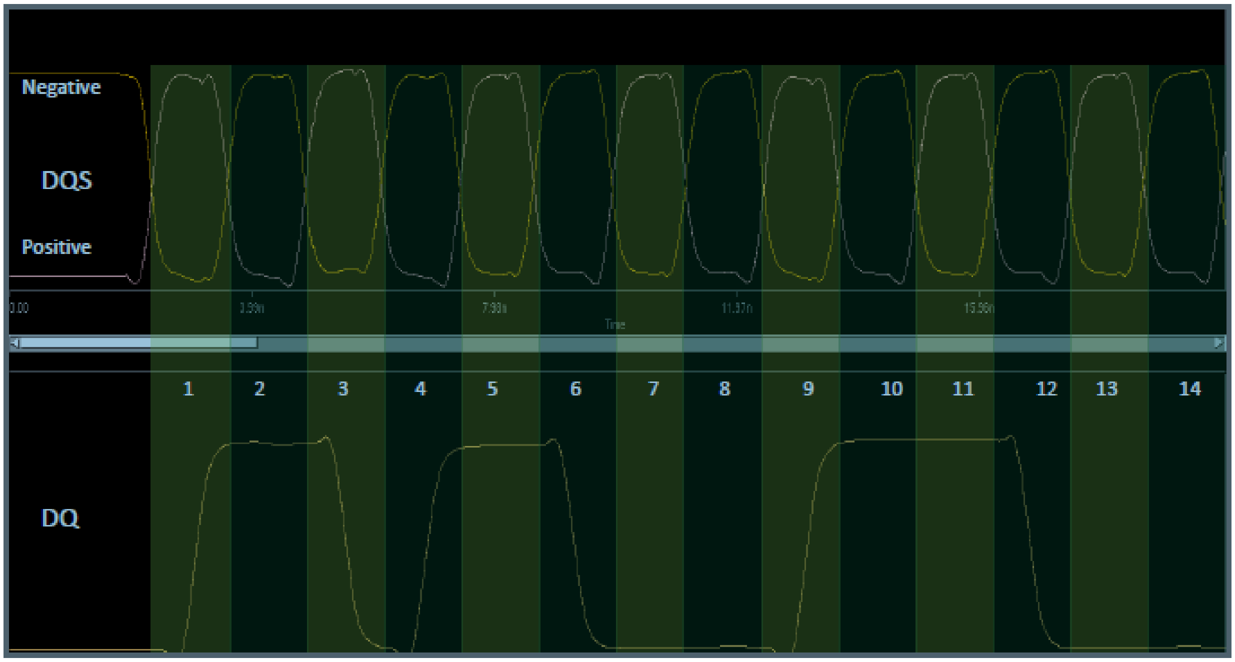
1. Overlap all the parts together to generate eye-diagram.
2. To report jitter.
3. 2UI eye jitter reports the max one of jitter1 & jitter2.



Trigger Jitter

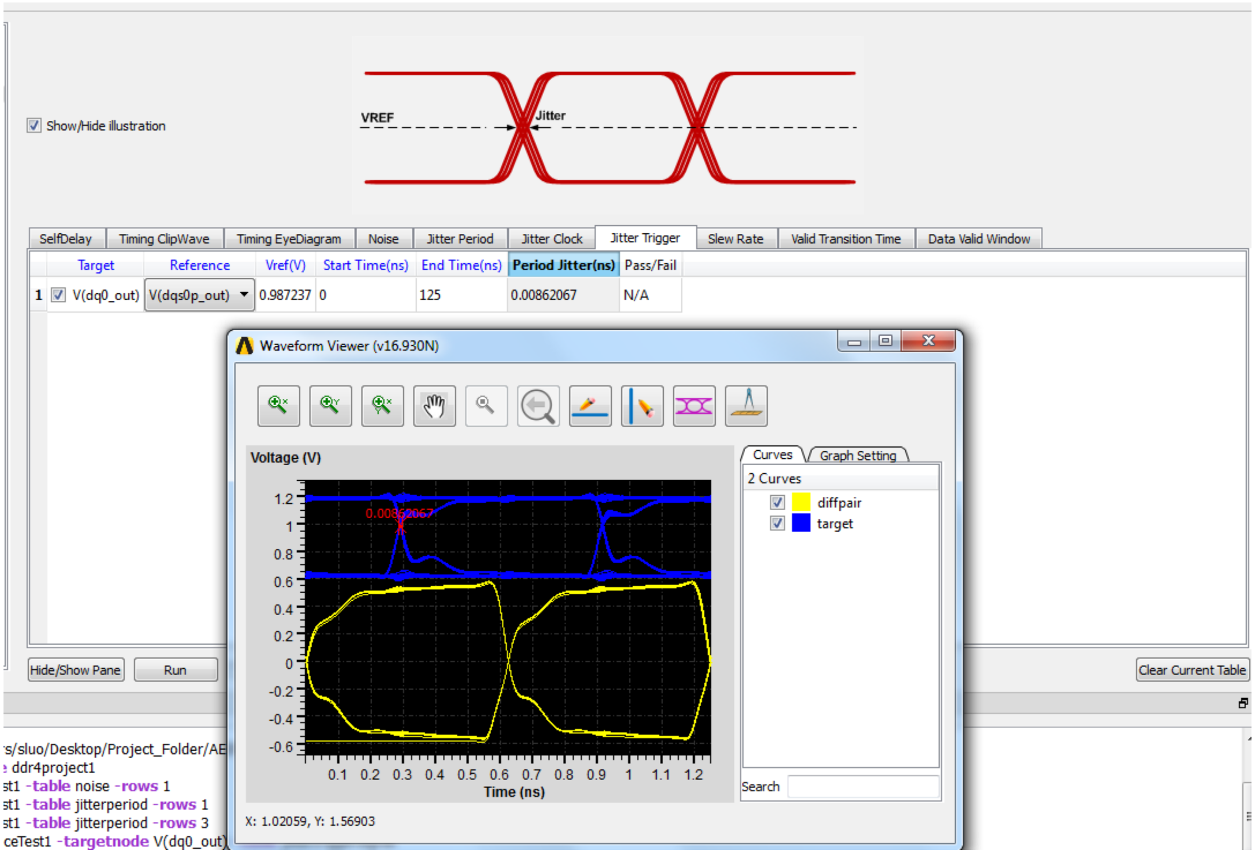
Trigger jitter is almost the same as period Jitter, only difference is the cut off type, trigger jitter is cut off the target by Strobes.

1. Select target's referred clock as trigger to cut off target: for example, using cross points of differential DQS to cut off DQ to 14 parts.



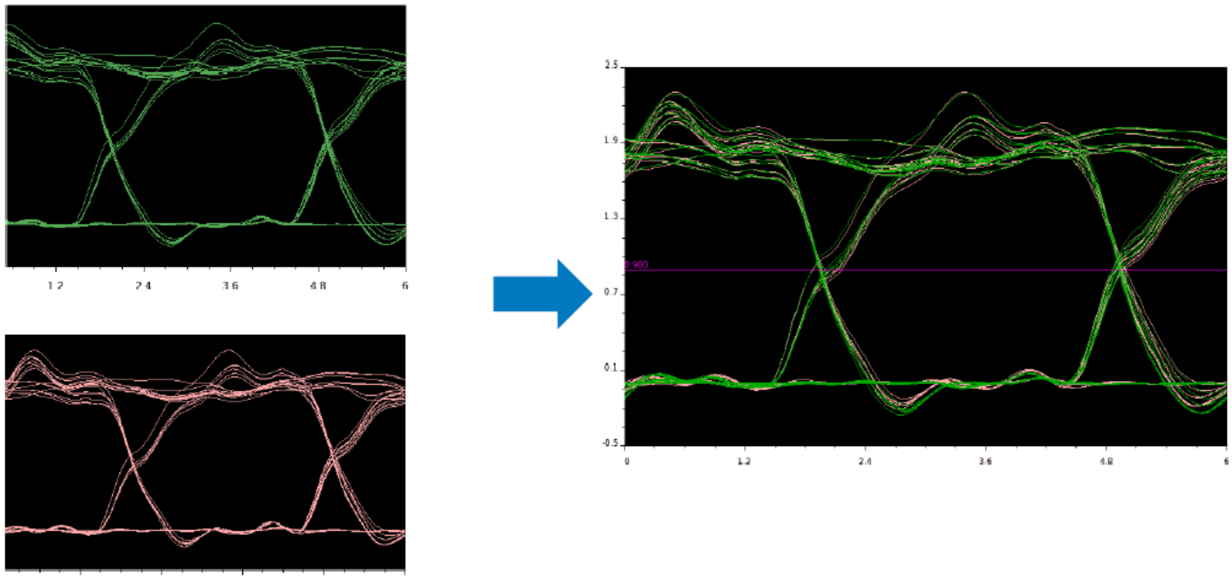
Trigger Jitter Example

1. Overlap all the parts together to generate eye-diagram.
2. To report jitter.
3. 2UI eye jitter reports the max one of jitter1 & jitter2.


Show/Hide illustration
VREF
Jitter
SelfDelay
Timing ClipWave
Timing EyeDiagram
Noise
Jitter Period
Jitter Clock
Jitter Trigger
Slew Rate
Valid Transition Time
Data Valid Window
Target
Reference
Vref(V)
Start Time(ns)
End Time(ns)
Period Jitter(ns)
Pass/Fail
1
V(dq0_out)
V(dqs0p_out)
0.987237
0
125
0.00862067
N/A
Waveform Viewer (v16.930N)
Voltage (V)
Curves
Graph Setting
2 Curves
diffpair
target
Time (ns)
Search
X: 1.02059, Y: 1.56903
Hide/Show Pane
Run
Clear Current Table

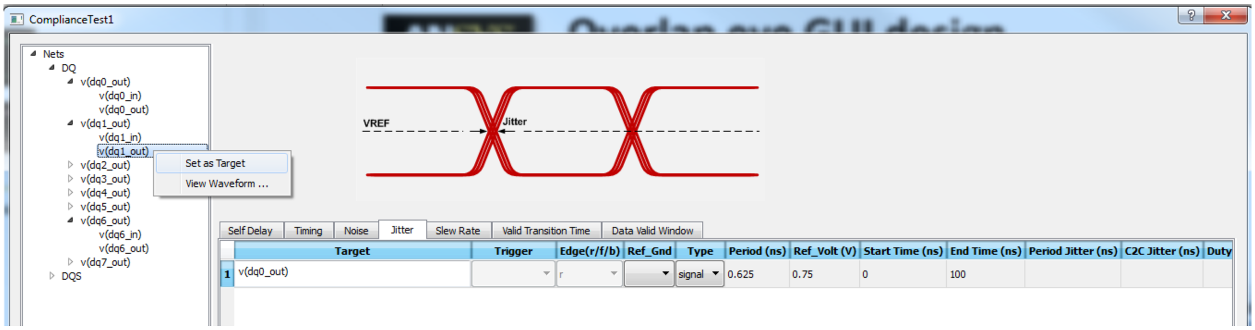
Overlap Eye Jitter

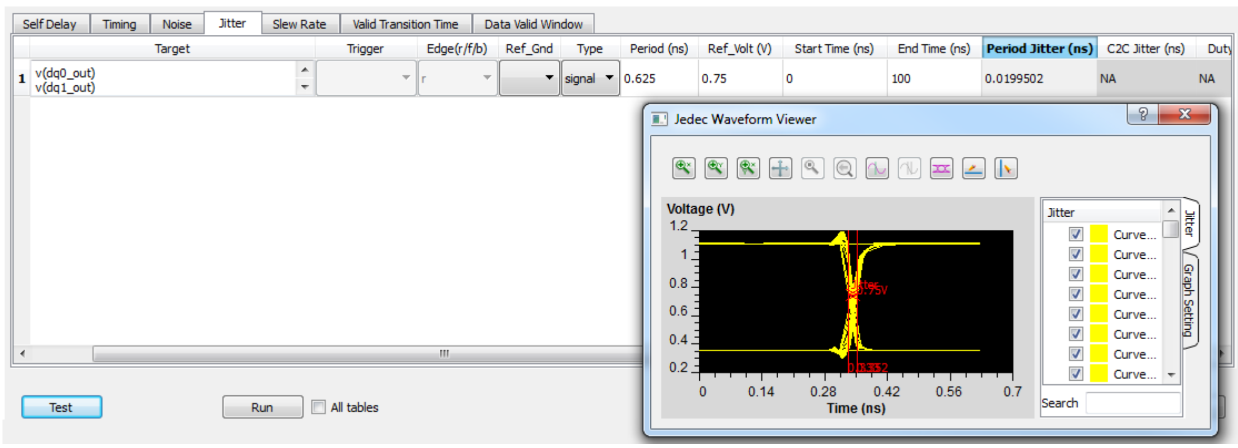
User can overlap multiple signal eyes , then measure the total jitter.



Overlap Eye Jitter Example

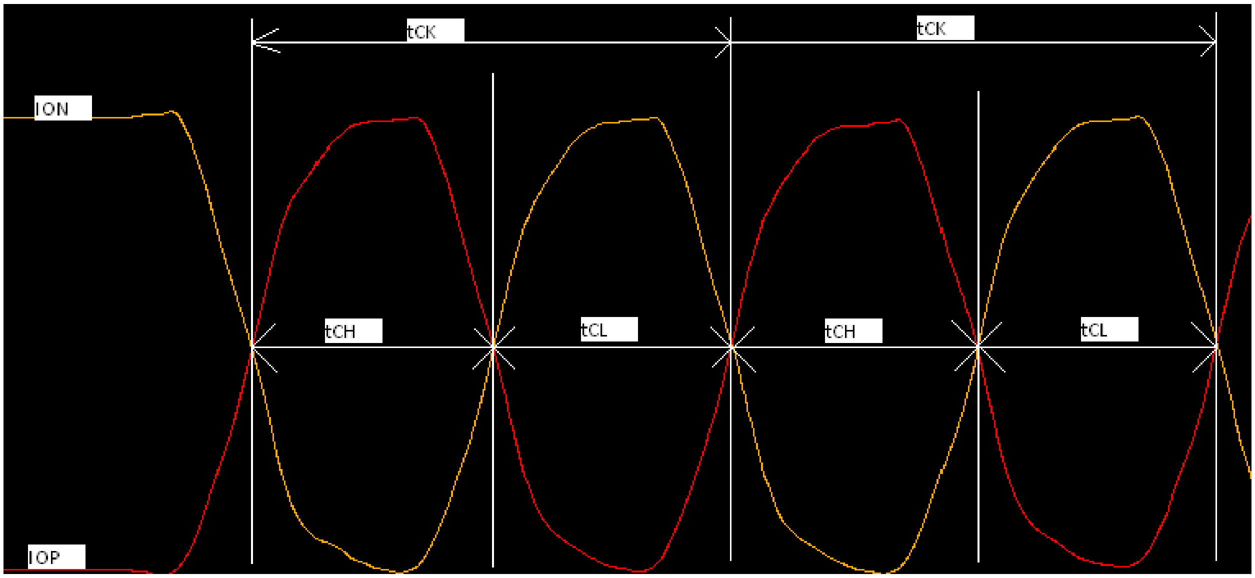
1. First set dq0_out as a target in table, and high-light dq0_out row.
2. Select another target dq1_out, dq0_out and dq1_out targets merges together.
3. Click Run to get the total Jitter for 2 targets.





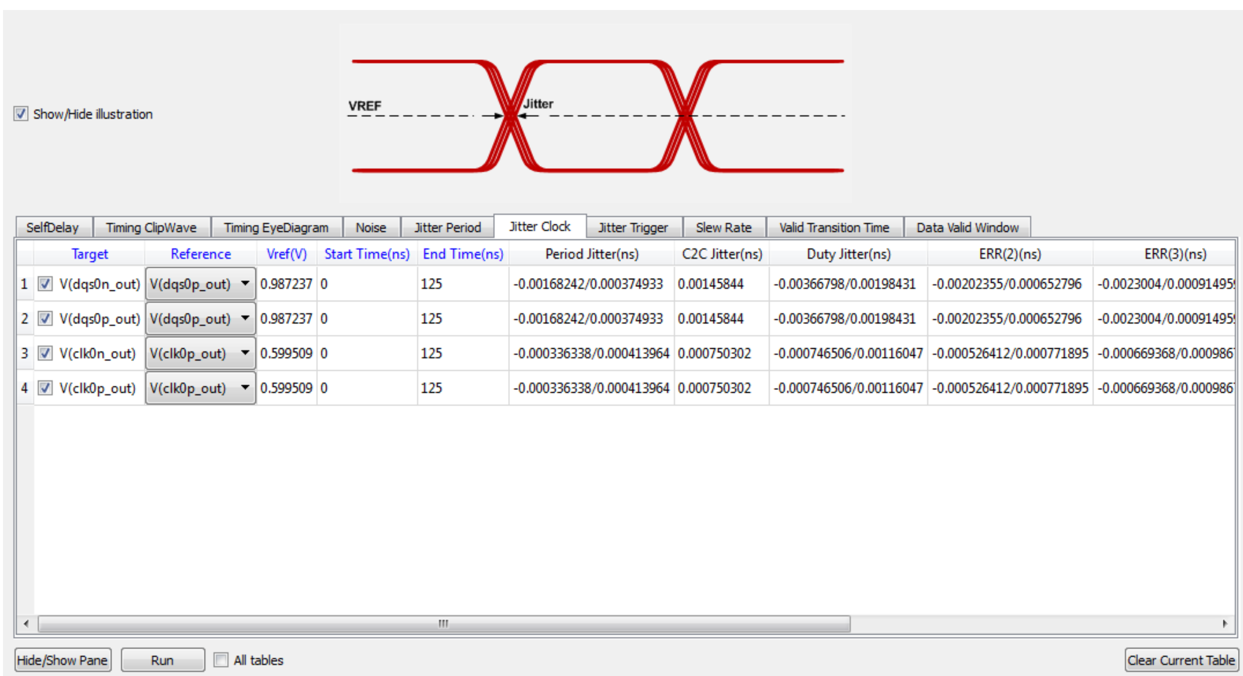
Clock Jitter

1. Period Jitter: tJIT(per) = min/max of (tCKi – tCK(avg)).
2. Duty jitter:
   a. tJIT(duty),min =MIN[(tCH(abs),min – tCH(avg),min),(tCL(abs),min – tCL(avg),min)] × tCK(avg)
   b. tJIT(duty),max =MAX[((tCH(abs),max – tCH(avg),max),(tCL(abs),max – tCL (avg),max)] × tCK(avg)
3. Cycle to cycle jitter: tJIT(cc) = max of |tCKi + 1 – tCKi|
4. ERR: tERR(nper) =(Σi+n-1 j=i tCKj) – (n*tCK(avg))



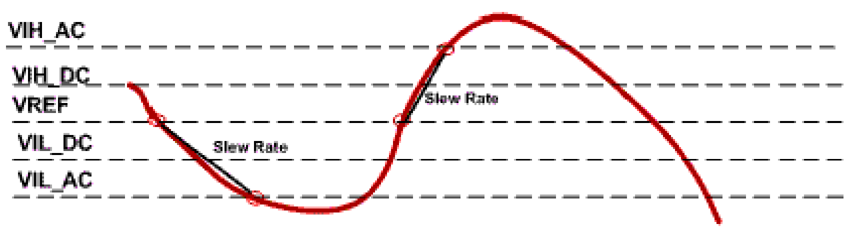
Clock Jitter Example

1. Select target from clock net or DQS net.
2. Run to generate report.



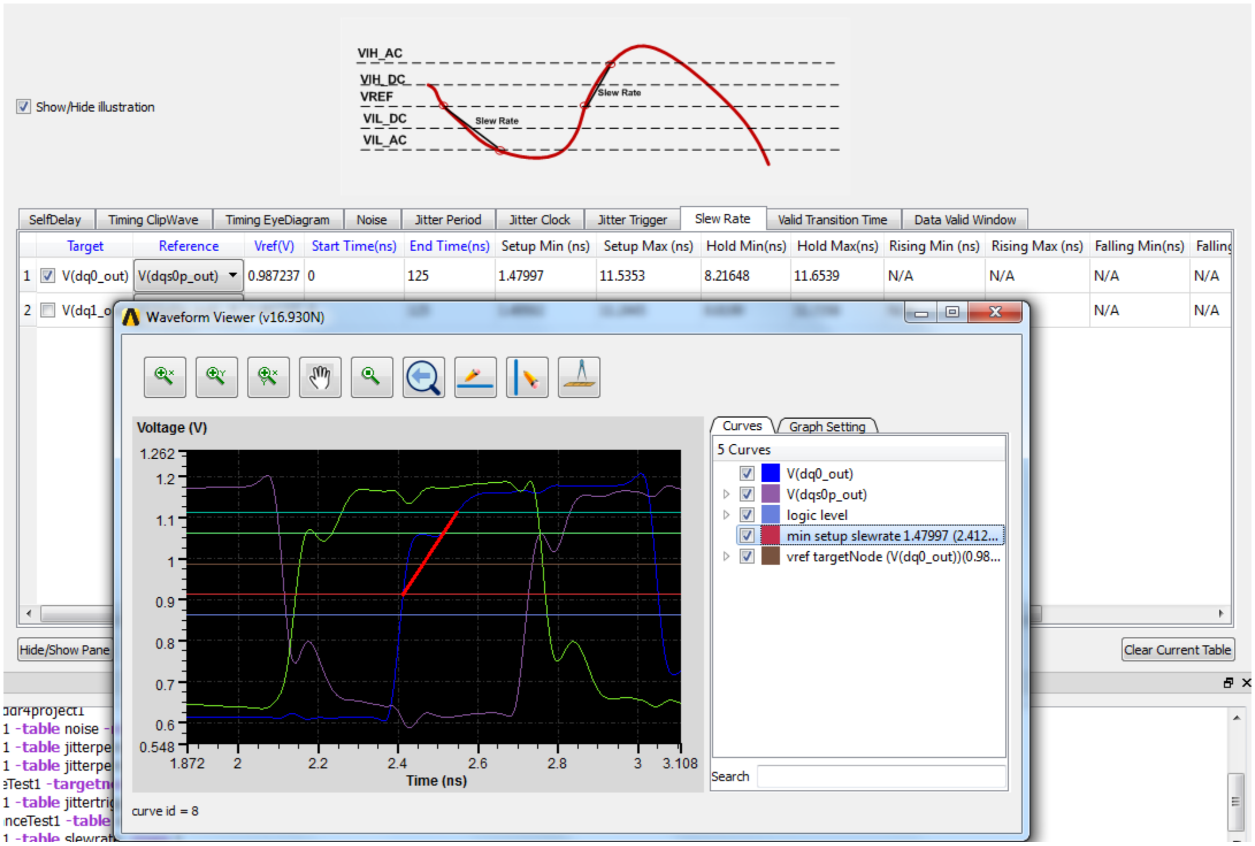
Slew Rate

1. Detect edges on target.
2. Get vil_ac & vih_ac cross-point on target edges.
3. Get setup slew rate: slew rate of area between ac point & Vref.
4. Get vil_dc & vih_dc cross-point on target edges.
5. Get hold slew rate: slew rate of area between dc point & Vref.
6. Report out min & max among setup & hold slew.



Slew Rate Example

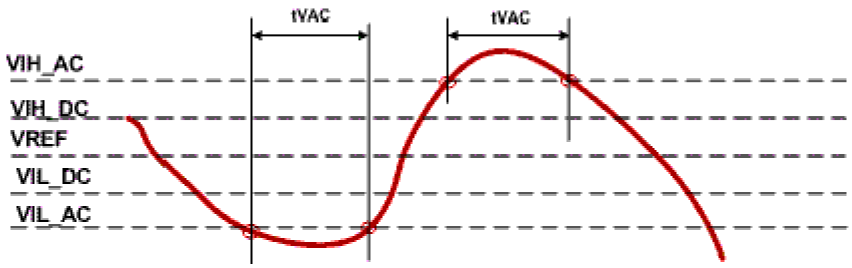
1. Select target.
2. Run to generate report



Validate Data Transition

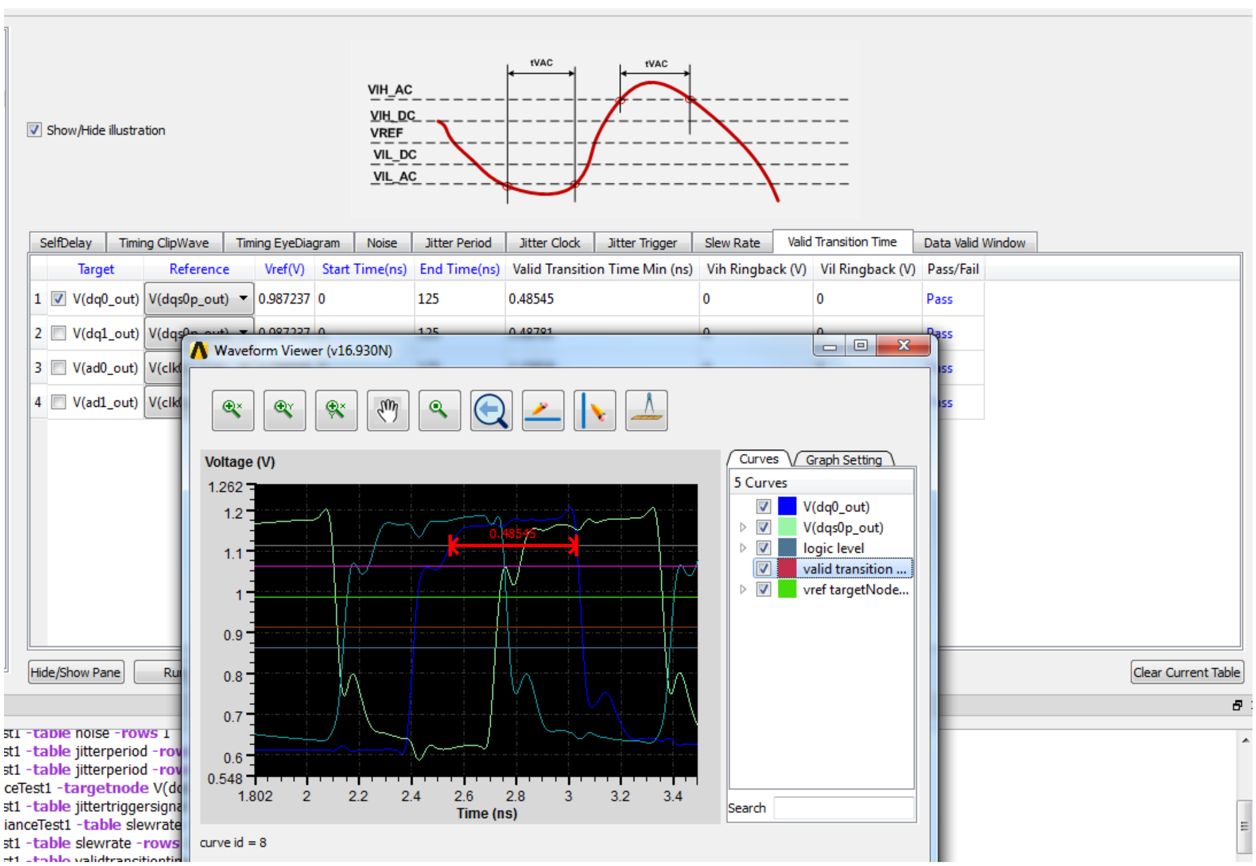
1. Detect edges on target.
2. Select the area under vil_ac or above vih_ac on between 2 continual edges.
3. Get the time length for each area.
4. Report out the min value.
5. Record waveform dips under the AC threshold while not crossing the DC threshold.

6. Report out the Vil Ringback and Vih Ringback.



Valid Data Transition Example

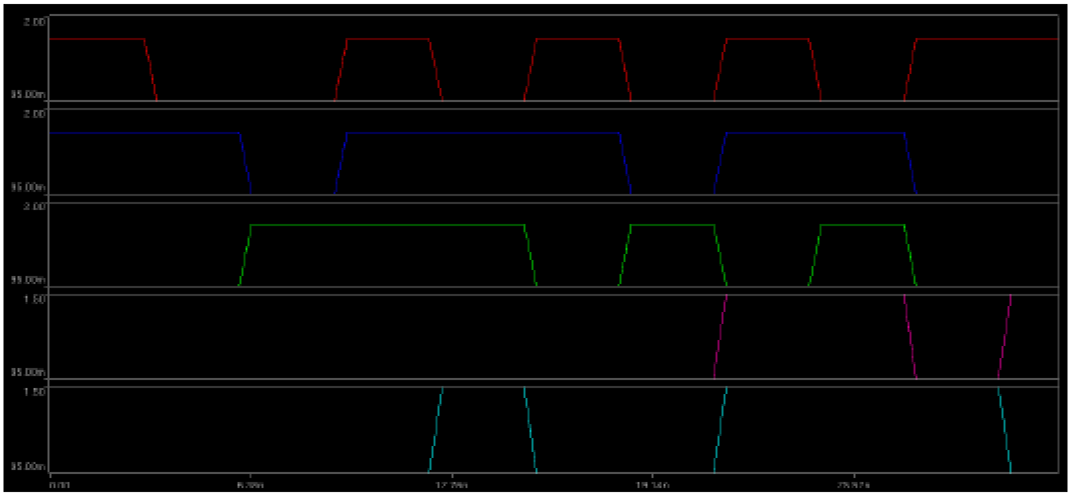
1. Select target.
2. Run to generate report.

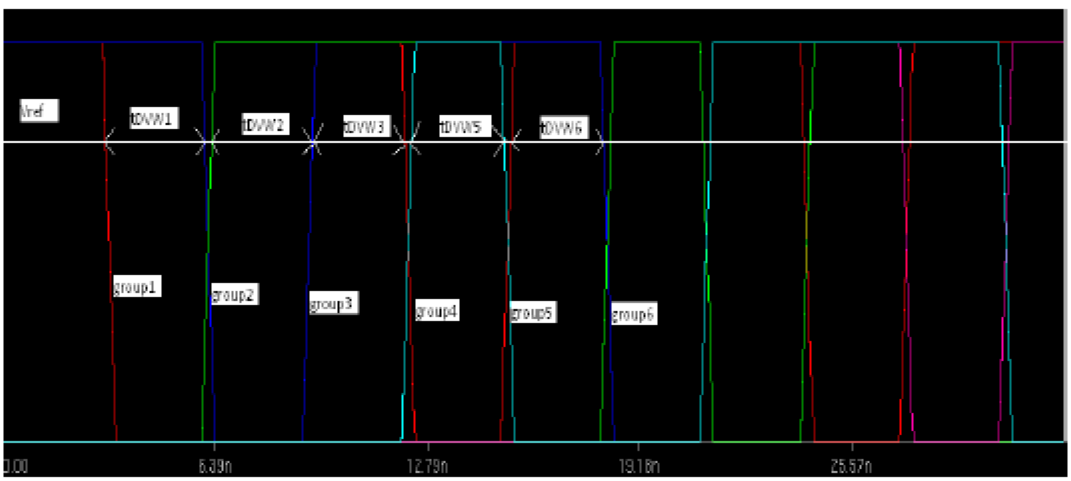


Validate Data Window

1. Select multiple targets.
2. Overlap them together.
3. Detect edges on all the targets.
4. Group all the edges based on the pulse width.
5. Get edge point on edge.
6. Calculate the window between 2 continual group.

7. Report out the min & max value.

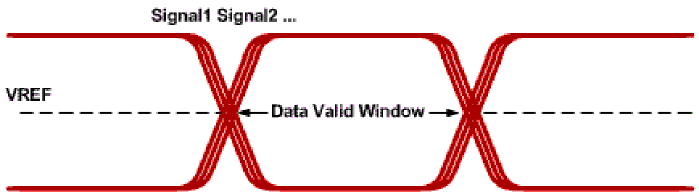


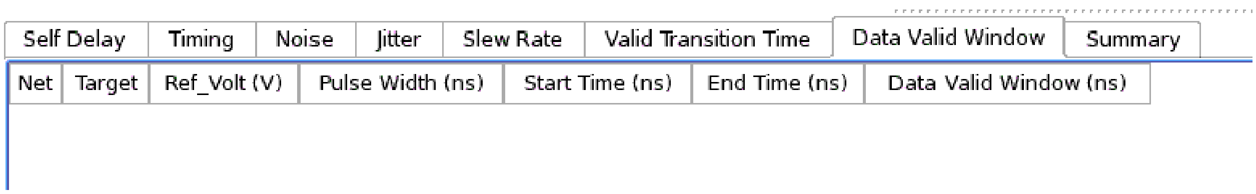


Valid Data Window Keywords

1. Target: report target.
2. Ref_Volt: reference voltage.
3. Pulse width: UI width.

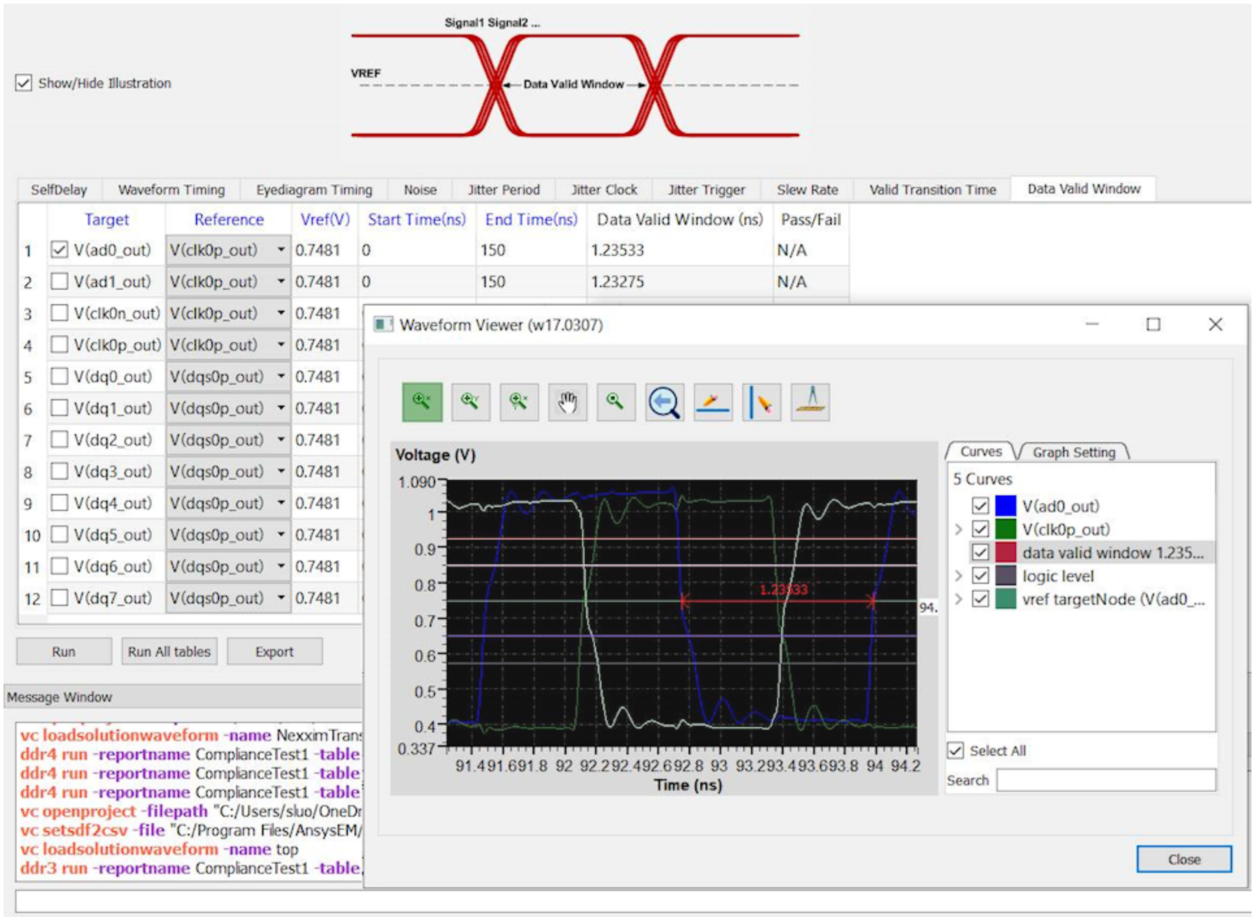
4. Start_time & End_time : Same as Self-delay.





Valid Data Window Example

1. Select multiple targets.
2. Run to generate report.



12 - Circuit Frequency Domain Analyses

Nexxim provides a variety of analysis methods for simulating circuit designs in the frequency domain.

Harmonic Balance Analysis

Harmonic balance analyzes the periodic or quasi-periodic steady-state response of a circuit to a periodic input by solving the circuit equations in the frequency domain. Time domain equations are represented by their Fourier series equivalents. In a one-tone or periodic analysis, the input is a sine wave at a specified frequency, and the response is measured over a specified range of harmonics of that frequency. In a multi-tone or quasi-periodic analysis, the input is a combination of sine waves at different frequencies, and the response is a spectrum containing the DC response, the harmonics of the input frequencies, and the sums and differences of the harmonic frequencies.

See Nexxim Harmonic Balance Analysis.

HB Load Pull Analysis

Load-pull analysis is used to determine the optimum load on a circuit. Nexxim load-pull analysis measures the output calculated by harmonic balance while varying the reflection coefficient of one input. The load-pull input is set up as a passive tuner.

See Nexxim Harmonic Balance Analysis.

Oscillator Analysis Tool

Oscillator analysis uses harmonic balance analysis techniques to find the oscillating frequency of a resonant circuit. The resonant frequency is an unknown, although the process of finding it can be aided by providing an estimate that is close to the true value.

The analysis has two phases. First, initial estimates of the oscillating frequency and test voltage are made, either from user input or as directed by simulator options, and those estimates are assigned to the probe. Second, multiple harmonic balance analyses are performed while adjusting the probe frequency and voltage, until a more accurate, final resonant frequency is found.

Oscillator analysis can be set up for single-tone or multi-tone calculations. In single-tone analysis, a single resonant frequency is calculated. In multi-tone analysis, two or more unknown oscillations are analyzed. Including the effects of one or more driving frequencies is optional.

The simulation can be directed to run just the initial estimate phase (resonant frequency search), or to run both the initial and the final phases of oscillator analysis.

The simulation can include phase noise analysis as part of the oscillator analysis.

See Nexxim Oscillator Analysis.

Envelope Analysis

Envelope analysis is commonly used to analyze systems where harmonic balance or transient analysis alone is not adequate. Such systems include circuits with two inputs, where one input is a fast-changing periodic or quasi-periodic source such as a clock or Local Oscillator (LO) and the other input is a non-periodic source such as a baseband RF modulator that changes on a timescale that is orders of magnitude slower than the timescale of the fast-changing input. Transient analysis requires a small timestep to capture the fast-changing input, but then requires a very large number of timesteps to simulate the slowly changing non-periodic input. Harmonic balance fails to analyze the non-periodic input.

Envelope analysis uses transient analysis to simulate the slowly moving signal in the time domain, plus harmonic balance to analyze the fast-moving signal in the frequency domain. The time-domain analysis can use a varying timestep that is appropriate to the slowly changing waveform. At each timestep of the transient analysis, an HB analysis is run. The frequency coefficients at each timestep are stored and returned as the result. From these coefficients, obtain a variety of results including a transient-like time-domain result.

See Envelope Analysis.

Time-Varying Noise Analysis

Time-varying noise analysis (TV-Noise) calculates the response of a circuit to shot, thermal, and flicker noise sources analyzed as small-signal AC perturbations around the periodic steady-state operating point calculated by harmonic balance. The circuit elements are linearized by a DC operating point calculation prior to the harmonic balance analysis. The user specifies the range of frequencies over which the noise is to be calculated, one output whose response is to be calculated, and the output harmonic frequencies of interest. Optionally, an input can be specified to use as an input-referred noise (IRN) source. The response matrix shows the noise power spectrum at the specified harmonic frequencies.

See Nexxim Time-Varying Noise Analysis.

Periodic Transfer Function Calculation

The TV Noise analysis can be extended to include periodic transfer function analysis. Periodic transfer function analysis computes the small-signal transfer function from multiple input sources at multiple frequencies to one output at one frequency, or using the sweep of output frequencies on the TV noise analysis setup. A typical application for periodic transfer function analysis is to determine image rejection.

See Nexxim Time-Varying Noise Analysis.

Linear Network Analysis

Linear network analysis (LNA) computes the frequency-dependent scattering, impedance, and admittance parameters for a linearized circuit. Linear network analysis performs a linear frequency-domain analysis. Circuit components are analyzed using Y-matrix analysis, and any nonlinear devices are linearized around their bias points when computing the bias values.

Optionally, LNA can include group delay analysis. Group delay analysis determines the delay of the propagation of energy at a given frequency point. This analysis is defined as the derivative of the phase of a network parameter with respect to frequency, using the S-parameters as a basis for calculation.

Optionally, LNA can include DC noise analysis. Noise analysis calculates the noise spectral density at designated outputs due to thermal, flicker, and shot noise sources in a circuit that has been linearized around the DC bias operating point.

Optionally, LNA can calculate the AC small-signal transfer functions relating outputs to inputs at selected test frequencies. AC analysis calculates the response of a circuit to small-signal AC perturbations around the DC bias operating point.

See Nexxim Linear Network Analysis.

Smith Tool

The Ansys Smith Tool is an interactive Smith chart utility for the design of amplifiers, oscillators, and matching networks. The Smith Tool is available after a successful linear network analysis.

See Smith Tool.

Nexxim Harmonic Balance

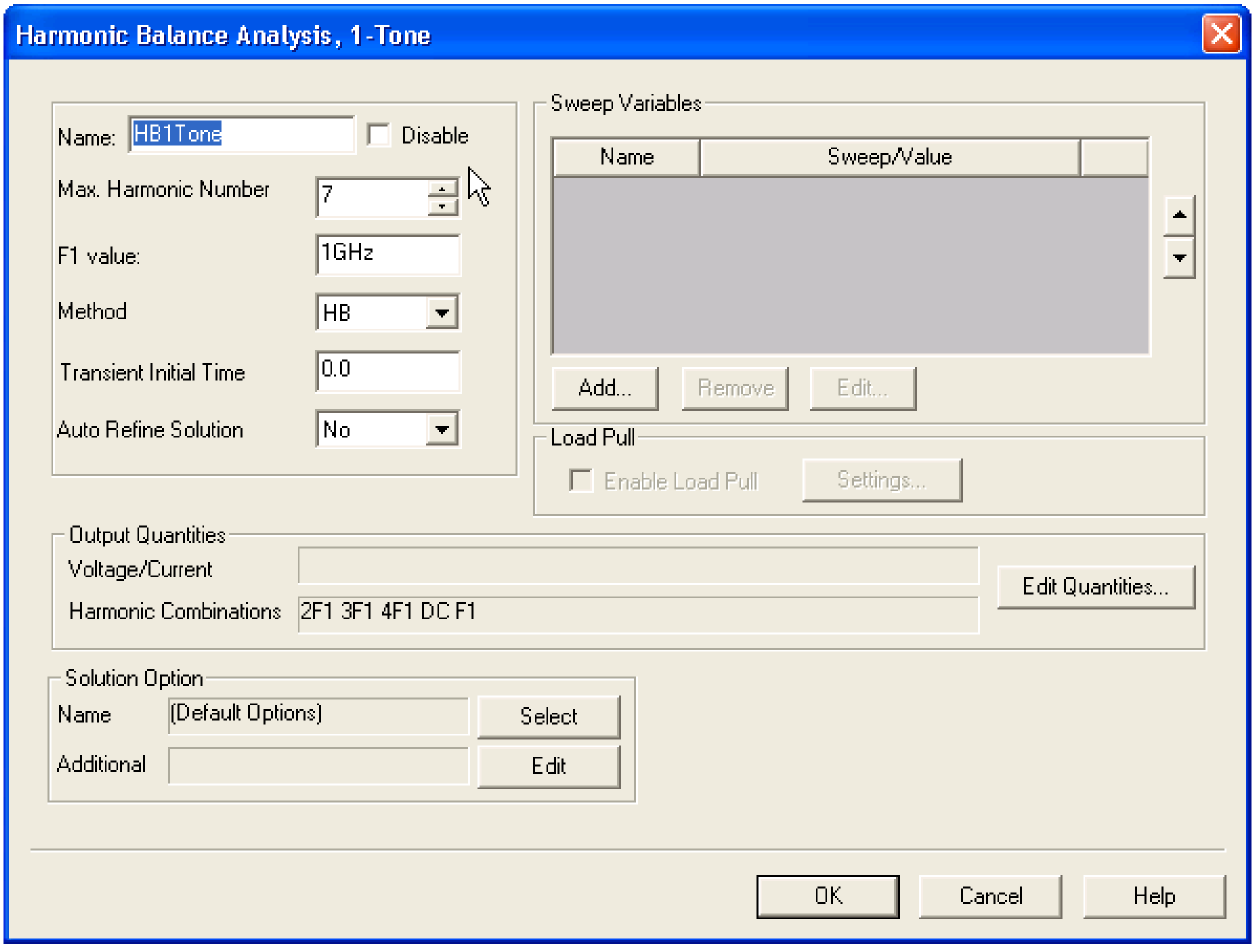
Harmonic balance analysis calculates the periodic or quasi-periodic steady-state response of a circuit to periodic inputs by solving the circuit equations in the frequency domain. Time domain equations are represented by their Fourier series equivalents. In a one-tone analysis, the input is a sine wave at a specified frequency *f*, and the response is usually measured over a specified range of multiples or sub-multiples of that frequency. In a multi-tone analysis, the inputs are at two or more frequencies (f_1, f_2, ...). The response is a spectrum containing the DC response, the harmonics of the input frequencies, and the sums and differences of the harmonic frequencies. An optional load-pull analysis is available for schematic designs.

Running Harmonic Balance on the Schematic Editor

From the **Schematic Editor**, perform the following steps to set up and run a harmonic balance analysis using Nexxim.

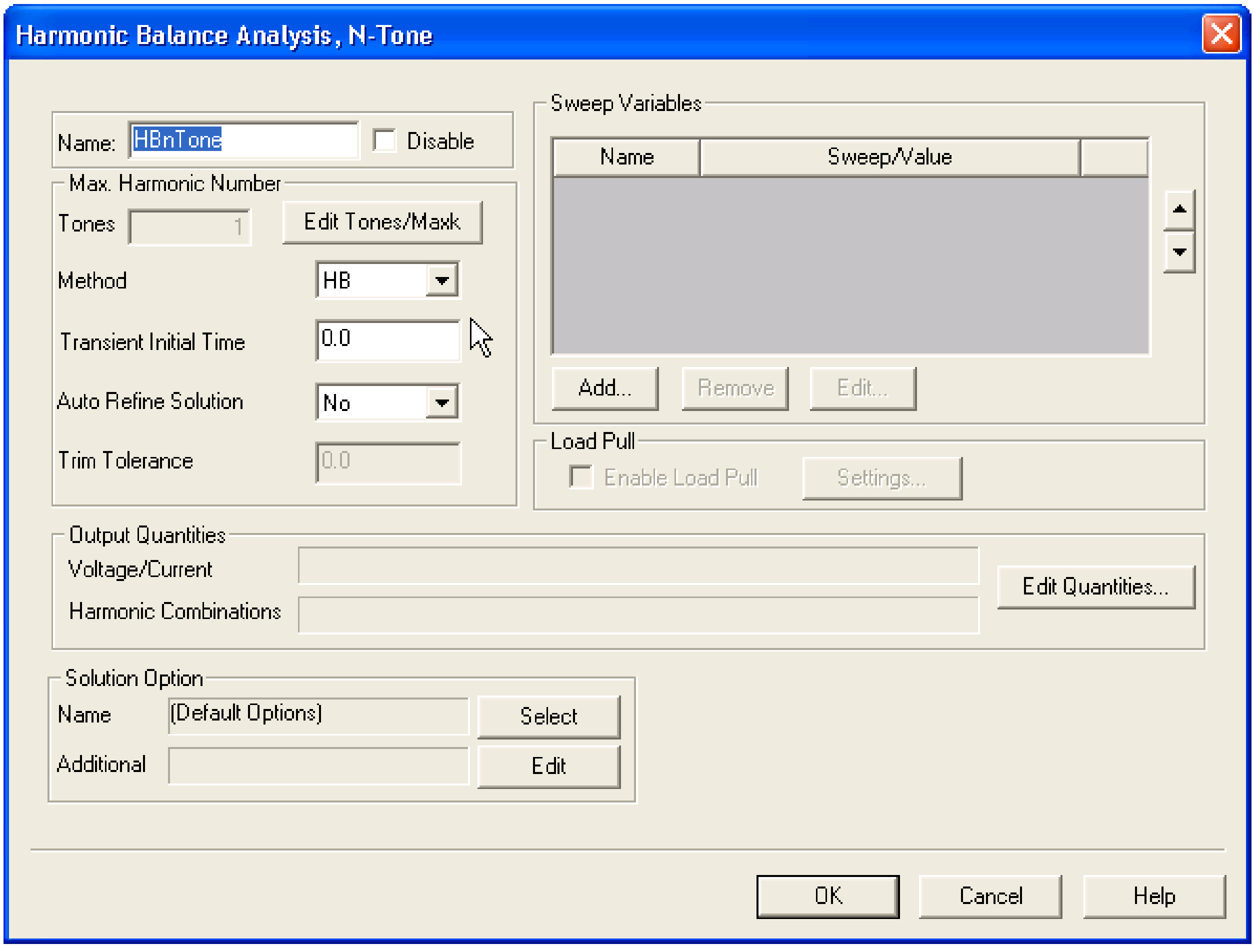
1. In the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim SolutionSetup** > **Harmonic Balance (1-Tone)** or **Harmonic Balance (N-Tone)**.

   Select **1-Tone Analysis** to open the **Harmonic Balance Analysis, 1-Tone** window.



- Type an **Analysis Name** (or accept the default name, for example “HB1Tone”).
- For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
- Specify the maximum harmonic number in the **Max Harmonic Number** field.
- Specify the one-tone frequency in Hz in the **F1 value** field.
- Optionally, select a **Method** (HB or Shooting; HB is the default). See Handling Strongly Nonlinear Circuits for details on the **METHOD** entry in the HB statement.
- Optionally, set **Transient Initial Time** to a positive non-zero value representing time required for Transient to stabilize before HB begins using the result.
- Optionally, set **Auto Refine Solution** to Yes (default is No). See Increasing Accuracy or Speed for details on the **AUTO_REFINE_SOLUTION** entry in the HB statement.

Select **N-tone Analysis** to open the **Harmonic Balance Analysis, N-Tone** window.



- Type an **Analysis Name** (or accept the default name, “HBnTone”).
- For most simulations, leave the **Disable** boxunselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
- Click **Edit Tones/Maxk** to open the **Edit Tones/Maximum Harmonic Number** window. Set the **No. of tones** ; the window shows that number of tones. Click the **Value** field for each tone to enter the appropriate frequency. Click the **MAXK** field for each tone to enter the maximum harmonic number. Click **OK** to close the **Edit Tones/Maximum Harmonic Number** window and return to the **Harmonic Balance Analysis, N-Tone** window.
- If you set the **No. of tones** to 1, the analysis becomes a single-tone analysis. With this setup, the **Method** field is enabled. Select HB or Shooting (HB is the default). See

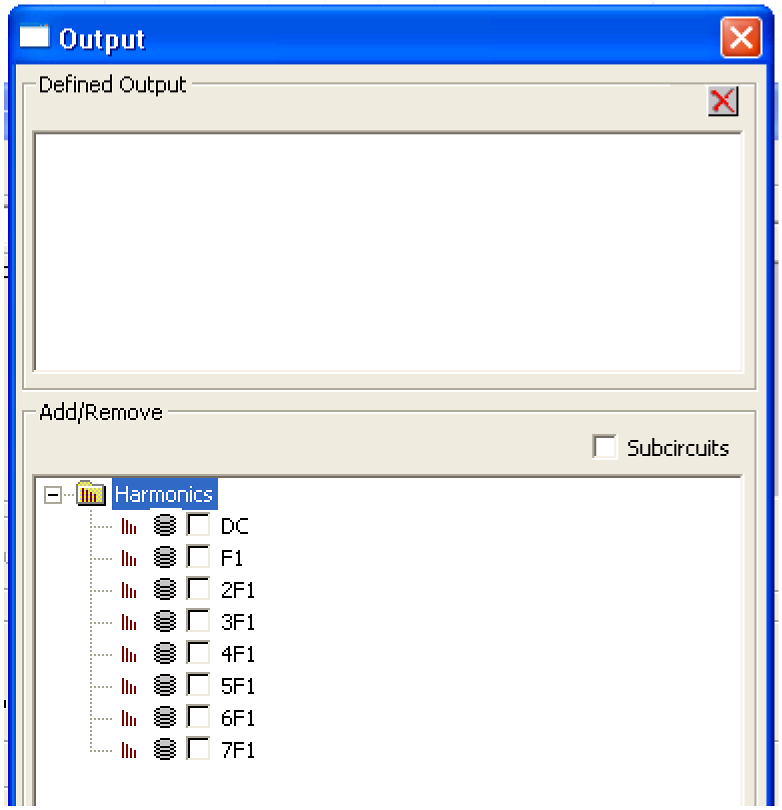
Handling Strongly Nonlinear Circuits for details on the **METHOD** entry in the HB statement.

- Optionally, set **Transient Initial Time** to a positive non-zero value representing time required for Transient to stabilize before HB begins using the result.
- Optionally, set **Auto Refine Solution** to Yes (default is No). See Increasing Accuracy or Speed for details on the **AUTO_REFINE_SOLUTION** entry in the HB statement.
- When the number of tones is greater than one, the **Trim Tolerance** field is activated. See Increasing Accuracy or Speed for details on the **TRIM_TOL** entry in the HB statement.

2. To add a sweep, locate the **Sweep Variable** area in the **Harmonic Balance Analysis**window.

- From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
- In the **Variable** list, select **Temp** or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**.
- Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
- For details on sweeps, see Variable Sweep.
- Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Harmonic Balance Analysis** window.

3. Click **Edit Quantities** to open the **DefinedOutput** window. In addition to the circuit nets and nodes, select the harmonics to include in the analysis:



- Click the check boxes on the **Output** window to select harmonics, and add net voltages and branch currents to the data generated by the harmonic balance analysis. (If no harmonics are selected, harmonics DC, F1, 2F1, 3F1, and 4F1 is automatically selected for output. In the **Report** window, only the selected harmonics are available for plotting.)
- Click **OK** to add the selected output values and return to the **Harmonic Balance Analysis** window.

4. Optionally, use the fields in the **Solution Options** group box to select or add HB analysis options and other Nexxim options to the design.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **HB Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Select the **HB Options** tab. Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window box. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **HB Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

The line can contain multiple options settings. Use spaces to separate the options settings.

To modify or delete an option once it has been added, just edit or delete its entry in the list.

To clear all options, delete the entire line.

Click **OK** to return to the **Harmonic Balance Analysis** window.

The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

- For more information, see Harmonic Balance Options.

5. Run the simulation:

- From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
- The **Message Manager** window signals success or failure.

Running Harmonic Balance on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist should contain the information for running the harmonic balance analysis. No Solution Setup is required.

Run Nexxim Harmonic Balance Analysis

Run the simulation:

- From the **Circuit** menu, select **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
- The **Message Manager** window signals success or failure.

Nexxim Harmonic Balance Netlist Format

Harmonic balance can be invoked on the netlist with a .HB statement.

The syntax for 1-tone or N-tone harmonic balance analysis uses parentheses to enclose the vector of tone frequencies and the corresponding vector of harmonics:

```
.HB TONES=(f1 [, ...fN]) MAXK=(M1 [, ...MN]) [METHOD=HB|SHOOTING]
[AUTO_REFINE_SOLUTION=YES|NO] [TRIM_TOL=val] [TSTAB=val]
```

The parentheses are required for multitone harmonic balance analysis; they can be omitted for one-tone harmonic balance analysis.

The **TONES** group box in the .HB statement specifies the frequencies to use in the analysis. The frequencies of the actual voltage or current inputs to the circuit must be equal to or integer multiples of the **TONES** frequencies.

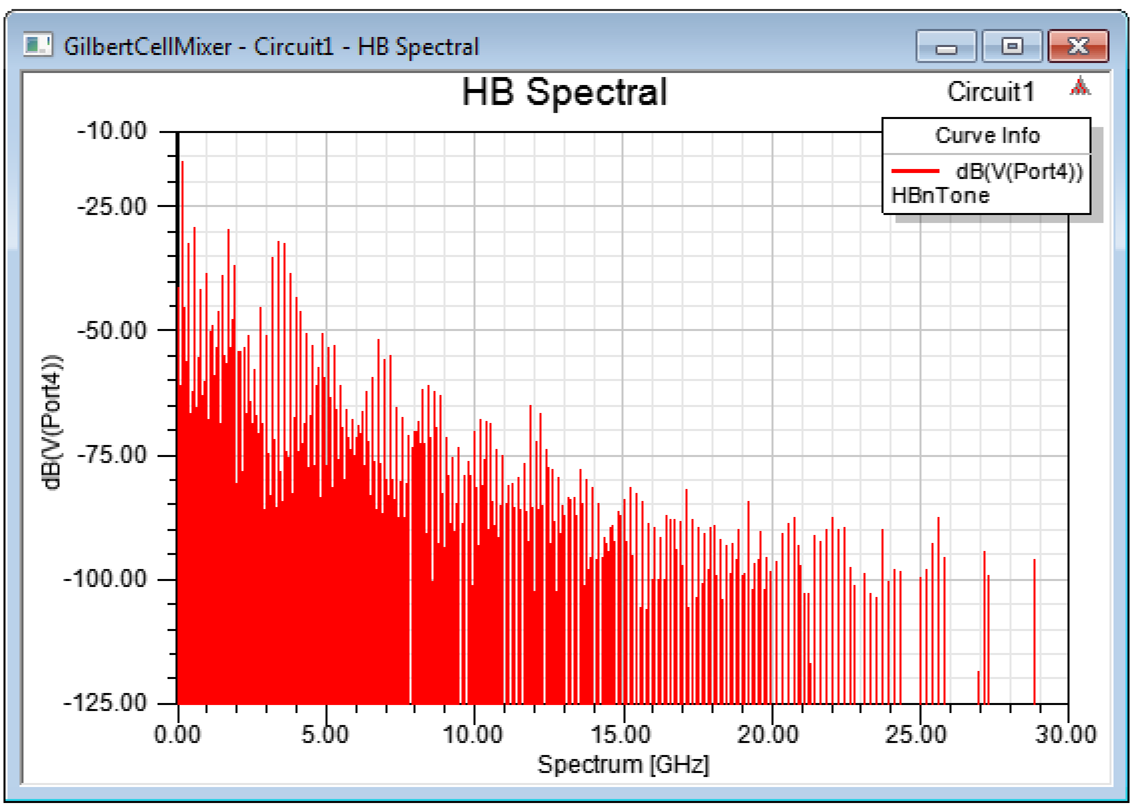
See Handling Strongly Nonlinear Circuits for details on the **METHOD** and **TSTAB** entries in the HB statement.

See Increasing Accuracy or Speed for details on the **AUTO_REFINE_SOLUTION** and **TRIM_TOL** entries in the HB statement.

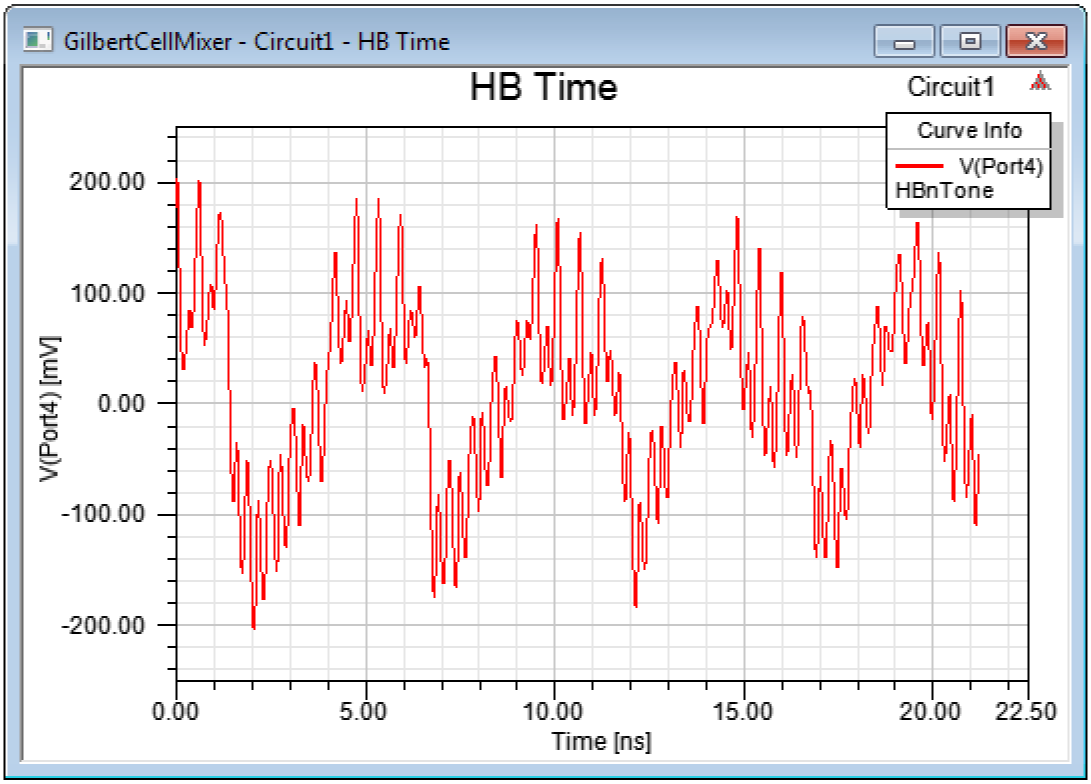
Harmonic Balance Analysis Outputs

Results on the harmonic balance analysis are available in the spectral domain and in the time domain.

Spectral domain results look like the following:



Time domain results look like the following graph:



For details on creating and modifying reports, see Generating Reports and Post-processing.

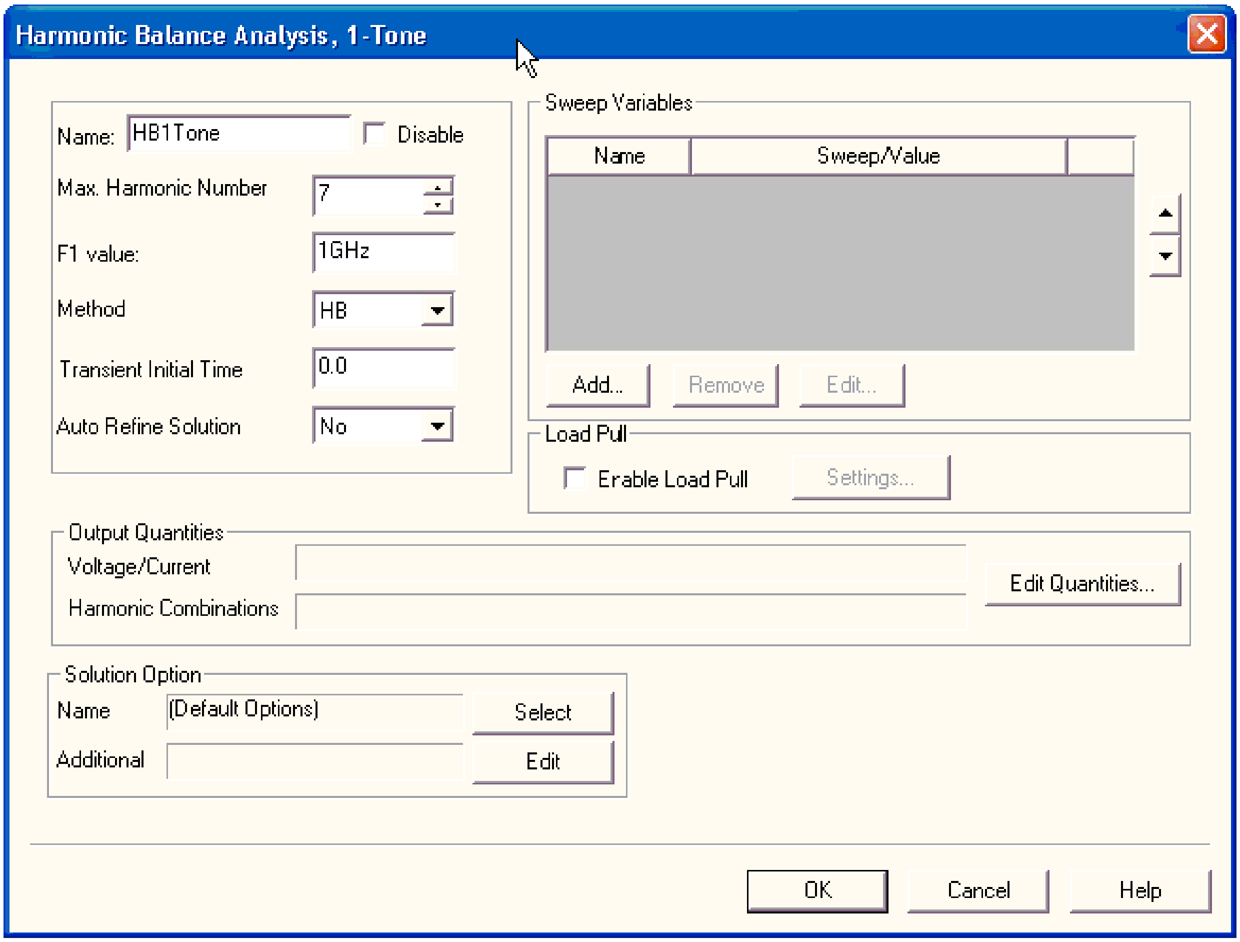
See HB Output Notes for details on the relationships between the responses in the frequency and time domains.

HB Load-Pull Analysis

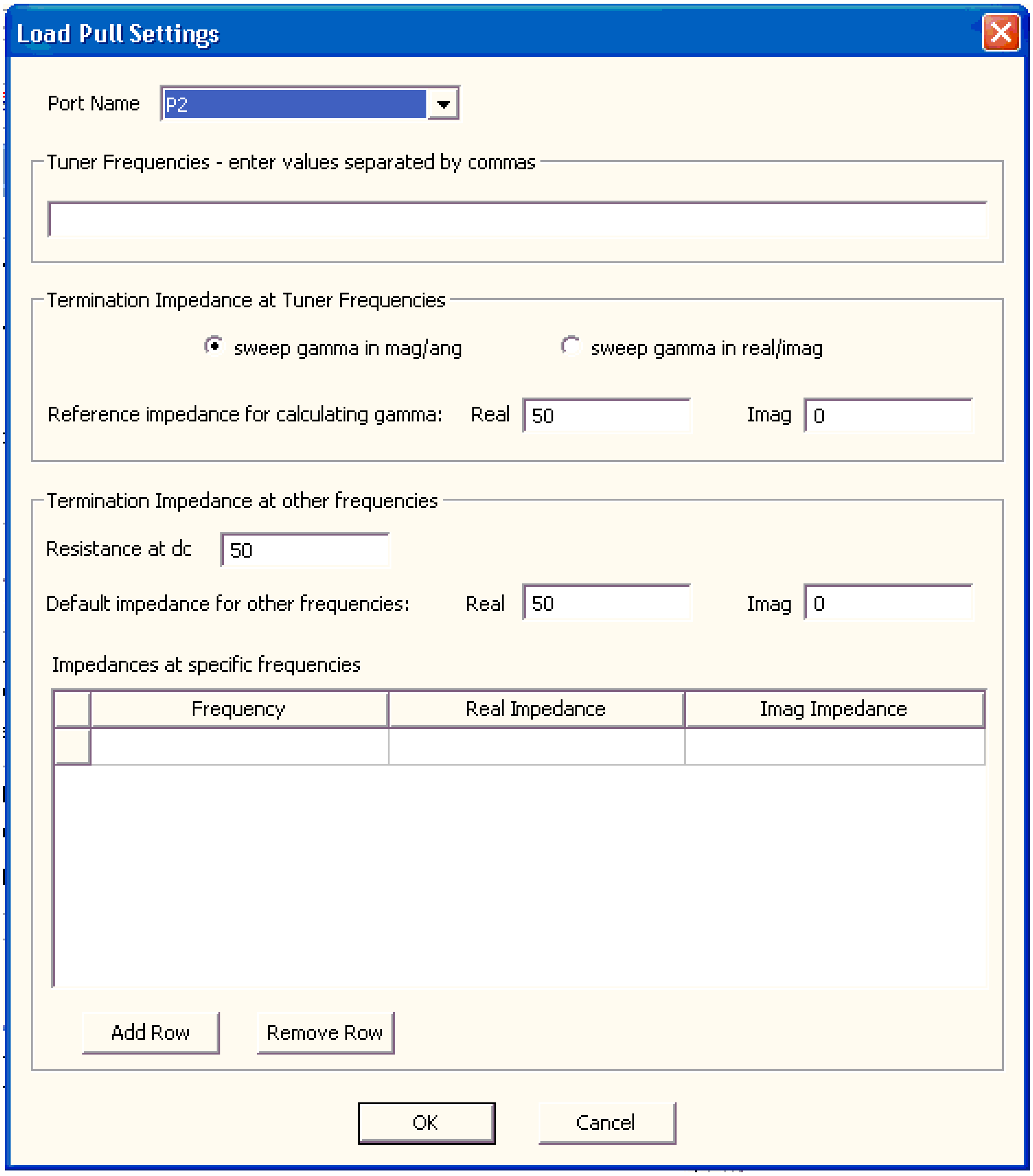
Load-pull analysis is used to determine the optimum load on a circuit. Nexxim load-pull analysis measures the output calculated by harmonic balance while varying the reflection coefficient of one load. The load-pull load impedance is set up as an ideal tuner.

Load-Pull Analysis in a Schematic

Load-Pull analysis is set up in the **Harmonic Balance Analysis, 1-Tone** window or **Harmonic Balance Analysis, N-Tone** window. The circuit must have at least one port to activate the **Load Pull** group box.



1. Click the **Enable Load Pull** check box in the **Load Pull** group box to activate load pull analysis. The **Load Pull Settings** window opens:



- Make a selection on the port drop-down menu.
- Specify one or more tuner frequencies to be applied in the analysis.
- Select **mag/ang** or **real/imag** as the format for the sweep of the tuner reflection coefficient Gamma. Set the reference impedance to use for the tuner at the specified tuner frequencies. Nexxim automatically creates local variables **ZRho** (sample impedance) and **ZAng** (angle of sample impedance) and sets up the selected port as an ideal tuner with a reflection coefficient defined by the ZRho and ZAng parameters.

- Set the termination impedances to use at DC and at frequencies other than the tuner frequencies. You can set specific impedances for particular frequencies in the table, and a default. Click **Add Row** and **Remove Row** to modify the table, as necessary.
- Click **OK** to close the **Load Pull Settings** window.

Nexxim creates linear sweeps of parameters **ZRho** and **ZAng**. The sweep setups are displayed in the **Sweep Variables** group box of the **Harmonic Balance Analysis** window. You can select a load-pull sweep and edit it, but you cannot remove it.

Disabling Load-Pull analysis removes the **ZRho** and **ZAng** sweeps on the setup and removes the **ZRho** and **ZAng** variables on the port.

2. From the **Project Manager** window, expand the **Project Tree** and **Analysis** folder. Then right-click the HB solution setup and select **Analyze**.

- If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
- If the analysis is not successful, check the **Message Manager** window for an explanation, and then take corrective action.

Load-Pull Analysis Netlist Format

The netlist can add a load-pull analysis by defining netlist parameters ZRho (sample impedance) and ZAng (angle of sample impedance), using these netlist parameters to set an ideal tuner element, and setting up sweeps of the ZRho and ZAng parameters as part of the HB analysis statement. The **bolded** lines in the folloeing example netlist, show the additions for the load-pull analysis.

```
* Nexxim Load-Pull Netlist Example

.PARAM Freq1=1.5e9
.PARAM Freq2=2.2e9
.PARAM ZRho=1 // Sample impedance for load-pull
.PARAM ZAng=0 // Sample impedance angle

V1 net_0 0 DC=0.5 SIN(0 0.5 Freq1 0 0 0)
V2 net_0 0 DC=0.5 SIN(0 0.5 Freq2 0 0 0)
R2 net_0 Port1 1000

RPort1 Port1 0 PORTNUM=1 // Ideal tuner element
+ GAMMA_MAG=ZRho GAMMA_ANG=ZAng // Reflection coefficient
+ REF_REAL=50 REF_IMAG=0 // Reference impedance
+ TUNER_FREQS=[Freq1, Freq2] // Main tuner frequencies
+ Z_FREQS=['2*Freq1-Freq2', '2*Freq2-Freq1'] // Cluster freqs
+ Z_REAL=[100, 100] Z_IMAG=[0.5, 0.5] // Cluster impedances
+ RDC=50 // Tuner Resistance at DC
+ RDEF=50 XDEF=0 // Default tuner impedance for all frequencies
```

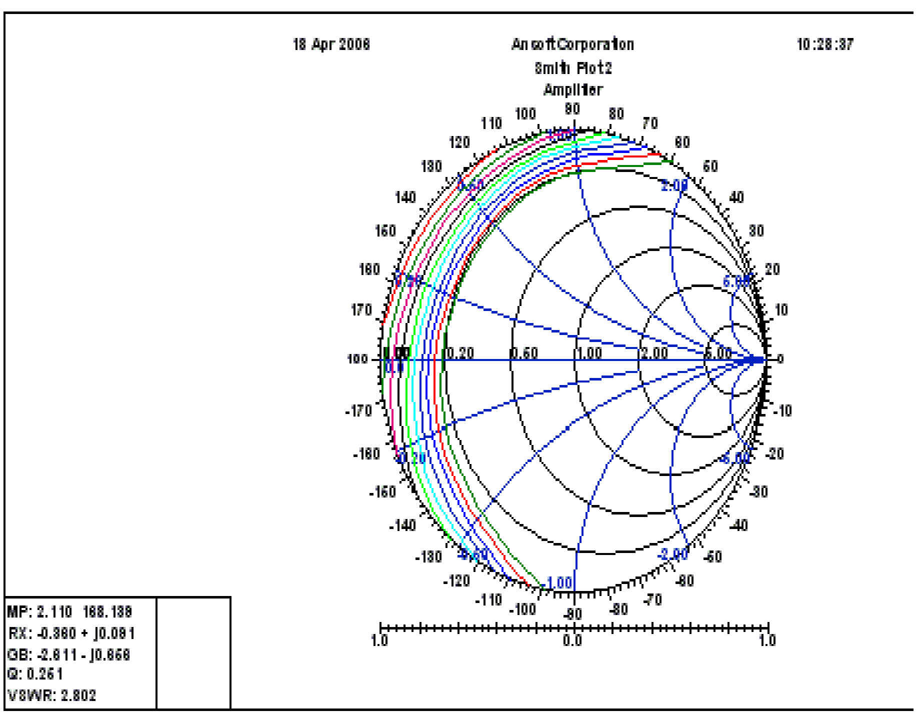
```
.HB
 + TONES=[Freq1, Freq2] MAXK=[2, 2]
 + SWEEP // Sweeps of ZRho and ZAng implement the load-pull
 + ZRho LIN 11 0 1
 + ZAng LIN 13 0 6.283185307
.END
```

For more information, see *Ideal Tuner* in the Nexxim Components help topics for details on the ideal tuner element.

Viewing Load-Pull Analysis Results

To display a Smith Contour chart of the load-pull results after a successful analysis:

1. Click the **Results** icon in the **Project Manager** window.
2. Select **Create Load Pull Report**, then select **Smith Contour Plot**. Click **OK** to open the **Reports** window.
3. From the **Reports** window, select the **HB** analysis and select **LoadPull Contour** as the **Domain**.
4. Select **Power**, select the appropriate port number, and select **<none>**. Click **New Report**. A Report window opens to display a graph of the analysis results.



For details on creating and modifying reports, see Generating Reports and Post-processing.

Harmonic Balance Options

The default values of the harmonic balance analysis options are listed in the Nexxim Harmonic Balance Options Reference. These default values can be changed with a .OPTIONS statement. The syntax is:

```
.OPTIONS [hb.]option1=value1 [[hb.]option2=value2 ... ]
```

Use the prefix **hb.** when an option applies only to harmonic balance but not to other tools that recognize the option.

See Analysis-Specific Options for a listing of options that are used by more than one Nexxim analysis.

> **Note:**
>
> Harmonic balance uses 1.0e-7 as the default for **vntol**, the absolute node voltage tolerance. Nexxim DC and transient analysis tools use 50e-6 for the default. Tightening **vntol** allows HB to accurately analyze circuits with normal values in microvolts.

For the subordinate options that are available when an option such as **initial_guess=transient** is in effect, use periods to separate the subordinate option field (e.g., **hb.initial_guess.transient.tmax**).

See Controlling Nexxim Harmonic Balance Analysis for a discussion of the effects of changing these options.

Harmonic Balance Troubleshooting Guide

The following topics are intended to help with specific harmonic balance issues.

A convergence issue can be caused by a circuit or improper solver setting. To improve convergence on the solver side:

1. Weaken nonlinearity by increasing the values of Alpha and Beta.
2. Improve the initial guess as mentioned in HB Convergence Aids and Frequency Divider Circuit Errors.
3. Increase max_newton_iterations and/or decrease limiting_step to aid convergence.
4. Loosening tolerances (abstol, vntol, reltol) may increase the chance of convergence, as mentioned in HB convergence Check Failures. However, loosening tolerances implies increased errors in the solution and such increased errors may adversely affect a convergence problem. Hence, loose tolerances slightly if necessary.

5. Use the shooting method, see Handing Strongly Nonlinear Circuits. (Nexxim tries the shooting method automatically, for single tone, if HB failed. The simulation time may take longer.)
6. Increase inner iterations, see Inner Iterations Limit Errors.
7. Reduce the finite truncation error by using nonzero values for CMIN, or increasing the number of harmonics.

> **Note:**
> 1. Modification on Alpha, Beta, and CMIN changes the circuit.
> 2. Different methods are used for the initial guess alternately. If one failed, try the next. This process may take long and sometimes display error messages in other types of analysis, like transient analysis. If a simulation succeeds at the end, try to manually set pseudo_transient as the continuation method to avoid long simulation time.

Inner Iterations Limit Errors

When the inner linear solver does not converge within the maximum number of iterations set by the option **hb.max_linearsolver_iterations** (the default is 100 iterations), Nexxim generates the following warning:

```
[warning]analysis:hb(warning): Inner iterations limit reached. If
analysis fails, try setting hb.max_linearsolver_iterations = 200 or
higher
```

The option **hb.max_linearsolver_iterations** can be set from in the **Schematic Editor** or in the netlist.

Frequency Divider Circuit Errors

Frequency divider circuits can present a problem for harmonic balance. The options that prove most useful depend somewhat on whether the harmonic balance is a single-tone or multitone analysis

With single-tone analyses, convergence can be achieved by setting the **method** option to **shooting**:

```
.option hb.method=shooting
```

This shooting method for single-tone problems is a time-domain method for achieving a steady state.

Another useful option for single-tone circuits is to use transient analysis to generate the initial guess at voltages:

```
.option hb.initial_guess=transient
```

With multitone problems, successful convergence may be achieved by using the **fewer_tones_ transient** strategy:

```
.option hb.initial_guess=fewer_tones_transient
```

The **fewer_tones** strategy in general calculates a single-tone solution as the initial guess for the 2-tone analysis, and proceeds to add more and more tones until the full multitone problem converges. With the **fewer_tones_transient** option, Nexxim uses transient analysis to generate the initial guess for the single-tone.

Convergence can also be improved by doubling or tripling the value of **tstop**:

```
.option hb.initial_guess=fewer_tones_transient.tstop=val
```
k,

HB Convergence Check Failures

The Linear Solver uses function (residue) and delta (update) checks similar to those described under DC Analysis.

- View Function Check.
- View Delta Check.

Sometimes a combination of roundoff error and tight tolerances causes convergence at a particular node to fail with a trace message such as:

```
#|F| = 2.14983e-011 |dx| = 1.11644e-015 |x| = 1.2
<trace>:Residue convergence check failed for node net_1979.
```

If this failure occurs, try loosening **abstol** (absolute current tolerance) to 4e-12 or higher (on the default of 1e-12 for harmonic balance):

```
hb.abstol=4e-12
```

This change affects the entire circuit, not just the problem node. Although it is not a robust solution, loosening **abstol** often achieves convergence on the correct solution.

Nexxim Oscillator Analysis

Oscillator analysis uses harmonic balance analyses to find the oscillating frequency of a resonant circuit. The process of finding the unknown resonant frequency can be aided by providing an estimate that is close to the true value.

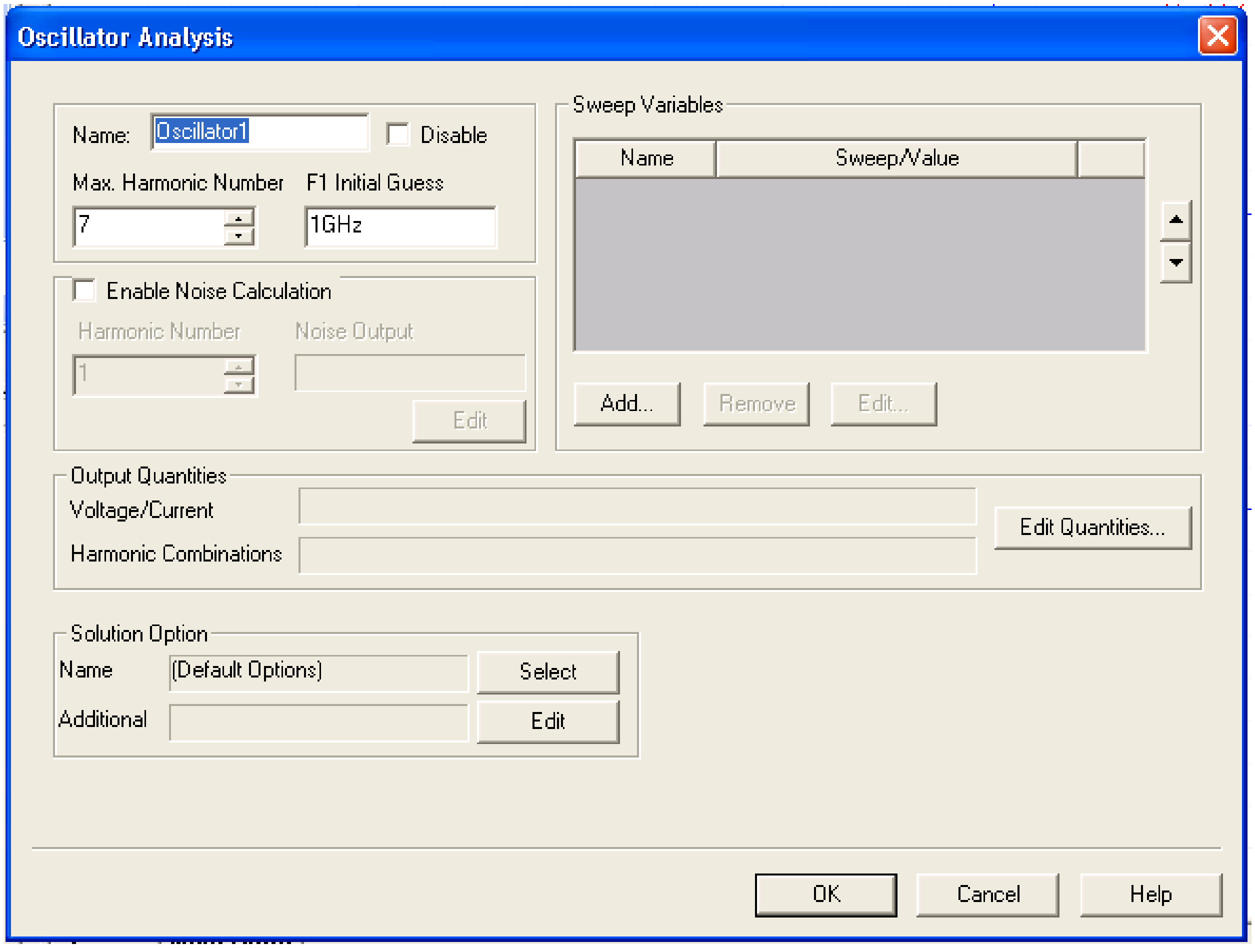
Oscillator analysis uses an oscillator probe element to provide a range of test voltages and frequencies. Use of the probe is explained in the Oscillator Analysis Technical Notes topic. The analysis has two phases. First, initial estimates of the oscillating frequency and test voltage are made, either from user input or as directed by simulator options, and those estimates are assigned to the probe. Second, multiple harmonic balance analyses are performed while adjusting the probe frequency and voltage, until a more accurate, final resonant frequency is found.

Oscillator analysis can be set up for single-tone or multi-tone calculations. In single-tone analysis, a single resonant frequency is calculated. In multi-tone analysis, two or more unknown oscillations are analyzed. Including the effects of one or more driving frequencies is optional. The simulation can be directed to run an initial estimate (resonant frequency search), or to run both the initial and the final phases of oscillator analysis. The simulation can include phase noise analysis.

Running Single-Tone Oscillator Analysis on the Schematic Editor

Single-tone oscillator analysis is typically used on a circuit that contains an oscillator source of unknown frequency, with no other frequency sources in the circuit. From the **Schematic Editor**, perform the following steps to set up and run a single-tone oscillator analysis using Nexxim.

1. Set up the schematic circuit for oscillator analysis by inserting an oscillator probe between two nodes in your circuit. Edit the probe properties and specify an initial guess for the unknown oscillating frequency (**FREQ** parameter) and an initial voltage (**A** parameter). See Using the Oscillator Probe for details.
2. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Oscillator Analysis (1-Tone)** to open the **Oscillator Analysis** window.



3. Type an **Analysis Name** (or accept the default name, for example, “Oscillator”).
4. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
5. The **Oscillator Analysis** window opens. This window allows you to specify the maximum harmonic number to include in the analysis, and a frequency in Hz to be used as the initial guess for the oscillator frequency:

- Specify the maximum harmonic number to use in the analysis by clicking the up and down arrows in the **Max. Harmonic Number** field.
- Specify the frequency for **F1 Initial Guess**. The initial guess should be the same frequency as the **FREQ** parameter on the oscillator probe. The **F1 Initial Guess** value overrides the probe **FREQ** parameter if they are not the same.

6. Optional: To run Phase Noise analysis, check the **Enable Noise Calculation** group box. See Running Phase Noise Analysis on the **Schematic Editor** for details on the other

steps in setting up the phase noise analysis.

7. To add a sweep, locate the **Sweep Variable** area in the **Oscillator Analysis** window.
- From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
- In the **Variable** list, select **Temp** or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**.
- Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
- For details on sweeps, see Variable Sweep.
- Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Oscillator Analysis** window.

7. Click **Edit Quantities** in the **Output Quantities** field to open the **Defined Output Quantities** window.

Expand the icons for the circuit components (**Nets** and individual devices), and click the check boxes to select quantities for output.

Expand the **Harmonics** icon to display a list of harmonics (the number of available harmonics is set by the **Max Harmonic Number** specified in Step 6). Select and **Add** the harmonics you want to analyze. If you do not select any harmonics, harmonics DC, F1, 2F1, 3F, and 4F1 are automatically selected. In the **Report** window, only selected harmonics are available for plotting.

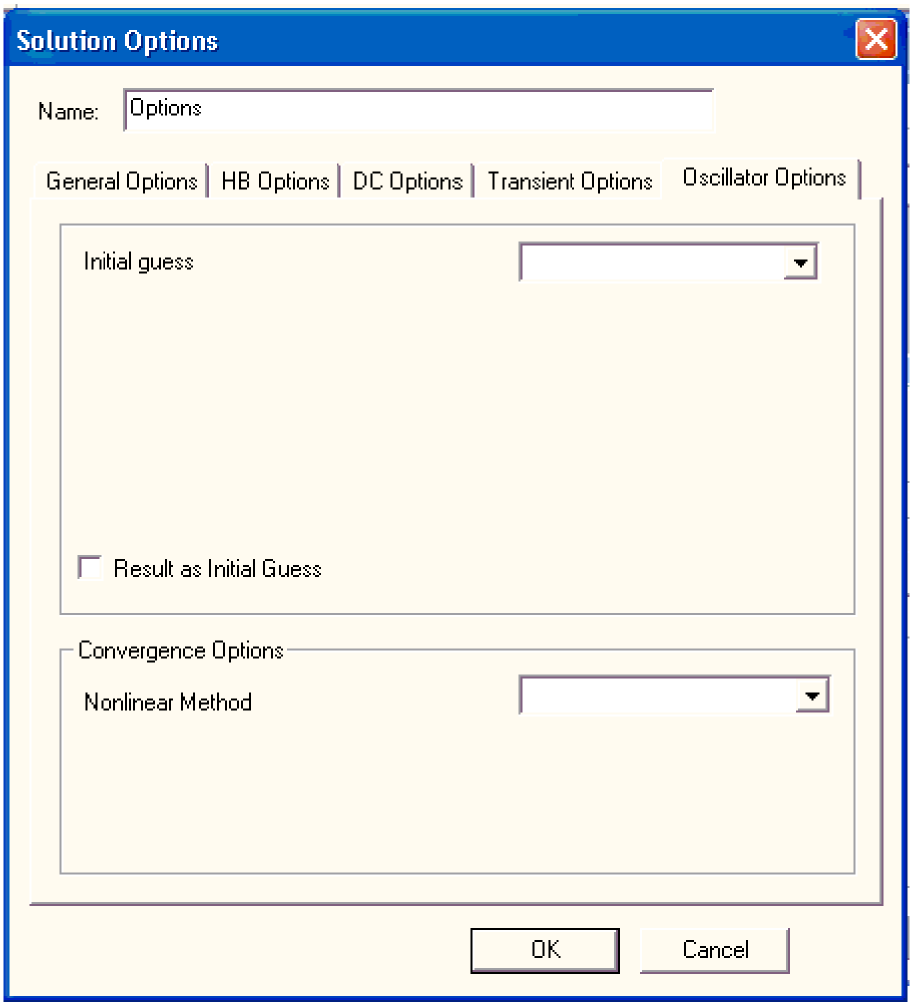
Click **OK** to return to the **Oscillator Solution Setup** window.

9. Optionally, use the fields in the **Solution Options** group box to select or add Oscillator analysis options and other Nexxim options to the design.
- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **Oscillator Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

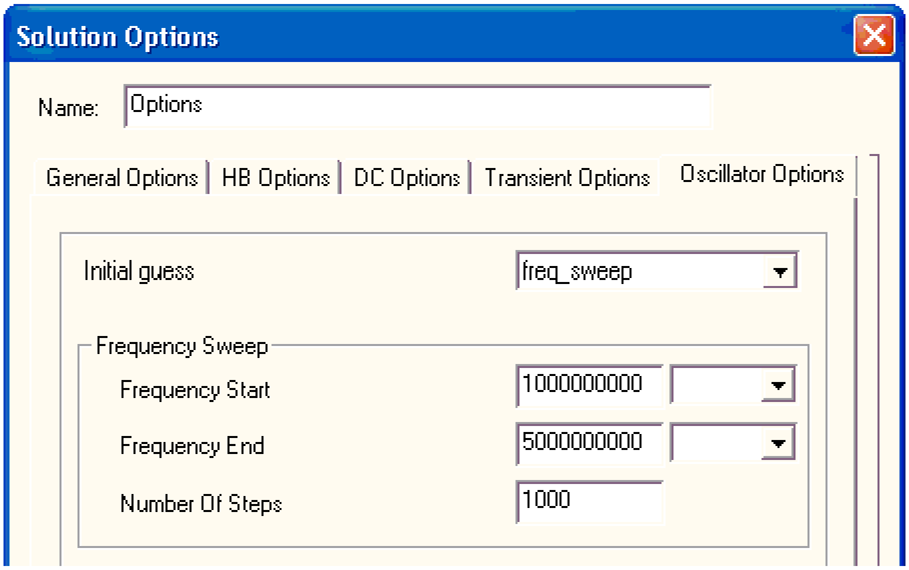
To create a new option set, click **New** to open the **Solution Options** window.

- Select the **Oscillator Options** tab:



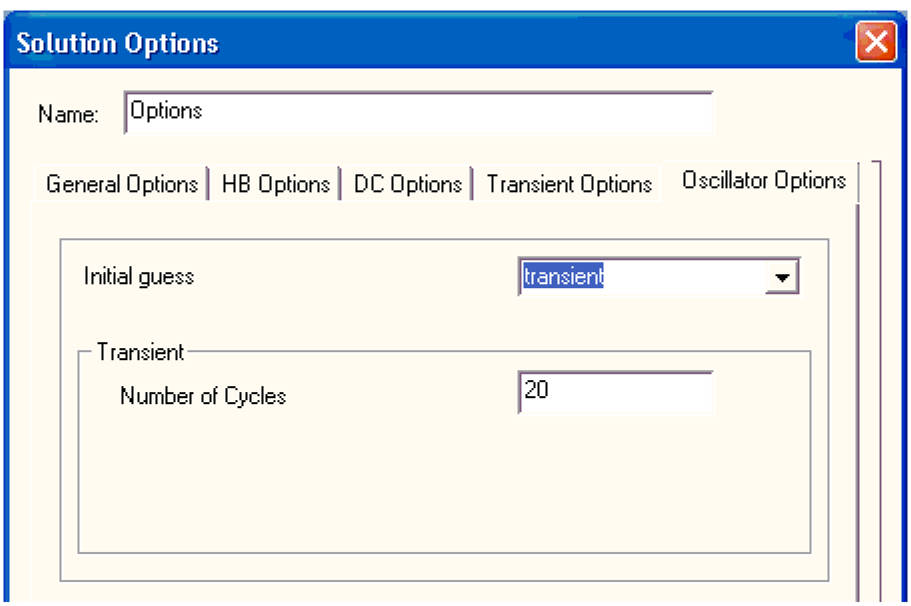
- Enter a name for the options, or accept the default name.
- To select an **Initial Guess** option, click in the drop-down menu field. The options are **Frequency Sweep**, **Transient**, and **DC**.

For an **Initial Guess** of **Frequency Sweep**, the window changes to:



Enter the **Frequency Start** and **Frequency End** values. Select the unit (**Hz** or **kHz**) using the drop-down menu.

For an **Initial Guess** of **Transient**, the window changes to:



Enter a **Number of Cycles** if the default is not as appropriate.

- Click the **Result as Initial Guess** group box to use the result on the previous oscillator sweep step as the initial guess for the current step. Use of this setting can avoid repeated analysis runs.

- To select a **Convergence Option**, click in the drop-down menu field. The options are **Newton**, **Descent**, and **Damped Newton**.
- Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window box. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **DC Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.
- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

The line can contain multiple options settings. Use spaces to separate the options settings.

To modify or delete an option once it has been added, just edit or delete its entry in the list.

To clear all options, delete the entire line.

Click **OK** to return to the **Oscillator Analysis** window.

The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

- For more information on the option settings, see Oscillator Analysis Options.

10. When all setup steps are as appropriate, click **Finish** on the **Oscillator Analysis** window. The solution setup is added to the **Project Manager** window under the **Analysis** icon.
11. Run the simulation:
    - Click the solution setup in the **Project Manager** window and click **Analyze** on the menu. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
    - If the analysis is not successful, an error message appears in the **Message Manager** window. Check the **Message Manager** window for an explanation, then take corrective action.
12. View results:
    - After a successful oscillator analysis, Nexxim displays a message like the following in the **Message Manager** window, near the end of the messages:

```
[info]analysis:osc(info):
Lowest oscillation frequency is 2.28514e_006 Hz (Time, Date)
```

The frequency value is the one calculated for your circuit, and the time and date shows the time of the analysis on your machine.

Note:

As analysis progresses, oscillator analysis may save messages like the following to the log file:

analysis:osc(status): Probe 0: selecting (*node_name1*,*node_name2*) as alternative location 1.

analysis:osc(status): Probe 0: selecting (*node_name3*, *node_name4*) as alternative location 2.

The *node_name*s are circuit-specific. If oscillator analysis fails, Nexxim automatically reruns the analysis after moving the probe to alternative location 1. If that analysis fails, Nexxim retries using alternative location 2.

Running Multitone Oscillator Analysis on the Schematic Editor

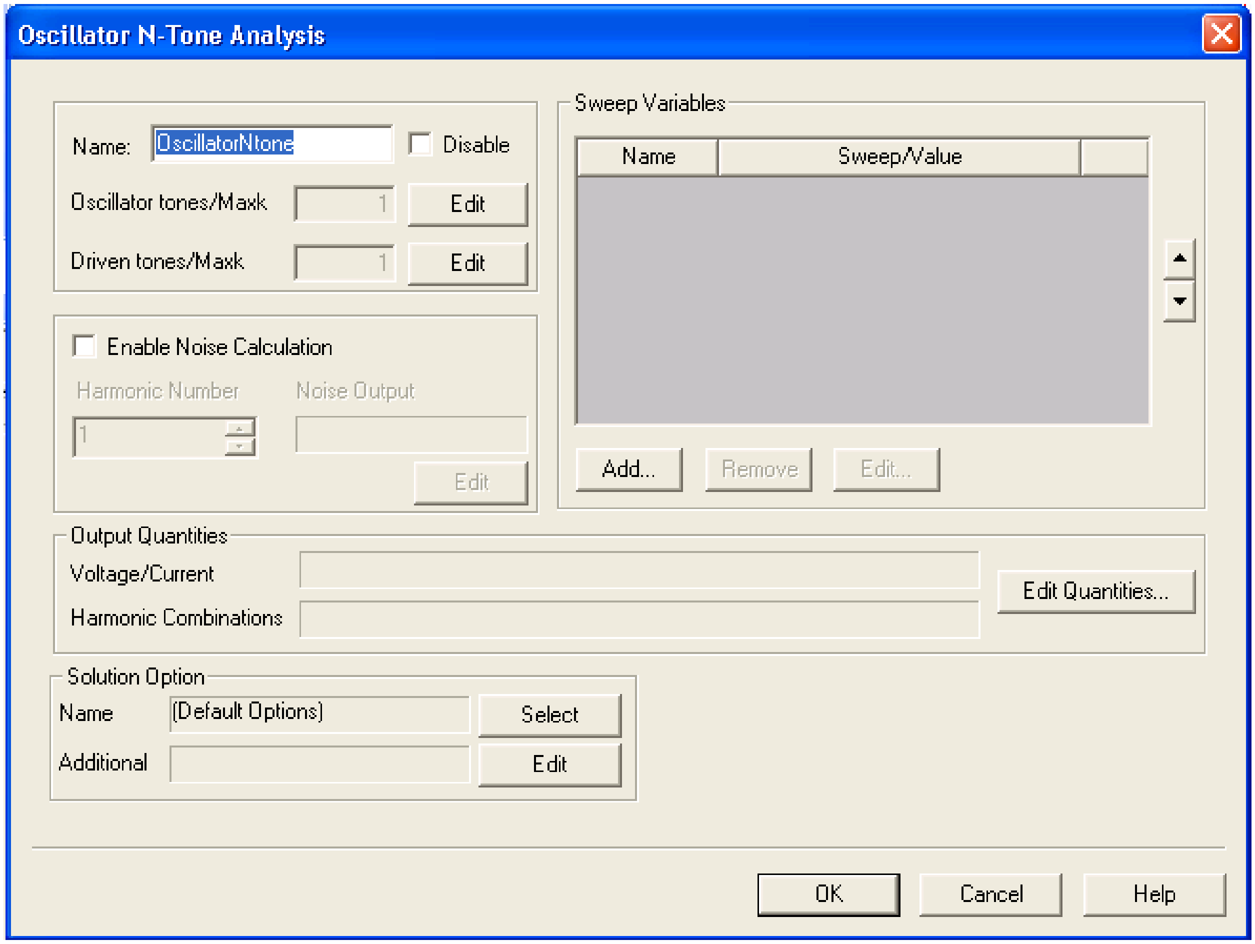
Multi-tone oscillator analysis is typically used on a circuit that includes one or more oscillation sources:

- One unknown resonant frequency (F_1) plus one or more known driven oscillations (RF_1 through RF_M) such as sinusoidal voltage sources.
- Two or more unknown resonant frequencies (F_1 through F_N) with no driven oscillations in the circuit.
- Two or more unknown resonant frequencies (F_1 through F_N) plus one or more known driven oscillations (RF_1 through RF_M).

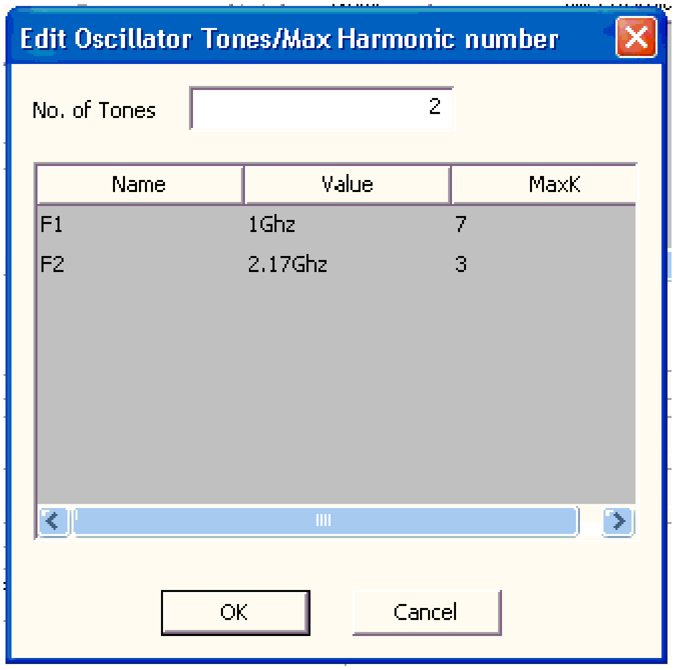
An example is a mixer circuit connecting one or more voltage-controlled oscillators (VCOs) of unknown frequencies with one or more RF sources of known frequencies to produce a single resultant IF output.

From the **Schematic Editor**, perform the following steps to set up and run a multi-tone oscillator analysis using Nexxim, and view the results.

1. Set up the schematic circuit for multi-tone oscillator analysis by inserting oscillator probes between pairs of nodes in your circuit. Each probe represents one unknown oscillation. Edit the probe properties and specify an initial guess for the frequency (**FREQ** parameter) and an initial voltage (**A** parameter) for each probe. See Using the Oscillator Probe for details.
2. From the **Project Manager** window, expand the **Project Tree** and right-click **Analysis**. Then select **Add Nexxim Solution Setup ...** > **Oscillator Analysis (N-Tone)** to open the **Oscillator N-Tone Analysis** window.

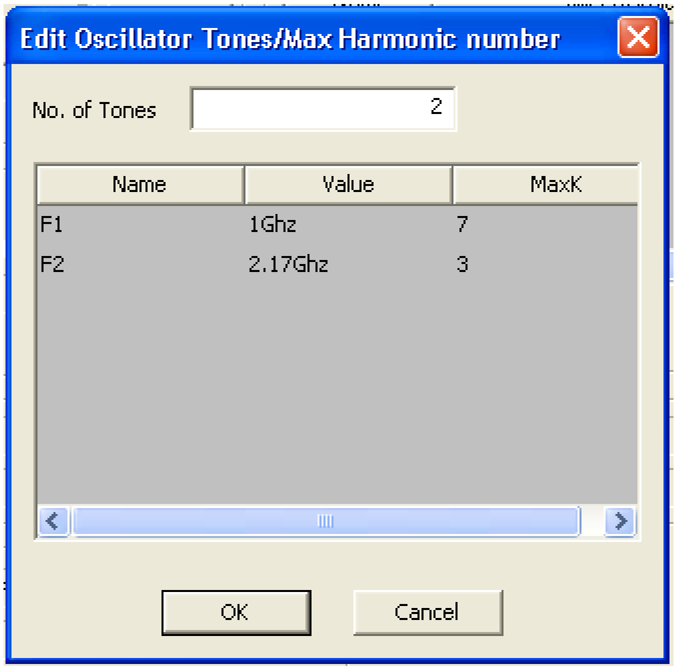


5. Type an **Analysis Name** (or accept the default name, for example “OscillatorNtone”).
6. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
7. Click the upper **Edit** button to open the **Edit Oscillator Tones/Max Harmonic Number** window:



- Specify the number of unknown frequencies to be analyzed in the **No. of Tones** field. The display changes to show that number of tones, labeled **F1** through **F*n***. The names cannot be changed. The illustration above shows two frequencies (however, most analyses have only one unknown frequency). The number of unknown frequencies must be the same as the number of oscillator probes in the circuit.
- Click the **Value** fields for the tones to specify the initial guess frequencies. The initial guess frequencies should be the same as the ones entered in the **FREQ** parameters of the oscillator probes in the circuit. The **Oscillator Tones** values override the probe **FREQ** parameters if they are not the same. The default frequency is 1 GHz.
- Click the **MaxK** field for a tone to specify its maximum harmonic number. The default is 7 harmonics.
- Click **OK** to return to the **Oscillator N-Tone Analysis** window.

8. Click the lower **Edit** button to open the **Edit Driven Tones/Max Harmonic Number** window:



- Specify the number of driven frequencies in the **No. of Tones** field. The display changes to show that number of tones, labeled **RF1** through **RF*n***. The names cannot be changed. The illustration above shows two driven frequencies. These frequencies represent the known, fixed inputs to the analysis.
- Click the **Value** fields for to specify the frequencies of the driven oscillations. The default frequency is 1 GHz.
- Click the **MaxK** field for a tone to specify its maximum harmonic number. The default is 7 harmonics.
- Click **OK** to return to the **Oscillator N-Tone Analysis** window.

9. Optional: To run Phase Noise analysis, check the **Enable Noise Calculation** group box. See Running Phase Noise Analysis on the **Schematic Editor** for details on the other steps in setting up the phase noise analysis.
10. To add a sweep, Click the **Sweep Variable** area in the **Oscillator N-Tone Analysis** window.

- From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
- In the **Variable** list, select Temp or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value, Linear step, Linear count, Decade count,** or **Octave count**.
- Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
- For details on sweeps, see Variable Sweep.

- Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Oscillator N-Tone Analysis** window.

11. Click **Edit Quantities** in the **Output Quantities** field to open the **DefinedOutput Quantities** window.

Expand the icons for the circuit components (**Nets** and individual devices), and click the check boxes to select quantities for output.

Expand the **Harmonic Combinations** icon to display a list of harmonics including combinations. The number of available harmonic combinations is set by the **No. of Frequencies** and the **MaxK** for each frequency set in the **Edit Oscillator Tones/Max Harmonic Number** window in Step 8. Select the harmonics you want to analyze. If you do not select any harmonics, harmonics DC, F1, 2F1, 3F, and 4F1 are automatically selected. In the **Report** window, only selected harmonics are available for plotting.

Click **OK** to return to the **Oscillator Solution Setup** window.

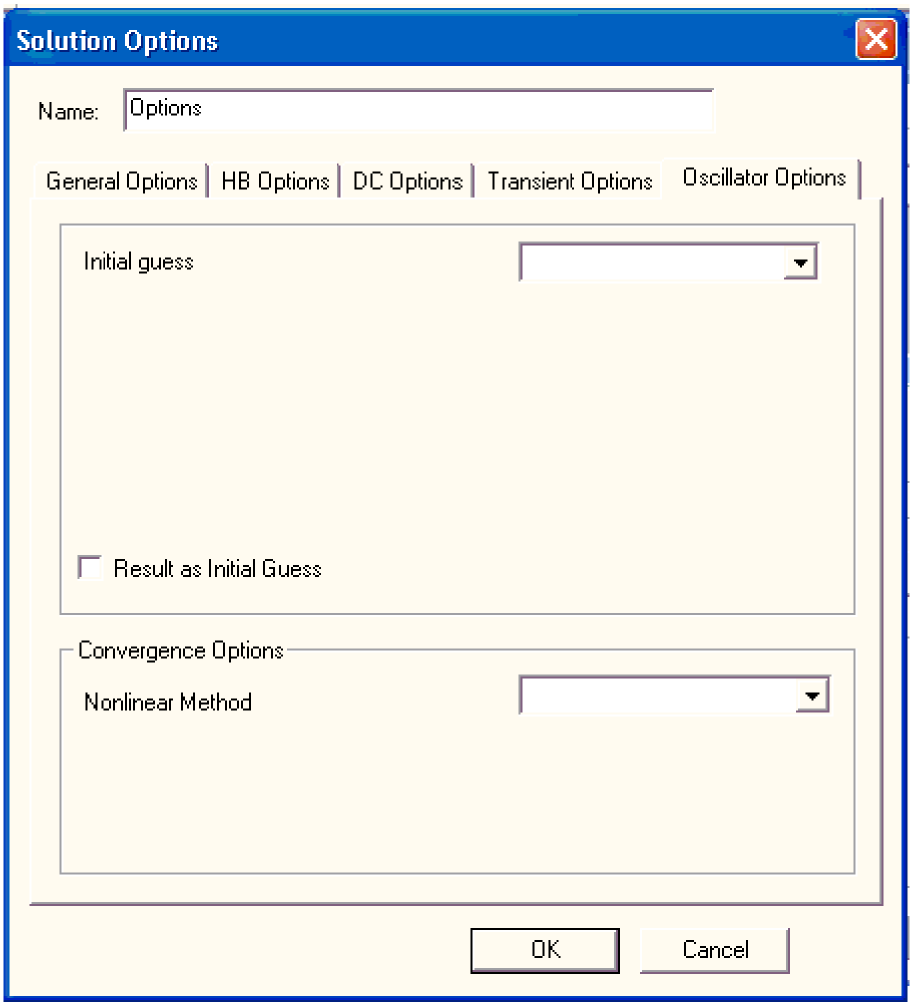
12. Optionally, use the fields in the **Solution Options** group box to select or add Oscillator analysis options and other Nexxim options to the design.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **Oscillator Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

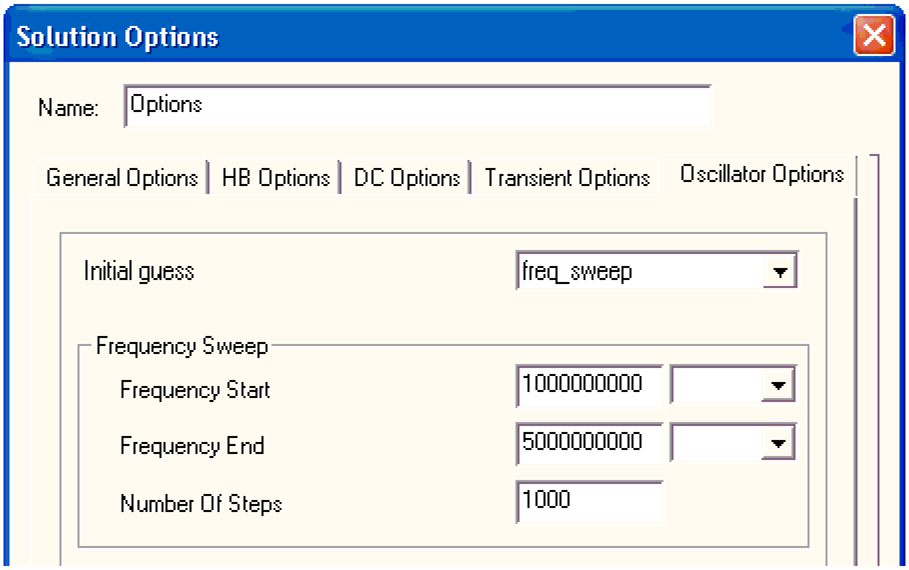
To create a new option set, click **New** to open the **Solution Options** window.

- Select the **Oscillator Options** tab:



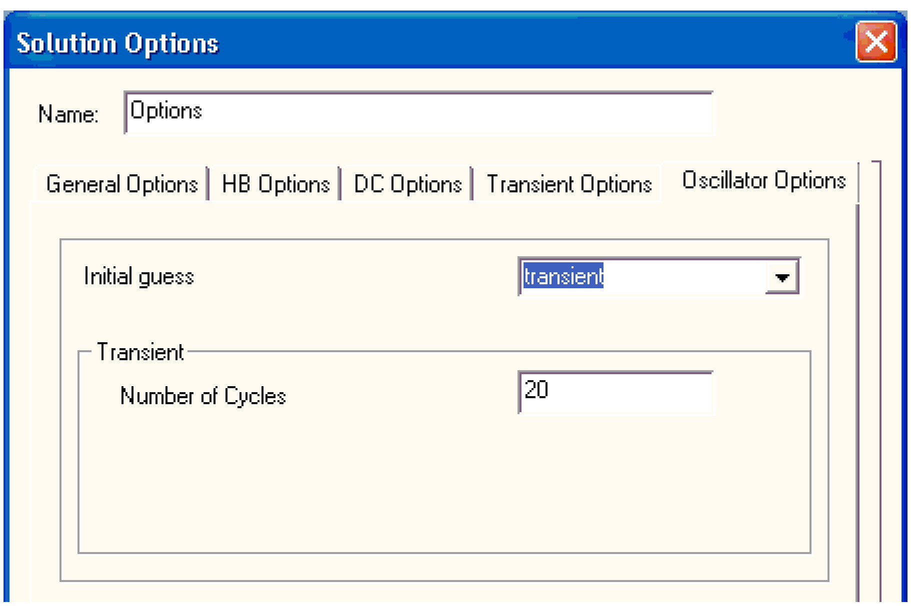
- Enter a name for the options, or accept the default name.
- To select an **Initial Guess** option, click in the drop-down menu field. The options are **Frequency Sweep**, **Transient**, and **DC**.

For an **Initial Guess** of **Frequency Sweep**, the window changes to:



Enter the **Frequency Start** and **Frequency End** values. Select the unit (**Hz** or **kHz**) using the drop-down menu.

For an **Initial Guess** of **Transient**, the window changes to:



Enter a **Number of Cycles** if the default is not as appropriate.

- Click the **Result as Initial Guess** group box to use the result on the previous oscillator sweep step as the initial guess for the current step. Use of this setting can avoid repeated analysis runs.

- To select a **Convergence Option**, click in the drop-down menu field. The options are **Newton**, **Descent**, and **Damped Newton**.
- Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window box. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **DC Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.
- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**. Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Click **OK** to return to the **Oscillator Analysis** window.
- For more information on the option settings, see Nexxim Oscillator Analysis Options.

13. When all setup steps are as appropriate, click **Finish** on the **Oscillator Analysis** window. The solution setup is added to the **Project Manager** window under the **Analysis** icon.
14. Run the simulation:
    - Click the solution setup in the **Project Manager** window and click **Analyze** on the menu. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
    - If the analysis is not successful, an error message appears in the **Message Manager** window. Check the **Message Manager** window for an explanation, then take corrective action.
15. View results:
    - After a successful oscillator analysis, Nexxim displays a message like the following in the **Message Manager** window, near the end of the messages:

```
[info]analysis:osc(info):
Lowest oscillation frequency is 2.28514e_006 Hz (Time, Date)
```

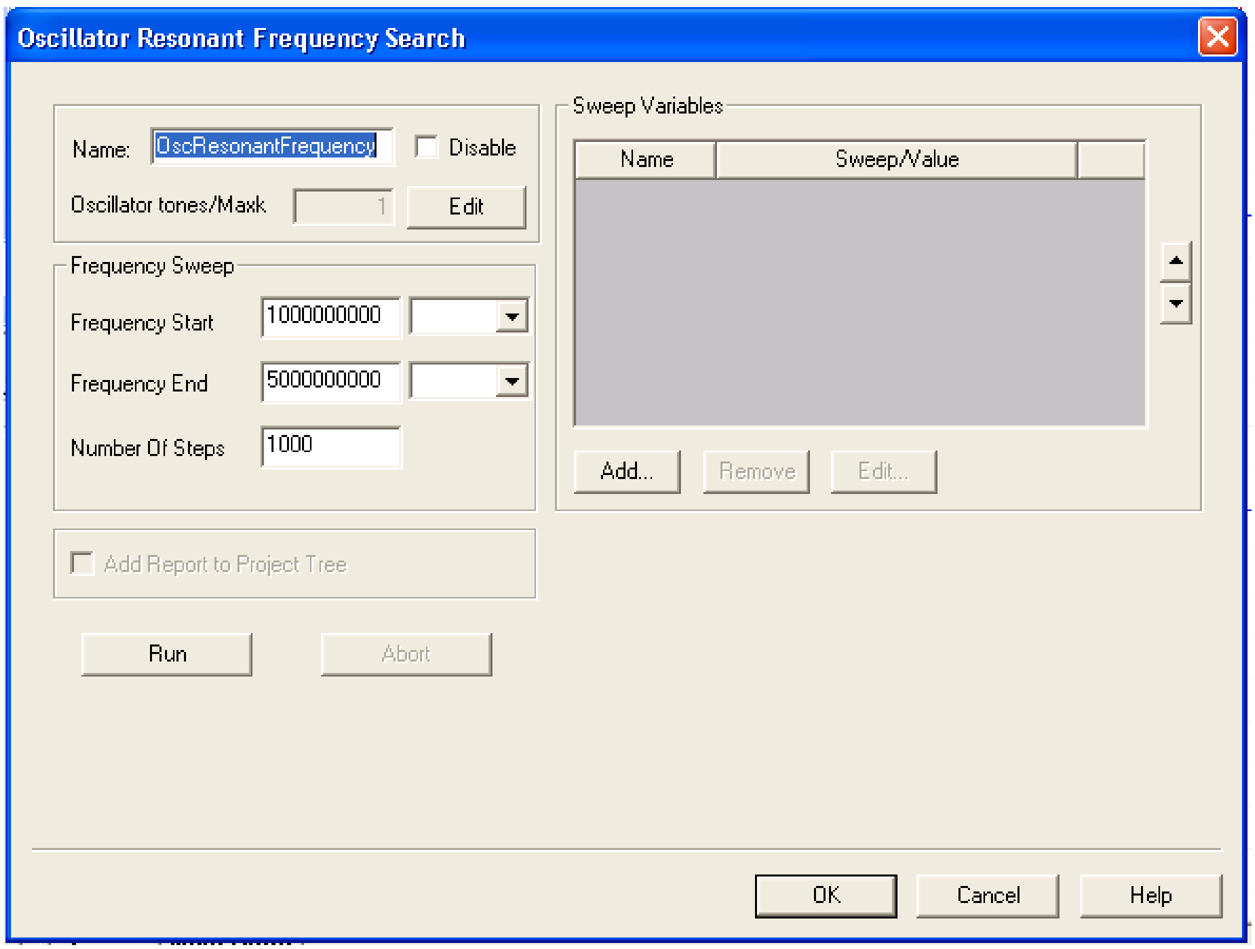
The frequency value is the one calculated for your circuit, and the time and date shows the time of the analysis on your machine.

> **Note:**
>
> As analysis progresses, oscillator analysis may save messages like the following to the log file:
>
> analysis:osc(status): Probe 0: selecting (*node_name1*,*node_name2*) as alternative location 1.
>
> analysis:osc(status): Probe 0: selecting (*node_name3*, *node_name4*) as alternative location 2.
>
> The *node_name*s are circuit-specific. If oscillator analysis fails, Nexxim automatically reruns the analysis after moving the probe to alternative location 1. If that analysis fails, Nexxim retries using alternative location 2.

Running Resonant Frequency Search on the Schematic Editor

The Resonant Frequency Search performs a modified form of oscillator analysis to obtain an initial estimate of the resonant frequency of the circuit. From the **Schematic Editor**, perform the following steps to set up and run an oscillator resonant frequency search, and view the results.

1. Set up the schematic circuit for oscillator analysis by inserting an oscillator probe between two nodes in your circuit. Edit the probe properties and specify an initial guess for the frequency (**FREQ** parameter) and an initial voltage (**A** parameter). See Using the Oscillator Probe for details.
2. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Oscillator Analysis (N-Tone)** to open the **Oscillator Resonant Frequency Search** window.



3. Type an **Analysis Name** (or accept the default name, for example “OscResonantFrequency”).

4. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
5. Click **Edit** to change the number of tones, the frequencies of the tones, and the maximum harmonic numbers to use for each tone. These settings are reflected in the **.OSC** statement in the netlist.
6. Use the fields in the **Frequency Sweep** group box to set the parameters for the initial guess, which always uses the frequency sweep method. The settings in this group box generate **.OPTIONS** statements in the netlist.
7. To add a sweep, use the fields in the **Sweep Variables** group box.
   - From the **Sweep Variable** area, select **Add** to open the **Add/Edit Sweep** window.
   - In the **Variable** list, select Temp or the name of a variable (when a variable has been defined for the design), then select one of the following: **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**.
   - Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
   - For details on sweeps, see Variable Sweep.
   - Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Oscillator Resonant Frequency Search** window.
8. Click **Run** to run the analysis directly on the window.
   - If the analysis is successful, Nexxim displays plots of the real and imaginary parts of the probe current. The estimated resonant frequency occurs where the imaginary part of the probe current is zero. This estimated resonant frequency is displayed in the **Resonant Frequencies** field at the top of the Report.
   - After a successful analysis, check **Add Report to Project Tree** to save the plot under the **Results** icon in the **Project Manager** window. The report is added when the setup window is closed with **Finish**.
9. Click **Finish** to add the solution setup (and the results if selected) to the **Project Manager** window and exit the window. You can then run the analysis again by clicking on it and selecting **Analyze** on the menu.
10. To run the Resonant Frequency Search analysis on the **Project Manager** window:
    - Locate the resonant frequency analysis setup icon under the **Analysis** icon on the **Project Manager** window. Click the resonant frequency analysis setup icon and select **Analyze** on the menu. The analysis begins immediately and a progress bar appears.
    - If the analysis is not successful, an error message appears in the **Message Manager** window. Check the **Message Manager** window for an explanation, then take corrective action.
11. View results:

- After a successful oscillator analysis, Nexxim displays a message like the following in the **Message Manager** window, near the end of the messages:

```
   [info]analysis:osc_analysis(info):
 Best chance for oscillation detected at 2.28514e_006 Hz (9:15 AM
May 20, 2005)
```

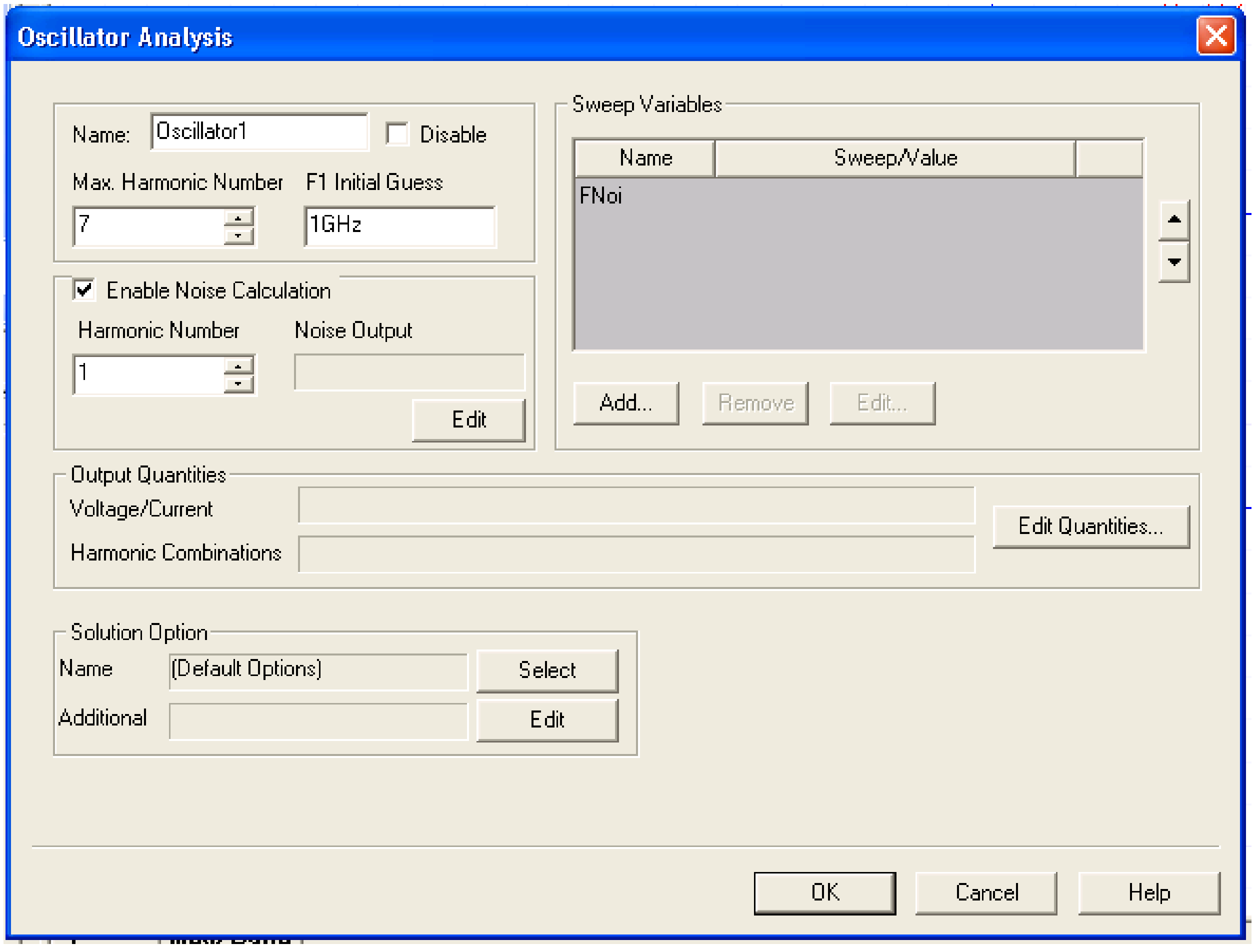
The frequency value is the one calculated for your circuit, and the time and date show the time of the analysis on your machine.

> **Note:**
>
> As analysis progresses, oscillator analysis may save messages like the following to the log file:
>
> analysis:osc(status): Probe 0: selecting (*node_name1*,*node_name2*) as alternative location 1.
>
> analysis:osc(status): Probe 0: selecting (*node_name3*, *node_name4*) as alternative location 2.
>
> The *node_name*s are circuit-specific. If oscillator analysis fails, Nexxim automatically reruns the analysis after moving the probe to alternative location 1. If that analysis fails, Nexxim retries using alternative location 2.

Running Phase Noise Analysis on the Schematic Editor

Nexxim can calculate phase noise as part of oscillator analysis.

Set up the circuit for single-tone or multi-tone oscillator analysis as described in the topics Running Single-Tone Oscillator Analysis on the **Schematic Editor** and Running Multi-Tone Oscillator Analysis on the **Schematic Editor**. Complete the following Phase Noise setup steps before clicking **Finish** on the **Oscillator Analysis** window:



1. From the **Oscillator Analysis** window, select (check) the box for **Enable Noise Calculation**.
2. In the **Noise Calculation** group box, select the **Harmonic Number**. The **Harmonic Number** value selects one of the harmonics of the oscillator frequency around which to calculate the phase noise. The default is harmonic number 1 (the fundamental oscillating frequency).
3. Click **Edit** in the **Noise Calculation** group box to open the **Noise Output** window. Expand the icons for the nets and devices, then click the check boxes to select one or more outputs for noise analysis.
4. In the **Sweep Variables** group box, you can add a sweep of the noise frequencies to be calculated, with or without a second sweep of temperature. The swept frequencies are offsets from the calculated oscillating frequency.

- To add a sweep of noise frequencies, select **FNoi** on the **Variables** group box and click **Edit** to open the **Add/Edit Sweep** window. Select **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**, then complete the specification by filling in the numeric fields. Click **Add** to add the sweep to the **Sweep Values** list.

- To add a sweep of temperature, click **Add** in the **Sweep Variables** group box to open the **Add/Edit Sweep** window. Select **Temp** on the **Variables** group box. Select **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**, then complete the specification by filling in the numeric fields. Click **Add** to add the sweep to the **Sweep Values** group box.
- Click **OK** to close the **Add/Edit Sweep** window and return to the **Oscillator Analysis** window.

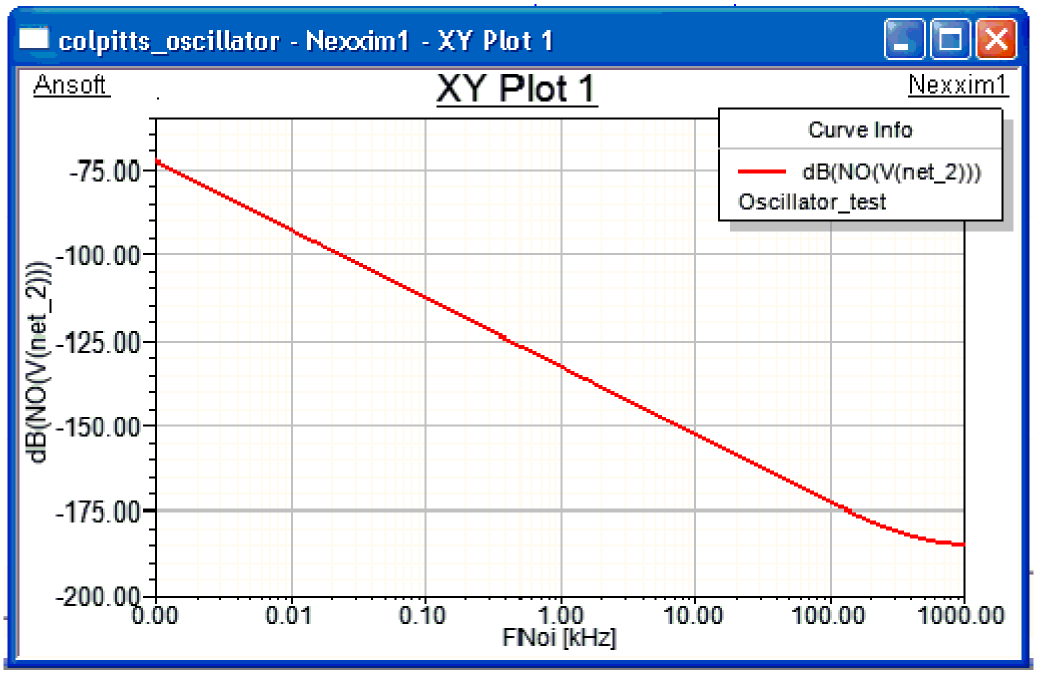
5. When all setup steps are as appropriate, click **Finish** on the **Oscillator Analysis** window. The solution setup is added to the **Project Manager** window under the **Analysis** icon.
6. Run the simulation:

- Click the solution setup in the **Project Manager** window and click **Analyze** on the menu. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
- If the analysis is not successful, an error message appears in the **Message Manager** window. Check the **Message Manager** window for an explanation, then take corrective action.

7. Display results:

- From the **Circuit** menu, select **Create Report** (Alternatively, select **Create Report** on the **Results** icon in the **Project Manager** window.) to open the **Create Report** window. Select Standard as the **Report Type** and Rectangular Plot as the **Display Type**. Click **OK**.
- When the **Report** window opens, select the appropriate oscillator analysis on the **Solution** drop-down. Select **Noise** on the **Domain** drop-down.
- Select the **Noise Output** category. The output quantities you specified in the **Noise Output** window (step 3 above) are available for selection. For most noise analyses, select **dB** as the function. Click **New Report**.
- A Report window opens to display a graph of the analysis results.
- Double-click the X-axis (the **FNoi** axis at the bottom of the report) to open the **X-Axis Properties** window. Select the **Scaling** tab, and Select **Log** for **Axis Scaling**. Click **OK** to close the window.

- The report now shows the phase noise in the common double-log format:



- For details on creating and modifying reports, see Generating Reports and Post-processing.

Running Oscillator Analysis on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist should contain the information for running the oscillator analysis. No Solution Setup is required.

1. Add one oscillator probe to the circuit for each unknown frequency to be calculated. Set the parameters on each probe to specify an initial guess for the unknown frequency (**FREQ** parameter) and an initial voltage to be used in the calculations (**A** parameter). See Using the Oscillator Probe for more information
2. Run the simulation:
   - From the **Circuit** menu, select **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
   - If the analysis is not successful, check the **Message Manager** window for an explanation, and then take corrective action.
3. Display results of oscillator analysis:
   - After a successful oscillator analysis, Nexxim displays a message like the following in the **Message Manager** window, near the end of the messages:

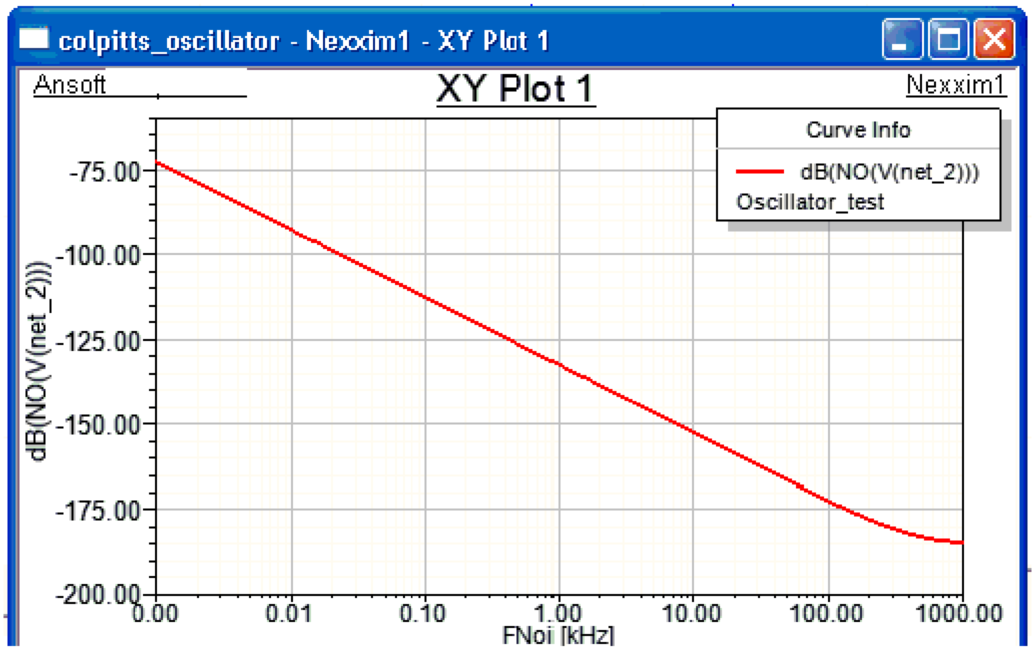
```
[info]analysis:osc_analysis(info):
Lowest oscillation frequency is 2.28514e_006 Hz (9:15 AM May 20,
2005)
```

The frequency value is the one calculated for your circuit, and the time and date shows the time of the analysis on your machine.

> **Note:**
>
> As analysis progresses, oscillator analysis may save messages like the following to the log file:
>
> analysis:osc(status): Probe 0: selecting (*node_name1*,*node_name2*) as alternative location 1.
>
> analysis:osc(status): Probe 0: selecting (*node_name3*, *node_name4*) as alternative location 2.
>
> The *node_name*s are circuit-specific. If oscillator analysis fails, Nexxim automatically reruns the analysis after moving the probe to alternative location 1. If that analysis fails, Nexxim retries using alternative location 2.

4. If the **.OSC** statement in the netlist includes phase noise analysis, perform the following steps to display the results of the phase noise analysis on a graph.
   a. From the **Circuit** menu, select **Create Report**. (Alternatively, select **Create Report** on the **Results** icon in the **Project Manager** window.) to open the **Create Report** window. Select Standard as the **Report Type** and **Rectangular Plot** as the **Display Type**. Click **OK**.
   b. Select the appropriate oscillator analysis on the **Solution** drop-down. Select **Noise** on the **Domain** drop-down.
   c. Select the **Noise Output** category. The output quantities specified in the **NOISE_ OUTPUT** entry in the **.OSC** statement that defined the phase noise analysis are available for selection. For most noise analyses, select **dB** as the function (the report uses $10\log_{10}$(magnitude) on the Y-axis). Click **Add Trace**.
   d. Repeat the previous bullet until all appropriate quantities have been selected, then click **Done** to open a Report window open to display a graph of the analysis results.
   e. Double-click the X-axis (the **Fnoi** axis at the bottom of the report) to open the **X-Axis Properties** window. Select the **Scaling** tab, and click **Log** (deselecting the **Linear** button). Click **OK** to close the window.

- The report now shows the phase noise in the common double-log format:



- If the X-axis units are not as appropriate, double-click the X-axis to open the X-Axis **Properties** window. Select the **Scaling** tab, and deselect the **Autounits** radio button. Select new **Unit**, **Field Width**, and **Precision** values until the X-axis is as appropriate. Click **OK** to close the window.
- For details on creating and modifying reports, see Generating Reports and Post-processing.

Oscillator Analysis Netlist Formats

The following topics provide information about netlist formats and statements related to oscillator analyses.

Oscillator Probe Syntax

To enable oscillator analysis, the circuit must include one oscillator probe device for each unknown frequency to be calculated. The netlist syntax for the oscillator probe device is:

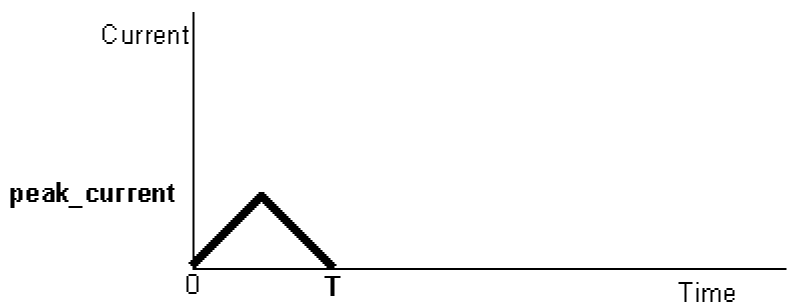
R*xxx n+ n-***OSC_PROBE=1 A=***val***FREQ=***val* [**T=***val* **peak_current=***val*]

The oscillator probe element can have any name beginning with **R**. The **OSC_PROBE** parameter is required to distinguish the oscillator probe from a resistor.

The **FREQ** parameter specifies a frequency to be used as the initial guess for the unknown oscillation. The **A** parameter specifies a voltage to be used in the calculations.

The probe behaves like an oscillating voltage source with amplitude given by the parameter **A** at the frequency given by the parameter **FREQ**. At all other frequencies, the probe current is zero. The default for parameter **A** is 10mV. The default for parameter **FREQ** is 1e6Hz.

Parameters **T** and **peak_current** are specific to transient analysis, and need to be set only when an oscillator circuit does not begin oscillation during time-domain analysis. Choosing a non-zero **peak_current** (Amperes) and positive **T** (seconds) adds a temporary triangular noise current source with a peak value of **peak_current** and duration **T**, starting at time = 0.



The PWL current source acts as extra noise which helps the oscillation to start and reach steady state soon. This current source does not affect the period of oscillation or any other quantity of interest in an oscillator analysis.

.OSC Single-Tone Statement

Oscillator analysis is invoked on the netlist with a **.OSC** statement. The single-tone syntax is used when the circuit has only DC input sources.

.OSC TONES=*val***MAXK=***val*

The **TONES** value in the **OSC** statement specifies an initial estimate of the resonant frequency to find in the analysis. The **TONES** frequency must equal the value of the **FREQ** parameter on the oscillator probe. The **TONES** value overrides the probe **FREQ** parameter if they are not the same.

The **MAXK** value specifies the highest harmonic to include in the analysis.

The entry **INITIAL_GUESS_ONLY=YES** limits the calculation to the initial guess phase. When this entry is present, the second phase with harmonic balance analysis is not performed. By default, or when the entry **INITIAL_GUESS_ONLY=NO** is present, both phases of calculation are performed.

Here is an example of a single-tone oscillator analysis netlist:

```
*Single-tone oscillator analysis
.param freq_guess=2e6
//
// circuit details not shown
//
Rosc_probe 2 3 osc_probe=1 A=0.001 FREQ=freq_guess
.OSC TONES=freq_guess MAXK=31
.END
```

.OSC Multi-Tone Statement

The multi-tone syntax for the **OSC** statement is used to analyze a circuit with multiple oscillating frequencies:

- One unknown resonant frequency (F_1) plus one or more known driven oscillations (RF_1 through RF_M) such as sinusoidal voltage sources. The circuit must contain one oscillator probe for the unknown resonance.
- Two or more unknown resonant frequencies (F_1 through F_N) with no driven oscillations in the circuit. The circuit must contain one oscillator probe for *each* of the unknown resonances.
- Two or more unknown resonant frequencies (F_1 through F_N) plus one or more known driven oscillations (RF_1 through RF_M). The circuit must contain one oscillator probe for *each* of the unknown resonances.

An example is a mixer circuit connecting one or more voltage-controlled oscillators (VCOs) of unknown frequencies with one or more RF sources of known frequencies to produce a single resultant IF output.

This form uses parentheses to enclose a comma-separated list of tone frequencies and the corresponding comma-separated list of harmonics:

```
.OSC TONES=[F1 [, ...FN] [, RF1 [, ...RFM]]]
MAXK=[maxkF1 [, ...maxkFN], [maxkRF1 [, ...maxkRFM]]
```

The left and right brackets are required in the multi-tone form of the **OSC** statement.

The first *N* of the **TONES** frequencies (*F1* through *FN* in the syntax above) must equal the **FREQ** parameter settings of the *N* oscillator probes in the circuit. These frequencies are the initial estimates of the unknown resonant frequencies for the circuit, and the corresponding **MAXK** entries (*maxkF1* through *maxkFN*) are the highest harmonics to calculate for the corresponding frequencies.

After the list of unknown frequency estimates, the **TONES** list should have entries for the *M* known, driven frequency sources in the circuit (*RF1* through *RFM* in the syntax above). The corresponding **MAXK** entries (*maxkRF1* through *maxkRFM*) are the highest harmonics to calculate for the corresponding driven oscillations.

The number of oscillator probes in the netlist determines the number of **TONES** frequencies that are treated as estimates of the unknown frequencies to be calculated. The remaining frequencies in the **TONES** group box are treated as the fixed frequencies of the driven oscillations.

The frequency estimates in the **TONES** group box should be the same as the ones entered in the **FREQ** parameters of the oscillator probes. The **TONES** values override the probe **FREQ** parameters if they are not the same.

Here is an example of a netlist using the multi-tone oscillator analysis syntax:

```
*Oscillator with mixer and driver
\\
\\ circuit details not shown
\\
.param freq_guess=2e6, RF_freq=0.5e6
Rosc_probe 2 3 osc_probe=1 A=0.001 FREQ=freq_guess
.OSC TONES=[freq_guess,RF_freq] MAXK=[31,3]
.END
```

.OSC Resonant Frequency Search Statement

The resonant frequency search syntax is used when only an initial estimate of the oscillating frequency is appropriate. The oscillator analysis line for resonant frequency search is:

```
.OSC TONES=valMAXK=val INITIAL_GUESS_ONLY=YES
```

The **TONES** and **MAXK** entries can use either the single-tone format or the multi-tone format. See .OSC Single-Tone Statement and .OSC Multi-Tone Statement for further information.

The entry **INITIAL_GUESS_ONLY=YES** limits the calculation to the initial guess phase. When this entry is present, the second phase with harmonic balance analysis is not performed. By default, the entry **INITIAL_GUESS_ONLY=NO** is present, and both phases of calculation are performed.

The netlist must also contain the following options:

```
.OPTION initial_guess=freq_sweep
.OPTION initial_guess.freq_sweep.start=val
.OPTION initial_guess.freq_sweep.stop=val
```

These options select a frequency sweep for the initial guess at the resonant frequency.

A fourth option may also be used:

.OPTION initial_guess.freq_sweep.num_steps=*val*

This option sets the number of steps to use in the frequency sweep. The default is 50 steps.

Here is an example using the resonant frequency search with two-tone oscillator analysis syntax:

```
*Resonant frequency search with mixer and driver
\\
\\ circuit details not shown
.param freq_guess=2e6, RF_freq=0.5e6
rosc_probe 2 3 osc_probe=1 A=0.001 FREQ=freq_guess
.option osc.initial_guess=freq_sweep
.option osc.initial_guess.freq_sweep.start=freq_guess
.option osc.initial_guess.freq_sweep.stop='4*freq_guess'
.OSC TONES=[freq_guess,RF_freq] MAXK=[31,4]
+ INITIAL_GUESS_ONLY=YES
.END
```

.OSC Phase Noise Analysis Statement

The Nexxim netlist syntax for phase noise analysis consists of extensions to the basic oscillator analysis.

.OSC TONES=*val***MAXK=***val*
NOISE_OUTPUT=[*outputs***] HARMONIC_NUMBER=***val frequency_sweep*

The **TONES** and **MAXK** entries can use either the single-tone format or the multi-tone format. See .OSC Single-Tone Statement and .OSC Multi-Tone Statement for further information.

The **NOISE_OUTPUT** parameter specifies the noise outputs of interest. A noise output can be:

- The voltage at a single node, format **V(***node***)**, for example, V(n1).
- The voltage differential between two nodes, format **V(***node,node***)**, for example, V(n1,n2).
- The current across a branch, format **I(***branch***)**, for example, I(V1)

The **HARMONIC_NUMBER** parameter selects one of the harmonics of the oscillator frequency around which to calculate the phase noise. The default is harmonic number 1 (the fundamental oscillating frequency).

The entry *frequency_sweep* specifies a range of frequencies (offsets on the frequency specified by the **HARMONIC_NUMBER** parameter) to be applied in the analysis.

The *frequency_sweep* field has several formats.

The sweep can be just a single frequency value.

A second form of *frequency_sweep* specifies the starting and ending frequencies, and the incremental step:

```
[START=]start_val [STOP=]stop_val [STEP=]step_val
```

If the sweep specification omits the **START=**, **STOP=**, and **STEP=** labels, the start value, stop value and step value must be entered in the order shown. If the labels are used, the values may be entered in any order, but then all three values must have labels.

A third form of the *frequency_sweep* field has the format:

```
freq_count_type numpoints startfreq stopfreq
```

This format specifies a set of analysis frequencies defined by *numpoints*, *startfreq* and *stopfreq*, using one of the predefined *freq_count_type* tokens **LIN**, **DEC**, or **OCT**:

LIN
Linear count sweep at *numpoints* different frequencies on the frequency given by *startfreq* to the frequency given by *stopfreq*. The spacing between test frequencies is

(*stopfreq*–*startfreq*) / (*numpoints*– 1).

DEC
Logarithmic count sweep at *numpoints* different frequencies in each decade (10× multiple of *startfreq*) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within decades.

OCT
Logarithmic count sweep at *numpoints* different frequencies in each octave (2× multiple of *startfreq*) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within octaves.

A fourth form of *frequency_sweep* has the format:

```
POInumpoints freq1 [freq2 ...]
```

This form for *frequency_sweep* provides a space-separated list of *numpoints* test frequencies.

The phase noise analysis can also include a sweep of temperature or any other parameter.

See Variable Sweep for details.

Here are examples of .OSC Phase Noise statements:

```
.OSC TONES=1E10 MAXK=3 NOISE_OUTPUT=[V(1,2),V(NET_3),I(VD)]
+ HARMONIC_NUMBER=2 LIN 100 0.1 2E10
```

```
.OSC TONES=1E6 MAXK=3 NOISE_OUTPUT=[V(NET_3),I(VD)]
+ HARMONIC_NUMBER=1 LIN 10 0.5e6 2E6
+ SWEEP TEMP POI 5 15 20 25 30 35
```

Oscillator Analysis Options

The oscillator analysis is based on harmonic balance, so all the harmonic balance analysis options are valid. (See Harmonic Balance Options for details on these options.)

Nexxim oscillator analysis has its own options, listed in the Oscillator and Phase Noise Options Reference. The default values of these options can be changed with a .**OPTIONS** statement. The syntax is:

```
.OPTIONSosc.option1=value1 [[osc.]option2=value2 ... ]
```

The prefix **osc.** applies an option only to oscillator analysis but not to other tools that may recognize the same option.

For subordinate options that are available when an option such as **osc.initial_guess=transient_init** is in effect, the netlist uses periods to separate the option fields (e.g., **osc.initial_guess.transient_init.tmax**).

See Controlling Nexxim Oscillator Analysis for a discussion of the effects of changing the oscillator analysis options.

Oscillator Analysis Outputs

After a successful oscillator analysis, Nexxim displays a message like the following in the **Message Manager** window:

```
[info]analysis:osc_analysis(info):
Lowest oscillation frequency is 2.28514e_006 Hz (9:15 AM May 20,
2017)
```

The frequency value is the one calculated for your circuit, and the time and date shows the time of the analysis on your machine.

The outputs on the harmonic balance analysis are also available for report generation. See Harmonic Balance Outputs for further information.

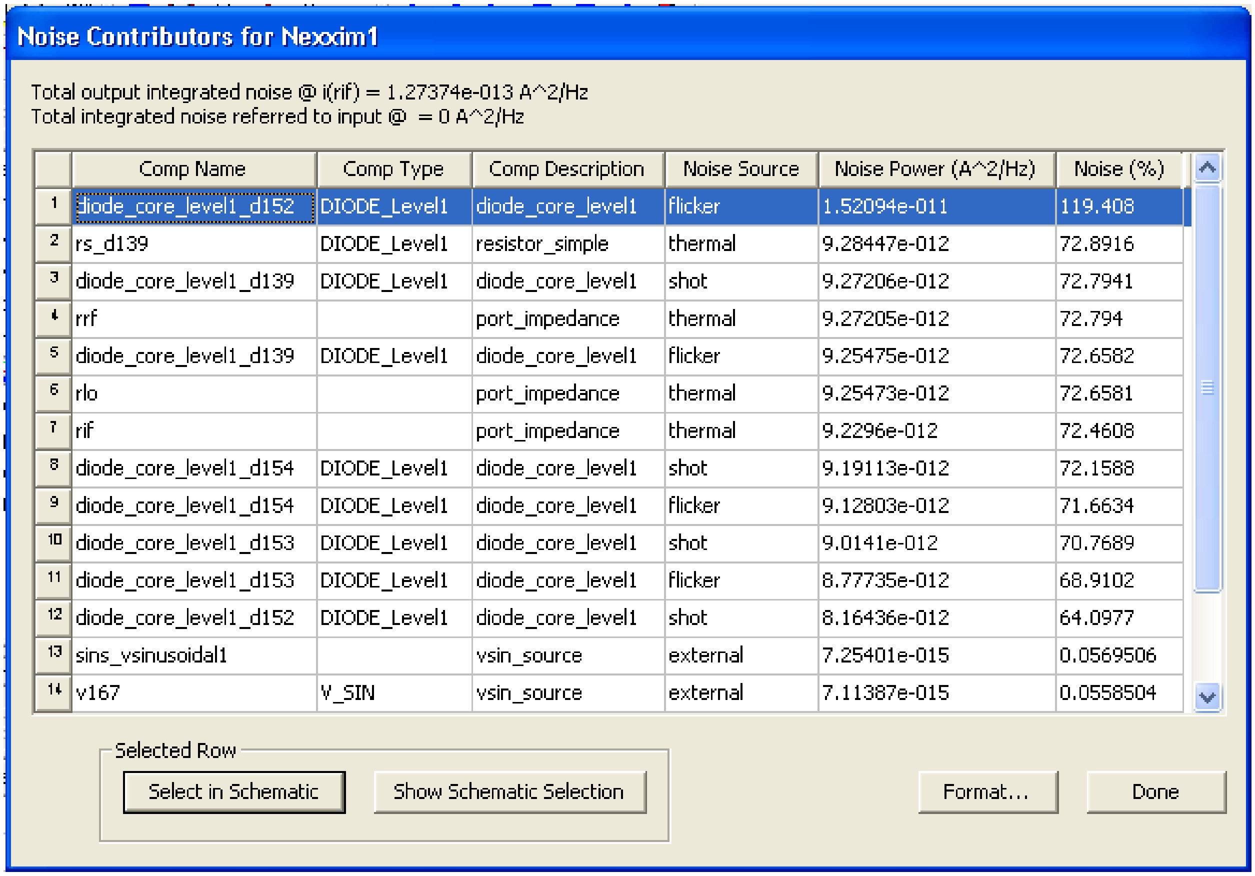
Viewing OSC Noise Contributors

When noise analysis has been specified in the solution setup or in the netlist, view a listing of noise contributors after a successful simulation. Noise contributors are available for Linear Network Analysis (LNA), Time-Varying Noise Analysis (TVNoise) and Oscillator Analysis (OSC).

To enable this feature, the **Oscillator Analysis** setup in a schematic design must have **Noise Analysis** enabled, and must specify at least one **Noise Output** device (node voltage or branch current). In a netlist design, the netlist must include the **NOISE_OUTPUT** entry in the .OSC statement.

- For schematic designs, right-click the solution setup icon in the **Project Manager** window and select **Show Noise Contributors** on the menu.
- For netlist designs, right-click the circuit name icon in the **Project Manager** window and select **Show Noise Contributors** from the menu.

The **Noise Contributors** window opens:



Initially, the noise contributors are listed in order by **Component Name**. (If the window is opened before an analysis has been performed, it is empty.)

- Click in a column header to sort the display by the contents of that column (the example above has been sorted by **Noise (%)**).

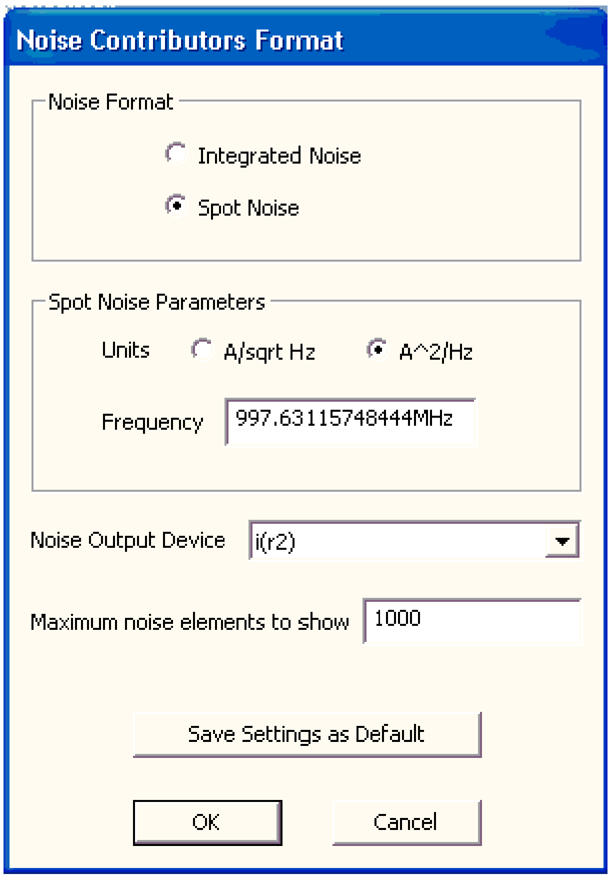
In a schematic design, the **Selected Row** group box is enabled.

- Select a contributor (row) and click **Select in schematic**. The contributing element is selected in the schematic window. The schematic window is brought to the front (under the **Noise Contributors** window) and scrolled so the selected object is visible. Only one element can be selected at a time.

- Select an object in the schematic window and click **Show schematic selection**. The corresponding contributing element is selected in the **Noise Contributors** window. The row containing that element is scrolled into view if necessary.

Initially, the display is set up to show spot noise contributors using default settings.

- Click **Format** to open the **Noise Contributors Format** window:



Choose one noise output device (voltage through node or current through branch) on the **Noise Output Device** field drop-down menu. The list is populated on the output devices selected in the **Oscillator Analysis** setup or in the **NOISE_OUTPUT** entry on the .OSC statement.

Enter a value in the **Maximum noise elements to show** field to restrict the display to that number of elements with the most noise contributions.

When **Spot Noise** is selected, the **Spot Noise Parameters** group box is activated.

- The **Units** button settings depend on the **Noise Output Device**:

When the device is a branch current (as in the illustration above), the choice of units is between Amps/(Hz)^{½} and (Amps)^2/Hz.

When the device is a node voltage, the choice of units is between Volts/(Hz)$^{1/2}$ and (Volts)^2/Hz.

- The **Frequency** field is initialized to a value drawn on the analysis setup. Change the frequency as appropriate to see how the spot noise contributions vary with frequency.

When **Integrated Noise** is selected, the **Spot Noise Parameters** group box is deactivated. Units are Volts for output device node voltages and Amperes for output device branch currents.

- Click **Save settings as default** to save the **Noise Contributors Format** window settings as the defaults for future displays.
- Click **OK** to close the **Noise Contributors Format** window and return to the **Noise Contributors** window.

Click **Done** to close the **Noise Contributors** window.

The **Noise Contributors** window becomes invalid if the design is modified. If the window is open when the analysis is re-run, the window is refreshed.

Multiple **Noise Contributor** windows may be opened, but only one for a given design.

Nexxim Envelope Analysis

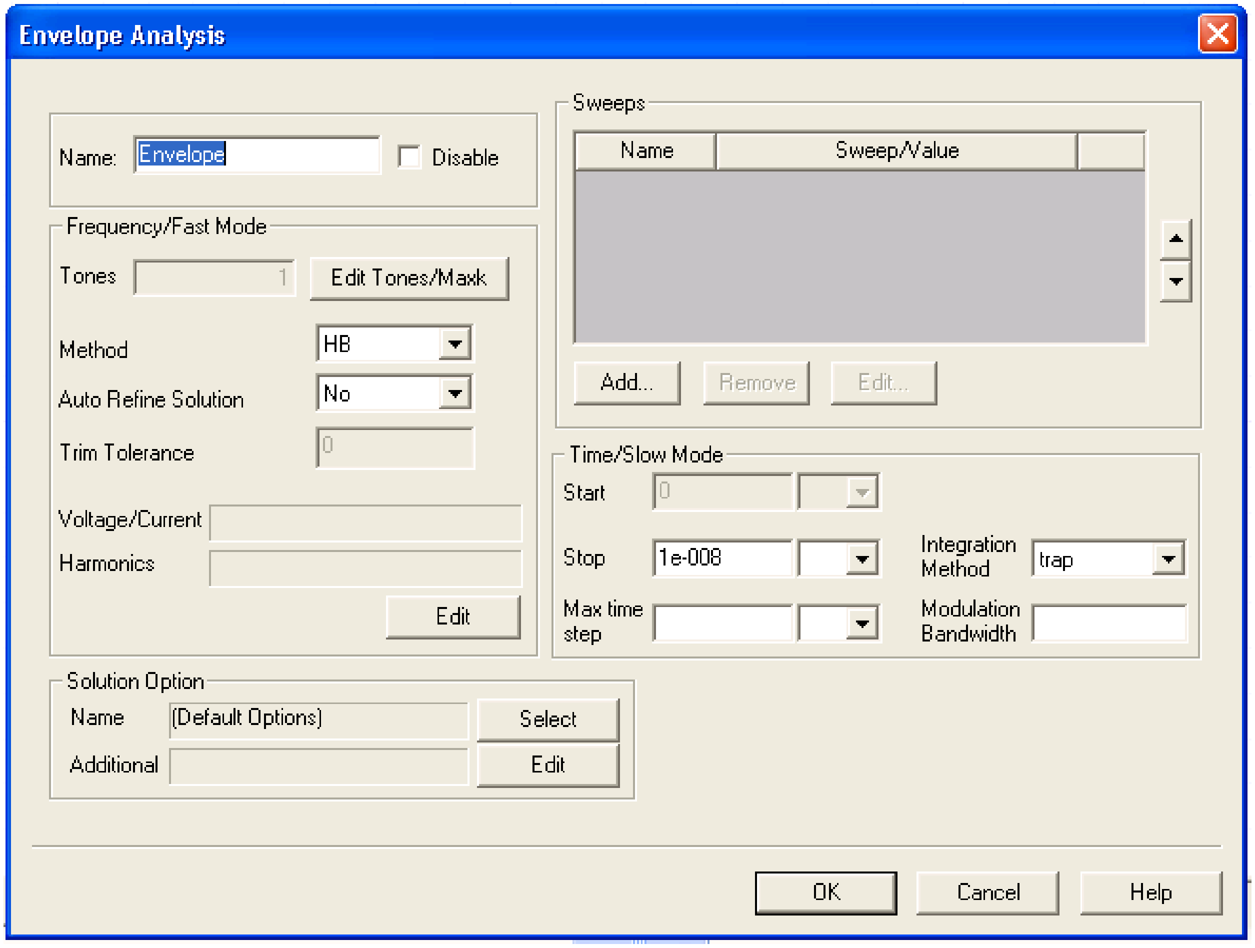
Envelope analysis is commonly used to analyze systems where harmonic balance or transient analysis alone is not adequate. Such systems include circuits with two inputs, where one input is a fast-changing periodic or quasi-periodic source such as a clock or Local Oscillator (LO) and the other input is a non-periodic source such as a baseband RF modulator that changes on a timescale that is orders of magnitude slower than the timescale of the fast-changing input. Transient analysis requires a small timestep to capture the fast-changing input, but then requires a very large number of timesteps to simulate the slowly changing non-periodic input. Harmonic balance fails to analyze the non-periodic input.

Envelope analysis uses transient analysis to simulate the slowly moving signal in the time domain, plus harmonic balance to analyze the fast-moving signal in the frequency domain. The time-domain analysis can use a varying timestep that is appropriate to the slowly changing waveform. At each timestep of the transient analysis, an HB analysis is run. The frequency coefficients at each timestep are stored and returned as the result. From these coefficients, obtain a variety of results including a transient-like time-domain result.

Running Nexxim Envelope Analysis on the Schematic Editor

From the **Schematic Editor**, perform the following steps to set up and run an envelope analysis using Nexxim.

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Envelope Analysis** to open the **Envelope Analysis** window.



3. Type an **Analysis Name** (or accept the default name, for example “Envelope”).
4. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
5. The **Frequency/Fast Mode** group box controls the harmonic balance analysis.
   - Click **Edit Tones/Maxk** to specify the periodic, fast-changing inputs for harmonic balance analysis. On the **Edit Tones/Max Harmonic Number** window, enter the number of inputs in the **No. of Tones** field. The field changes to reflect the number of frequencies specified. The names of the frequencies are F1, F2, etc. and cannot be changed. Click the **Value** field for a frequency to enter the input frequency. Click the **MAXK** field for a frequency to enter the appropriate maximum harmonic number. Click **OK** to close the window and return to the **Envelope Analysis** window.
   - When one-tone analysis has been specified, the **Method** field is activated. Use this field to select **HB** or **shooting** (HB is the default). See Handling Strongly Nonlinear Circuits in the HB topic for details on the **Method** entry.

- Optionally, Click the **Auto Refine Solution** field to select Yes or No (No is the default). See Increasing Accuracy or Speed in the HB topic for details on the **Auto Refine Solution** entry.
- When multi-tone analysis has been specified, the **Trim Tolerance** field is activated. Use this field to set a tolerance value. See Increasing Accuracy or Speed in the HB topic for details on the **Trim Tolerance** entry.
- Optionally, click **Edit** to open the **Output** window. Click the check boxes on the **Output** window to add net voltages, branch currents, and harmonics to the data generated by the envelope analysis. Click **OK** to add the selected output values and return to the **Envelope Analysis** window.

6. The Time/Slow mode group box controls the transient analysis.

- Set the **Stop** field to the stop time for the transient analysis. The **Start** time is always zero.
- Optionally, set the **Maximum timestep** field to suggest a maximum transient step size.
- Optionally, select a transient analysis integration method on the **Integration Method** drop-down. **TRAP** (trapezoidal rule) is the default, **BDF2** (Gear) is the alternative.
- Optionally, set the **Modulation Bandwidth** field to an estimate of the bandwidth of the RF modulation input.

7. Optionally, use the fields in the **Solution Options** group box to select or add Harmonic Balance analysis options and other Nexxim options to the design. Envelope analysis uses the HB options.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a previously defined option set to be used in the current analysis, click the name of the option set, then click **OK** to return to the **Envelope Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Select the **HB Options** tab. Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **Envelope Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

- The line can contain multiple options settings. Use spaces to separate the options settings.
- To modify or delete an option once it has been added, just edit or delete its entry in the list.
- To clear all options, delete the entire line.

Click **OK** to return to the **Envelope Analysis** window.

The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

- For more information, see Nexxim Envelope Analysis Options.

8. Run the simulation:

- From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
- If the analysis is not successful, check the **Message Manager** window for an explanation, and then take corrective action.
- For details on creating and modifying reports, see Generating Reports and Post-processing.

Running an Envelope Analysis on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist should contain the information for running the envelope analysis. No Solution Setup is required.

Run Nexxim Envelope Analysis

1. From the **Circuit** menu, select **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
2. The **Message Manager** window signals success or failure.

- For details on creating and modifying reports, see Generating Reports and Post-processing.

Nexxim Envelope Analysis Netlist Format

Envelope analysis is invoked on the netlist using a variation of the harmonic balance (**.HB**) statement:

```
.HB TONES=(f1 [, ...fN]) MAXK=(M1 [, ...MN]) STOP=val
 [STEP=val] [ENVMETHOD=TRAP|BDF2] [MODULATIONBW=val]
[METHOD=HB|SHOOTING] [AUTO_REFINE_SOLUTION=YES|NO]
[TRIM_TOL=val]
```

- The **TONES** group box specifies the estimated frequency or frequencies of the periodic inputs to the circuit for harmonic balance analysis.
- Each **MAXK** entry M_i specifies the maximum harmonic number to use for the corresponding tone f_i. The harmonic balance analysis is performed using harmonics 1 through M_1 of tone f_1 (frequency f_1 itself is harmonic 1), harmonics 1 through M_2 of tone f_2, up to harmonics 1 through M_N of tone f_N.

- The **STOP** entry specifies the stop time in seconds for the transient analysis. The **STOP** entry is required to distinguish an envelope analysis statement from an ordinary harmonic balance statement. All transient analyses start at time 0.
- The optional **STEP** entry affects the maximum time step used in the transient analysis. See the Transient Analysis topic for details on the timestep calculation.
- The optional **ENVMETHOD** entry selects the integration method to be used in the transient analysis. The choices are **TRAP** (trapezoidal rule) and **BDF2** (Second Order Backward Difference Formula, also called Gear.) The default is **TRAP**. See the Transient Analysis topic for details on integration methods.

- The optional **MODULATIONBW** entry is an estimate of the bandwidth of the non-periodic RF modulation input signal.

Harmonic Balance Controls

The Harmonic Balance controls **METHOD**, **AUTO_REFINE_SOLUTION**, and **TRIM_TOL** can be used to refine the HB phase of the analysis.

See Handling Strongly Nonlinear Circuits in the Harmonic Balance topic for details on the **METHOD** entry in the HB statement.

See Increasing Accuracy or Speed for details on the **AUTO_REFINE_SOLUTION** and **TRIM_TOL** entries in the HB statement.

Nexxim Envelope Analysis Options

Envelope analysis options Click the **.hb** prefix. See Envelope Analysis Options Reference for further information. Envelope analysis also uses the "Harmonic Balance Options" on page 12-15, to control both the harmonic balance analysis and the transient analysis.

The default values of these options can be changed in the netlist with one or more .OPTIONS statement. The syntax is:

```
.OPTIONS hb.option1=value1 [... ]
```

Nexxim Envelope Analysis Outputs

To open the results of an envelope analysis, click on the Results icon in the **Project Manager** window and select **Create Report** to open the **Create Report** window. There are several ways to display the data.

To plot data as an eye diagram:

1. Select **Eye Diagram** as the Report Type and **Rectangular** as the Display Type on the **Create Report** window. Click **OK** to close the **Create Report** window and open the **Report** window.

2. Make a selection on the Solution drop-down menu. The Domain is always **Time**.
3. Select one or more quantities on the **IQ Voltage** or **IQ Current** listing, then click **Add Trace**.
4. Click **Done** to close the **Report** and open a new window opens with a graph of the results.

To plot data as a constellation chart:

1. Select **Constellation** as the Report Type and **Rectangular** as the Display Type on the **Create Report** window. Click **OK** to close the **Create Report** window and open the **Report** window.
2. Make a selection on the Solution drop-down menu. The Domain is always **Time**.
3. Select one or more quantities on the **IQ Voltage** or **IQ Current** listing, then click **Add Trace**.
4. Click **Done** to close the **Report** window. The Report window opens with the graph of the results.

To plot a normal voltage or current (not I or Q) against time:

1. Select **Standard** as the Report Type and **Rectangular** as the Display Type on the **Create Report** window. Click **OK** to close the **Create Report** window and open the **Report** window.
2. Make a selection on the Solution drop-down menu and select **Time** as the Domain. This domain plots one HB result per envelope time point.
3. Select one or more quantities on the **Voltage** or **Current** listing, then click **Add Trace**.
4. Click **Done** to close the **Report** window. The Report window opens with the graph of the results.

To plot a normal voltage or current (not I or Q) against transient time:

1. Select **Standard** as the Report Type and **Rectangular** as the Display Type on the **Create Report** window. Click **OK** to close the **Create Report** window and open the **Report** window.
2. Make a selection on the Solution drop-down menu and select **Transient-Time** as the Domain. This domain plots using all the transient time points.
3. Select one or more quantities on the **Voltage** or **Current** listing, then click **Add Trace**.
4. Click **Done** to close the **Report** window. The Report window opens with the graph of the results.

To view the amplitude of a particular frequency over time:

1. Select **Standard** as the Report Type and **Rectangular** as the Display Type on the **Create Report** window. Click **OK** to close the **Create Report** window and open the **Report** window.
2. Make a selection on the Solution drop-down menu and select **Time** as the Domain.
3. Select one or more quantities from any category, then click **Add Trace**.

4. Click **Done** to close the **Report** window. The Report window opens with the graph of the results.

To view the spectrum of a particular signal or node:

1. Select **Standard** as the Report Type and **Rectangular** as the Display Type on the **Create Report** window. Click **OK** to close the **Create Report** window and open the **Report** window.
2. Make a selection on the Solution drop-down menu and select **Spectral** as the Domain.
3. Select one or more quantities on the **Voltage** or **Current** listing, then click **Add Trace**.
4. Click **Done** to close the **Report** window. The Report window opens with the graph of the results.

Nexxim Time-Varying Noise Analysis

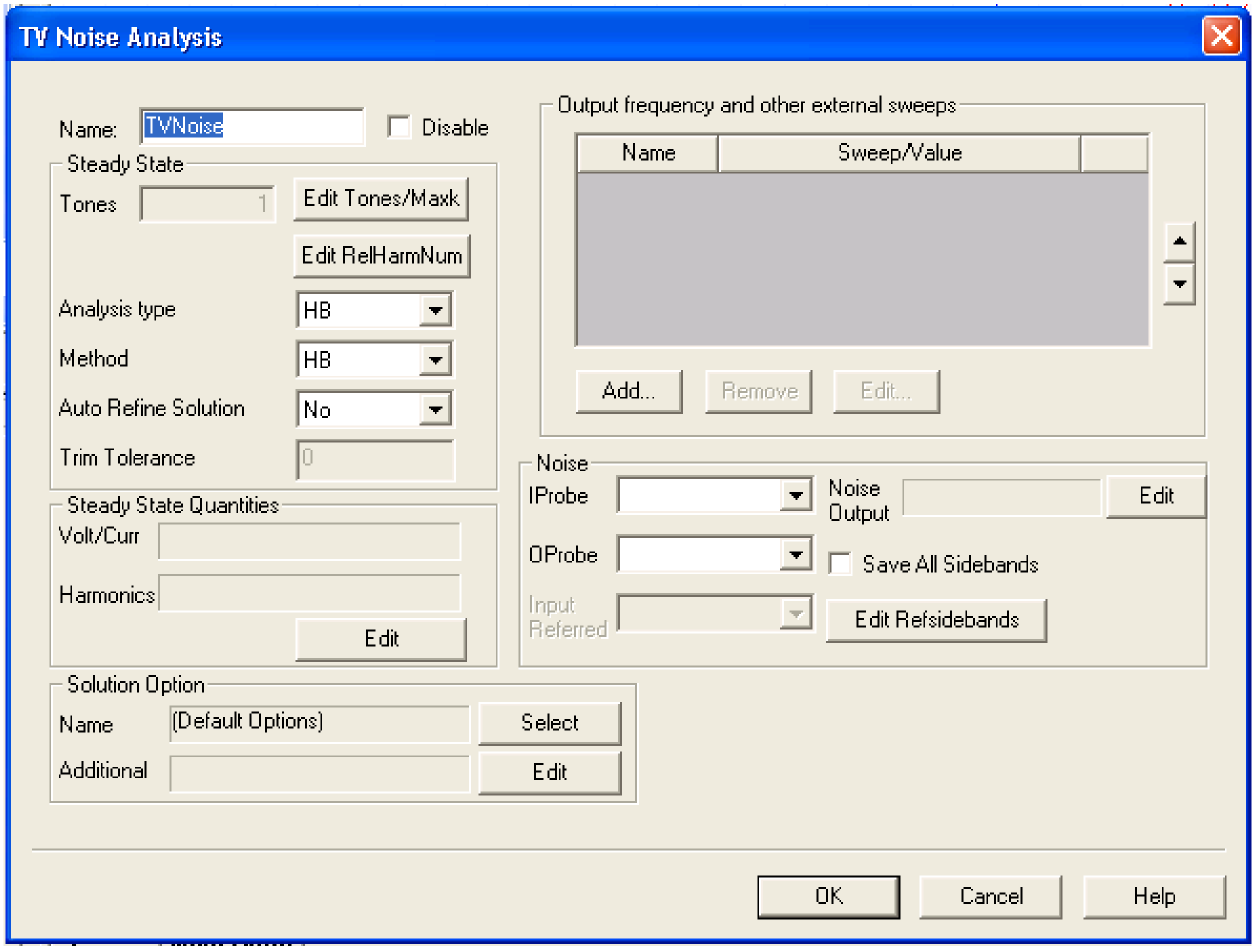
Time-varying noise analysis (TV Noise) calculates the response of a circuit to thermal, shot, and flicker noise sources analyzed as small-signal AC perturbations around the periodic steady-state operating point calculated by harmonic balance or oscillator analysis. The circuit elements are linearized by a DC operating point calculation prior to the steady-state analysis.

The TV Noise analysis can be extended to include periodic transfer function analysis. Periodic transfer function analysis computes the small-signal transfer function from multiple input sources at multiple frequencies to one output at one frequency, or using the sweep of output frequencies on the TV noise analysis setup. A typical application for periodic transfer function analysis is to determine image rejection.

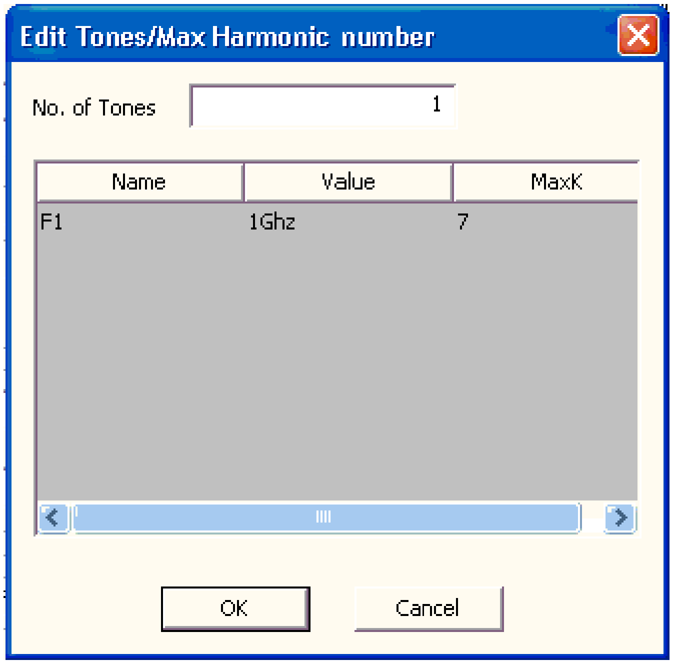
Running TV Noise Analysis on the Schematic Editor

From the **Schematic Editor**, perform the following steps to set up and run a Time-Varying Noise analysis using Nexxim.

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **TV Noise Analysis** to open the **TV Noise Analysis** window.

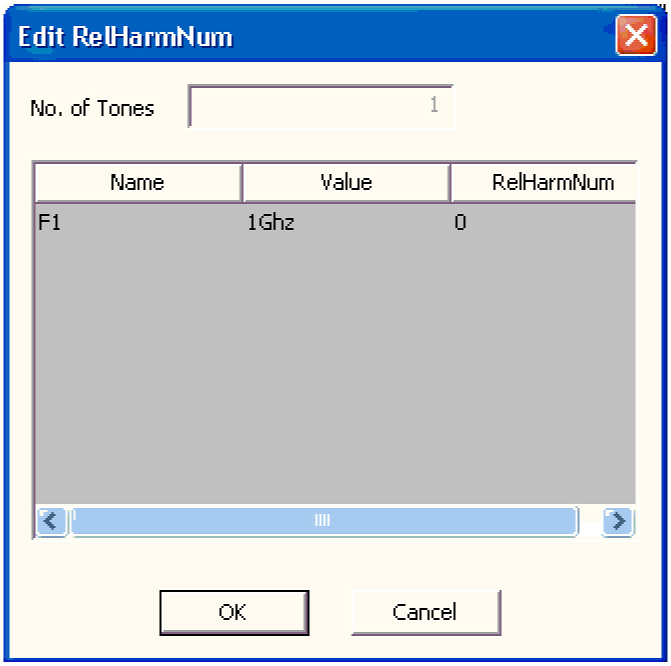


2. Type an **Analysis Name** (or accept the default name, for example “TVNoise”).
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
4. In the **Steady State** group box, click **Edit Tones/MaxK**. The **Edit Tones/Max Harmonic number** window opens:



- Enter the number of tones to be used in the steady-state phase of the analysis. The tones are listed with names **F1**... **F*n***. These names cannot be changed.
- Click a **Value** field in the row for one of the tones to enter the frequency for the steady-state analysis for that tone.
- Click a **MaxK** field in the row for one of the tones to enter the **Max. Harmonic Number** for the steady-state analysis for that tone.
- Click **OK** to close the **Edit Tones/Max Harmonic Number** window and return to the **TV Noise Analysis** window. The **Number of tones** field now shows the number of tones you specified in the window.

5. Optionally, click **Edit RelHarmNum** to open the corresponding window:



- The **No. of tones**, **Name**, and **Value** fields reflect the settings on the **Edit Tones/Maxk** window. The number, name, and value of tones cannot be changed in the **Edit RelHarmNum** window.
- Click in the **RelHarmNum** field for a frequency to set the relative harmonic number offset for that tone. See the TV Noise Technical Notes topic for information on the use of the **RelHarmNum** offsets.

Click **OK** to return to the **TV Noise Analysis** window.

6. Select the **Analysis type** on the drop-down. The default is harmonic balance (**HB**). Choose oscillator analysis (**OSC**) instead.
   - When **HB** is the **Analysis type** and one-tone analysis has been specified, the **Method** field is activated. Use this field to select **HB** or **shooting** (HB is the default). See Handling Strongly Nonlinear Circuits in the HB topic for details on the **Method** entry.
   - When **HB** is the **Analysis type**, the **Auto Refine Solution** field is activated. Use this field to select Yes or No (No is the default). See Increasing Accuracy or Speed in the HB topic for details on the **Auto Refine Solution** entry.
   - When **HB** is the **Analysis type** and multi-tone analysis has been specified, the **Trim Tolerance** field is activated. Use this field to set a tolerance value. See Increasing Accuracy or Speed in the HB topic for details on the **Trim Tolerance** entry.
7. Optionally, click **Edit** at the lower-right of the **Steady State** group box to add voltages, currents, and harmonic outputs on the underlying analysis (HB or OSC). These additional

output quantities are available in the Report window after simulation has been performed.

Expand the **Nets** and devices icons to display lists of the available quantities. To have Nexxim calculate outputs on the steady state analysis step, click the check boxes to select the outputs.

Expand the **Harmonics** icon to display a list of the available harmonics. To specify particular harmonics for Nexxim to use when calculating outputs on the steady state analysis step, click the check boxes to select the harmonics. If you do not select any harmonics, harmonics DC, F1, 2F1, 3F, and 4F1 are automatically selected. In the **Report** window, only selected harmonics are available for plotting.

Click **OK** to return to the **TV Noise Analysis** window.

8. TV Noise analysis requires a sweep of input frequencies. Click the **F** entry in the **Sweep Variables** field and click **Edit** (or **Add**) to open the **Add/Edit Sweep** window.
   - Select one of the following: **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**.
   - Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
   - Click **Add**, then click **OK** to close the window and reopen the **TV Noise Analysis** window. The sweep definition appears in the **Sweep Variables** field. (The **Sweep Variables** display includes fields labeled **Offset of F1** and **Sync**. These fields are not used by Nexxim TV Noise analysis.)
9. In the **Noise** group box, use the drop-down menu in the **IProbe** field to select a port, current source, or voltage source for the small-signal input to the circuit. If a port is selected, Nexxim computes the noise figure and conversion gain. For current and voltage sources, Nexxim computes the gain, but not the noise figure.

> **Note:** The port definition window has fields to specify noise parameters (See Editing an Interface Port Definition.) If the Input Port is defined with noise parameters (in particular, with a Noise Temperature), the analysis uses the temperature value specified. If these parameters are not in the definition, Nexxim adds thermal noise to the specified port for computing the noise factor and conversion gain with the formula 4KT/R, where K is Boltzmann’s constant, R is the port impedance, and T is the noise temperature, using 290° Kelvin for T.

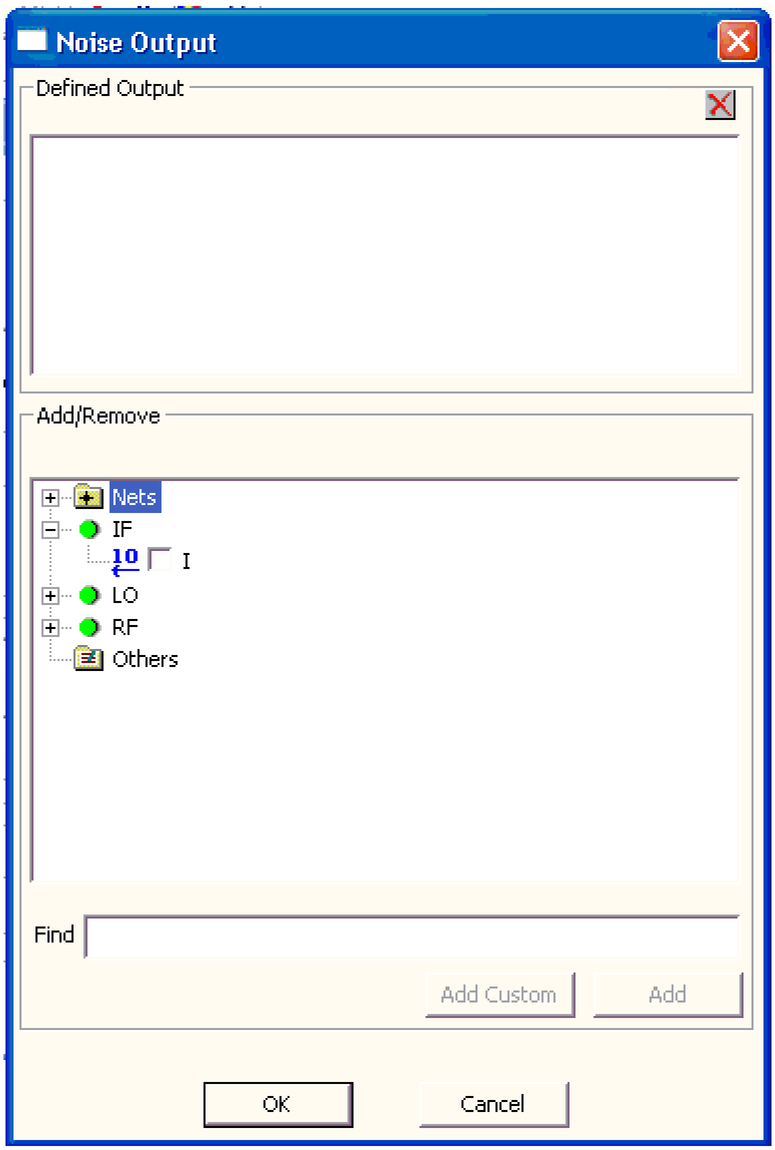
Gain for a port input is calculated in Volts, in the sense of available power. The input units are Volts for a voltage source and Amperes for a current source. If no **IProbe** input is selected, Nexxim computes the output noise only.

10. Use the drop-down menu in the **OProbe** field to select a port, resistor, current source, voltage source, or current probe for the small-signal output on the circuit.

> **Note:** The port definition window has fields to specify noise parameters (See Editing an Interface Port Definition.) The Output Port may be defined with noise parameters, but the analysis ignores the noise data for the output port.

When a port, current source, or resistor is selected, output units are Volts. When a voltage source or current probe is selected, output units are Amperes.

11. Click **Edit** next to the **Noise Output** field to open the **Noise Output** window.



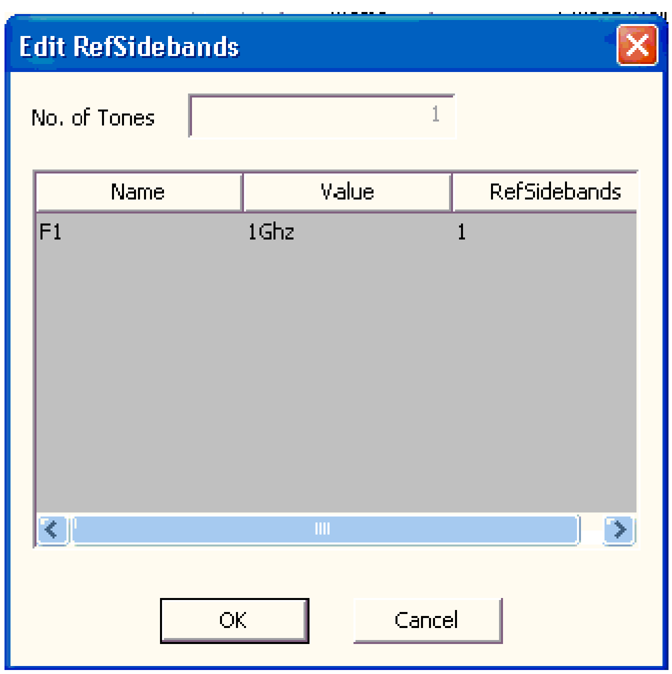
- Expand the **Nets** and devices icons to display lists of the available quantities. Click the check boxes to select noise outputs. (The **IProbe** and **OProbe** devices are always included as noise data outputs, so they do not appear in the list).
- **Adding a Custom Quantity (Differential Voltage Output)** To specify a noise output other than a single node voltage or branch current—for example, a differential voltage such as V(net1,net2)—type the output into the **Find** field , then click **Add Custom**. (**Add Custom** becomes active when a noise output in the **Find** field cannot be found in the list of Nets and devices.)

The custom output quantity is passed to the netlist as a **NOISE_OUTPUT** entry in the **TV_ NOISE** statement. For example, the differential voltage example above netlists as:

```
NOISE_OUTPUT=[v(net1,net2)]
```

No validation is performed on the custom output quantity before it is written out to the netlist.

12. Click **OK** to close the **Noise Output** window and return to the **TV Noise Analysis** window.
13. After one or more elements have been added in the Output Noise field, use the drop-down menu in the **Input Referred Noise** field to select a node voltage or branch current for input-referred noise analysis. The **Input Referred Noise** selection is active only when at least one item has been selected in the **Noise Output** field. The **Input Referred Noise** entry does not have to be the same as any of the **Noise Output** selections.
14. Optionally, click **Edit RefSidebands** to open the corresponding window:



- The **No. of tones**, **Name**, and **Value** fields reflect the settings on the **Edit Tones/MaxK** window. The number, names, and values of tones cannot be changed in the **Edit Refsidebands** window.
- Click in the **RefSidebands** field for a frequency to set the reference sideband coefficient for that frequency. The **REFSIDEBAND** coefficients are positive or negative integers plus zero. See the TV Noise Technical Notes topic for information on the use of the **Refsidebands** quantities.

> **Note:** When an **IProbe** device is specified, the netlist must also have the **Refsideband** list of coefficients for the **Edit Tones** frequencies in order to perform the conversion gain calculation. When no **IProbe** device is specified, the **Refsideband** entry is optional; the default coefficient value is 1.

15. Optionally, click the **Save All Sidebands** box to have Nexxim store the contributions from all noise sidebands. The default is not to save all contributions to save analysis time and memory usage, since the file is potentially very large.
16. Optionally, use the fields in the **Solution Options** group box to select or add HB or OSC analysis options and other Nexxim options to the design.

- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **TV Noise Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Select the **HB Options** or **OSC Options** tab. Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **TV Noise Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

The line can contain multiple options settings. Use spaces to separate the options settings.

To modify or delete an option once it has been added, just edit or delete its entry in the list. To clear all options, delete the entire line.

Click **OK** to return to the **TV Noise Analysis** window.

The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

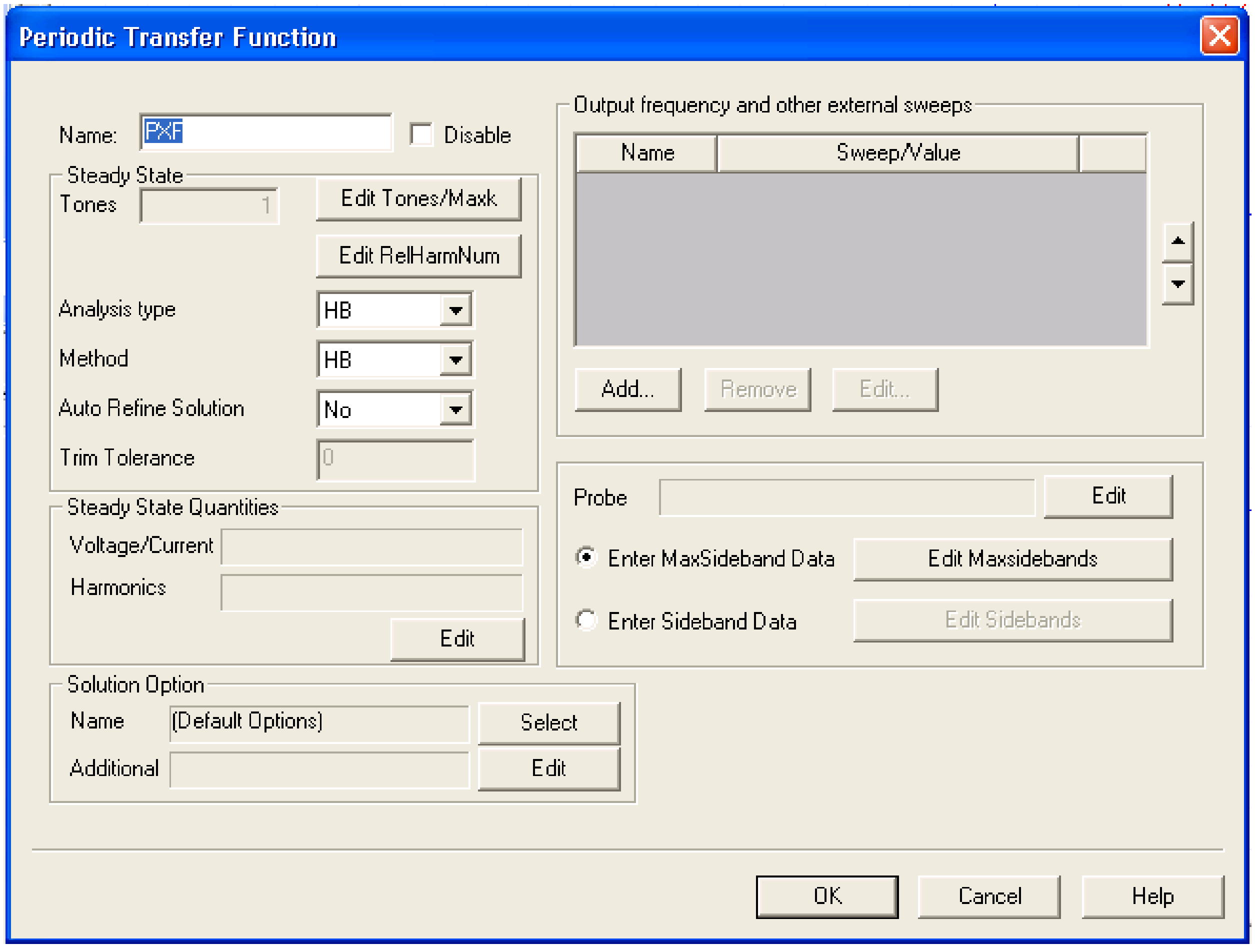
17. Click **Finish** to close the **TV Noise Analysis** window. The solution setup is added to the Project tree under the **Analysis** icon.
18. Run the simulation:
    - Expand the **Analysis** icon on the Project tree, click the appropriate solution setup, and select **Analyze** from the menu. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
    - The **Message Manager** window signals success or failure.
    - For details on creating and modifying reports, see Generating Reports and Post-processing.

Running Periodic Transfer Function Analysis from the Schematic Editor

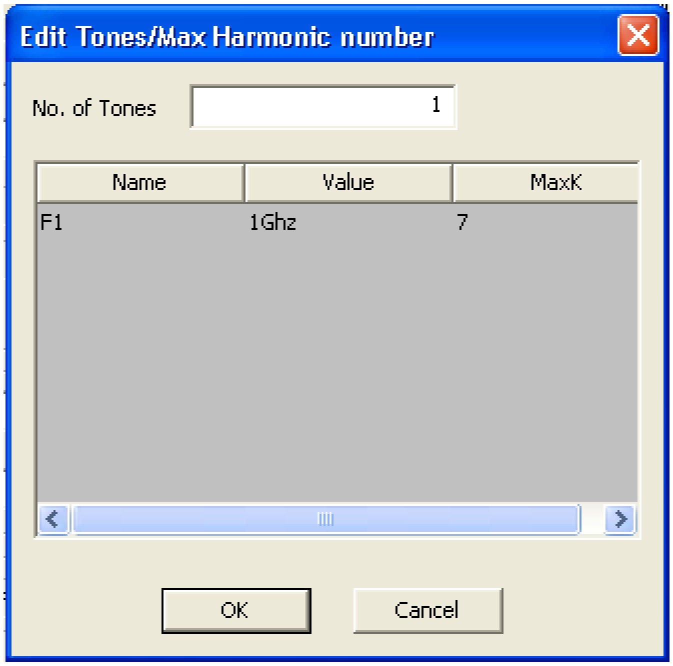
Periodic transfer function (PXF) analysis computes the small-signal transfer function from selected voltage and current sources in the circuit to a specified single output at a frequency that can be swept. A steady-state HB or OSC analysis first computes the periodic or quasi-periodic operating point, on which the small-signal PXF analysis is calculated. In Nexxim, PXF analysis is an extension of the time-varying noise analysis.

To run a periodic transfer function analysis on a Nexxim schematic design, perform the following steps:

1. In **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Periodic Transfer Function (PXF)**. The **Periodic Transfer Function** solution setup window opens:

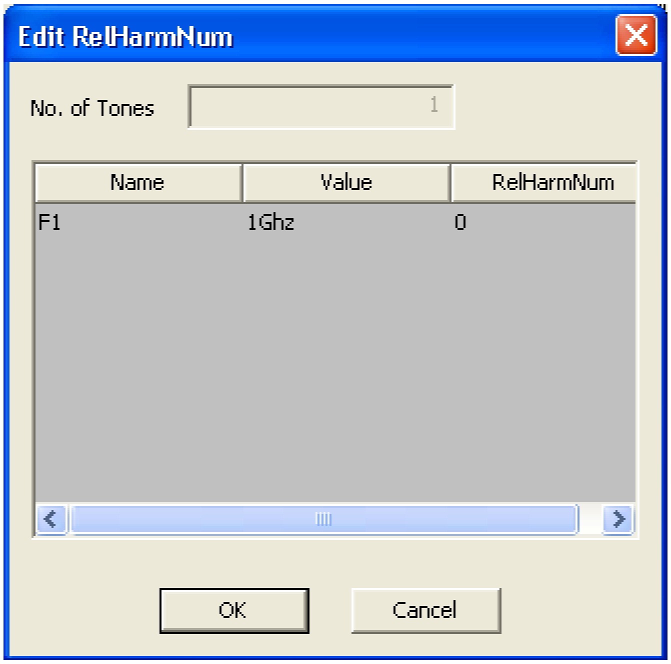


2. Type an **Analysis Name** (or accept the default name, for example “PXF”).
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
4. In the **Steady State** group box, click **Edit Tones/MaxK** to open the **Edit Tones/Max Harmonic Number** window.



- Enter the number of tones to be used in the steady-state phase of the analysis. The tones are listed with names **F1**... **F*n***. These names cannot be changed.
- Click a **Value** field in the row for one of the tones to enter the frequency for the steady-state analysis for that tone.
- Click a **MaxK** field in the row for one of the tones to enter the **Max. Harmonic Number** for the steady-state analysis for that tone.
- Click **OK** to close the **Edit Tones/Maxk** window and return to the **Periodic Transfer Function** window. The **Number of tones** field now shows the number of tones you specified in the window.

5. Optionally, click **Edit RelHarmNum** to open the corresponding window.



- The **No. of tones**, **Name**, and **Value** fields reflect the settings on the **Edit Tones/Maxk** window. The number, name, and value of tones cannot be changed in the **Edit RelHarmNum** window.

Click **OK** to return to the **Periodic Transfer Function** window.

- Click in the **RelHarmNum** field for a frequency to set the relative harmonic number offset for that tone. See the TV Noise Technical Notes topic for information on the use of the **RelHarmNum** offsets.

Click **OK** to return to the **Periodic Transfer Function** window.

6. Select the **Analysis type** on the drop-down. The default is harmonic balance (**HB**). Choose oscillator analysis (**OSC**) instead.

- When **HB** is the **Analysis type** and one-tone analysis has been specified in the **Edit Tones/Maxk** window, the **Method** field is activated. Use this field to select **HB** or **shooting** (HB is the default). See Handling Strongly Nonlinear Circuits in the HB topic for details on the **Method** entry.
- When **HB** is the **Analysis type**, the **Auto Refine Solution** field is activated. Use this field to select Yes or No (No is the default). See Increasing Accuracy or Speed in the HB topic for details on the **Auto Refine Solution** entry.
- When **HB** is the **Analysis type** and more than two tones have been specified in the **Edit Tones/Maxk** window, the **Trim Tolerance** field is activated. Use this field to set a

tolerance value. See Increasing Accuracy or Speed in the HB topic for details on the **Trim Tolerance** entry.

7. Optionally, click **Edit** at the lower-right of the **Steady State Quantities** group box to add voltages, currents, and harmonic outputs on the underlying analysis (HB or OSC). These additional output quantities are available in the Report window after simulation has been performed.

Expand the **Nets** and devices icons to display lists of the available quantities. To have Nexxim calculate outputs on the steady state analysis step, click the check boxes to select the outputs.

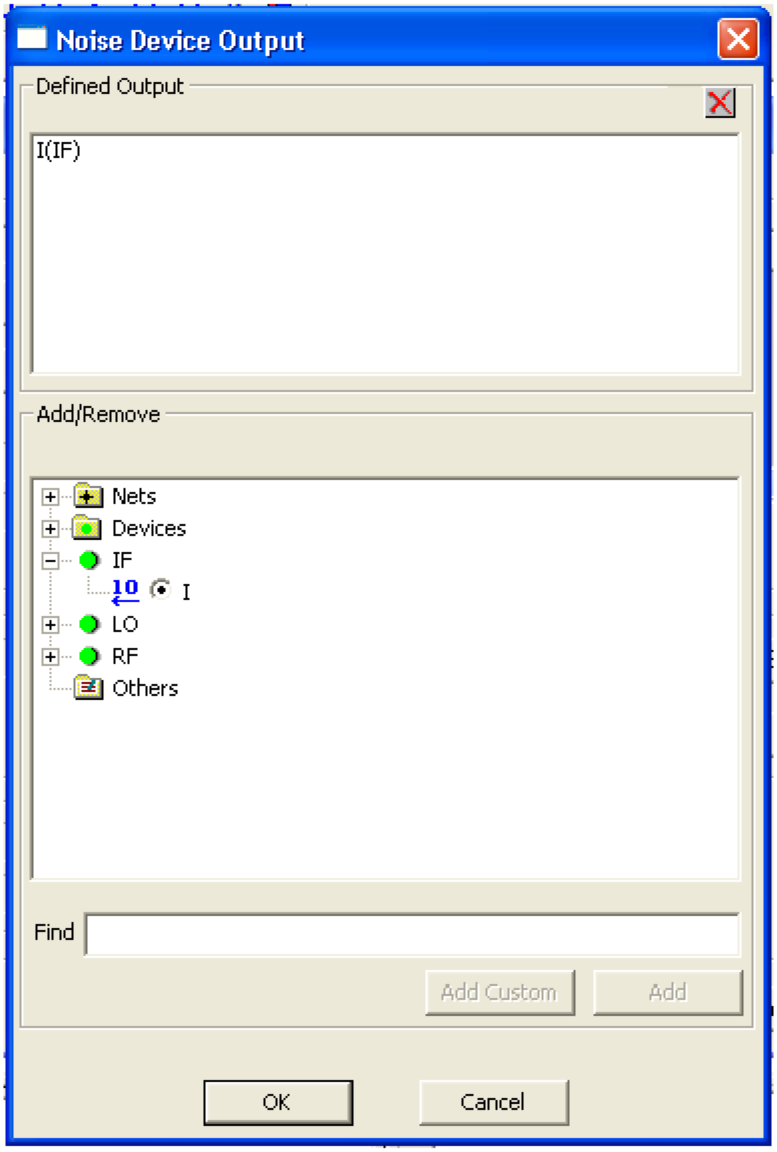
Expand the **Harmonics** icon to display a list of the available harmonics. To specify particular harmonics for Nexxim to use when calculating outputs on the steady state analysis step, click the check boxes to select the harmonics. If you do not select any harmonics, harmonics DC, F1, 2F1, 3F, and 4F1 are automatically selected. In the **Report** window, only selected harmonics are available for plotting.

Click **OK** to return to the **Periodic Transfer Function** window.

8. The underlying TV Noise analysis requires a sweep of output frequencies. Click **Add** in the **Output frequency and other sweeps** field to open the **Add/Edit Sweep** window.

- Ensure **F** (frequency) is the entry in the **Variable** field.
- Use the radio buttons to select one of the following: **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**.
- Type the sweep values into the **Value** text box (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units are selected for each.
- Click **Add**, then click **OK** to close the window and reopen the **Periodic Transfer Function** window.
- The sweep definition appears in the **Output frequency and other sweeps** field. (The **Output frequency and other sweeps** display includes fields labeled **Offset of F1** and **Sync**. These fields are not used by Nexxim PXF analysis.)

9. Click **Edit** next to the **Probe** field to open the **Noise Device Output** window.



Expand the **Nets** and **Devices** icons to display lists of the available quantities. Click in the check box for the single output to be used in the PXF analysis.

Click **OK** to return to the **Periodic Transfer Function** window.

10. If you have specified a single frequency in the Steady State **Tones** field, use the radio buttons to select **Enter MaxSideband Data** or **Enter Sideband Data**. See the TV Noise Technical Notes topic for details.
    - To enter a maximum sideband to use in calculating the input frequencies, select **Enter MaxSideband Data** and click **Edit Maxsidebands**. Click in the **MaxSidebands** field at

the right and enter the number of sidebands to use. (The **Name** and **Value** fields reflect your selection in the Steady State **Tones** field, and cannot be changed in the **Edit MaxSidebands** window). Click **OK** to return to the **Periodic Transfer Function** window.

- To enter specific sidebands to use, select **Enter Sideband Data** and click **Edit Sidebands**. Enter the first sideband value, then click **Add Row** to open another line in the window. To delete a row, select it and click **Delete Row**. When you have finished entering sidebands, click **OK** to return to the **Periodic Transfer Function** window.

11. If you have specified multiple frequencies in the Steady State **Tones** field, only the radio **Enter Sideband Data** button is active. See the TV Noise Technical Notes topic for more information on the use of the **Sideband Data** entries.
    - To enter specific sidebands to use, click **Edit Sidebands**. Enter the first set of sideband values (one value per frequency), then Click **Return** or **Tab** to open another line in the window. When you have finished entering sidebands, click **OK** to return to the **Periodic Transfer Function** window.
12. Optionally, use the fields in the **Solution Options** group box to add analysis options and other Nexxim options to the design.
    - Click **Select** on the **Name** field to open the **Select Solutions** window. From the **Select Solutions** window, click **New** to open the **Solution Options** window.

Click the **Name** field to name the new option settings. Select the **HB Options** or **OSC Options** tabs, depending on the selection for the **Underlying analysis type** made earlier (Step 8). Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window.

From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **Periodic Transfer Function** window. The name of the new option settings appears in the **Name** field of the **Solution Option** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**. Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Click **OK** to return to the **Periodic Transfer Function** window.

13. Click **Finish** to close the **Periodic Transfer Function** window. The solution setup is added to the Project tree under the **Analysis** icon.
14. Run the simulation:
    - Expand the **Analysis** icon on the Project tree, click the appropriate solution setup, and select **Analyze** from the menu. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
    - The **Message Manager** window signals success or failure.
    - For details on creating and modifying reports, see Generating Reports and Post-processing.

Running TV Noise Analyses on the Netlist Editor

TV Noise analysis is controlled from in the netlist using the **.TV_NOISE** analysis statement.

Running TV Noise Analyses

1. From the **Circuit** menu, select **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
2. The **Message Manager** window signals success or failure.
3. Display results of successful analysis:

- From the **Circuit** menu, select **Create Report** to open the **Create Report** window.
- See TV Noise Results for details on the outputs from TV Noise analysis.
- For details on creating and modifying reports, see Generating Reports and Post-processing.

TV Noise Netlist Syntax

The syntax of the **.TV_NOISE** statement in Nexxim is:

```
.TV_NOISE
 + output_freq_sweep
 + SS_ANALYSIS=HB|OSC
 + SS_TONES=[val [,val]...]
 + SS_MAXK=[val [,val]...]
 + [RELHARMVEC=[val [,val]...]]
+ [IPROBE=device_name]
+ [OPROBE=device_name]
+ [NOISE_OUTPUT=[nets_currents]
 + [INPUT_REFERRED_NOISE=[nets_currents]]]
+ [REFSIDEBAND=[val [,val]...]]
+ [SAVEALLSIDEBANDS=NO|YES]
+ [METHOD=HB|SHOOTING]
+ [AUTO_REFINE_SOLUTION=YES|NO]
+ [TRIM_TOL=val]
```

In this syntax:

- For clarity, elements are shown one per line using the continuation character (+). Elements can be entered on a single line using spaces as separators.
- Brackets shown in bold (**[ ]**) are required when more than one item is specified.
- Brackets in plain font ([ ]) denote optional entries or fields.

Output Frequency Sweep

The required entry *output_freq_sweep* specifies a range of output frequencies to apply in the noise calculation. The sweep specification should be the first field in the TV_NOISE statement (after the **.TV_NOISE** keyword).

(See the TV Noise Technical Notes topic for information on the use of the swept output frequencies.)

The *output_freq_sweep* field has several formats.

The sweep can be just a single frequency value.

A second form of *output_freq_sweep* specifies the starting and ending frequencies, and the incremental step:

```
[START=]start_val [STOP=]stop_val [STEP=]step_val
```

If the sweep specification omits the **START=**, **STOP=**, and **STEP=** labels, the start value, stop value and step value must be entered in the order shown. If the labels are used, the values may be entered in any order, but then all three values must have labels.

A third format for *output_freq_sweep* is:

```
count_type numpoints startfreq stopfreq
```

The *count_type* entry (**LIN**, **DEC**, or **OCT**) specifies a pattern of frequencies to be swept with *numpoints*, *startfreq* and *stopfreq*:

The **LIN** (linear) count type specifies a linear sweep at *numpoints* different frequencies on the frequency given by *startfreq* to the frequency given by *stopfreq*. The spacing between test frequencies is (*stopfreq*–*startfreq*) / *numpoints*.

The **DEC** (decade) count type specifies a logarithmic sweep at *numpoints* different frequencies in each decade (10× multiple of *startfreq*) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within decades.

The **OCT** (octave) count type specifies a logarithmic sweep at *numpoints* different frequencies in each octave (2× multiple of startfreq) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within octaves.

A fourth format for *output_freq_sweep* is:

```
POInumpoints freq1 [freq2 ...]
```

The **POI** format specifies a space-separated list of individual frequency points. The entry *numpoints* is the number of frequency points in the list.

SS_ANALYSIS

The **SS_ANALYSIS** entry selects between harmonic balance (**HB**) and oscillator analysis (**OSC**) to compute the steady-state values. **HB** is the default.

SS_TONES

The **SS_TONES** entry specifies one or more frequencies to use in the steady-state harmonic balance calculations.

SS_MAXK

The **SS_MAXK** entry specifies the maximum harmonic numbers to use for the corresponding **SS_TONES** frequencies in the steady-state harmonic balance calculations. The number of **SS_MAXK** entries should be the same as the number of **SS_TONES** frequencies.

RELHARMVEC

The **RELHARMVEC** entry specifies offset multipliers for the corresponding **SS_TONES** frequencies. The number of **RELHARMVEC** entries should be the same as the number of **SS_TONES** frequencies.

> **Note:**
>
> When just one offset is specified, the netlist can use the entry **RELHARMNUM** instead of **RELHARMVEC**, and can omit the brackets around the offset value.
>
> See the TV Noise Technical Notes topic for information on the use of the **RELHARMVEC** offsets.

IPROBE

The **IPROBE** entry specifies the input RF device to use for the small-signal input to the circuit.

The device can be a port, a current source, or a voltage source.

When a port is specified as the input device, Nexxim computes the noise figure and the conversion gain. Gain for a port is in units of Volts, in the sense of available power.

> **Note:**
>
> The definition of the port in the netlist uses the Port Impedance Resistor syntax.
>
> If the port definition includes the noise parameters NOISE and NOISETEMP, the analysis uses the temperature value specified. If these parameters are not in the definition, Nexxim adds thermal noise to the specified port for computing the noise factor and conversion gain with the formula 4KT/R, where K is Boltzmann's constant, R is the port impedance, and T is the noise temperature, using 290° Kelvin for T.

For current and voltage source input devices, only the gain is computed. The input units are Amperes and Volts, respectively.

If **IPROBE** is not specified, only the **Onoise** figure is calculated.

OPROBE

The **OPROBE** entry specifies the device to use for the small-signal output on the circuit.

The device can be a port, a current source, a voltage source, a resistor, or a current probe.

> **Note:**
>
> The definition of the port in the netlist uses the Port Impedance Resistor syntax.
>
> The port definition can include noise parameters NOISE and NOISETEMP, but the analysis ignores the noise data for the output port.

When a port, current source, or resistor is specified as the **OPROBE** device, the output units are in Volts. When a voltage source or current probe is specified, the units are in Amperes.

REFSIDEBAND

The **REFSIDEBAND** entry specifies coefficients to apply to the **SS_TONES** frequencies in the calculation of the RF/input frequency. The **REFSIDEBAND** coefficients are positive or negative integers plus zero. See the TV Noise Technical Notes topic for more information on the use of the **REFSIDEBAND** coefficients.

> **Note:**
>
> When an **IPROBE** device is specified, the netlist must also have the **REFSIDEBAND** list of coefficients for the **SS_TONES** frequencies in order to perform the conversion gain calculation. When no **IPROBE** device is specified, the **REFSIDEBAND** entry is optional; the default coefficient value is 1.

NOISE_OUTPUT

The **NOISE_OUTPUT** entry is a comma-separated list of node voltages and branch currents for which noise output is appropriate, in addition to the noise data calculated for the **IPORT** and **OPORT** devices. (No noise factor or conversion gain is computed for these additional outputs.

Voltage outputs are specified using the format:

`V(`*`net1`*`,`*`net2`***`)`**

Noise output for nets has units V^2/Hz.

Branch currents are specified using the format:

`I(`*`device`***`)`**

Noise output for branch currents has units A^2/Hz.

See TV Noise Results for details on the outputs from TV Noise analysis.

INPUT_REFERRED_NOISE

The **INPUT_REFERRED_NOISE** entry specifies a single node voltage or branch or current to be used as the source for an input-referred noise calculation.

A node voltage is specified using the format:

V(*net1*,*net2***)**

A branch current is specified using the format:

I(*device***)**

The **INPUT_REFERRED_NOISE** entry can be specified only when the **NOISE_OUTPUT** entry is also specified.

Harmonic Balance Controls

When **SS_ANALYSIS=HB** is specified, the additional controls **METHOD**, **AUTO_REFINE_SOLUTION**, and **TRIM_TOL** can be used.

See Handling Strongly Nonlinear Circuits in the Harmonic Balance topic for details on the **METHOD** entry in the HB statement.

See Increasing Accuracy or Speed for details on the **AUTO_REFINE_SOLUTION** and **TRIM_TOL** entries in the HB statement.

TV Noise Netlist Example

```
.TV_NOISE
+ START=1000000000 STOP=2000000000 STEP=500000000
+ OPROBE=[Z167]
+ SS_MAXK=[7]
+ SS_TONES=[1000000000]
+ SS_ANALYSIS=HB
+ NOISE_OUTPUT=[V(IF),I(LO)]
+ INPUT_REFERRED_NOISE=[RIF]
+ REFSIDEBAND=[1]
+ SAVEALLSIDEBANDS=no
```

Periodic Transfer Function Netlist Syntax

The syntax of the **.TV_NOISE** statement in Nexxim is:

```
.TV_NOISE
 + output_freq_sweep
 + SS_ANALYSIS=HB|OSC
 + SS_TONES=[val [,val]...]
 + SS_MAXK=[val [,val]...]
 + [RELHARMVEC=[val [,val]...]]
```

```
+ PXF=1
 + PROBE=[net_current]
 + MAXSIDEBAND=val | SIDEBAND=[val [,val]...]
 + [METHOD=HB|SHOOTING]
+ [AUTO_REFINE_SOLUTION=YES|NO]
+ [TRIM_TOL=val]
```

In this syntax:

- For clarity, elements are shown one per line using the continuation character (+). Elements can be entered on a single line using spaces as separators.
- Brackets shown in bold (**[ ]**) are required when more than one item is specified.
- Brackets in plain font ([ ]) denote optional entries or fields.

Output Frequency Sweep

The required entry *output_freq_sweep* specifies a range of output frequencies to apply in the noise calculation. The sweep specification should be the first field in the TV_NOISE statement (after the **.TV_NOISE** keyword).

(See the TV Noise Technical Notes topic for information on the use of the swept output frequencies.)

The *output_freq_sweep* field has several formats.

The sweep can be just a single frequency value.

A second form of *output_freq_sweep* specifies the starting and ending frequencies, and the incremental step:

```
[START=]start_val [STOP=]stop_val [STEP=]step_val
```

If the sweep specification omits the **START=**, **STOP=**, and **STEP=** labels, the start value, stop value and step value must be entered in the order shown. If the labels are used, the values may be entered in any order, but then all three values must have labels.

A third format for *output_freq_sweep* is:

```
count_type numpoints startfreq stopfreq
```

The *count_type* entry (**LIN**, **DEC**, or **OCT**) specifies a pattern of frequencies to be swept with *numpoints*, *startfreq* and *stopfreq*:

The **LIN** (linear) count type specifies a linear sweep at *numpoints* different frequencies on the frequency given by *startfreq* to the frequency given by *stopfreq*. The spacing between test frequencies is (*stopfreq*–*startfreq*) / *numpoints*.

The **DEC** (decade) count type specifies a logarithmic sweep at *numpoints* different frequencies in each decade (10× multiple of *startfreq*) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within decades.

The **OCT** (octave) count type specifies a logarithmic sweep at *numpoints* different frequencies in each octave (2× multiple of startfreq) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within octaves.

A fourth format for *output_freq_sweep* is:

`POI``numpoints freq1 [freq2 ...]`

The **POI** format specifies a space-separated list of individual frequency points. The entry *numpoints* is the number of frequency points in the list.

SS_ANALYSIS

The **SS_ANALYSIS** entry selects between harmonic balance (**HB**) and oscillator analysis (**OSC**) to compute the steady-state values. **HB** is the default.

SS_TONES

The **SS_TONES** entry specifies one or more frequencies to use in the steady-state harmonic balance calculations.

SS_MAXK

The **SS_MAXK** entry specifies the maximum harmonic numbers to use for the corresponding **SS_TONES** frequencies in the steady-state harmonic balance calculations. The number of **SS_MAXK** entries should be the same as the number of **SS_TONES** frequencies.

RELHARMVEC

The **RELHARMVEC** entry specifies offset multipliers for the corresponding **SS_TONES** frequencies. The number of **RELHARMVEC** entries should be the same as the number of **SS_TONES** frequencies.

> **Note:**
>
> When just one offset is specified, the netlist can use the entry **RELHARMNUM** instead of **RELHARMVEC**, and can omit the brackets around the offset value.
>
> See the TV Noise Technical Notes topic for information on the use of the **RELHARMVEC** offsets.

PXF

The entry **PXF=1** is required to invoke periodic transfer function analysis.

PROBE

The **PROBE** entry specifies the device to use for the small-signal transfer function output.

The *device* can be a port, a current source, a voltage source, or the voltage drop across a pair of nodes, using the format **V(***n1***,***n2***)**.

When a port, voltage source, or voltage drop is specified as the **PROBE** device, the output units are in Volts. When a current source is specified, the units are in Amperes.

MAXSIDEBAND or SIDEBAND

The netlist must specify the combination of harmonics of the steady state **SS_TONES** frequencies, using either **MAXSIDEBAND** or **SIDEBAND**.

(See the TV Noise Technical Notes topic for information on the use of the **MAXSIDEBAND** and **SIDEBANDS** entries.)

When the **SS_TONES** group box has only one frequency, click the **MAXSIDEBAND** entry to specify the maximum harmonic of that frequency.

Alternatively, specify a set of specific sidebands to use for the single output frequency, using the **SIDEBAND** entry.

When the **SS_TONES** group box specifies more than one frequency, you must Click the **SIDEBAND** entry to specify the combinations of harmonics that are of interest as input frequencies (**MAXSIDEBANDS** is not valid with multiple **SS_TONES** frequencies).

Harmonic Balance Controls

When **SS_ANALYSIS=HB** is specified, the additional controls **METHOD**, **AUTO_REFINE_SOLUTION**, and **TRIM_TOL** can be used.

See Handling Strongly Nonlinear Circuits in the Harmonic Balance topic for details on the **METHOD** entry in the HB statement.

See Increasing Accuracy or Speed for details on the **AUTO_REFINE_SOLUTION** and **TRIM_TOL** entries in the HB statement.

Periodic Transfer Function Netlist Example

```
.TV_NOISE
+ START=1000000000 STOP=2000000000 STEP=500000000
+ SS_MAXK=[7]
+ SS_TONES=[1.0e9, 2.17e9]
+ SS_ANALYSIS=HB
+ PXF=1
+ PROBE=[V22]
+ SIDEBAND=[1,0,2,-1,1,1]
```

TV Noise Results

TV Noise analysis can compute several kinds of outputs. To view the available quantities for output in a report, specify **Noise** in the **Domain** field on the **Report** window. The following categories of outputs can be displayed in a report:

Variables

This category contains the swept frequency variable **F** and any other variables defined in the design.

Output Variables

This category contains the output variables on the steady-state harmonic balance calculation specified in the solution setup. In a schematic, the Output Variables quantities are specified by editing the **Output Variables** field in the **TV Noise Analysis** window. In a netlist, the Output Variables quantities are specified with a **.PRINT HB** statement.

Noise Output

This category contains the noise outputs at the nodes or branches identified in the solution setup. In a schematic, the Noise Output quantities are specified by editing the **Noise Output** field in the **TV Noise Analysis** window. In a netlist, the Noise Output quantities are specified with the **NOISE_OUTPUT** entry in the **.TV_NOISE** statement. For voltages, the noise output is expressed in V^2/Hz. For currents, the noise output is expressed in A^2/Hz. No noise factor or conversion gain is computed for these outputs. (The noise factor and conversion gain are computed for the following Onoise figure.)

Input Referred Noise

The solution setup can specify one input (port, resistor, or source) to use as an input-referred noise source. Nexxim creates a result for each noise output. For voltage sources, the input-referred noise is expressed in V^2/Hz. For all other devices, the input-referred noise is expressed in A^2/Hz.

Noise

This category includes the Onoise, Noise Factor, and Noise Figure outputs.

Onoise is the noise output for the specified output frequency at the specified output port, in dB. The Noise Factor/Noise Figure and Conversion Gain outputs are referenced to Onoise.

The Noise Factor and Noise Figure are computed when both input and output ports are specified. The calculation uses the thermal noise parameters on the input port specification, if present, otherwise defaulting to a noise temperature of 290°K. Noise parameters for the output port are ignored in the calculation.

The Nexxim TV Noise Factor is the IEEE single-sideband figure. The Noise Figure is the Noise Factor converted to dB.

- to open the Noise Factor, select **NF11** on the **Quantities** group box, and select **<none>** from the **Functions** group box in the **Report** window.
- to open the Noise Figure, select **NF11** on the **Quantities** group box, and select **dB** on the **Functions** list in the **Report** window.

Conversion Gain

The Conversion Gain output (quantity **CG11**) is the power conversion gain between the input port at the input frequency and the output port at the output frequency, in dB.

The Conversion Gain is computed when both input and output ports are specified. The calculation uses the thermal noise parameters on the input port specification, if present, otherwise defaulting to a noise temperature of 290°K. Noise parameters for the output port are ignored in the calculation.

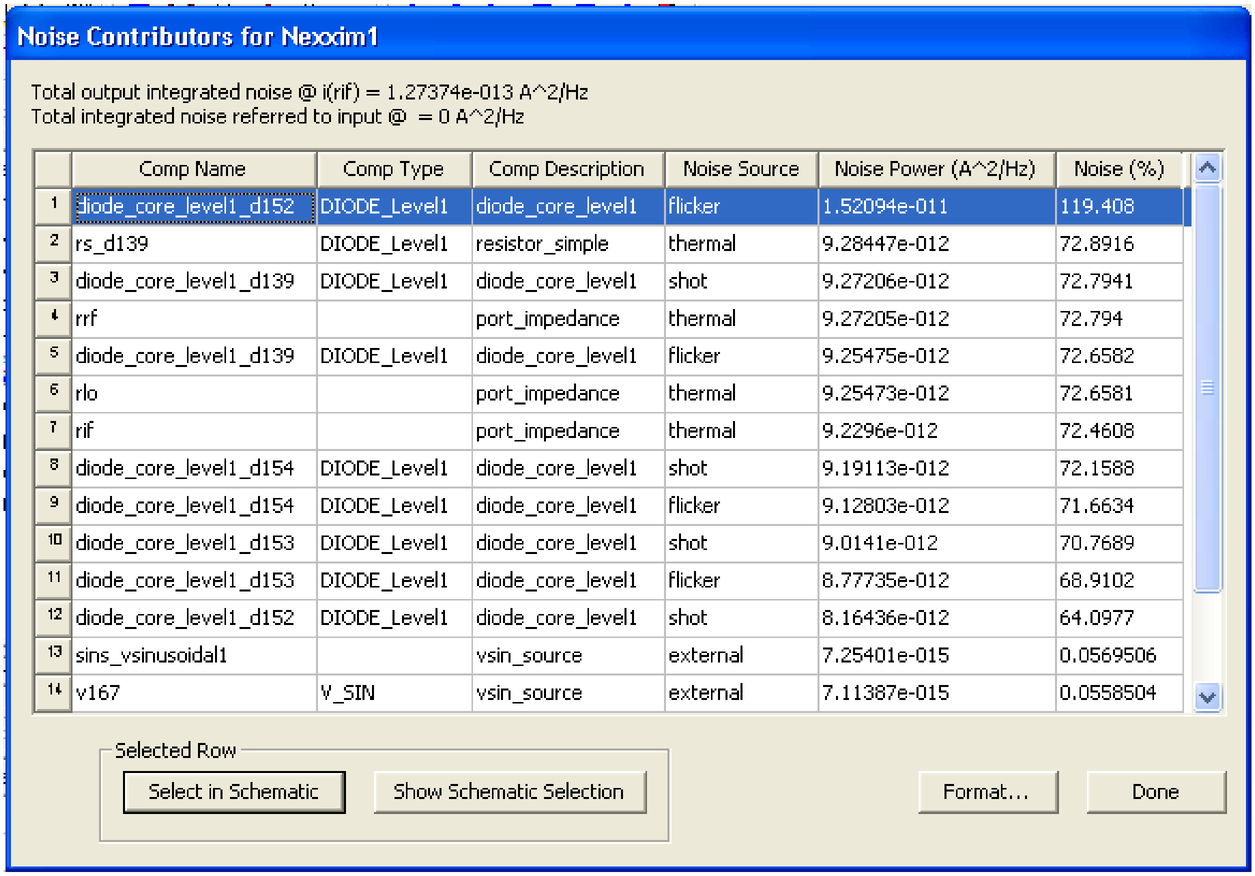
Viewing TV Noise Contributors

When noise analysis has been specified in the solution setup or in the netlist, view a listing of noise contributors . Noise contributors are available for Linear Network Analysis (LNA), Time-Varying Noise Analysis (TVNoise) and Oscillator Analysis (OSC).

To enable this feature, the **TV_Noise Analysis** setup in a schematic design must specify at least one **Noise Output** device (node voltage or branch current). In a netlist design, the netlist must include the **NOISE_OUTPUT** entry in the .TV_NOISE statement.

- For schematic designs, right-click the solution setup icon in the **Project Manager** window and select **Show Noise Contributors** on the menu.
- For netlist designs, right-click the circuit name icon in the **Project Manager** window and select **Show Noise Contributors** from the menu.

The **Noise Contributors** window opens:



Initially, the noise contributors are listed in order by **Component Name**. (If the window is opened before an analysis has been performed, it is empty.)

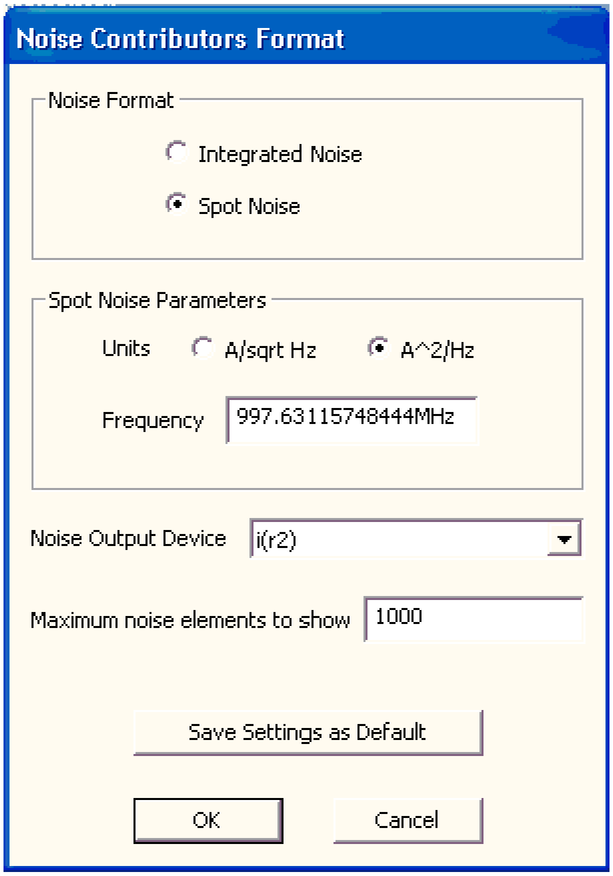
- Click in a column header to sort the display by the contents of that column (the example above has been sorted by **Noise (%)**).

In a schematic design, the **Selected Row** group box is enabled.

- Select a contributor (row) and click **Select in schematic**. The contributing element is selected in the schematic window. The schematic window is brought to the front (under the **Noise Contributors** window) and scrolled so the selected object is visible. Only one element can be selected at a time.
- Select an object in the schematic window and click **Show schematic selection**. The corresponding contributing element is selected in the **Noise Contributors** window. The row containing that element is scrolled into view if necessary.

Initially, the display is set up to show spot noise contributors using default settings.

- Click **Format** to open the **Noise Contributors Format** window:



Choose one noise output device (voltage through node or current through branch) on the **Noise Output Device** field drop-down menu. The list is populated on the output devices selected in the **TV_Noise Analysis** setup or in the **NOISE_OUTPUT** entry on the .TV_NOISE statement.

Enter a value in the **Maximum noise elements to show** field to restrict the display to that number of elements with the most noise contributions.

When **Spot Noise** is selected, the **Spot Noise Parameters** group box is activated.

- The **Units** button settings depend on the **Noise Output Device**:

  When the device is a branch current (as in the illustration above), the choice of units is between Amps/(Hz)^{½} and (Amps)^2/Hz.

  When the device is a node voltage, the choice of units is between Volts/(Hz)^{½} and (Volts)^2/Hz.

- The **Frequency** field is initialized to a value drawn on the analysis setup. Change the frequency as appropriate to see how the spot noise contributions vary with frequency.

When **Integrated Noise** is selected, the **Spot Noise Parameters** group box is deactivated. Units are Volts for output device node voltages and Amperes for output device branch currents.

- Click **Save settings as default** to save the **Noise Contributors Format** window settings as the defaults for future displays.
- Click **OK** to close the **Noise Contributors Format** window and return to the **Noise Contributors** window.

Click **Done** to close the **Noise Contributors** window.

The **Noise Contributors** window becomes invalid if the design is modified. If the window is open when the analysis is re-run, the window is refreshed.

Multiple **Noise Contributor** windows may be opened, but only one for a given design.

TV Noise Options

The TV Noise and Periodic Transfer Function (PXF) analysis options are listed in the TV Noise and PXF Analysis Options Reference. The TV Noise and PXF option values can be changed with a **.OPTIONS** statement in a netlist.

> **Note:**
>
> The TV Noise option values cannot be changed in a schematic design.

The syntax for the .OPTIONs statement is:

```
.OPTIONS tv_noise.option1=val1 [tv_noise.option2=val2 ... ]
```

Enter the prefix **tv_noise.** to apply the option value only to time-varying noise analysis but not to other tools that recognize the option.

Nexxim Linear Network Analysis

Linear network analysis (LNA) computes the frequency-dependent scattering (S), admittance (Y), and impedance (Z) parameters for a circuit linearized around the DC bias point. Optionally, group delay analysis and DC noise analysis can be included.

LNA Circuit Configuration

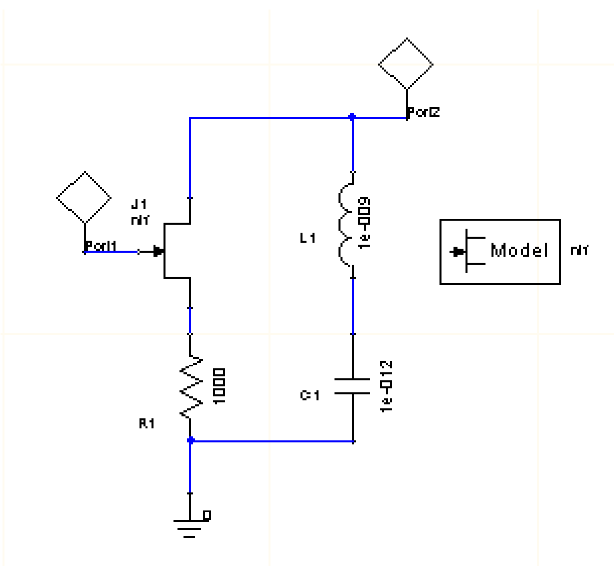
To configure the circuit for linear network analysis, you must add test ports to the circuit. The procedure depends on the type of design — circuit or netlist.

> **Note:**
>
> A port or port impedance resistor is not required for a linear network analysis that consists only of noise calculations.

LNA Circuit Schematic Configuration

In the **Schematic Editor**, you add the editor's built-in I/O ports. Select the **Interface Port** icon  on the Schematic Editor menu. Drag and place one or more ports, then connect them to the circuit at the appropriate places.

The following figure shows a simple JFET circuit configured with Schematic ports for LNA:



LNA Circuit Netlist Configuration

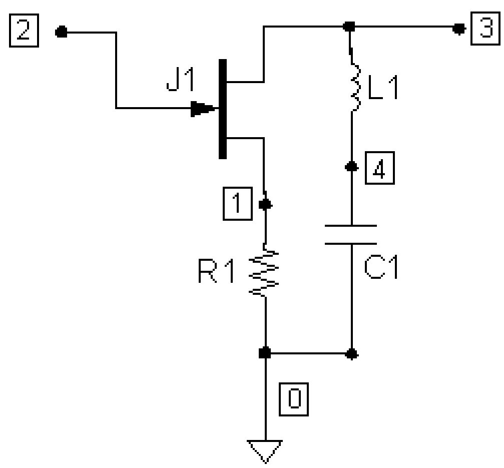
If you are using the Netlist Editor to create a design, you add the test ports to the netlist with port impedance resistors. See *Resistor, Port Impedance* in the Nexxim Components help for further information.

Suppose the circuit consists of the four elements in the following netlist:

```
* JFET circuit example
J1 3 2 1 njf
L1 3 4 1e-9
C1 4 0 1e-12
R1 1 0 1000
```

```
.MODEL njf NJF level=1 $ parameters not shown
.END
```

The corresponding schematic is illustrated in the following figure:



To create a test port, add a port impedance resistor from an input node to ground. The port impedance resistor includes a **PORTNUM** parameter specifying a port number, and specifies the port impedance with parameters **RZ** and **IZ**. (See Resistor, Port Impedance in the Nexxim Component Library documentation for details on the **PORTNUM**, **RZ**, and **IZ** parameters.) The **ZERO_PORT_VALUES=1** option tells Nexxim to treat the port resistors as external to the circuit (See LNA Options).

In combination with the network analysis, Nexxim can perform an analysis of the AC small-signal transfer functions. (See LNA-Associated AC Analysis for further information.) To enable AC analysis, the netlist includes one or more voltage or current sources with the AC parameter to specify the magnitude and phase of the AC value. The netlist also includes a **.PRINT AC** statement to specify the outputs to be analyzed.

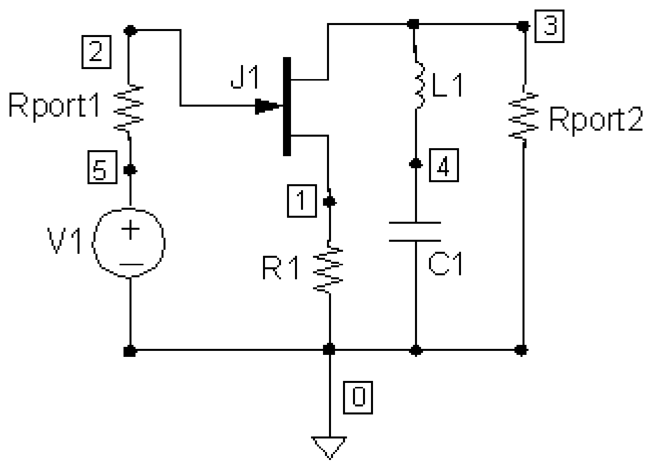
> **Note:**
>
> Port impedance resistors are not required for a linear network analysis that consists only of AC small-signal analyses.

Here is the netlist with added port resistors with **ZERO_PORT_VALUES** option, and statements to provide for AC transfer function analysis:

```
* JFET circuit with Port Resistors for Linear Network Analysis
J1 3 2 1 njf
L2 3 4 1e-9
C1 4 0 1e-12
R1 1 0 1000
```

```
V1 5 0 5 AC 1 0 $ Voltage source for AC analysis
Rport1 2 5 PORTNUM=1 RZ=50 IZ=0 $ Port 1 impedance resistor
Rport2 3 0 PORTNUM=2 RZ=50 IZ=0 $ Port 2 impedance resistor
.MODEL njf NJF level=1 $ parameters not shown
.LNA dec 10 5e8 5e9 flag='LNA'
.OPTIONS ZERO_PORT_VALUES=1
.PRINT AC V(1) I(V1) $ Outputs for AC analysis
.END
```

The following figure illustrates the netlist as a schematic:

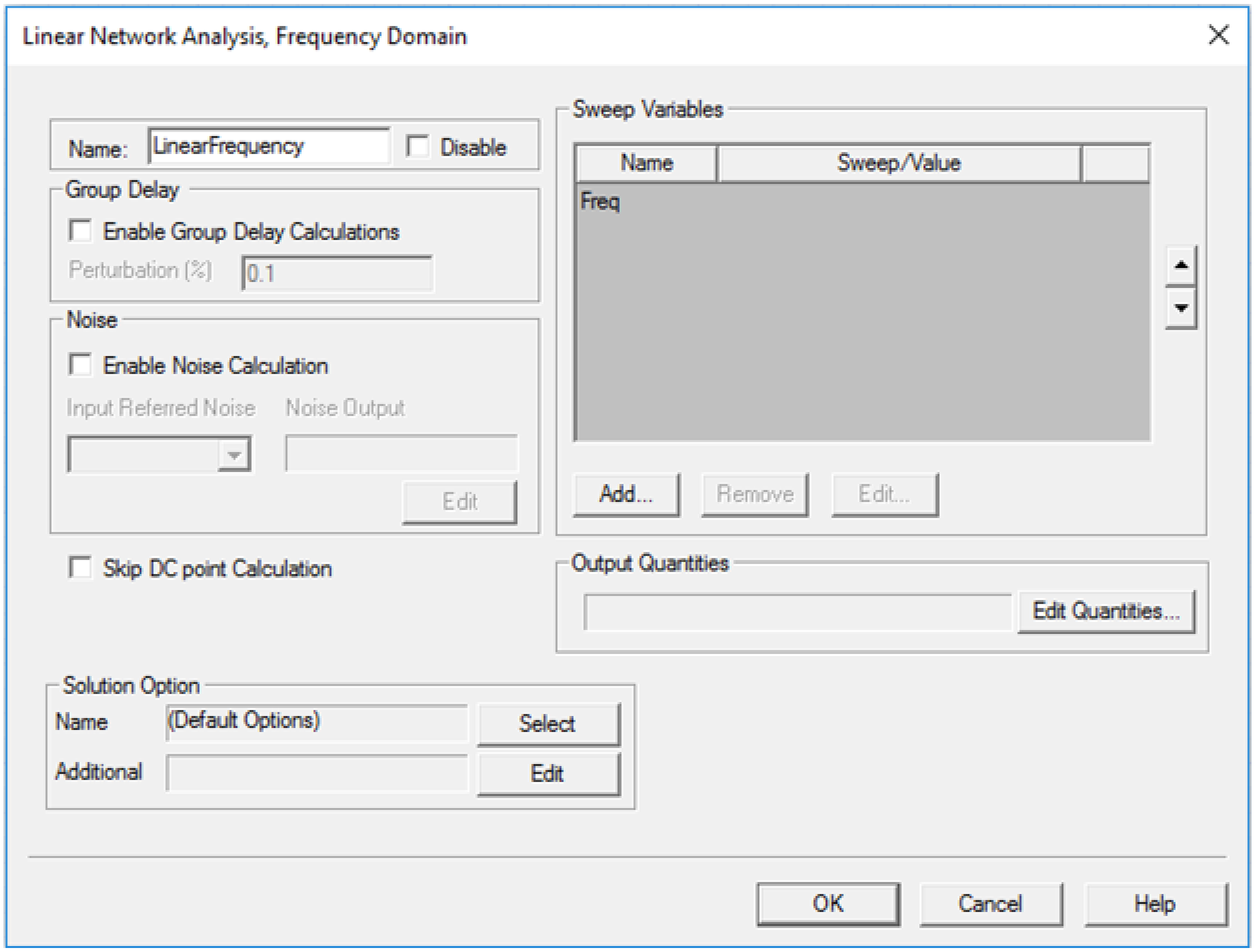


In a design entered with the schematic editor, the schematic uses Interface (I/O) ports instead of the port impedance resistors, and uses a voltage probe instead of the voltage source.

Running LNA on the Schematic Editor

From the **Schematic Editor**, perform the following steps to set up a Nexxim linear network analysis, and run the analysis.

1. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Setup ...** > **Linear Network Analysis** to reopen the **Linear Network Analysis, Frequency Domain** window.



2. Type an **Analysis Name** (or accept the default name, for example “LinearFrequency1”).
3. For most simulations, leave the **Disable** group box unselected (the default setting). Selecting this box lets you store multiple solution setups for later use. (Note that if a solution setup is deactivated before the analysis is run, any changes made to the design invalidates the simulation results.)
4. Depending on the requirements of the project, select **Enable Group Delay Calculations** and specify a **Perturbation** as a percentage of the test frequency. (See Group Delay Analysis later in this topic).
5. Depending on the requirements of the project, select **Enable Noise Calculation**. When the **Enable Noise Calculation** group box is checked, the **Noise Output** field is activated. When at least one Noise Output has been added, the **Input Referred Noise** field is activated.
   a. Click **Edit** under the **Noise Output** field to open a **Noise Output** window. Select one or more currents through devices or voltages through nets, then click **OK** to close the **Noise Output** window (or click **Cancel** to close the window without selecting any outputs).

b. After adding one or more **Noise Output** elements, select a source for the input-referred noise calculation from the list in the **Input Referred Noise** drop-down.

See the LNA Technical Notes for more information on LNA noise calculations.

6. By default, the field solvers solve for the DC frequency. To deactivate the DC calculation, check **Skip DC Point Calculation**.The results without the DC calculation are reliable only for circuits with linear devices and no non-zero-valued sources (AC sources are OK).
7. To add a basic frequency sweep, locate the **Sweep Variable** area in the **Linear Network Analysis, Frequency Domain** window box.
   a. From the **Sweep Variable** area, select the frequency **Freq** > **Edit** to open the **Add/Edit Sweep** window.
   b. In the **Variable** list, frequency **Freq** is selected (and cannot be changed). Select one of the following: **Single value**, **Linear step**, **Linear count**, **Decade count**, or **Octave count**.
   c. Type the sweep values into the **Value** field (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units (GHz, MHz, kHz) are selected for each.
   d. Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Linear Network Analysis, Frequency Domain** window.
   f. For more information, see Variable Sweep.
7. Click **Edit Quantities** in the **Output Quantities** field to open the **Output Quantities** window. Select and **Add** the outputs to be analyzed, then click **OK** to return to the **Linear Network Analysis, Frequency Domain** window.

> **Note:** Any voltage or current probes that are present in the schematic is included as output quantities.

8. Optionally, use the fields in the **Solution Options** group box to select or add DC analysis options and other Nexxim options to the design.
- Click **Select** on the **Name** field to open the **Select Solution Options** window.

If any options sets have been defined, their names appear in the **Select Solution Options** group box. To select a named option set that you have previously defined, click the name of the option set, then click **OK** to return to the **LNA Analysis** window. The named option set appears in the **Name** field in the **Solution Options** group box.

To create a new option set, click **New** to open the **Solution Options** window. Click the **Name** field to name the new option set. Select the **DC Options** tab. Make the appropriate changes to option values, then click **OK** to return to the **Select Solutions** window box. From the **Select Solutions** window, click the name of the new option settings, then click **OK** to return to the **DC Analysis** window. The name of the new option settings appears in the **Name** field in the **Solution Options** group box.

- Click **Edit** on the **Additional** field to open a text-entry window, **Edit additional options**.

Use the field to enter any Nexxim options exactly as they are to appear in the netlist. Do not include the keyword .OPTIONS; the .OPTIONS keyword is automatically inserted at the beginning of the line in the netlist statement.

The line can contain multiple options settings. Use spaces to separate the options settings.

To modify or delete an option once it has been added, just edit or delete its entry in the list.

To clear all options, delete the entire line.

Click **OK** to return to the **Linear Network Analysis, Frequency Domain** window.

The options in the **Additional** field are automatically added to the netlist when the solution containing them is performed.

For more information, see DC Analysis Options.

11. Click **Finish** to close the **Linear Network Analysis, Frequency Domain** window.
12. Run the simulation:
    - From the **Circuit** menu, select **Analyze**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
    - The **Message Manager** window signals success or failure.
    - For details on creating and modifying reports, see Generating Reports and Post-processing.

> **Note:**
>
> In the **Schematic Editor**, nodes to be output are identified by attaching Ports in the schematic (See LNA Circuit Configuration for an example).

Running LNA on the Netlist Editor

If you use the Netlist Editor to create a design, the netlist should contain the information for running the linear network analysis. No Solution Setup is required.

Running Nexxim Linear Network Analysis

1. From the **Circuit** menu, select **Analyze with Nexxim**. If the circuit is set up correctly, the analysis begins immediately and a progress bar appears.
2. The **Message Manager** window signals success or failure.
   - For details on creating and modifying reports, see Generating Reports and Post-processing.

LNA Netlist Syntax

The netlist syntax of the Linear Network Analysis statement is:

.LNA*frequencies* [**FLAG=***analysis_flag*] [**GROUP_DELAY_PERT=***val*]
[**NOISE_OUTPUT=[***list_of_outputs***]**]
[**INPUT_REFERRED_NOISE=[***source***]**]

Optionally, the netlist may also contain an output statement of the form:

.PRINT AC*list_of_outputs*

LNA Frequency Sweep

The entry *frequencies* specifies a sweep of frequencies to be applied in the analysis.

The *frequencies* field has several formats.

The sweep can be just a single frequency value.

A second form of *frequencies* specifies the starting and ending frequencies, and the incremental step:

[**START=**]*start_val* [**STOP=**]*stop_val* [**STEP=**]*step_val*

If the sweep specification omits the **START=**, **STOP=**, and **STEP=** labels, the start value, stop value and step value must be entered in the order shown. If the labels are used, the values may be entered in any order, but then all three values must have labels.

A third form for the *frequencies* field has the format:

freq_sweep_type numpoints startfreq stopfreq

This format specifies a count of analysis frequencies defined by *numpoints*, *startfreq* and *stopfreq*, using one of the predefined *freq_sweep_type* tokens **LIN**, **DEC**, or **OCT**:

LIN
Linear sweep at *numpoints* different frequencies on the frequency given by *startfreq* to the frequency given by *stopfreq*. The spacing between test frequencies is

```
(stopfreq - startfreq) / (numpoints - 1).
```

DEC
Logarithmic sweep at *numpoints* different frequencies in each decade (10× multiple of *startfreq*) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within decades.

OCT
Logarithmic sweep at *numpoints* different frequencies in each octave (2× multiple of *startfreq*) from *startfreq* to *stopfreq*. The spacing between test frequencies is logarithmic within octaves.

A fourth format for the *frequencies* field is:

POI*numpoints freq1* [*freq2* ...]

This form provides a space-separated list of *numpoints* test frequencies.

LNA Analysis Flag

The *analysis_flag* is a string (enclosed in single quotes) that specifies the analyses to be performed. The valid entries for *analysis_flag* are:

'LNA'
Linear network analysis

'NOISE_DC'
DC noise analysis

'LNAN'
Linear network and DC noise analyses

'LNAG'
Linear network and group delay analyses

'LNANG'
(Default) Linear network, DC noise, and group delay analyses (**LNAGN** is a synonym for **LNANG**)

Group Delay Perturbation Factor

When the *analysis_flag* on the LNA statement is ‘LNAG’ or ‘LNANG’, group delay analysis is enabled. The entry **GROUP_DELAY_PERT** specifies the perturbation to be applied in calculating group delay. The perturbation is specified as a *percentage* of the test frequency (See the LNA Technical Notes).

Noise Contributions

The .LNA statement can include specifications for noise output and for input-referred noise.

The syntax entry:

NOISE_OUTPUT=[*list_of_outputs***]**

specifies that noise output is to be calculated for the list of outputs in square brackets. The outputs in the list are node voltages or device currents separated by commas, for example:

```
NOISE_OUTPUT=[V(1), I(R3), V(Port2)]
```

The syntax entry:

INPUT_REFERRED_NOISE=[*source***]**

specifies that the single node voltage or device current in square brackets is to be used as the source for an input-referred noise calculation, as in these examples:

```
INPUT_REFERRED_NOISE=[V(1)]
```

```
INPUT_REFERRED_NOISE=[I(R3)]
```

> **Note:**
>
> A port or port impedance resistor is not required for a linear network analysis that consists only of noise calculations.

AC Calculation

On a line that is separate on the .LNA statement, the netlist can contain a PRINT AC statement of the form:

.PRINT AC*list_of_outputs*

To request the output of small-signal AC analysis that is calculated by the linear network analysis tool. The outputs in the list are node voltages or device currents separated by commas. See LNA-Associated AC Analysis for further information.

Support for Small Signal Analysis in Imported Netlists

In addition to the linear network analysis netlist syntax, Nexxim supports small-signal AC analysis in HSPICE™ syntax in netlists imported from other design environments. The supported statement type is:

.AC

LNA Results

The results for linear network, DC noise, and group delay analyses are summarized in the following tables. The parameters for N-ports are listed first, the results for 2-port networks.

Netlist Parameters for N-Port Circuits

TF	Transfer function. When ports are used as described earlier, the transfer function describes the relationship between input voltage and output current.
Y	Complex *Y* admittance parameter matrix, I = YV where I = vector of output currents, V = vector of input voltages
Z	Complex *Z* impedance parameter matrix, V = ZI
S	Complex *S* scattering parameter matrix, B = SA where E_i = vector of incident waves, E_r = vector of reflected waves, $V = E_i + E_r$, $I = (E_i - E_r)/Z_0$, Z_0 = characteristic impedance, $A = E_i/Z_0$, $B = E_r/Z_0$
GD	Real Group Delay,

$$GD_{ij} = \frac{-d\phi_{ij}}{d\omega}$$

where ϕ_{ij}=phase(S_{ij})

Single Port Netlist Parameters for N-Port Circuits

RHO*i* Complex Reflection coefficient,

$$\mathrm{RHO}i = S_{ii}$$

RTL*i* Real Return loss

$$\mathrm{RTL}i = |S_{ii}|$$

VSWR*i* Real Voltage standing wave ratio,

$$\mathrm{VSWR}i = \left|\frac{1+|S_{ii}|}{1-|S_{ii}|}\right|$$

Noise Parameters for N-Port Circuits

Onoise N×N matrix of square roots of noise power spectra (Volt/Hz$^{1/2}$ or Amp/Hz$^{1/2}$).

$$Syy = [TF(\omega) \times Sxx(\omega) \times TF(\omega)^H]$$

$$Onoise = \sqrt{Syy}$$

Inoise Input-referred noise.

$$Inoise(out, in) = \frac{Onoise(out, out)}{\sqrt{TF(out, in) \times (TF(out, in))^*}}$$

Noise Parameters for Two-Port Circuits

FMIN Real Minimum noise figure power ratio. FMIN is derived from fundamental noise quantities.

NF Real Noise figure power ratio. NF is derived from fundamental noise quantities. The reference temperature for the source (port) used in NF calculations is 27 degrees Celsius.

NT Real Equivalent noise temperature,

$$NT = (NF - 1) * 290$$

RN Real Equivalent normalized noise resistance ratio. RN is derived from fundamental noise quantities.

RNU Real Equivalent un-normalized noise resistance,

$$RNU = RN * Z_{REF}$$

GOPT Complex Optimum noise figure reflection coefficient,

$$GOPT = \frac{Zopt - Zs}{Zopt + Zs}$$

YOPT Complex Optimum noise figure source admittance,

$$YOPT = Go + jBo$$

where *Go* and *Bo* are derived from fundamental noise quantities.

ZOPT Complex Optimum noise figure source impedance,

$$ZOPT = 1/YOPT$$

Netlist Parameters for Two-Port Circuits

ABCD *ij* Complex ABCD parameters (chain parameters).

$$\begin{bmatrix} v_1 \\ i_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} v_2 \\ -i_2 \end{bmatrix}$$

Complex Hybrid parameters

H*ij*

$$\begin{bmatrix} v_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} i_1 \\ v_2 \end{bmatrix}$$

G*ij* Complex Inverse Hybrid parameters, $G = H^{-1}$

Gain and Matching Parameters for Two-Port Circuits

Available power gain,

GA

$$\mathrm{GA} = \frac{1-|\Gamma_S|^2}{|1-S_{11}\Gamma_S|^2}\,|S_{21}|^2\,\frac{1}{1-|\Gamma_{OUT}|^2},\quad \textit{where}$$

$$\Gamma_{OUT} = S_{22} + \frac{S_{12}\,S_{21}\,\Gamma_S}{1-S_{11}\Gamma_S}$$

GFMN Gain when the input impedance (Zopt) is used to achieve minimum noise figure (FMIN)

Real Maximum available gain,

GMAX

$$\mathrm{GMAX} = \begin{cases} \dfrac{1}{1-|S_{11}|^2}\,|S_{21}|^2\,\dfrac{1}{1-|S_{22}|^2}, & \textit{unilateral case} \\[2ex] \dfrac{|S_{21}|}{|S_{12}|}\left(K-\sqrt{K^2-1}\right), & \textit{bilateral case} \end{cases}$$

Complex Optimum gain reflection coefficient for Load at maximum available gain (GMax),

GML

$$\mathrm{GML} = \frac{B_2 \pm \sqrt{B_2^2 - 4|C_2|^2}}{2C_2}$$

where

$$B_2 \;=\; 1+|S_{22}|^2-|S_{11}|^2-|\Delta|^2$$

$$C_2 \;=\; S_{22}-\Delta S_{11}^{*}$$

and

$$\Delta = S_{11}S_{22}-S_{12}S_{21}$$

Complex Optimum gain reflection coefficient for Source at maximum available gain (GMax), ,

$$\mathrm{GMS} = \frac{B_1 \pm \sqrt{B_1^2-4|C_1|^2}}{2C_1}$$

where

GMS

$$B_1 \;=\; 1+|S_{11}|^2-|S_{22}|^2-|\Delta|^2$$

$$C_1 \;=\; S_{11}-\Delta S_{22}^{*}$$

and

$$\Delta = S_{11}S_{22}-S_{12}S_{21}$$

GP Power gain,

$$\mathrm{GP} = \frac{1-|\Gamma_L|^2}{|1-S_{22}\Gamma_L|^2}\,|S_{21}|^2\,\frac{1}{1-|\Gamma_{IN}|^2},\ \ where$$

$$\Gamma_{IN} = S_{11} + \frac{S_{12}\,S_{21}\,\Gamma_L}{1-S_{22}\Gamma_L}$$

Transducer power gain,

TG

$$\mathrm{TG} = \frac{1-|\Gamma_S|^2}{|1-S_{11}\Gamma_S|^2}\,|S_{21}|^2\,\frac{1-|\Gamma_L|^2}{|1-\Gamma_{OUT}\Gamma_L|^2}$$

Real Maximum stable gain,

MSG

$$\mathrm{MSG} = \frac{|S_{21}|}{|S_{12}|}$$

Unilateral power gain,

UPG

$$\mathrm{UPG} = \frac{1}{1-|S_{11}|^2}\,|S_{21}|^2\,\frac{1}{1-|S_{22}|^2},\ \ when\ S_{12} = 0$$

Source admittance at maximum available gain (GMAX),

YMS

$$\mathrm{YMS} = \frac{1}{Z_S}\left(\frac{1-\mathrm{GMS}}{1+\mathrm{GMS}}\right)$$

where Z_S is the source impedance.

YML Load admittance at maximum available gain (GMAX),

$$\mathrm{YML} \;=\; \frac{1}{Z_L}\left(\frac{1-\mathrm{GML}}{1+\mathrm{GML}}\right)$$

where Z_L is the load impedance.

Source impedance at maximum available gain (GMAX)

ZMS $$ZMS = \frac{1}{YMS}$$

Load impedance at maximum available gain (GMAX),

ZML $$\mathrm{ZML} = \frac{1}{\mathrm{YML}}$$

Port Parameters for Two-Port Circuits

Input admittance with port 2 terminated,

YIN $$\mathrm{YIN} \;=\; Y_{11} - \frac{Y_{12}\,Y_{21}}{Y_{22}+Y_L}$$

where Y_L is the load admittance.

Output admittance with port 1 terminated,

YOUT $$\mathrm{YOUT} \;=\; Y_{22} - \frac{Y_{12}\,Y_{21}}{Y_{11}+Y_S}$$

where Y_S is the source admittance.

Input impedance with port 2 terminated,

ZIN $$\mathrm{ZIN} \;=\; Z_{11} - \frac{Z_{12}\,Z_{21}}{Z_{22}+Z_L}$$

where Z_L is the load impedance.

Output impedance with port 1 terminated,

ZOUT $$\mathrm{ZOUT} = Z_{22} - \frac{Z_{12}Z_{21}}{Z_{11}+Z_S}$$

where Z_S is the source impedance.

Stability Parameters for Two-Port Circuits

B1 B1 term of the stability factor,.

$$\mathrm{B1} = 1+|S_{11}|^2+|S_{22}|^2 - |\Delta|^2$$

where.

$$\Delta = S_{11}S_{22} - S_{12}S_{21}$$

K Real Stability factor k

$$\mathrm{K} = \frac{1-|S_{11}|^2-|S_{22}|^2+|\Delta|^2}{2|S_{21}S_{12}|}$$

MU Real Stability factor mu

$$\mathrm{MU} = \frac{1-|S_{22}|^2}{|S_{11}-(\Delta)S_{22}^*|+|S_{21}S_{12}|}$$

KCSR Real Stability circle radius for Source,

$$\mathrm{KCSR} = \left|\frac{S_{12}S_{21}}{|S_{11}|^2-|\Delta|^2}\right|$$

KCLR Real Stability circle radius for Load,

$$\mathrm{KCLR} = \left| \frac{S_{12} S_{21}}{\left|S_{22}\right|^2 - \left|\Delta\right|^2} \right|$$

Complex Stability circle origin for Source,

KCSO

$$\mathrm{KCSO} = \frac{\left(S_{11} - \Delta S_{22}^{*}\right)^{*}}{\left|S_{11}\right|^2 - \left|\Delta\right|^2}$$

Complex Stability circle origin for Load

KCLO

$$\mathrm{KCLO} = \frac{\left(S_{22} - \Delta S_{11}^{*}\right)^{*}}{\left|S_{22}\right|^2 - \left|\Delta\right|^2}$$

Two-Port Voltage Gain Parameters

Complex Voltage gain input-output,

VGIO

$$\mathrm{VGIO} = V_2 / V_1$$

Complex Voltage gain insertion,

VGIN

$$VGIN = VGSL\left(1 + Z_S / Z_L\right)$$

Complex Voltage gain source-load,

VGSL

$$VGSL = V_2 / V_S$$

Exporting LNA Results

You can export the results of a linear network analysis to a Touchstone™ file for use by other analyses. Results can be exported from both Schematic Editor designs and Netlist Editor designs.

See Exporting LNA Results to a File in the Importing and Exporting Data topic for further information.

Please See the documentation for your design environment for the syntax definitions of this statement.

> **Note:**
>
> Port impedance resistors are not required for an AC small-signal analysis that uses the .AC statement.

LNA Options

In a netlist, options are specified with the **.OPTIONS** statement. In **Schematic Editor** designs, options are specified with the **Add Solution Options** window. See Linear Network Analysis Options Reference for details on the LNA options.

ZERO_PORT_VALUES

Port impedance resistors specify the port impedances with their PORTNUM parameters.

See *Resistor, Port Impedance* in the Nexxim Component help for further information.

LNA needs to treat these impedances as external to the circuit. The model option **ZERO_ PORT_VALUES=1** is therefore required for LNA.

The **ZERO_PORT_VALUES** option is automatically enabled with any design entered through the **Schematic Editor**. It does not appear on the list of Analysis Options in the **Schematic Editor**. For designs entered with the Netlist Editor, the **ZERO_PORT_VALUES** option must be entered explicitly with a **.OPTIONS** statement.

Nexxim DC Analysis Options in LNA

All DC Analysis options apply to the operating point analysis that is performed at the start of LNA. See DC Analysis Options in Nexxim Help.

Nexxim S-Parameter Element Options in LNA

For designs that incorporate s-parameter elements, setting the s-parameter option **time_ domain_s_model=1** causes Nexxim to create a state-space system for the analysis. Normally, a state-space system is not generated for frequency-domain analyses such as LNA.

LNA References

[1] Guillermo Gonzales, *Microwave Transistor Amplifiers: Analysis and Design, 2nd Edition.*Prentice-Hall, Inc., Upper Saddle River, NJ, Copyright © 1997, 1984.

Smith Tool

The Ansys Smith Tool is an interactive Smith chart utility for the design of amplifiers, oscillators, and matching networks. The Smith Tool is available after a successful linear network analysis.

Smith Tool Capabilities

The Smith Tool addresses amplifier design for gain and noise, design of general matching circuits, and initial design of oscillators. Circles of constant gain, noise, reflection, and stability regions are used to select circuit impedances and design parameters. Capabilities of the Smith Tool include:

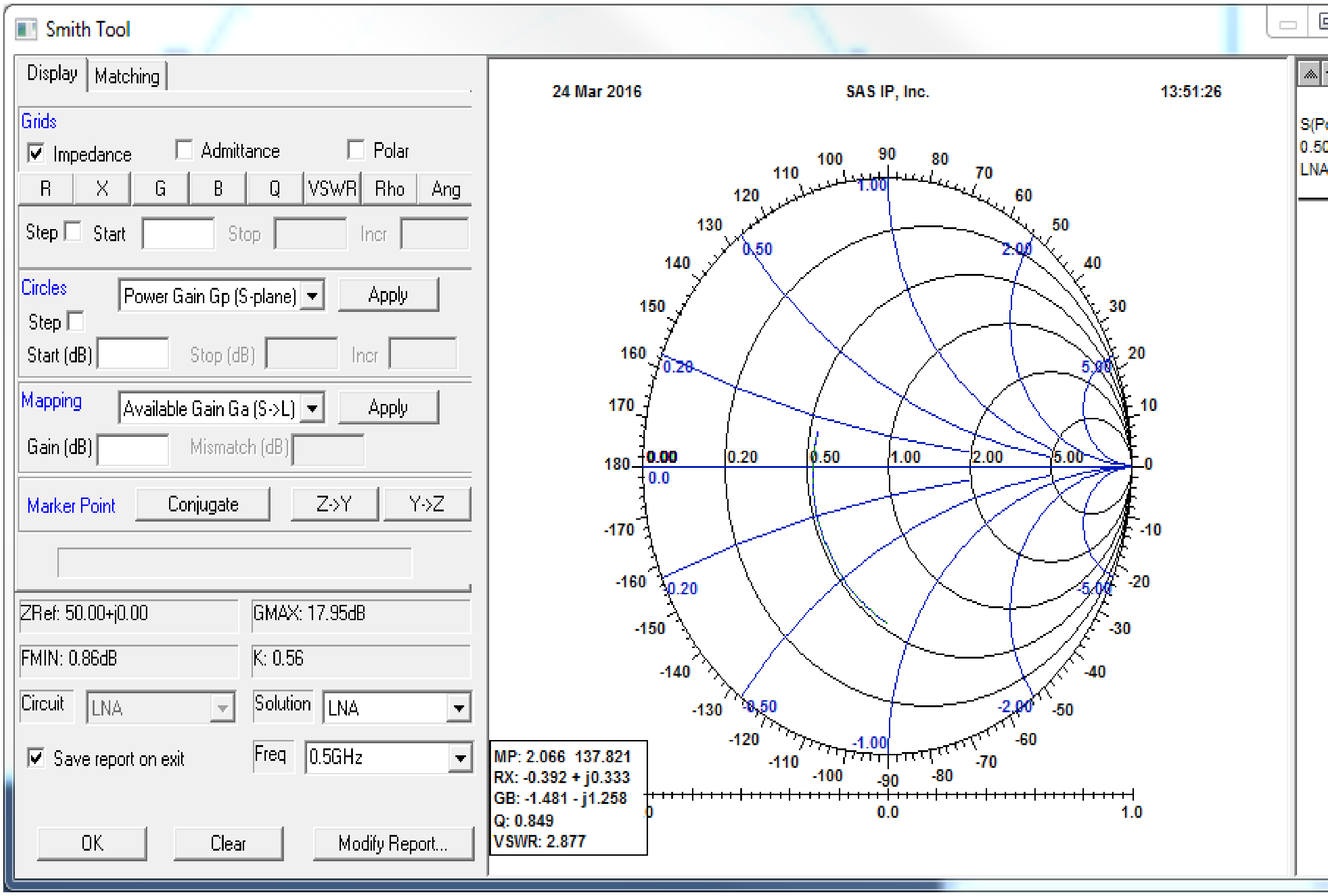
- Arbitrary grids for impedance, admittance, *Q*, VSWR, et cetera
- Constant available gain and power gain circles
- Constant noise circles
- Stability circles
- Circles of constant reflection for oscillator design
- Bilateral mapping between source and load planes with gain mismatch circles
- Ladder matching circuits using discrete and distributed elements

> **Note:**
>
> The Smith Tool is available to analyze the results of linear network analyses performed by the Nexxim simulator on schematic-based designs. Netlist-only Nexxim designs cannot be analyzed with the Smith Tool.

Running the Smith Tool

To start a Smith Tool session, after running a successful linear network analysis, do either of the following to open the **SmithTool** window:

- Click **Product>Smith Tool**.
- With a polar report or Smith-chart report selected and active, click **Report2D** > **Smith Tool**.

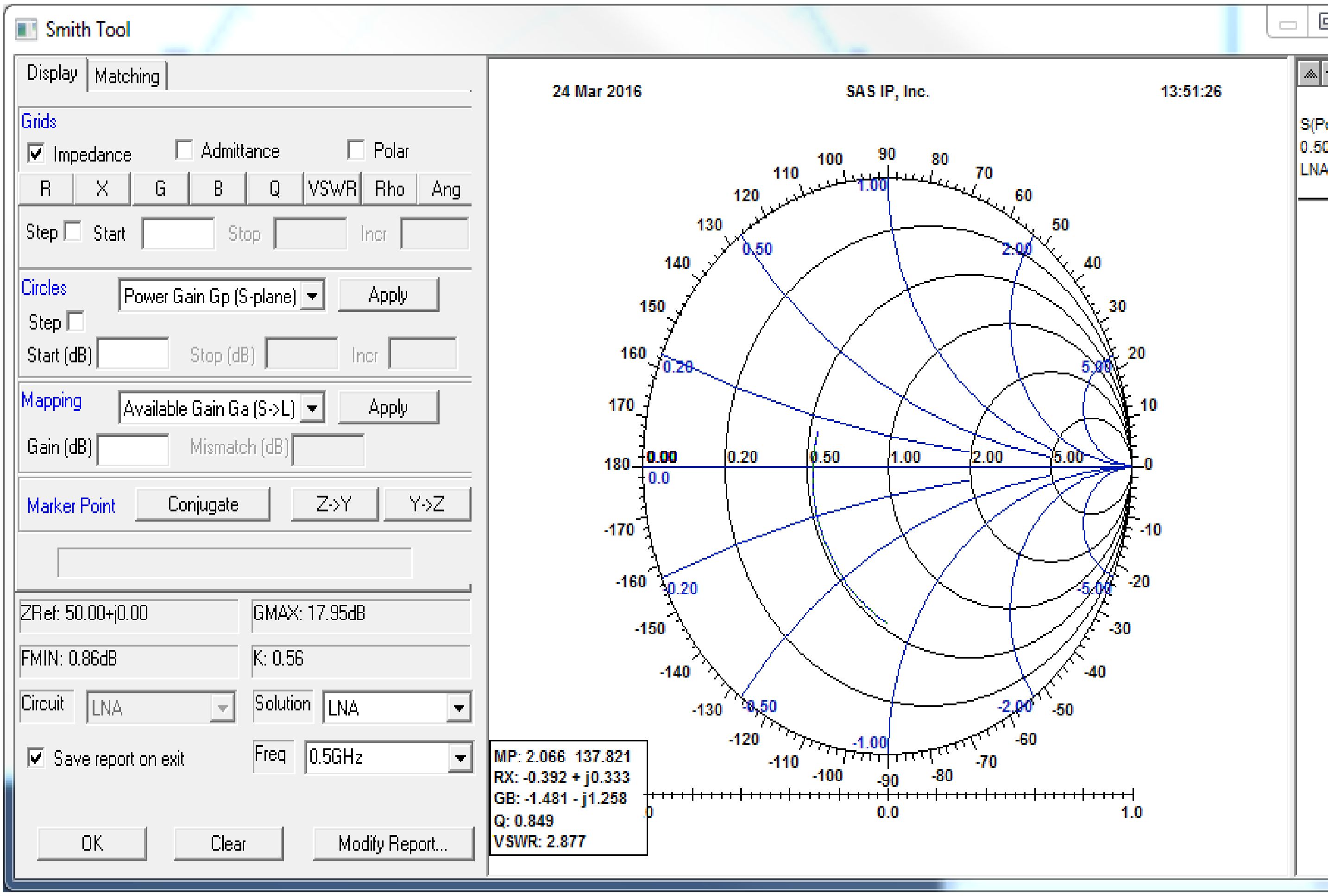


The **SmithTool** performs interactive matching-network synthesis at a single analysis frequency for a particular analysis of a specified circuit/subcircuit. Because multiple analyses may be available for multiple circuits when the Smith Tool window opens, be sure to begin each session by adjusting/confirming these settings at the bottom of the **SmithTool** window:

- In the **Circuit** group box, select the circuit to which you want to add a matching network or networks. The matching network(s) you create becomes as subcircuits of the circuit you select.
- In the **Solutions** group box, select the analysis on which you want to base your Smith Tool session.
- In the **Freq** group box, select the frequency for which your matching network(s) is designed.

Smith Tool window

The **Smith Tool** window is re-sizable and consists of control sections at left and a Smith chart display at right.



- The control section at left contains **Display** and **Matching** tabs and controls settings for session parameters.
- The display group box at right contains a Smith chart representation of the parent circuit's S_{11} response.

Settings for Smith Tool Session Parameters

The bottom portion of the Smith Tool control window displays the following information:

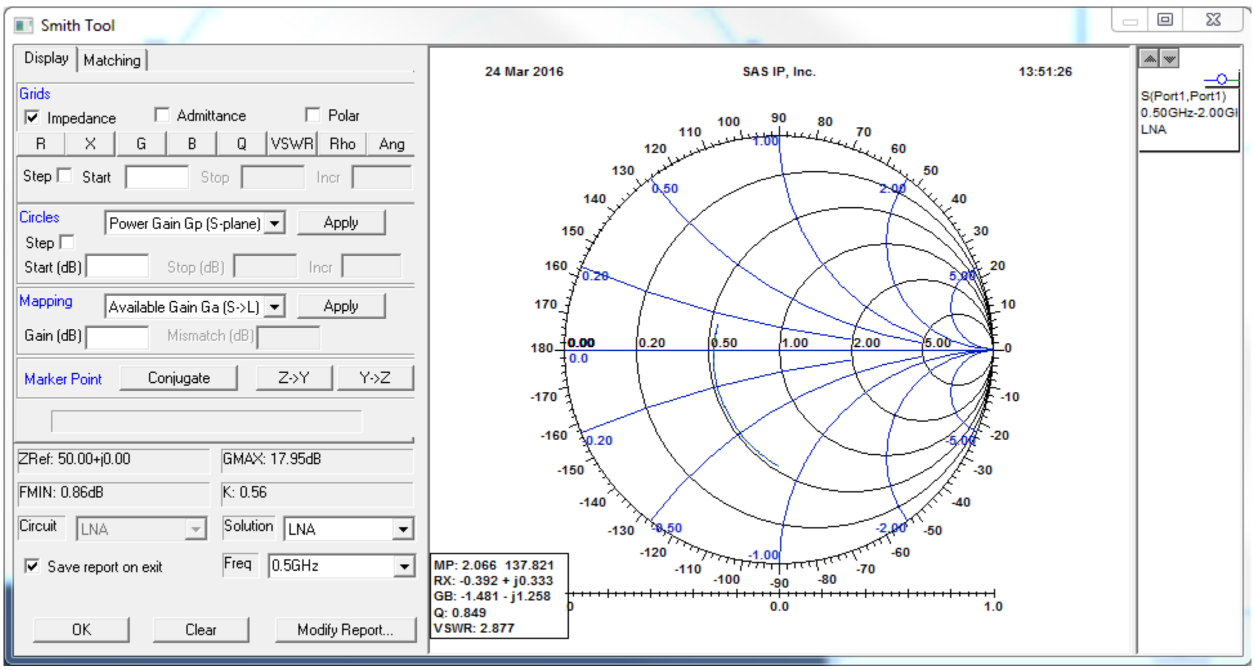
- Reference impedance (**ZRef**)
- Minimum possible noise figure (**FMIN**)

- Maximum gain (**GMAX**)
- Stability factor (**K**) for the selected **Circuit**, **Solution**, and **Freq** uency.

> **Note:** For a Smith Tool session based on a swept-frequency analysis, the **Freq** selection defaults to the lowest frequency in the sweep. Be sure to select the correct frequency as you begin your Smith Tool session.

Smith Tool Display Tab

The **Display** tab is divided into group boxes that enable control of the grid-circle display and reference-plane mapping. The group boxes are: Grids, Circles, Mapping, Marker Point.



Grids in Smith Charts

This group controls the display of grids.

Display any combination of available grids by selecting the following check boxes:

- **Impedance**
- **Admittance**
- **Polar**

The buttons under the **Grid** check boxes allow you to display grids of constant value for eight parameters:

- **R** (resistance)
- **X** (reactance)
- **G** (conductance)
- **B** (susceptance)
- **Q** (quality factor)
- **VSWR** (voltage standing wave ratio)
- **Rho** (reflection coefficient magnitude)
- **Ang** (reflection coefficient angle)

You can specify a grid by typing its value or by clicking in the graph.

To display a grid by typing its value:

1. In the **Start** group box, type the normalized value of the grid you want to display.
2. Click the button in the appropriate **Value** field.

To display a family of grids:

1. Select **Step**.
2. In the **Start** group box, type the normalized value of the lowest-value grid you want to display.
3. In the **Stop** group box, type the normalized value of the highest-value grid you want to display.
4. In the **Incr** group box, type the step increment.
5. Click the button in the appropriate **Value** field.

To display a grid by clicking in the graph:

1. Click the button in the appropriate **Value** field. The mouse cursor moves to the graph center, and the corresponding parameter value (in normalized form) appears in the **Start** group box.
2. Click the graph at any point through which the appropriate grid must pass.

Circles in Smith Charts

This group controls the display of circles in the chart display of the **Smith Tool** window.

The display of constant circles is controlled by the following parameters. All quantities except circles of constant noise-figure are available at the source and load planes.

Parameter	Symbol	Reference Plane
Power Gain	Gp	Source and load planes

Available Gain	Ga	Source and load planes
Stability Factor	K	Source and load planes
	Noise	Source plane
Reflection Coefficient	Refl	Source and load planes

To display a circle:

1. Select a circle type.
2. Type a value in the **Start** group box.
3. Click **Apply**.

To display a family of circles:

1. Select a circle type.
2. Select **Step**.
3. In the **Start** group box, type the normalized value of the lowest-value grid you want to display.
4. In the **Stop** group box, type the normalized value of the highest-value grid you want to display.
5. In the **Incr** group box, type the step increment.
6. Click in the appropriate parameter **Value** field.

Mapping in Smith Charts

This group controls the relationship between the circuit's source and load planes.

To map impedances at one plane to another:

1. Select a mapping function.
2. In the **Gain (dB)** group box, type the appropriate gain at the known plane.
3. Click **Apply:**

   — Circles corresponding to the specified gain in the known and target planes appear in the graph. If you have already plotted a gain circle for the specified value, the known-plane mapping circle overlays.

   — Tracking markers on the known- and target-plane circles indicate points of correspondence between them. As you use your mouse to move the known-plane marker to various impedances on its circle, the target-plane marker tracks to indicate the target-plane impedance that produces the specified gain.

4. When you have moved the known-plane marker to the impedance you want to map, click to mark the source and load-plane impedances on the graph. The source and load-plane gain circles remains on the graph.

Mismatch Mapping in Smith Charts

Mismatch S →L and **Mismatch L →S** mapping functions allow you to explore the sensitivity of a network's gain to the variations in source and load impedance.

- **Mismatch S →L** plots a *load*-mismatch circle.
- **Mismatch L →S** plots a *source*-mismatch circle.
- If you enter 0 in **Mismatch** or leave it blank, Smith Tool assumes a tolerance of 0 and displays a mismatch circle of zero radius.
- Terminating the network with an impedance within a **Mismatch S →L** circle, or exciting the network with a source having an impedance with a **Mismatch L →S** circle, results in a gain in excess of **Gain (dB)** –|**Mismatch**| dB.

To implement mismatch mapping:

1. Select the appropriate response in the mapping response list.
2. In the **Gain (dB)** group box, enter the target gain.
3. In the **Mismatch** group box, enter the tolerable deviation on the target gain value, then click **Apply**.

   — Known-plane and target-plane circles appear; tracking markers on the circles indicate points of correspondence between them.

4. Click the appropriate known-plane impedance to mark the target-plane impedance and plot a mismatch circle around it.

Marker Points in Smith Charts

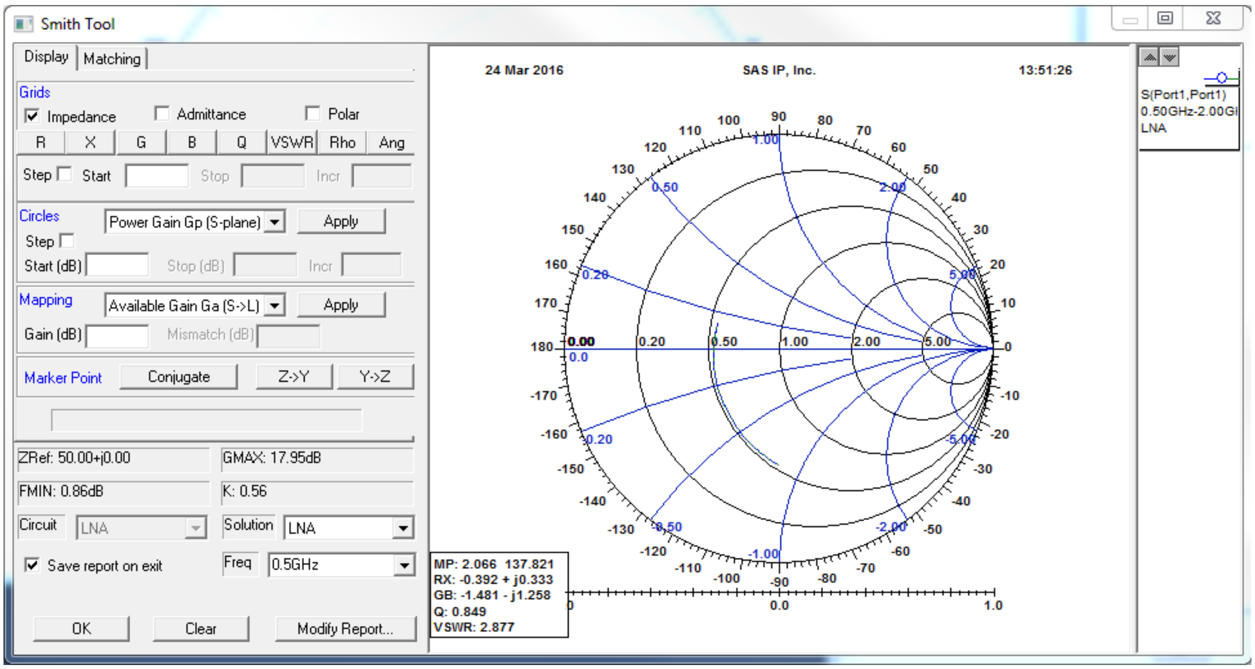
This group allows you to mark the conjugate of a point, transform an admittance to an impedance, and transform an impedance to an admittance.

To mark the conjugate of a point:

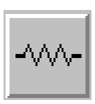
1. Click **Conjugate**.
2. Move the cursor to the appropriate point (for example, 0.5–*j*0.2) and click. A circle appears at the conjugate point (in this case, 0.5+*j*0.2). The **X →Y** and **Y →Z** functions work similarly.

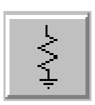
Smith Tool Matching Tab

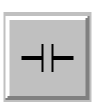
The **Matching** tab in the **SmithTool** window allows you to add matching networks made up of various components including series and shunt resistors, capacitors, inductors, transmission lines, stubs, and transformers.

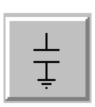


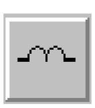
Each component is selectable by clicking a corresponding button in the **Matching** tab window:

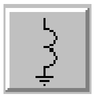

Resistor

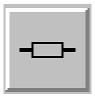

Grounded Resistor

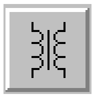

Capacitor

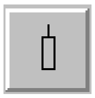

Grounded Capacitor

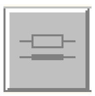

Inductor

 Grounded Inductor

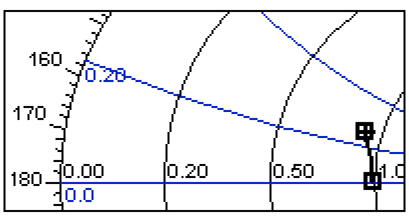
 Transmission Line Shorted Stub

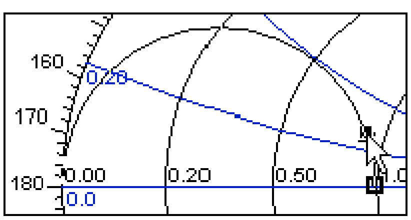
 Ideal Transformer

 Open Stub

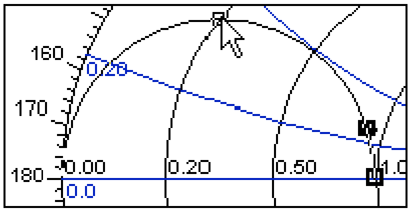
 Transmission Line with Reference Node (Nexxim only)

These matching components become available when you start a matching-network design (by clicking **New Match**). Once you have begun a match, add it to the network by clicking any matching component. When you add a component to the network, an arc that depicts a segment of its impedance-transformation range appears on the graph:

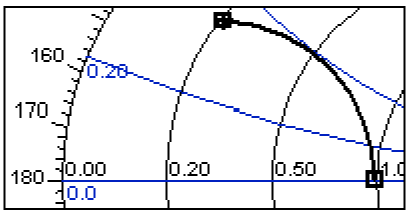




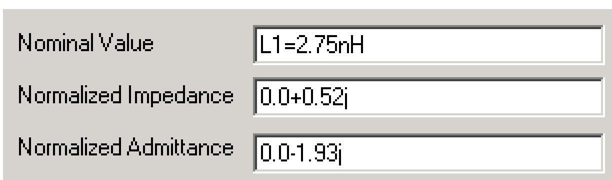
Drag the adjustment box on the arc to select a value:



Release the adjustment box to set the value:



The **Nominal Value**, **Normalized Impedance**, and **Normalized Admittance** group boxes update to show the value of the new component:



- To change the value of an existing matching component, do either of the following:

  — Drag its adjustment box to a new value.

  — Select it by clicking its adjustment box, type a new value into the **Nominal Value**, **Normalized Impedance**, or **Normalized Admittance** field, then press **Enter**.

- To remove the most recently placed matching component, click **Delete Last**. Click **Delete Last** repeatedly to remove multiple components.
- To copy your matching network to the current circuit as a subcircuit, click **Export**.

Note:

- You can create multiple matching networks during a Smith Tool session.
- To selectively export one of the networks, click on the network to select/highlight it , then click **Export**.
- If multiple matching networks exist and you do not select one before clicking **Export**, Smith Tool automatically exports the most recently created network.

Creating Matching Networks in Smith Tool

To create a matching network with the Smith Tool:

1. Click **New Match** to place a starting-point impedance on the Smith chart. The cursor moves to the graph center.
2. Move the cursor to the impedance you want to match, and click it.
3. Click a component button.

An arc on the Smith chart shows the effect of the impedance transformation the component performs. Crosshairs in the box at one end of the arc identify that end as adjustable.

4. Click the adjustment box ⊞ to open the locus of values available for that component.
5. Adjust the component’s value by dragging its adjustment box to the appropriate point on its arc.

As you place and adjust matching components, the values shown in the **Nominal Value**, **Normalized Impedance** and **Normalized Admittance** group boxes update to show the value of the current component.

Modify the value of an existing matching component with your mouse or your keyboard.

- To adjust a matching component with your mouse:

— Click the component’s adjustment box with your mouse.

— Drag the box adjustment box to a new value.

- To adjust a matching component with your keyboard:

— Click the component’s adjustment box to select it.

— Type a new value in the **Nominal Value**, **Normalized Impedance**, or **Normalized Admittance** group box.

— Press **Enter**.

6. Continue placing appropriate components to transform the impedance until you reach the chart origin (or your target impedance).

— At any time, click the endpoint of any existing matching component and drag it to a new value. The display updates to show the new match.

— To delete the current component, click **Delete Last**. Click **Delete Last** repeatedly to undo the current matching session element by element.

7. Once you're satisfied with the match, click **Export**.

Your new matching network is copied to the current circuit as a subcircuit.

> **Note:** You can create multiple matching networks during a given Smith Tool session. To selectively export any one among multiple existing networks, select it—click it to highlight it — before you click **Export**. If multiple matching networks exist and you do not select one before clicking **Export**, Smith Tool exports the most recently created network.

13 - Circuit Nexsys Analyses

The Circuit simulator designs can incorporate behavioral components on the Nexsys component library. Circuits composed of electrical and behavioral components can be simulated using Nexsys time domain and frequency domain analyses

Nexsys Discrete Time-Domain Analysis

In a Nexsys design, functional and electrical components and sub-designs may be connected arbitrarily. Nexsys analysis seamlessly integrates with Nexxim Transient analysis and Envelope analysis to solve mixed-mode design problems at differing levels of abstraction. Nexsys time domain analysis allows the simulation of arbitrary wired/wireless communications system topologies and other system-level applications.

For details on Nexsys Analysis, see Nexsys Discrete Time Domain Analysis.

System Frequency-Domain Analysis

System (Nexsys) frequency domain analysis allows you to evaluate and troubleshoot RF/high-frequency designs for wireless and microwave applications.

Results available with System Frequency Domain analysis include:

- The output power (**Pout**) at all external output ports.
- The voltage and power spectra and steady state time domain responses at the points where Nexsys voltage probes have been placed.
- The analysis results may be viewed in the spectral domain or in the time domain.

For more information, see System Frequency Domain Analysis.

For more information, see Frequency Dependent Sources in Nexxim.

Nexsys Discrete Time Domain Analysis

Nexsys discrete time domain simulation is a comprehensive simulation environment for mixed-mode systems containing both functional and electrical components.

- *Functional* components are time-domain behavioral components with an input-output transformation and a unidirectional data flow (signals flow in one direction on the input to the output).
- *Electrical* components are bi-directional components (signals propagate in both directions). Linear electrical networks can be described in the frequency domain using S-parameters or the equivalent. Nonlinear components such as transistors and diodes use time domain models.

In a Nexsys design, functional and electrical components and sub-designs may be connected arbitrarily. Nexsys analysis seamlessly integrates with Nexxim Transient analysis and Envelope analysis to solve mixed-mode design problems at differing levels of abstraction. Nexsys time domain analysis allows the simulation of arbitrary wired/wireless communications system topologies and other system-level applications. A Matlab user-defined model capability allows the designer to take advantage of the enormous existing Matlab functionalities in DSP, communications, vector manipulation and optimization to co-simulate with Nexsys.

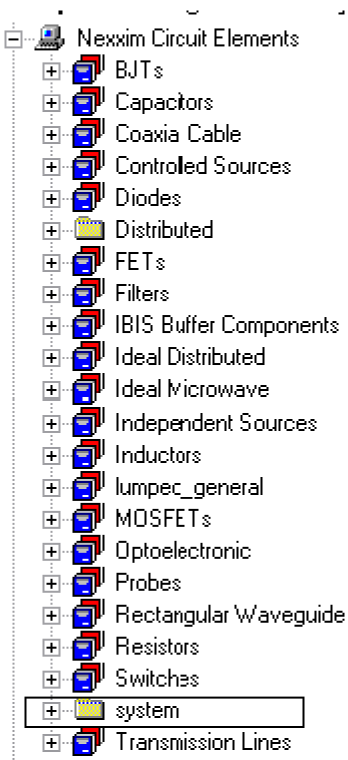
Creating and Simulating a Nexsys Design

A Nexsys design starts with a Circuit design project.

Open an existing Circuit design, or start a new Circuit Design. Then use the Components window to select and place both Circuit components for Nexxim and System components for Nexsys. When the circuit is complete, set up and run the analyses.

Add Nexsys and Nexxim Components

The Nexsys functional components are in the System folder under Nexxim Circuit Elements.



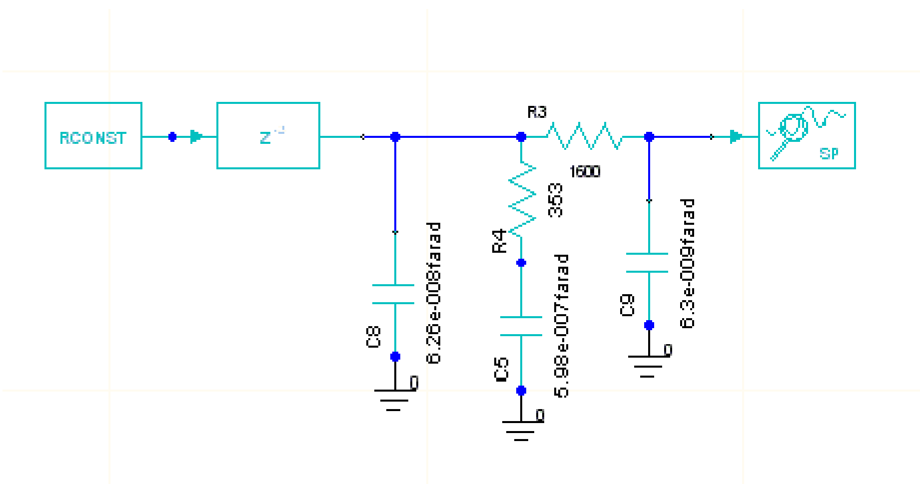
Place one or more of the Nexsys components in the Nexxim design. Nexxim automatically sets up the Nexsys component for discrete time domain simulation.

When mixing Nexxim and Nexsys components, two important issues are the selection of sources and the need for special interface elements.

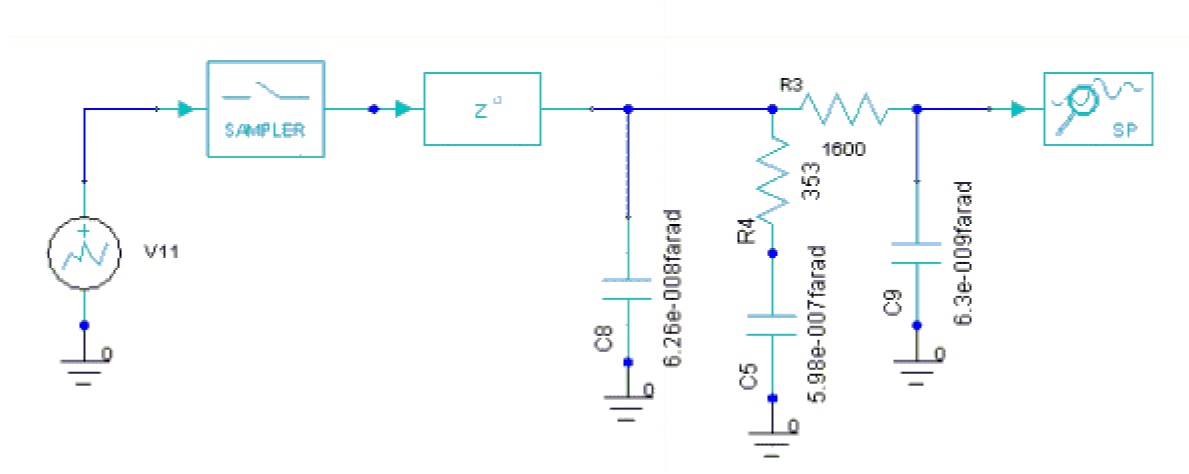
Nexxim and Nexsys Sources

The library used for voltage, current, and power sources determines the timestep and duration to be used in the time domain analysis.

- When sources are on the System folder, the time step is fixed and is taken on the sample rates in the System sources.
- Any timestep or stop time set in the Transient analysis setup is ignored.
- In the loop_filter_1 example, the RCONST source determines the simulation length and the timestep with its parameters NSAMP and SAMPLE_RATE.



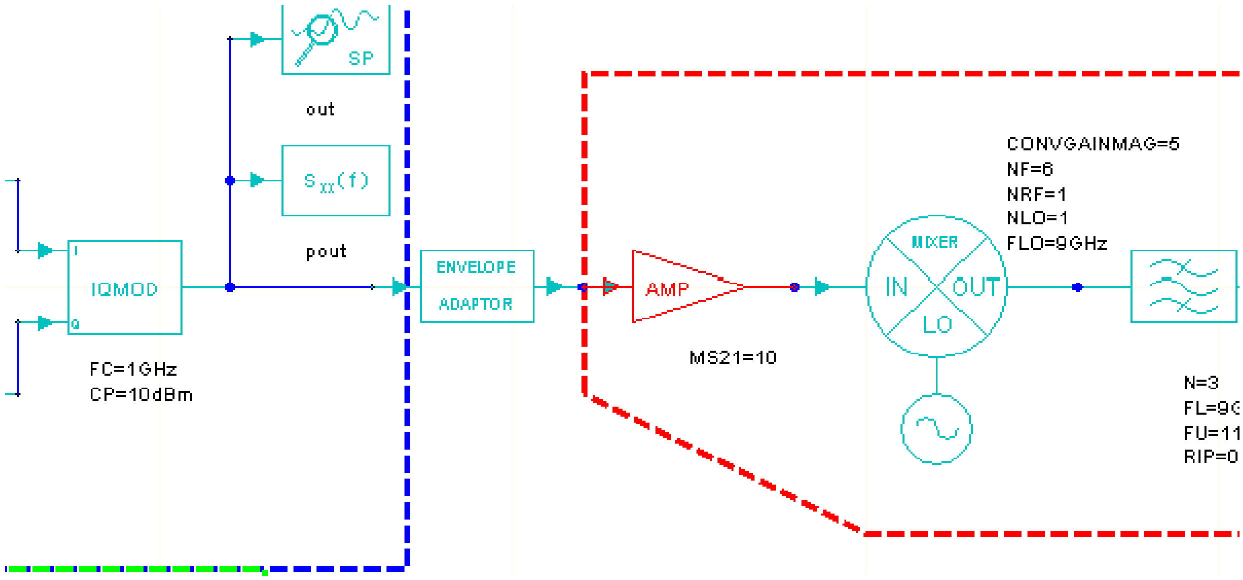
When any source is an interface source, the stop time taken on the Transient analysis setup, and a SAMPLER component is required to interface the interface source to the Nexsys behavioral subsection and provide the timestep (inverse of the SAMPLER sample rate). In the loop_filter_2 following example, the interface source sets the simulation length to the Stop setting in the Transient analysis setup. The timestep is provided by the SAMPLER component.



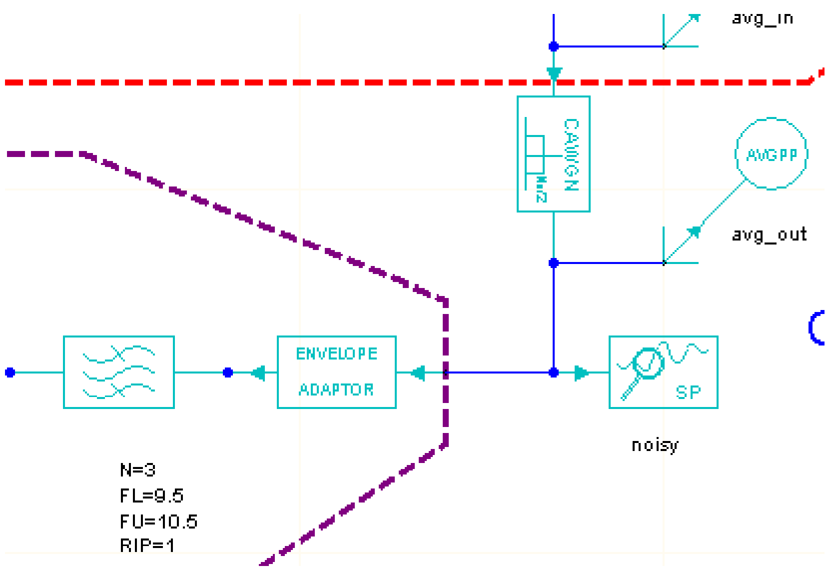
Interfacing Elements

When a Nexxim continuous source connects to a Nexsys behavioral component, a **SAMPLER** component must be placed between them to prevent a fatal simulation error (See the loop_filter_2 example above). The **SAMPLER** component is located under the System subdirectory of Nexxim components in the **Miscellaneous** folder. The sampling rate and number of samples set for the **SAMPLER** component control the time domain analysis.

Following are two examples using the **ENVELOPEADAPTOR** component, on the MPSK_Nexxim_System example in the Examples directory.



In the following circuit, the Envelope adaptor interfaces the behavioral channel element with the electrical elements in the RF Receiver.



- In the circuit fragment above, the Envelope adaptor is required to connect the Baseband transmitter stage.

- With only behavioral components, the RF transmitter stage includes electrical components.
- The Envelope adaptor identifies the waveform as an envelope to the Nexxim portion of the simulation.
- The Nexxim simulation then knows to send on an envelope waveform.

Set Up and Run Nexxim and Nexsys Analyses

The Nexsys time domain simulation shares the same analysis setup with transient analysis.

To add a Transient analysis solution setup.

1. Right-click the **Analysis** menu in the **Project Manager** window to open a menu. Select **Add Nexxim Solution Setup** on the menu, then slide the cursor to select **Transient Analysis** on the subordinate menu.
2. Set the **Step** and **Stop** times to control the analysis when an interface source is driving the system. These analysis control parameters are ignored by the Nexsys simulator if there are no interface sources present in the design.
3. Click **OK** to close the Transient Analysis Setup.
4. Click the project and select **Analyze All** on the menu.

Display Nexsys Simulation Results

The results of Nexsys discrete time simulations may be viewed in one of the following domains in the Reporter:

- Sweep Domain
- Time Domain (Standard, Constellation, Eye Diagram and Statistical report types)
- Spectral Domain

Nexsys Probes

Obtaining measurements for discrete time analysis requires the use of **probes**. Some probes are applicable only to the **Sweep Domain.** Examples of such probes include the **BERP** and **ACPRP** probes (since the BER and ACPR responses are typically plotted against some swept parameter). Signal, voltage, and power probes are applicable only to the **Time** and **SpectralDomains**, since the response obtained on these probes is typically plotted against time or frequency. Time and Spectral displays may also be viewed for different swept values. A special class of probes provides statistical displays of time domain signals, such as histogram and CCDF probes.

Currently, the following probes are available:

ACPRP: Adjacent Channel Power Ratio Probe (Sweep Domain).

AVGPP: Average Power Probe (Sweep Domain)

BERP: Bit Error Rate Probe (Sweep Domain)

CCDFP: Complementary Cumulative Distribution Function Probe (Statistical Time Domain)

CDFP: Cumulative Distribution Function Probe (Statistical Time Domain)

CFP: Crest Factor Probe (Sweep Domain)

EVMP: Error Vector Magnitude Probe (Sweep Domain)

FTRAJP: Frequency Trajectory Probe (Time Domain) for tracking frequency changes in a PLL. This probe may be connected to the output of a VCO.

HISTP: Histogram Probe (Statistical Time Domain)

PAPP: Peak-to-Average Power Probe (Sweep Domain)

PDFP: Probability Density Function Probe (Statistical Time Domain)

PSDP: Power Spectral Density Probe (Spectral Domain).

RMSP: Root Mean Square Probe (Sweep Domain)

SNDP: Signal to Noise and Distortion Probe (Sweep Domain).

SP: Signal Probe (Time and Spectral Domains)

PNP: Phase Noise Probe.

For more information on the measurements obtained by each probe, See the probe component help topics.

Generating Reports

Once the signal analysis is performed, bring up the **Create Report** editor to open the results for the Standard, Constellation, Eye Diagram or Statistical Reports. After selecting the solution (analysis) of interest and one of the available domains (**Time**, **Spectral** or **Sweep**) on the **Domain** field, select the probe responses available for display on the **Quantity** field of the editor. Once a specific probe response is selected, a list of functions and units in the **Function** field is displayed. The user can then view the probe response in any of the forms available in the **Function** list.

For details see Generating Reports and Post-processing.

Overview of Nexsys Discrete Time Domain Analysis

A typical communications system for Nexsys discrete time domain simulation (Figure 1) consists of:

- A propagation channel, such as AWGN and Multi-path Fading channels
- RF functional and electrical bandpass components for the RF transceiver
- Functional baseband components for transmitter/receiver digital-signal processing

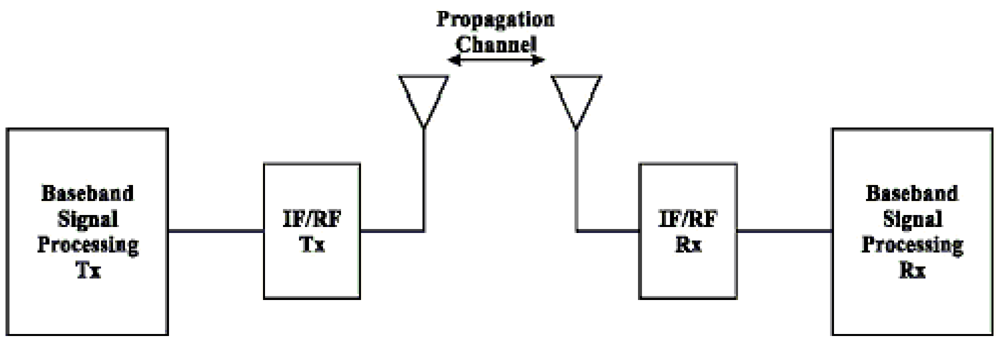


Figure 1 – General Wireless Communications System

Arbitrary transmit/receive topologies may be simulated using discrete time analysis, including:

- Carrier recovery
- Phase locked loops
- Bit synchronization
- Frequency synthesizers
- Symbol/bit-timing recovery
- Arbitrary feedback and digital signal processing topologies

Discrete time-signal analysis yields system performance (BER/ACPR) and a probable response in the time/spectral/sweep domains at any node. Discrete time-signal analysis applies both to mixed mode systems and also to systems with functional components only.

Signal and Noise Waveforms

The signals and waveforms present in a system are Baseband Signals, Bandpass Signals, and Noise Waveforms

Baseband Signals

The general form of any baseband signal may be represented in the following form:

$$S(t) = V_S(t)$$

where $V_S(t)$ represents a time-varying voltage signal.

Bandpass Signals

A baseband signal becomes a bandpass signal once it modulates an RF carrier. The general form of any bandpass signal is:

$$S(t) = V_S(t)\cos(2\pi f_c t + \theta_S(t))$$

$$= I_S(t)\cos(2\pi f_c t) - Q_S(t)\sin(2\pi f_c t)$$

Where f_c is the carrier frequency and θ_S is the phase of the input modulated signal.

The quantity

$$\tilde{V}_S(t) = V_S(t)\exp\{2\pi j\theta_S(t)\} = I_S(t) + jQ_S(t)$$

is known as the complex envelope of the bandpass signal $S(t)$. $I_S(t)$ and $Q_S(t)$ are the In-phase and Quadrature-phase baseband information-bearing signals.

Noise Waveforms

Noise is random fluctuations of a signal. This randomness is typically governed by a statistical distribution. For example, a White Gaussian noise process has a Gaussian distribution. The noise processes supported by the Nexsys discrete time analysis are assumed stationary (the statistical distribution of the noise level does not vary with time).

A baseband random noise signal may be represented in the following form:

$$N(t) = V_N(t)$$

where $V_N(t)$ is a time-varying noise voltage.

A bandpass random noise signal is represented in the following form:

$$N(t) = V_N(t)\cos(2\pi f_c t + \theta_N(t))$$

where f_c is the carrier frequency.

The quantity

$$\tilde{V}_N(t) = V_N(t)\exp\{2\pi j\theta_N(t)\} = I_N(t) + jQ_N(t)$$

is known as the complex envelope of the bandpass Noise $N(t)$. $I_N(t)$ and $Q_N(t)$ are the In-phase and Quadrature-phase baseband noise-bearing signals.

Any noise process may be classified as uncorrelated or correlated. An uncorrelated noise process implies that the noise samples $N(t)$ at time t and $N(t + dt)$ at time $t + dt$ are not correlated. A correlated process implies that the samples $N(t)$ and $N(t + dt)$) tend to be correlated. An example of an uncorrelated process is White Gaussian noise. Examples of correlated processes are colored Gaussian noise and Rayleigh fading.

The power spectral density of a noise process typically represents the correlation. For weakly correlated processes, this power spectral density tends to be wideband. A strongly correlated process tends to have a narrowband power spectral density.

Discrete Time Simulation of Signals and Noise

In discrete time simulation, all waveforms (signal and noise) are represented in a discrete (sampled) form. This implies that baseband signal and noise waveforms may be simulated using the discrete signals

$$S(nt_s) = V_S(nt_s) \quad n \geq 0,$$

and

$$N(nt_s) = V_N(nt_s) \quad n \geq 0,$$

where

f_s = Waveform sampling frequency or sampling rate,

and $t_s = \frac{1}{f_s}$ = Waveform simulation timestep.

The Nyquist criterion states that if a continuous time signal $S(t)$ is sampled at a rate that is at least twice its highest frequency content (i.e., bandwidth), this signal may be completely reconstructed from its sampled version $S(nt_s)$. The minimum sampling rate needed to completely recover a continuous signal from its sampled version is known as the Nyquist rate.

It is the responsibility of the user to be aware of the signal $S(t)$ bandwidth. It is highly recommended that the actual simulation sampling rate be chosen well above the Nyquist rate for more accurate simulations. This added accuracy typically results in a longer simulation time.

Most bandpass signals encountered in practical communications systems are narrowband signals. A narrowband bandpass signal has a bandwidth that is much less than its carrier frequency. The Nexsys discrete time simulator can utilize envelope modulation techniques where only the complex envelope of the bandpass signal or band-limited noise is sampled, as opposed to sampling the carrier as in Nexxim transient analysis. As a result, the Nexsys simulation sampling rate may be orders of magnitude less than that of a Nexxim simulation. This translates into a much faster simulation time.

The information of a continuous bandpass signal and noise may be recovered from knowledge of their corresponding sampled complex envelope signals:

$$\tilde{V}_S(nt_s) = V_S(nt_s)\exp\{2\pi j\theta_S(nt_s)\}$$

and

$$\tilde{V}_N(nt_s) = V_N(nt_s)\exp\{2\pi j\theta_N(nt_s)\}$$

respectively, and the carrier frequency f_c.

Thus, the information of a continuous bandpass signal and noise may be recovered on the triplets ($I_S(nt_s)$, $Q_S(nt_s)$, f_c) for the signal information and ($I_N(nt_s)$, $Q_N(nt_s)$, f_c) for the noise information.

To ensure that the Nyquist criterion still holds for narrowband bandpass signals, the user needs to choose a simulation sampling rate that is at least twice the highest frequency content (bandwidth) of the complex envelope .

$$\tilde{V}_S(t) = V_S(t)\exp\{2\pi j\theta_S(t)\}$$

The choice for the simulation sampling rate f_s affects the outcome of the simulation. Choosing a simulation sampling ratef_s that is higher than $2f_c$ results in the sampling of the carrier. In this case, the bandpass signal with a complex envelope representation is converted to a baseband real signal of the form:

$$S(nt_s) = V_s(nt_s)\cos(2\pi f_c nt_s + \theta_S(nt_s)) \quad n \geq 0$$

,

$$N(nt_s) = V_N(nt_s)\cos(2\pi f_c nt_s + \theta_N(nt_s)) \quad n \geq 0$$

,

Conversely, time domain simulation results for bandpass signals with a sampling rate

$$f_s \leq 2f_c$$

may be viewed only in terms of their complex envelopes.

Discrete Time Simulation of Nonlinear Behavioral Components

The nonlinear discrete time simulation technique described here is used for processing bandpass modulated signals through nonlinear behavioral components (e.g., amplifiers, mixers, frequency multipliers). Generally, nonlinear behavioral components are not only power dependent, but also frequency and/or temperature dependent. An equivalent model is used for nonlinear components in discrete time simulation to separate the power dependent characteristics from frequency and temperature dependent characteristics.

All nonlinear behavioral two-port components are assumed unidirectional (e.g., S_{12} = 0). Each nonlinear component is partitioned into three segments: a linear electrical (active) input component, a nonlinear functional component, and a linear electrical (passive and noiseless) output component. This arrangement is shown in Figure 2.

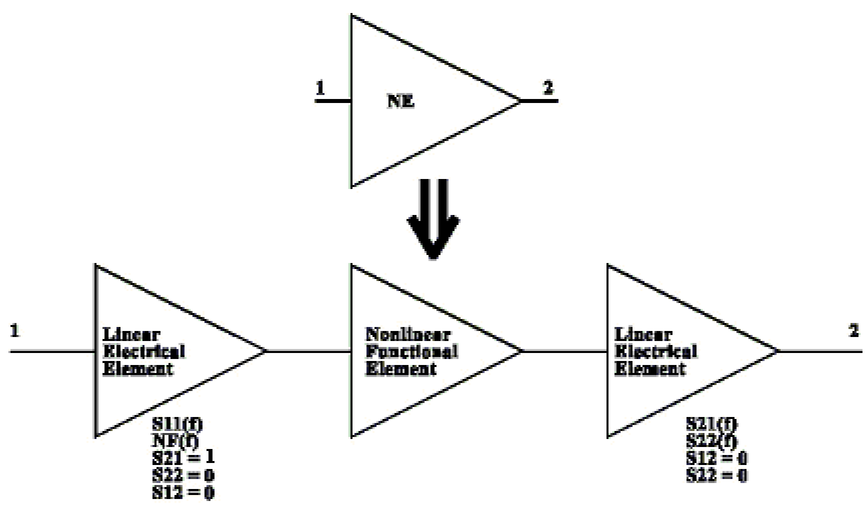


Figure 2– Equivalent Model for Discrete Time Simulation of Nonlinear Electrical Components

In this model, the power dependent characteristics of S_{11}, S_{12} and S_{22} are ignored. The frequency and temperature dependent small signal S_{11} and NF are associated with the input linear electrical component. The power dependent characteristics of S_{21} are associated with the nonlinear functional component. The frequency and temperature dependent small signal S_{21} and S_{22} are associated with the output linear electrical component.

For discrete time simulation, the linear electrical input and output components are associated with other connected linear electrical components and simulated as a linear electrical sub-design as discussed previously.

The nonlinear characteristic of a two-port component is described by its nonlinear figures-of-merit or nonlinear measured data. Only the nonlinear characteristic of S_{21} is considered in discrete time analysis techniques even though S_{11}, S_{12} and S_{22} may possibly be power dependent too.

> **Note:** For the two-port nonlinear mixer model, the arrangement in Figure 2 still holds. The nonlinear functional component is associated with the actual frequency conversion and discrete time phase noise simulation. As a result of that, the input linear electrical component in Figure 2 is associated with the mixer's input frequency while the output linear electrical component is associated with the mixer's output frequency.

Modeling Nonlinearity with Polynomial Power Series

It is always assumed that nonlinear measurements are obtained when the input and output ports of the nonlinear component are terminated in 50Ω. Nonlinear measurements of a two-port component typically include the AM-AM and AM-PM effects. This nonlinear relationship between $S_{21}$and P_1 (the available input power) or, equivalently, between P_{out} and P_1 is represented by the following power series polynomials:

$$\sqrt{P_{out}}\cos\phi = a_1X + a_3X^3 + \ldots$$

$$\sqrt{P_{out}}\sin\phi = b_1X + b_3X^3 + \ldots$$

where

P_1 = the available input power on the source (with RS = 50Ω)

$P_{out} = |S_{21}|^2 P$
= the output load power (assuming RL = 50Ω)

$$X = \sqrt{P_1}$$

S_{21ss} = the small signal gain.

$$\phi = \angle S_{21} - \angle S_{21ss}$$

The coefficients a_1, a_3, a_5... and b_1, b_3, b_5... are calculated using a least-squares curve fitting technique based on the user-supplied measurements P_1 - P_{out} data or P_1 - S-parameters data. For example, a set of coefficients can be obtained based on the following power amplifier P_1 - P_{out} measured data (in 50ohm terminations).

RTH_PA 2-port

POUT dBm

P1 dBm, FREQ = 900MHz

* P1	Pout	Phase (degrees)
5.00	25.68	88.75
7.00	27.67	88.75
9.00	29.66	88.75
11.00	31.64	88.75
13.00	33.61	88.75
15.00	35.56	88.75
17.00	37.48	88.76
19.00	39.35	88.76
21.00	41.16	88.76
23.00	42.86	88.75
25.00	44.38	88.71
27.00	45.65	88.66
29.00	46.53	88.76
31.00	47.17	91.85
33.00	47.50	97.08
35.00	47.66	102.81

Note:

If the error obtained using the least squares curve fitting technique for the a and b coefficients exceeds 1e-5, an alternate approach based on cubic spline interpolation is used to compute the output power level for a given input signal power

If the input signal power to the nonlinear component exceeds the maximum supplied measured input power P1, the simulator assumes the last supplied output power entry in the measured data to be the saturation power.

The above mentioned data is given at FREQ = 900MHz. Additional nonlinear measured data at other frequencies may be provided. The discrete time analysis is capable of locating the actual operating point using multi-dimensional data interpolation. For more information, please See the Nonlinear RF Component Models documentation.

If the user chooses to provide the nonlinear figures-of-merit (OIP3 or P1dB and Psat) instead of measurement data, the power series coefficients are approximated by

$$a_1 = |S_{21}|$$

$$a_3 = \frac{-\left(\frac{4}{3}\right)|S_{21}|^3}{OIP3}$$

$a_i = 0$, $i = 5,7, \ldots$

$b_i = 0$, $i = 1,3,5, \ldots$

where:

S_{21} is the linear small signal gain

OIP3 is the output power at the third order intercept point.

In this case, the simulations tend to be less accurate.

Calculating a Nonlinear Output Voltage

Assuming a bandpass modulated input signal of the form:

$$S(t) = V_s(t)\cos(\omega_c t + \theta_s(t)) = I_s(t)\cos\omega_c t - Q_s(t)\sin\omega_c t$$

where

$V_s(t)$ is the baseband input voltage.

$$\theta_s(t)$$

is the phase of the input modulated signal.

$$I_s(t) = V_s(t)\cos\theta_s(t)$$

is the In-phase envelope of the input signal.

$$Q_s(t) = V_s(t)\sin(\theta_s(t))$$

is the Quadrature-phase envelope of the input signal.

$$\omega_c = 2\pi f_c$$

is the carrier frequency in radians per second, where f_c is the carrier frequency in Hz.

The nonlinear output voltage is calculated as:

$$S_{out}(t) = V_{out}(t)\cos(\omega_c t + \theta_{out}(t)) = I_{out}(t)\cos\omega_c t - Q_{out}(t)\sin\omega_c t$$

where:

$$V_{out}(t) = \sqrt{P_{out}(t)R_{out}} = \sqrt{I_{out}^2(t) + Q_{out}^2(t)}$$

$$\theta_{out}(t) = \operatorname{atan}\left(\frac{Q_{out}(t)}{I_{out}(t)}\right)$$

$$I_{out}(t) = \sqrt{Rout}\{ GIV(t)\cos\theta_s(t) - GQV(t)\sin\theta_s(t)\}$$

$$Q_{out}(t) = \sqrt{Rout}\{ GIV(t)\sin(\theta_s(t)) - GQV(t)\cos(\theta_s(t))\}$$

with

$$GIV(t) = \left(\frac{a_1}{MS_{21}}\right)X(t) + \left(\frac{a_3}{MS_{21}}\right)X(t)^3 + \left(\frac{a_5}{MS_{21}}\right)X(t)^5 + \dots$$

$$GQV(t) = \left(\frac{b_1}{MS_{21}}\right)X(t) + \left(\frac{b_3}{MS_{21}}\right)X(t)^3 + \left(\frac{b_5}{MS_{21}}\right)X(t)^5 + \dots$$

and

$$X(t) = \sqrt{P_1(t)}$$

$$P_1(t) = \frac{V_s^2(t)}{4Rin}$$

$$MS_{21} = \sqrt{a_1^2 + b_1^2}$$

The source resistance *Rin* and the load resistance *Rout* are always defaulted to 50Ω, but they can be individually specified for each input and output, respectively.

Nexsys Simulation of a Mixed Mode Topology

A mixed-mode communications system is composed of functional (time based) and electrical (frequency based) components. Functional components are time-domain behavioral components with an input-output transformation and a unidirectional data flow. Electrical components are physical and bidirectional, and may use frequency-domain descriptions such as S-parameters.

This topic provides information on how Nexsys handles mixed mode systems.

Nexsys Partitioning and Scheduling Process

Prior to running the discrete time system simulation, Nexsys partitions the mixed mode system, converting the mixed-mode topology to a purely functional topology where the signal flow is unidirectional. Electrical subcircuits must be modeled functionally or behaviorally after accounting for all impedance mismatches, noise, and other electrical effects.

After finishing the partitioning is completed, the process of scheduling all components inside the system for discrete time simulation begins. Since discrete time signals flow unidirectionally from input to output, the order in which components are scheduled is critical during discrete time simulation.

The scheduling process begins with the source components in the system , then iteratively schedules each remaining component (after ensuring that all components leading to the input of that component have already been scheduled). A **SAMPLER** component is placed at the physical/functional interface where each interface source connects to a Nexsys component. The sampling rate specified for the **SAMPLER** sets the Nexsys timestep.

During partitioning, Nexsys packs each set of connected electrical components into an internal functional model based on the signal type (baseband or bandpass) of the input to the electrical components.

Nexsys Transient Co-simulation Model

When the input signal type is baseband, Nexsys uses a Transient Co-simulation model (tran_ cosim_model). When the Nexsys time domain simulation encounters a tran_cosim_model, the transient analysis engine is invoked and stays open during the course of the simulation. Data is exchanged at each timestep between the top-level Nexsys simulator and the transient engine. (See Figure 3)

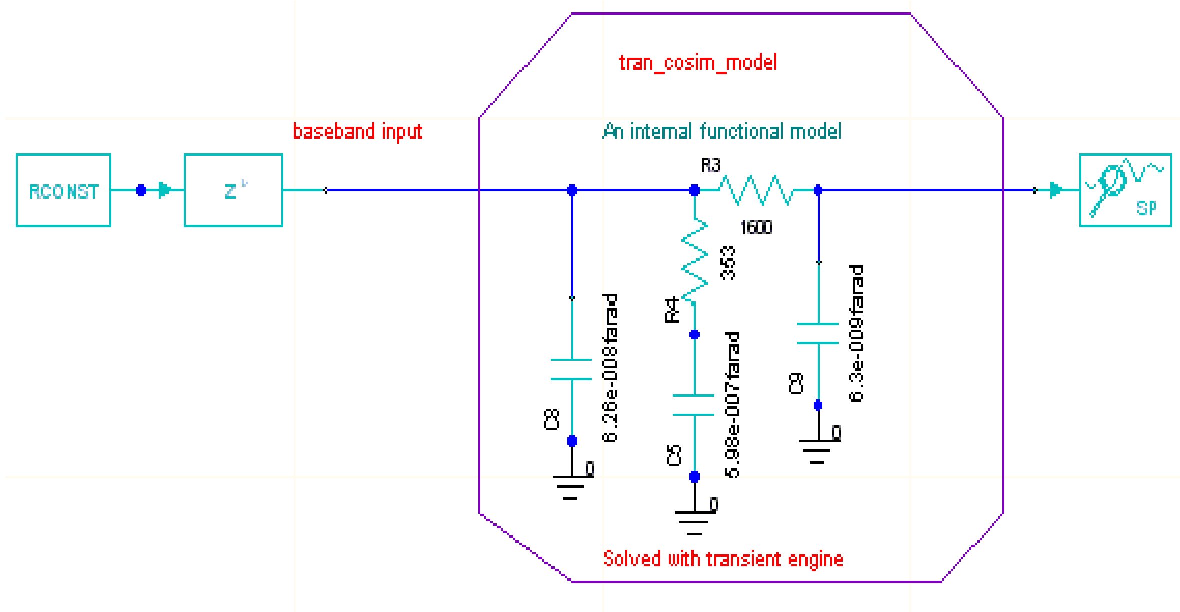


Figure 3 – Nexsys Transient Cosimulation Model

Envelope Co-simulation Model

When the input signal type is bandpass, Nexsys uses an Envelope Co-simulation model (envelope_cosim_model).

An **ENVELOPEADAPTOR** component must be placed at the interface of a bandpass or modulated signal entering a Nexxim subcircuit from a Nexsys functional component. The **ENVELOPEADAPTOR** component helps the simulation engine to partition the system for simulation and to select the appropriate analysis for the Nexxim subcircuit.

The envelope_cosim_model employs two simulation strategies, one for linear subcircuits and another for nonlinear subcircuits such as amplifiers, mixers, and frequency multipliers.

Linear Subcircuits. When the sub-circuit is linear, the top level Nexsys simulation engine calls Nexxim Linear Network analysis (LNA) to obtain the frequency response and convert it to the time domain impulse response. Convolution is then used to calculate the output signal. (See Figure 4).

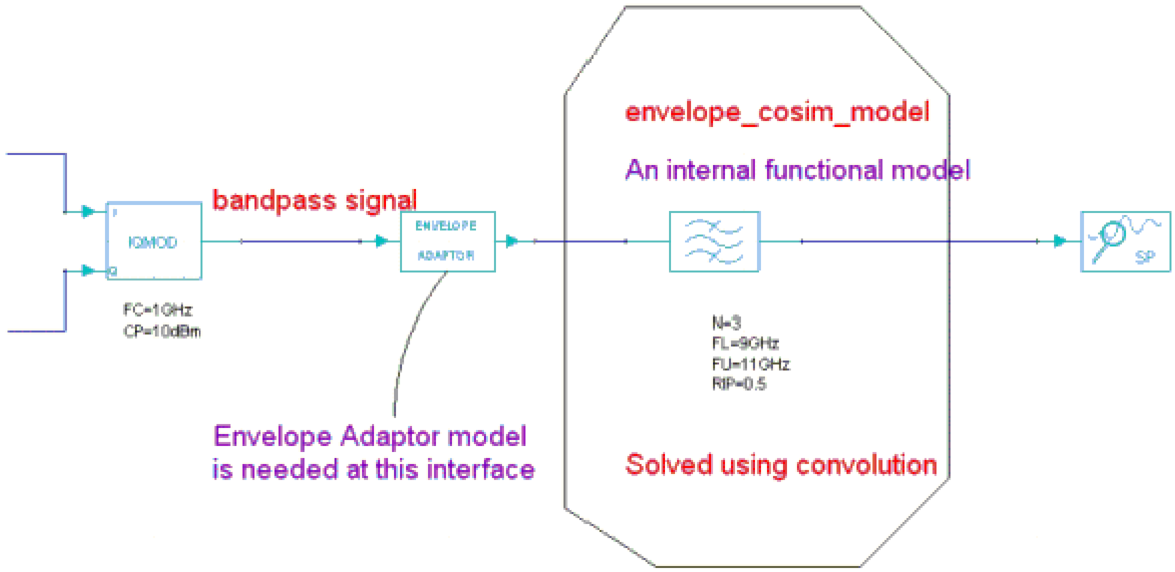


Figure 4 – Nexsys Envelope Cosimulation Model: Convolution Analysis of Linear Electrical Sub-Circuit with Envelope Input

Nonlinear Subcircuits. When the Envelope Co-simulation model sub-circuit is nonlinear, the top level Nexsys simulation engine runs Nexxim Envelope Analysis to solve the circuit and interactively exchange data as described above for the tran_cosim_model. (See Figure 5).

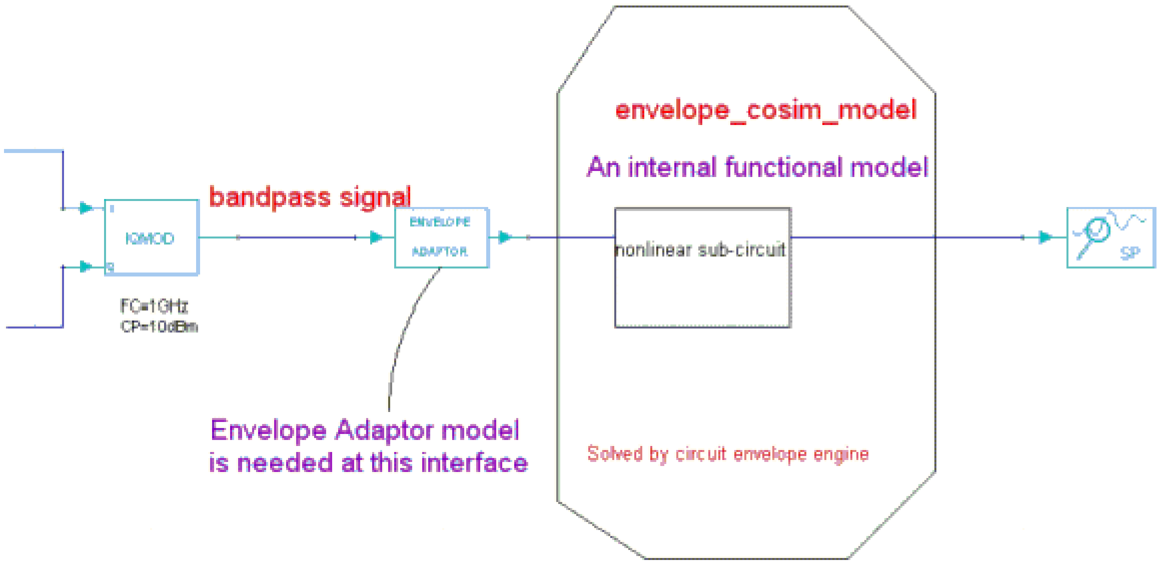


Figure 5 – Nexsys Envelope Cosimulation Model: Circuit Analysis of Nonlinear Electrical Sub-Circuit with Envelope Input

Scheduling Feedback Loops

Scheduling of feedback loops starts with the first component in the feed-forward path and continues around the loop until the feedback input is reached. To prevent a possible deadlock in the discrete time simulation of a feedback loop, the feedback signal in the loop is primed at time *t*=0 with one discrete time sample having a zero value. The user instead can force the priming to occur anywhere inside the feedback loop by placing an RDELAY or CDELAY component at the appropriate priming point. If the simulator does not detect the presence of any RDELAY or CDELAY components in the feedback loop, Nexsys automatically performs the minimum number of primings needed to resolve the deadlock.

An example of a simple feedback loop is the phase locked loop example shown in Figure 6.

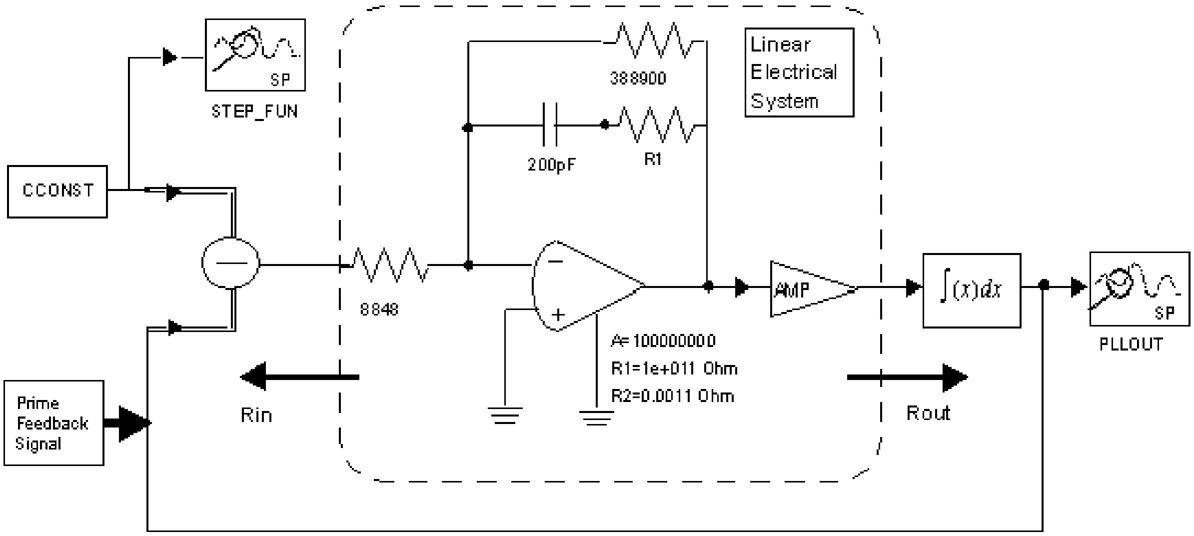


Figure 6 – PLL Example with a Primed Feedback Loop

To break the deadlock, the feedback signal leading to the input of the complex subtractor CSUB (the component with the minus sign) is initially primed with a zero sample.

Assumptions for Nexsys Partitioning

The following assumptions govern the partitioning process.

1. The impedance seen by an electrical component or sub-design looking into an input port of a functional component is assumed to be **Rin**, always defaulted to an open or infinity Ω. In other words, when an electrical component or sub-design port is terminated into the input port of a functional component, the termination resistance is **Rin**. This load resistance can be adjusted by assigning a different value to the **Rin** parameter of the functional component. This resistance has no noise contribution.
2. The impedance seen by an electrical component or sub-design looking into an output port of a functional component is assumed to be **Rout** (defaulted to a short or 0Ω). In other words, when an electrical component or sub-design input port is connected to the output port of a functional component, the assumed source resistance is **Rout**. This source resistance can be adjusted by assigning a different value to the **Rout** parameter of the functional component. Note that in reference image, the input voltage to the electrical sub-design V2 equals f(V1) if the impedance seen by the functional component looking into the electrical sub-design is 50Ω, provided that **Rout** is set to 50Ω. The notation **f(.)** represents the equivalent signal processing operation of the functional component.

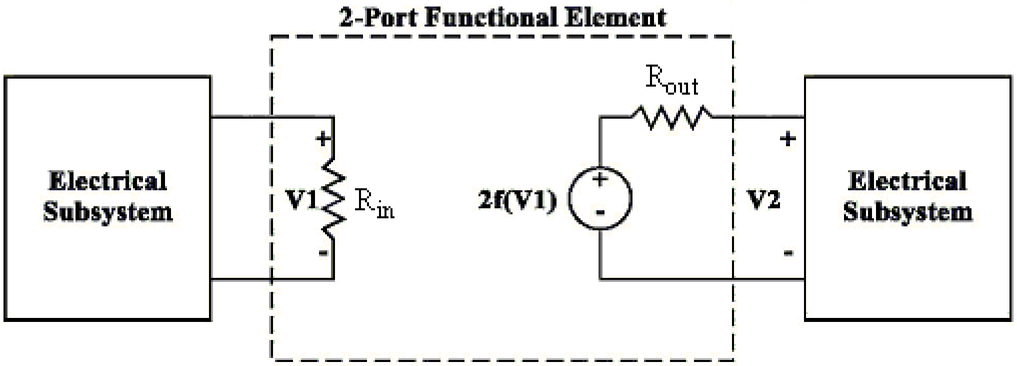


Figure 7– Equivalent Model for Functional-Electrical Connections

3. The impedance seen by a functional component looking into an input port of another functional component is assumed to be **Rin** (typically defaulted to infinity) and the impedance seen by a functional component looking into an output port of another functional component is assumed to be Rout (typically defaulted to zero).

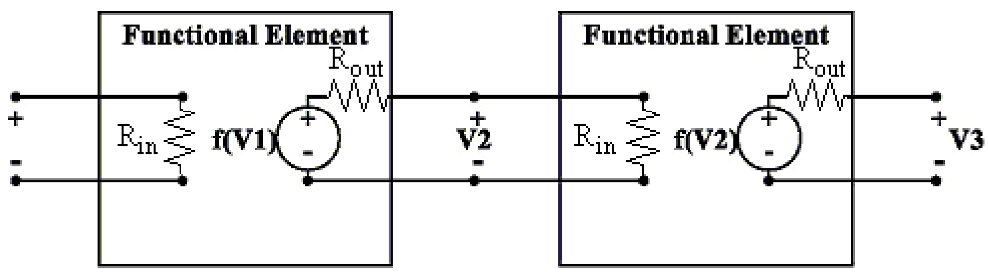


Figure 8 – Equivalent Model for Functional-Functional Connections

4. All nonlinear electrical two-port components (AMP and MIXER) are assumed to be unidirectional (e.g., $S_{12} = 0$). Each nonlinear electrical component is partitioned into three segments: a linear active input stage, a nonlinear functional stage, and a linear electrical passive output stage. This arrangement is discussed later in more detail.
5. During signal analysis, any signal path in the system is described by its complex envelope and carrier frequency (the triplet ($I(t)$, $Q(t)$, f_c)) and the simulation timestep t_s. In general, the signal (($I(t)$, $Q(t)$)) information bandwidth, carrier frequency f_c, and simulation timestep t_s vary from one point in the system to another. The user must measure the signal (($I(t)$, $Q(t)$)) information bandwidth, the carrier frequency f_c, and simulation timestep t_s at the correct points in the system to generate meaningful results. A baseband signal is described by its real signal value and the timestep.

6. If interface sources are interfaced with Nexsys components, a **SAMPLER** component must be placed at the physical/functional interface, otherwise the simulation engine issues an error. The user specifies the sampling rate for the **SAMPLER** component.
7. An **ENVELOPEADAPTOR** component must be placed at the interface of bandpass signal (or modulated signal) entering a Nexxim sub-circuit from a Nexsys functional component. The **ENVELOPEADAPTOR** component helps the simulation engine to recognize what circuit simulation analysis is needed to call to solve the Nexxim sub-circuit.

Nexsys MATLAB user-defined Models

Nexsys can simulate with user-defined models (UDMs) for behavioral (functional) components using the MATLAB® interpretive language (MATLAB® is a registered trademark of The Mathworks, Inc.) The user supplies a model file defining the model name and the corresponding MATLAB functionality. When Nexsys encounters a component defined with a MATLAB UDM, Nexsys calls the MATLAB function defined in the model file, and the MATLAB engine carries out the computations for the model.

Using MATLAB co-simulation, the user can combine the Nexsys system design environment and report generator with the enormous existing MATLAB functionality in DSP, communications, vector manipulation, and optimization. MATLAB can be also used to display the simulation results in real time.

Setting Up Nexsys Co-simulation with MATLAB

Nexsys/MATLAB co-simulation requires the MATLAB software installed on the same system with Nexsys. If MATLAB is not installed properly, Nexsys analysis containing MATLAB models fails with an error message.

All files related to MATLAB UDMs should be put in *InstallDirectory***userlib**, where *InstallDirectory* is the Nexsys installation directory.

The Nexsys MATLAB Model File

The file containing the MATLAB code for the model must be named *model_name***.m**. The **.m** file defining a Nexsys MATLAB UDM has two required elements:

- The model definition line (a comment line at the top of the file).
- The MATLAB function definition specifying the model computations.

MATLAB Nexsys Model Definition Line

The model definition line identifies the name of the model, the inputs and outputs on the model, and the names and initial values of any model parameters. The netlist syntax for the model definition line is:

% NEXSYS Functional_Model("*model_name*", "*In_N*", "*In_Types*", "*In_Num_Sample_Needed*", "*Out_N*", "*Out_types*", "*Out_Num_Sample_Generated*", "*data_flag*", "*probe_domain*", "*parameter_string*")

> **Note:**
>
> Each entry in the parentheses ( ) must be double quoted, or else an error occurs.

Nexsys MATLAB UDM Netlist Entries

Netlist Entry	Description
model_ name	The name for the MATLAB UDM
In_N	The number of input ports
In_Types	A character string of length In_N specifying the data type for each corresponding input port. The characters are **r** for real, **c** for complex.
In_Num_ Sample_ Needed	The number of input samples needed per invocation (execution) of the model. This entry can be an expression using the parameters listed in the **parameter_string** entry.
Out_N	The number of output ports
Out_Types	A character string of length Out_N specifying the data type for each corresponding output port. The characters are **r** for real, **c** for complex.
Out_Num_ Samples_ Generated	The number of output samples generated per invocation (execution) of the model. This entry can be an expression using the parameters listed in the **parameter_string** entry.
data_flag	Flag. Set to DATA_REQ when external data needed, NO_DATA_REQ to reset.
probe_ domain	Domain for displaying probe data: TF_DOM for time and frequency domain FREQ_DOM for frequency domain only SWEEP_DOM for sweep domain NOT_APPLICABLE for non-probe elements
parameter_ string	List of keywords and their with default values (*keyword*=*val*) separated by commas.

MATLAB Nexsys Model Function Definition

After the model definition line, the model file contains the definition of the function that implements the model code.

The second line in the model file must be the MATLAB function declaration line. The format of the function declaration line depends on the model type. Nexsys supports modeling of sources or input/output (I/O) model types, defined by the numbers of inputs and outputs.

A **SOURCE** model has no input ports (**In_N** =0) and one or more output ports (**Out_N** >0). A **SOURC** E model must have parameters **NSAMP** and **SAMPLE_RATE** in the *parameter_list* in the model definition line.

An **I/O** model has one or more input ports (**In_N** >0) and one or more output ports (**Out_N** >0). The format for the function declaration of an I/O model depends on whether or not it has a *parameter_list* in the model definition line.

For **SOURCE** UDMs the syntax for the function declaration line is:

```
function[outputs,SV] = function_name(parameters, SV, status_in)
```

For **I/O** UDMS with no *parameter_list* in the model definition line:

```
function[outputs,SV] = function_name(inputs, SV, status_in)
```

For **I/O** UDMS (In_N > 0) with a non-empty *parameter_list* in the model definition line:

```
function[outputs,SV] = function_name(inputs, parameters, SV, status_
in)
```

The entries in the function declarations are defined as follows:

- ***outputs*** is the user-defined name of a two dimensional array with ***Out_N*** rows, each row corresponding to an output port in the same order specified in the UDM definition.
- ***inputs*** is a the user-defined name of a two dimensional array with ***In_N*** rows, each row corresponding to an input port in the same order specified in the UDM definition.
- ***parameters*** is the user-defined name of a one dimensional array containing the parameter values entered for the model instance in the same order defined in ***parameter_ string***.
- **SV** is a system-defined structure that retains the state variables of a MATLAB UDM. State variables are any structured data defined by the model file that must be saved between calls to the model. The first time a model is called, the simulator assigns a zero value to **SV** before passing it to the user-defined MATLAB function. During initialization (**status_in** = 0), the model saves any needed data structures in **SV** before passing the **SV** data back to Nexsys with the output of the function. Nexsys retains the **SV** information and passes it back to the MATLAB UDM function on the next invocation. If data retention is not required, the function does not need to initialize or set **SV**.
- **status_in** is a system-defined flag to indicate the input status when calling a MATLAB UDM function. (**status_in** = 0 for the initialization stage, nonzero otherwise.

Global Variables for Timestep and Center Frequency

The MATLAB model can access and change the timestep and center frequency on a port-by-port basis using MATLAB global variables **TS_IN**, **TS_OUT**, **CF_IN**, and **CF_OUT**.

- **TS_IN** is a vector containing the timesteps associated with all input signals. **TS_IN** (j) is the timestep associated with the signal of the *jth* input port.
- **TS_OUT** is a vector containing the time steps associated with all output signals. **TS_OUT** (j) is the time step associated with the signal of the *jth* output port.
- **CF_IN** is a vector containing the center (carrier) frequencies associated with all input signals. **CF_IN** (j) is the center frequency associated with the signal of the jth input port.
- **CF_OUT** is a vector containing the center (carrier) frequencies associated with all output signals. **CF_OUT** (j) is the center frequency associated with the signal of the jth output port.

The global Statement

To access these Nexsys global variables during simulation, the model includes a **global** statement on the line after the function declaration, listing the global variables to be accessed. For example, the statement

```
global TS_IN TS_OUT
```

allows users to access the global input and output time steps.

Accessing and Changing the Time Step

To access or set the timestep for output signals the model reads and writes the global vectors TS_IN and TS_OUT. For example, the statement

```
TS_OUT (2) = 2 * TS_IN (1)
```

sets the timestep of the second output port to be twice the timestep of the first input port. If the timestep for the output ports (**TS_OUT**) of a functional SOURCE UDM is not set explicitly in the code, the engine assigns 1/sample_rate to each of them. If **TS_OUT** is not set by a functional I/O model, the program assumes the output sample rates are equal to the first input sample rate.

Accessing and Changing the Center Frequency

Setting up the center frequencies for output signals can be done by accessing the vectors CF_IN and CF_OUT. For example, the statement

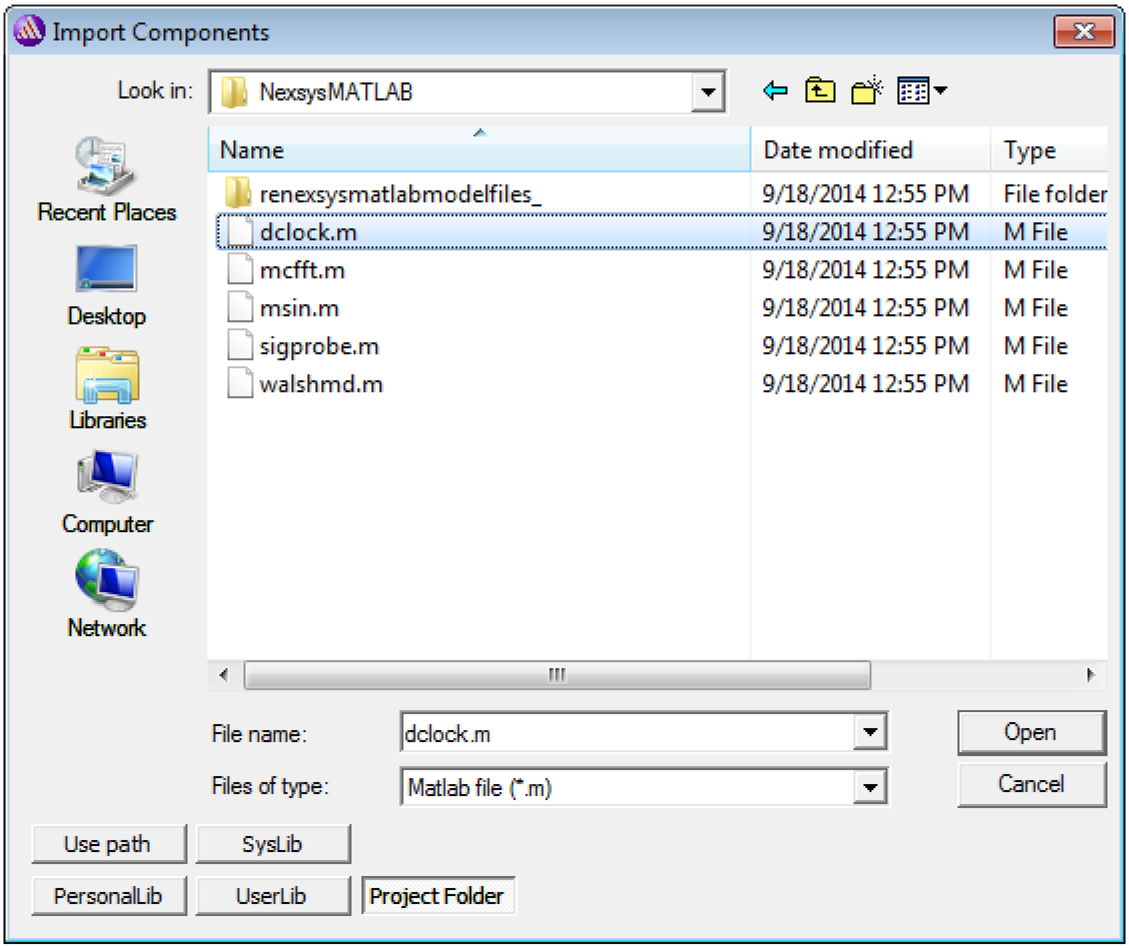
```
CF_OUT(2) = 2 * CF_IN(1)
```

sets the center frequency of the second output port to be twice the center frequency of the first input port. If CF_OUT is not set by a MATLAB functional UDM, all output port frequencies assumes the center frequency of the first input port for an I/O model and a zero center frequency for a SOURCE model.

Creating a MATLAB UDM Schematic Component

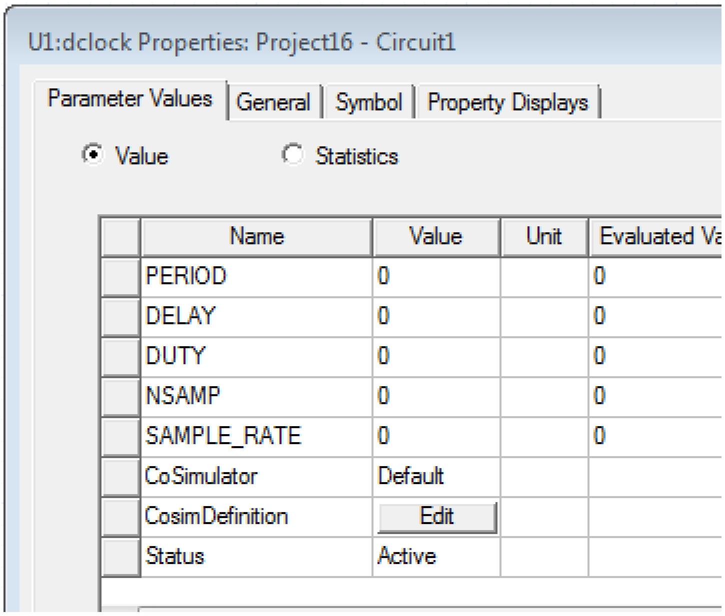
To create a schematic component for the Nexsys Matlab UDM, follow these steps.

1. In a Circuit design project, click the **View** menu to open the **Components Manager**. Then select the **Symbols** tab and click **Import Models**. Finally, click the **Matlab** icon to open the **Import Components** window.

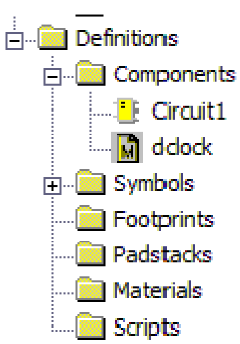


3. Browse to the directory that contains the MATLAB model file, select the file, then click **Open**.
4. An instance of the component is attached to the cursor. Drag and drop it into the schematic.

5. Double-click the component to select it and display its **Properties** window.



Click **OK** to close the **Properties** window. The MATLAB UDM (**dclock** in this case) appears in the **Components** group box.



Example MATLAB Nexsys Model Files

The following topics describe various examples of Nexsys MATLAB model files.

Nexsys Digital Clock Model

Here is a model for a digital clock source. This SOURCE type model has no inputs and one output. Here is a listing of the model file, **dclock.m**, with explanations.

The initial comment line (identified by **%NEXSYS**) is the model definition line. Nexsys reads the model definition into a structure that contains various information the simulator needs.

```
% NEXSYS FUNCTIONAL_MODEL ("dclock", "0","", "", "1","r", "1","NO_
DATA_REQ", "NOT_APPLICABLE", "PERIOD,DELAY=0,DUTY=0,NSAMP,SAMPLE_
RATE")
```

Here are explanations for the entries in the **dclock** model definition

“dclock”	Model name. The model name, dclock, is also the Matlab function name.
“0”	No inputs
“”	No input data types
“”	No input number of samples
“1”	One output
“r”	The single output is real
“1”	One (1) sample is to be generated per invocation
“NO_DATA_REQ”	No extra data reference is required
“NOT_APPLICABLE”	No probe domain applies
"PERIOD,DELAY=0,DUTY=0,NSAMP,SAMPLE_RATE"	Parameters for the model

Here is the Matlab function declaration for the digital clock. The function name, **dclock** in this example, is the same as model name.

```
function [ y, SV ] = dclock (p, SV, status_in)
```

The input and output variables for the function are:

p	Parameter values of the model
SV	State variable (in)
status_in	Input status
y	Output signal
SV	State variable (out)

The first step is to read the values on the parameters passed to the function.

```
% Get parameters from function argument p
period = p(1)
delay = p(2)
```

```
duty = p(3)
sample_rate = p(5)
time_step = 1.0/p(5)
```

During the initialization call, the **dclock** model sets up its state variables structure **SV**. During later calls, the model retrieves the data.

```
total_num_out = 0;
% (status_in == 0) => Analysis initialization stage
if status_in == 0
delay_mode = 0;
if (delay > 0)
delay_mode = 1;
end
time_pos = 0.0;
total_out_num = 0;
SV = [delay_mode time_pos total_num_out]
else
delay_mode = SV(1);
time_pos = SV(2);
total_num_out = SV(3);
end
```

Here is how **dclock** calculates the output (digital clock) values:

```
out_value = 0.0;
if delay_mode == 1
time_elapsed = (total_num_out + 1) / sample_rate;
if (delay > time_elapsed)
out_value = 0.0;
else
delay_mode = 0;
end
else
if ((time_pos>=0) && (time_pos < (duty * period)))
out_value = 1.0;
else
out_value = 0.0;
end
%re-calculate the time position
```

```
time_pos = time_pos + 1.0 / sample_rate;
if (time_pos >= period)
time_pos = time_pos - period;
elseif (time_pos == period)
time_pos = 0.0;
end
end
y(1) = out_value;
total_num_out = total_num_out + 1;
```

Finally, **dclock** saves its state variables **SV** and returns.

```
SV = [delay_mode time_pos total_num_out];
return
```

Nexsys Sinusoidal Source Model

Here is a model of a sinusoidal source, **sin.m**. This SOURCE type model has no inputs and one output.

The model name, **sin**, matches the Matlab function name.

```
% NEXSYS FUNCTIONAL_MODEL ("sin", "0","", "", "1","r", "1","NO_DATA_
REQ", "NOT_APPLICABLE", "NSAMP=256,SAMPLE_
RATE=102.4e6,AMPLITUDE=1,FOSC=10e6,PHASE=0" )
```

Here are explanations for the entries in this line:

“dclock”	Model name and function name
“0”	No inputs
“”	No input type
“”	No samples needed per invocation
“1”	One output
“r”	Output is real
“NSAMP”	“NSAMP” sample to be generated per invocation (all samples are sent to the output port at once)
“NO_DATA_REQ”	No extra data reference is required
“NOT_APPLICABLE”	No probe domain
" NSAMP=256, SAMPLE_RATE=102.4e6, AMPLITUDE=1, FOSC=10e6, PHASE=0"	Parameters for the model

Here is the Matlab **sin** function definition.

```
function [ output, SV ] = sin (parameters, SV, status_in)
```

The input and output variables for the function are:

parameters	Parameter values of the model
SV	State variable (in)
status_in	Input status
output	Output signal
SV	State variable (out)

Get parameters.

```
% Get parameters from function argument p
samples = parameters(1);
sample_rate = parameters(2);
amplitude = parameters(3);
fosc = parameters(4);
phase = parameters(5)*pi/180;
```

Calculate and set output and return:

```
time = (0:samples-1)/sample_rate;
output = amplitude*cos(2*pi*fosc*time + phase);
return
```

The sinusoidal source does not require any state variables (**SV**) to be saved or retrieved.

Nexsys Walsh Modulator Model

This example is a Walsh modulator model, **walshmd.m**. This **I/O** type model has one input and one output. The model definition line is:

```
% NEXSYS Functional_Model ("walshmd", "1","r", "N", "1","r", "2^N",
"NO_DATA_REQ", "NOT_APPLICABLE", "N" )
```

Here are explanations for the entries:

“walshmd”	Model name (and function name)
“1”	One input
“r”	Input type is real

“N”	N samples needed per invocation
“1”	One output
“r”	Output is real
“2^N”	2^N samples generated per invocation
“NO_DATA_REQ”	No extra data reference is required
“NOT_APPLICABLE”	No probe domain
"N"	Parameter for the model

Here is the **walshmd** function declaration:

```
function [ y, SV ] = walshmd ( x, p, SV, status_in)
```

The input and output variables for the function are:

x	Input signal
p	Parameter values of the model
SV	State variable (in)
status_in	Input status
y	Output signal
SV	State variable (out)

Access input and output timestep through global variables.:

```
% Declare the global variables this model need to access from Nexsys
engine
global TS_OUT TS_IN
```

Get parameters:

```
% Get parameters from function argument p
N = p(1)
```

Generate the Walsh Matrix once initially and save it into the **SV** state_variables, then retrieve it on later passes.

```
nrows = 2^N
ncols = 2^N
num_mod_symbols = 1
outnsamp = num_mod_symbols*ncols
% (status_in == 0) => Analysis initialization stage
if status_in == 0
```

```
size = 1
mask = 1
WM(1,1) = 0
for i = 1:N
for m = 1:size
for n = 1:size
WM(m,n+size) = WM(m,n)
WM(m+size,n) = WM(m,n)
if rem(WM(m,n),2) == 1
WM(m+size,n+size) = 0
else
WM(m+size,n+size) = 1
end
end
end
size = size*2
end
SV = WM
else
WM = SV
end
```

Set output timestep:

```
TS_OUT(1) = TS_IN(1)*N/ncols
```

Calculate output values and return:

```
cnt = 1
for j = 1:num_mod_symbols
factor = 1
mod_index = 1
for i = 1:N
mod_index = mod_index + factor * x(i+N*(j-1))
factor = 2*factor
end
for i = 1:ncols
y(cnt,1) = WM(i,mod_index)
cnt = cnt + 1
```

```
end
end
return
```

Nexsys Complex FFT Model

This **I/O** model, **cfft.m**, performs a Fast Fourier transform (FFT) on the incoming complex signal.

```
% NEXSYS Functional_Model ("cfft", "1","c", "length", "1","c",
"length", "NO_DATA_REQ", "NOT_APPLICABLE", "length" )
```

Here are the inputs and outputs for this model:

“cfft”	Model name and function name
“1”	One input
“c”	Input type is complex
“length”	“length” samples needed per invocation
“1”	One output
“c”	Output is complex
“length”	“length” samples generated per invocation
“NO_DATA_REQ”	No extra data reference is required
“NOT_APPLICABLE”	No probe domain
“length”	Parameter for the model

Here is the **cfft** function definition.

```
function[output, SV] = cfft(input, parameters, SV, status_in)
```

The input and output variables for the function are:

input	Input signal
parameters	Parameter values of the model
SV	State variable (in)
status_in	Input status
output	Output signal
SV	State variable (out)

Get Parameters:

```
length = parameters(1);
```

Calculate output and return:

```
output = fft(input(1:length));
return
```

Nexsys Options

The Nexsys behavioral simulator Nexsys uses an option to prevent division by zero. The syntax is:

.OPTIONSbehavioral_minimum_divisor=*val*

Setting **behavioral_minimum_divisor** to a nonzero value prevents division by zero when evaluating behavioral devices in a Nexsys analysis.

System Frequency Domain Analysis

System frequency domain analysis allows you to evaluate and troubleshoot RF/high-frequency designs for wireless and microwave applications.

Results available with System Frequency Domain analysis include:

- The output power (**Pout**) at all external output ports.
- The voltage and power spectra and steady state time domain responses at the points where Nexsys voltage probes have been placed.
- The analysis results may be viewed in the spectral domain or in the time domain.

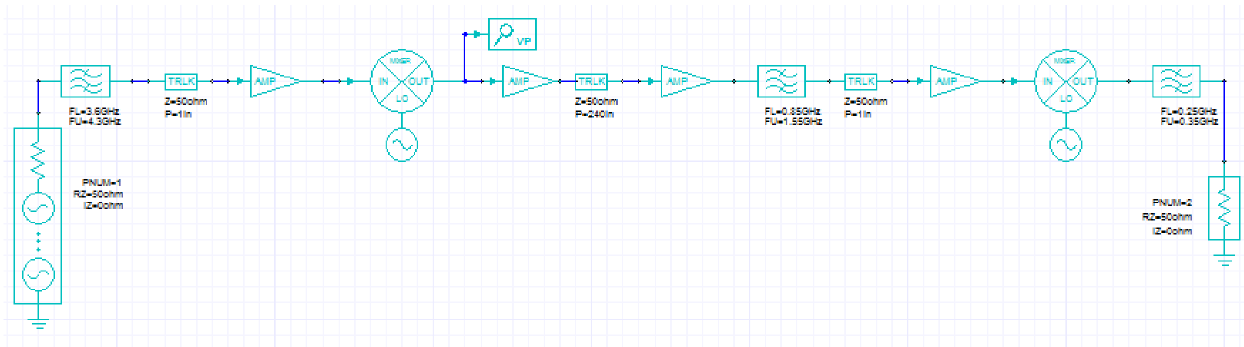
Running Frequency Domain Analysis

To set up a design and run frequency domain analysis

1. Open a new Circuit design.
2. Place the Components.

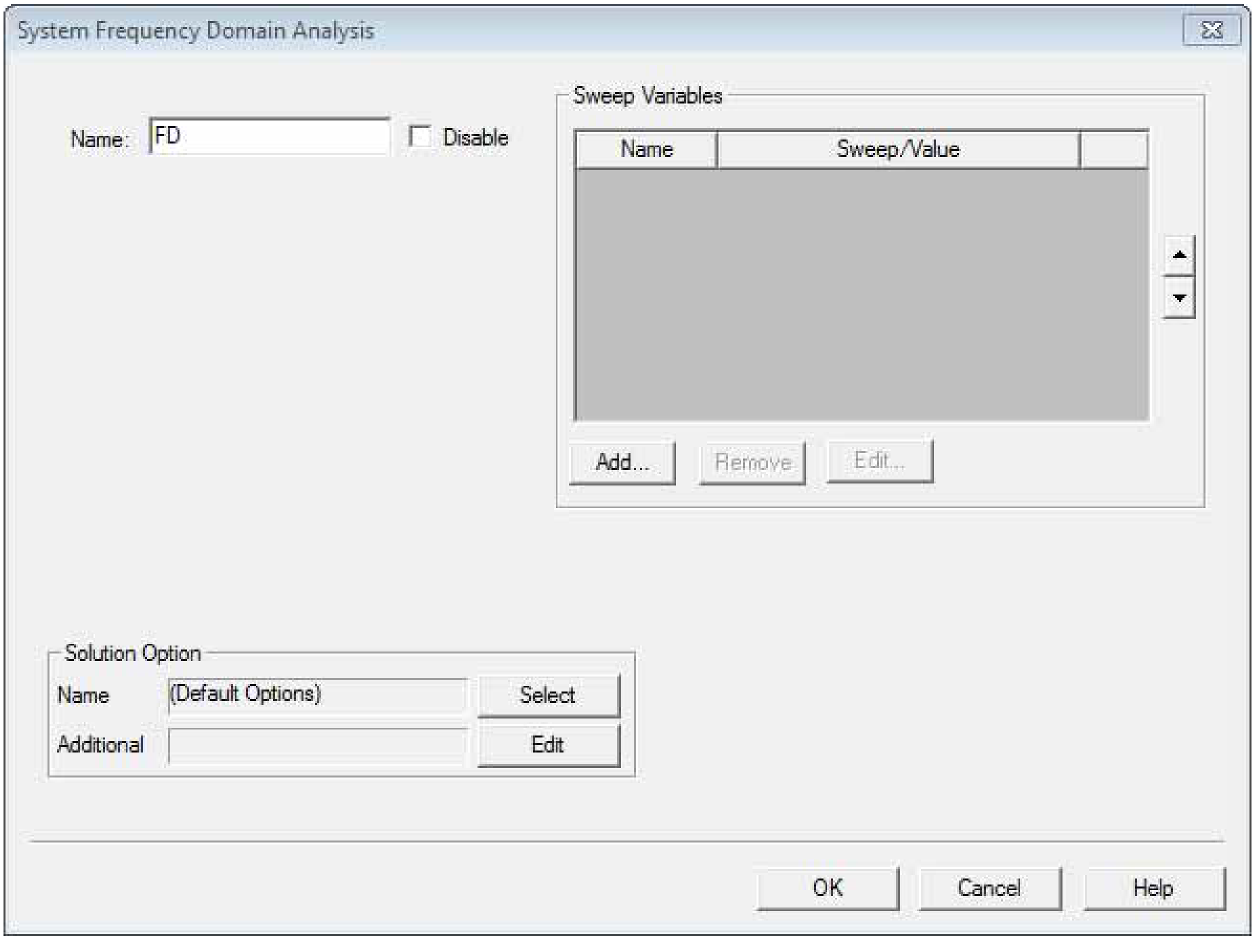
- Nexsys behavioral MIXER and AMP elements are located in **Components>System>nonlinear_RF**.

- The behavioral AMP and MIXER components can be connected directly to the electrical (Nexxim) components, as in the following example:



3. Add ports to external output nodes.
   - Ports are added by clicking the port symbol in the menu. See Ports in Schematics for further information.
   - Power and voltage results are calculated for every output port.
4. Place and name Nexsys voltage probes (VP) on internal nodes of interest.
   - Nexsys voltage probes are in **Components>System>Probes**.
   - Voltage and power spectra and steady state time domain responses are calculated at every probe location.

5. Select **Circuit>Add Nexxim Solution Setup>System Frequency Domain Analysis**.



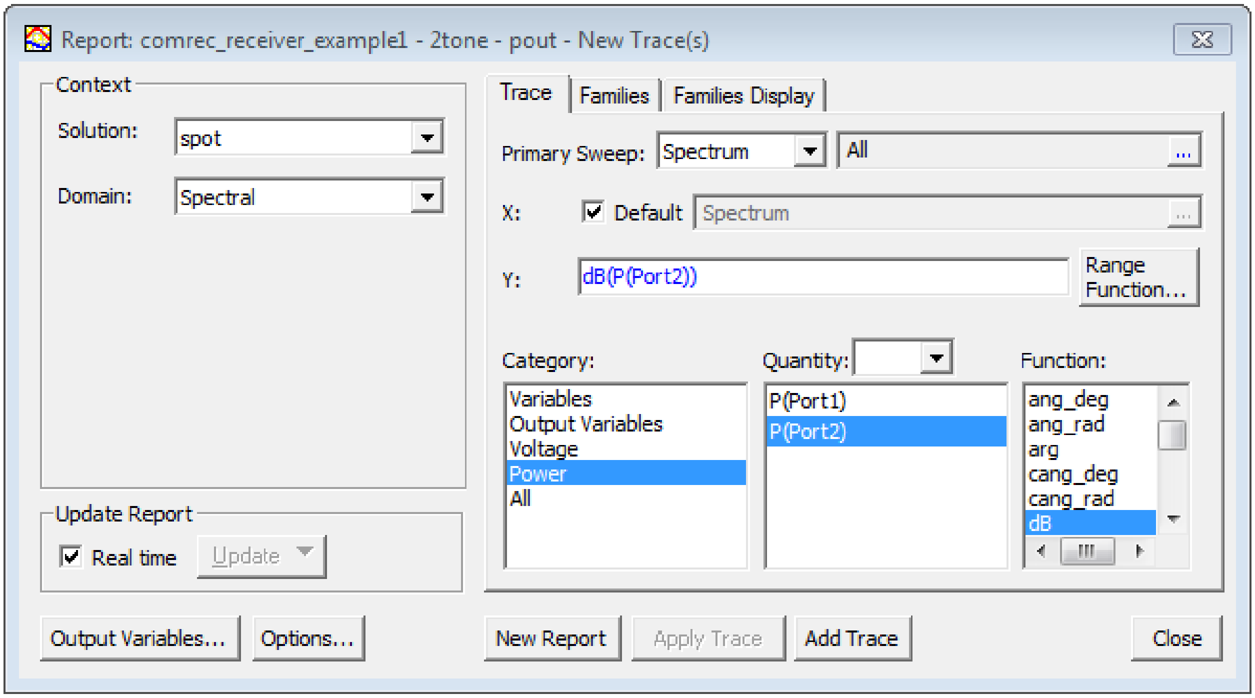
- Select an analysis **Name** or use the default (SystemFDAnalysis*n*).
- Click Add in the Sweep Variables group box. Add a frequency sweep:
  a. In the **Variable** list, frequency **F**is selected (and cannot be changed). Select one of the following: **Single value, Linear step, Linear count, Decade count,**or **Octave count**.
  b. Type the sweep values into the **Value** field (for **Single value**), or into the **Start**, **Stop**, and **Step** text fields (for **Linear**, **Decade**, or **Octave count**), and ensure that the appropriate units (**GHz, MHz, kHz**) are selected for each.
  c. Click **Add**, then click **OK** to close the **Add/Edit Sweep** window and reopen the **Frequency Domain Analysis**window.
  e. For more information, see Variable Sweep.
- Click **OK** to complete the Frequency Domain analysis setup.

6. Expand the **Project Manager** window and the Analysis folder. Right-click the **FDAnalysis** setup and select **Analyze** on the menu. The progress bar shows the analysis proceeding

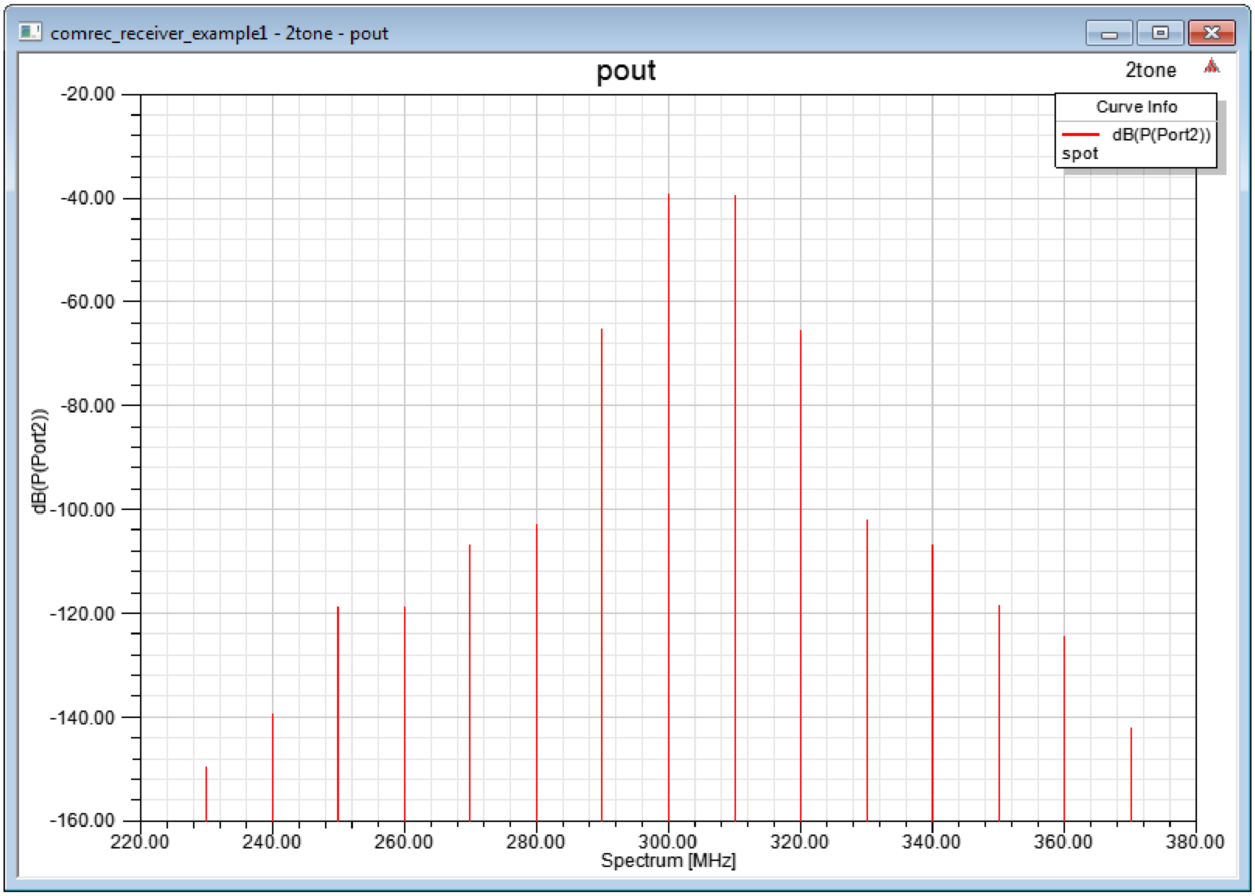
to completion.

7. When the analysis has run to completion, view the results.

- Click **Results>Create Standard Report>Rectangular Plot**.
- Select the Solution and the Domain. The FD analysis results may be viewed in the spectral domain or in the time domain.
- Select Power or Voltage, the output ports, and dB. Here is an example report setup:



- Click **New Report**. Here is an example of an output power plot.



Frequency Domain Analysis

Frequency domain analysis is a system-level simulation for electrical systems of linear elements (including S-parameter elements) and nonlinear behavioral electrical elements such as amplifiers and mixers.

Figure 1 shows a multi-channel nonlinear electrical topology made up of multi-port linear passive elements and two-port linear or nonlinear active elements.

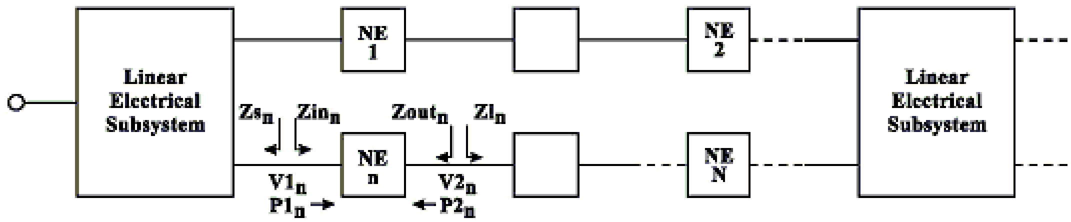


Figure 1. General nonlinear electrical topology containing N nonlinear elements (NE = nonlinear element)

Frequency-domain analysis is applicable only to electrical systems made up of linear/nonlinear frequency-based elements like the example in Figure 1. Systems containing functional time-based elements may not be analyzed in the frequency domain.

The multi-port system can be excited from any set of external sources. Frequency domain analysis can be performed with both single-tone and multi-tone excitations. When the source provides multi-tone excitations, frequency domain analysis can predict the inter-modulation distortion (IMD) or harmonics generated by nonlinear elements (mixers and amplifiers) in the system.

Note:

Assumptions for Frequency Domain Analyses:

1. All external ports are assumed to have a complex termination (defaulted to 50Ω). This load impedance can be adjusted by assigning a different value in the port definition.
2. All two-port nonlinear components are treated as unidirectional components (e.g., S12 = 0).
3. The IMD and harmonic measurements for each individual nonlinear element are obtained when the input and output ports have 50Ω terminations and the applied RF sources represent a single-tone bandpass input.
4. Nonlinear measurements are obtained when the input and output ports of the nonlinear component are terminated in 50Ω.

The Equivalent Model for Nonlinear Behavioral Components

Generally, nonlinear electrical components are not only power dependent, but also frequency and temperature dependent. The equivalent model used for nonlinear components in discrete time simulation separates the power-dependent characteristics on the frequency-dependent and temperature-dependent characteristics.

Each nonlinear electrical component is partitioned into three elements: a linear electrical (active) input component, a nonlinear functional component modeled in the time domain, and a linear electrical (passive and noiseless) output component. This arrangement is shown in Figure 2.

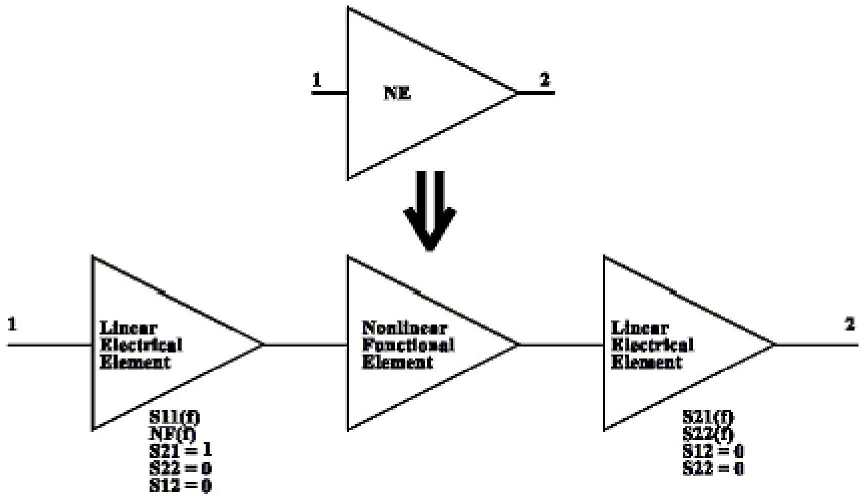


Figure 2 – Equivalent Model for Nonlinear Behavioral Components

All nonlinear electrical two-port components are assumed to be unidirectional (e.g., S_{12} = 0). In this model, the power-dependent characteristics of S_{11}, S_{12} and S_{22} are ignored. The frequency and temperature dependent small-signal S_{11} and noise figure NF are associated with the input linear electrical component. The power-dependent characteristics of S_{21} are associated with the nonlinear functional component. The frequency and temperature dependent small signal S_{21} and S_{22} are associated with the output linear electrical component.

The linear electrical input and output components are associated with other connected linear electrical components and simulated as a linear electrical sub-design, as described in The Algorithm for Response Evaluation of Nonlinear Systems. See Time Domain Representation of a Bandpass Envelope Multitone RF Signal for the calculation formulas.

The nonlinear characteristic of a two-port component is described by its nonlinear figures-of-merit or, alternatively, by measured nonlinear data. Only the nonlinear characteristic of S_{21} is considered, even though S_{11}, S_{12} and S_{22} may also be power-dependent. The topic Modeling Nonlinearity with Polynomial Power Series describes the modeling of this nonlinear functional component.

The measurements obtained for a two-port nonlinear amplifier typically represent the AM-AM and AM-PM distortion (e.g., output power and phase vs. input power). These quantities are typically measured for a given input frequency or set of frequencies while the available input power is swept. These measurements are then used during frequency domain analysis (taking the operating point for the nonlinear element into account) for computing the IMD generated by this element when it is embedded in a general topology such as the one shown in Figure 1 in the topic "Frequency Domain Analysis" on page 13-41. See IMD Calculations for Nonlinear Two-Port Elements for more on this calculation.

The harmonics generated by a mixer can be measured when the RF and IF ports are both terminated in 50Ω and a single RF tone with a given frequency and available input power is applied to the RF port of the mixer. The measured harmonics (called MIXERSPURS data) may then be used to predict the harmonics generated by this element when it is embedded in a general topology such as the one in Figure 1 in the topic "Frequency Domain Analysis" on page 13-41. See IMD Calculations for Mixers with MIXERSPURS Data Tables for more on this calculation.

The Algorithm for Response Evaluation of Nonlinear Systems

The nonlinear frequency domain algorithm described here accounts for all nonlinearities and inter-stage mismatches in the system. For a multi-channel nonlinear topology with N nonlinear elements (refer to Figure 1 in the topic "Frequency Domain Analysis" on page 13-41), it is always assumed that parallel nonlinear channels connected to the same linear multi-port subsystem are not coupled (i.e., non-interacting). This is ensured during multi-tone analysis by assuming that S_{12} is very close to zero for all nonlinear elements. This effectively implies that the topology has a feed-forward nature. As a result, the impedances Zin_n and $Zout_n$ for the n^{th} nonlinear element (1<=n<=N) are only a function of S11 and S22 respectively for the n^{th} nonlinear element (i.e., each nonlinear element is effectively unidirectional). With that important assumption, the algorithm used for evaluating the IMD and harmonic responses, in the system, proceeds as follows:

1. The nonlinear behavioral models are decomposed into three pieces (as shown in Figure 2 in the topic "The Equivalent Model for Nonlinear Behavioral Components " on page 13-42: one front end linear input component, one nonlinear functional behavioral model and one back end linear output component.
2. The topology is then partitioned into linear electrical subsystems (touching linear models) and nonlinear functional elements and all the partitioned components are scheduled on the left (input) to the right (output).
3. The linear electrical subsystems are solved with Nexxim linear network analysis (LNA) cosimulation, and the results are sent back to the top level frequency domain solver. The simulation takes into account all impedance mismatches Zin_n, $Zout_n$ seen looking into nonlinear elements connected to each linear subsystem (See Figure 1 in the topic "Frequency Domain Analysis" on page 13-41).
4. Harmonics at the input of each nonlinear element are transformed based on the nonlinear transformation characteristics of that element. First, the frequency domain signal at the input is transformed into time domain data. Next, the time domain output is calculated

based on the power series model described in Modeling Nonlinearity. Finally, the time domain signal is transformed back to frequency domain and sent to the next state. This calculation takes into account all impedance mismatches Zs_n, Zl_n seen looking into the ports of other elements connected to each nonlinear element (See Figure 1 in the topic "Frequency Domain Analysis" on page 13-41). These impedances are evaluated at all input and output harmonics. Calculations of harmonics and IMD products generated by mixers and nonlinear amplifiers is discussed later. See IMD Calculations for Nonlinear Two-Port Elements.

When two or more carriers are applied to a system that includes one or more nonlinear components, inter-modulation frequency components is generated. The frequency domain analysis predicts these inter-modulation distortions (IMD) at all external output ports and at internal nodes where voltage and/or power probes are placed.

The frequency domain analysis accounts for all the inter-modulation frequency components generated by all the nonlinear elements within a system. This analysis tends to be more accurate if nonlinear measurements such as Pout vs. Pin are provided for nonlinear electrical components, as opposed to just providing the nonlinear figures of merit OIP3, P1dB, or Psat.

Frequency Domain Measurements

Two kinds of measurements are available with frequency domain analysis.

- The output power (**Pout**) at all external output ports.
- The voltage and power spectra and steady state time domain responses at the points where voltage probes have been placed. A voltage probe may be placed anywhere inside a nonlinear electrical system prior to running the analysis. Each probe must be given a label or a name. Careful placement of voltage probes can help identify the effect on the overall IMD results produced by each individual component in the system.

The analysis results may be viewed in the spectral domain or in the time domain.

Modeling Nonlinearity with Polynomial Power Series

Nonlinear measurements of a two-port component typically include the AM-AM and AM-PM effects. The nonlinear relationship between $S_{21}$and P_1 (the available input power) or, equivalently, between P_{out} and P_1 may be represented by power series polynomials:

$$\sqrt{P_{out}}\cos\phi = a_1X + a_3X^3 + \ldots$$

$$\sqrt{P_{out}}\sin\phi = b_1X + b_3X^3 + \ldots$$

where:

$$P_{out} = |S_{21}|^2P$$

= the output load power (assuming RL = 50Ω)

$$X = \sqrt{P_1}$$

P_1 = the available input power on the source (with RS = 50Ω)

$$\phi = \angle S_{21} - \angle S_{21ss}$$

S_{21ss} = the small signal gain.

The coefficients a_1, a_3, a_5... and b_1, b_3, b_5... are calculated using a least-squares curve fitting technique based on the user-supplied measurements P_1 - P_{out} data or P_1 - S-parameters data. For example, a set of coefficients can be obtained based on the following power amplifier P_1 - P_{out} measured data (in 50-ohm terminations).

	RTH_PA 2-port Pout dBm P1 dBm, FREQ=900MHz	
P1	**Pout**	**Phase(degrees)**
5.00	25.68	88.75
7.00	27.67	88.75
9.00	29.66	88.75
11.00	31.64	88.75
13.00	33.61	88.75
15.00	35.56	88.75
17.00	37.48	88.76
19.00	39.35	88.76
21.00	41.16	88.76
23.00	42.86	88.75
25.00	44.38	88.71
27.00	45.65	88.66
29.00	46.53	88.76
31.00	47.17	91.85
33.00	47.50	97.08
35.00	47.66	102.81

Figure 3. Pin Pout Data Example

If the user chooses to provide the nonlinear figures-of-merit (OIP3 or P1dB and Psat) instead of measurement data, the power series coefficients are approximated by:

$$a_1 = |S_{21}| \quad a_3 = \frac{-\left(\frac{4}{3}\right)|S_{21}|^3}{OIP3}$$

$a_i = 0$, $i = 5,7, \ldots$

$b_i = 0$, $i = 1,3,5, \ldots$

where:

S_{21} is the linear small signal gain

OIP3 is the output power at the third order intercept point.

In this case, the simulations tend to be less accurate.

Time Domain Representation of a Multitone RF Source

The nonlinear functional component part of a nonlinear behavioral element (See Figure 2 in the topic "The Equivalent Model for Nonlinear Behavioral Components " on page 13-42) is modeled in the time domain. An important step for calculating IMD products in the time domain is to derive an equivalent envelope time domain representation for a given bandpass RF multi-tone source.

Complex Envelope of an RF Source

An RF multi-tone source typically contains a fundamental carrier and at least one additional carrier. The equivalent source voltage S(t) of an RF source that has a total of *n* carriers (including the fundamental carrier) may be expressed as:

$$S(t) = \sum_{i=1}^{n} V_i \cos(2\pi f_i t + \theta_i) = I_s(t)\cos(\omega_c t) - Q_s(t)\sin(\omega_c t)$$

$$I_s(t) + jQ_s(t)$$

= The complex envelope of the signal *S(t)*

$$I_s(t) = \sum_{i=2}^{n} V_i \cos(2\pi(f_i - f_1)t + \theta_i - \theta_1)$$

= In-phase envelope signal

$$Q_s(t) = \sum_{i=2}^{n} V_i \sin(2\pi(f_i - f_1)t + \theta_i - \theta_1)$$

= Quadrature-phase envelope signal

$$V_i = \sqrt{4R_s P_i}$$

$$1 \le i \le n$$

Where:

f_i = The ith carrier frequency

$f_1 = f_c$ = Fundamental carrier frequency

$\theta_1 = 0$

θ_i = The ith carrier phase

P_i = The ith carrier available input power

P_1 = Fundamental carrier available input power

Nonlinear Output Voltage

The nonlinear output voltage is calculated in the time domain as:

$$S_{out}(t) = V_{out}(t)\cos(\omega_c t + \theta_{out}(t)) = I_{out}(t)\cos(\omega_c t) - Q_{out}(t)\sin(\omega_c t)$$

Where:

$$V_{out}(t) = \sqrt{P_{out}(t)R_{out}} = \sqrt{I_{out}^2(t) + Q_{out}^2(t)} \quad \theta_{out}(t) = \operatorname{atan}\left(\frac{Q_{out}(t)}{I_{out}(t)}\right)$$

$$I_{out}(t) = \sqrt{R_{out}} \times [GIV(t)\cos(\theta_s(t)) - GQV(t)\sin(\theta_s(t))]$$

$$Q_{out}(t) = \sqrt{R_{out}} \times [GIV(t)\sin(\theta_s(t)) - GQV(t)\cos(\theta_s(t))]$$

$$GIV(t) = \left(\frac{a_1}{MS_{21}}\right)X(t) + \left(\frac{a_3}{MS_{21}}\right)X(t)^3 + \left(\frac{a_5}{MS_{21}}\right)X(t)^5 + \ldots$$

$$GQV(t) = \left(\frac{b_1}{MS_{21}}\right)X(t) + \left(\frac{b_3}{MS_{21}}\right)X(t)^3 + \left(\frac{b_5}{MS_{21}}\right)X(t)^5 + \ldots \quad X(t) = \sqrt{P_1(t)} \quad P_1(t) = \frac{V_s^2(t)}{4R_{in}}$$

$$MS_{21} = \sqrt{a_1^2 + b_1^2}$$

The source resistance R_{in} and the load resistance R_{out} default to 50Ω, but they can be individually specified for each input and output, respectively.

Constraints on the Carrier Frequency

The accuracy of the envelope multi-tone analysis puts constraints on the minimum spacing between carrier frequencies:

$$f_c = f_1 + k\Delta f \Delta f = min(f_i - f_j)$$

$i \neq j$

, , $i,j = 1,2...,n$

k = positive or negative integer.

Envelope multitone frequency domain analysis results may lose accuracy when Δf is too large. In general, the condition $\Delta f << f_1$ must hold for accurate results.

IMD Calculations for Nonlinear Two-port Elements

Intermodulation distortion (IMD) calculations through a nonlinear two-port element (e.g., amplifier, mixer, frequency multiplier) are carried out in the time domain based on the envelope modulation technique discussed in Time Domain Representation of a Bandpass Envelope Multitone RF Signal. The envelope modulation technique is based on the assumption that the frequency separation between the input carriers is much less than the fundamental input carrier frequency. For the envelope modulation technique, all input carriers can be resolved in terms of the fundamental input carrier to yield a time-domain envelope signal, which modulates the fundamental carrier. The non-linearity may then be applied to this envelope signal in the time domain to generate the distorted time domain output envelope. This output envelope can then be transformed back to the frequency domain to yield the output IMD frequency components. This technique is much more effective (especially for a large number of input carriers) than performing the nonlinear transformation in the frequency domain and having to account for all the possible IMD products.

IMD Calculations for Mixers with MIXERSPURS Data Tables

When a carrier mixes with the local oscillator frequency in a mixer, intermodulation and harmonic frequency components is generated at the output of the mixer. These harmonic frequency components are called Mixer Spurs.

Generally, mixing can produce energy at all frequencies equal to:

$$N_{RF} f_{RF} + N_{LO} f_{LO}$$

where:

f_{RF} is the frequency of the carrier at the input of the mixer

f_{LO} is the frequency of the local oscillator

N_{RF} and N_{LO} are integers representing the RF and LO multipliers respectively.

You can control how many spurs get generated by specifying the MAXORDER, MINF, MAXF, and MINP parameters for each mixer, where:

MAXORDER: maximum order of spurs, $|N_{RF}| + |N_{LO}| \leq$ MAXORDER

MINF: minimum spur frequency (spurs below MINF are ignored)

MAXF: maximum spur frequency (spurs above MAXF are ignored)

MINP: minimum spur power, (spurs below MINP are ignored)

The power of a spur at the output of a mixer is calculated using the MIXERSPURS data table provided for the mixer. Once generated, each spur is carried through the remainder of the system with all mismatches taken into account. Figure 4 shows a representative mixer spurs table. Essentially, the rows in Figure 4 represent relative power levels for harmonic RF multipliers, and the columns represent relative power levels for harmonic LO multipliers. The image power level can also be specified.

```
BEGIN BLOCK
VAR MODELTYPE   = MIXERSPURS
VAR             SPURINFO    = REF
VAR             MATRIXTYPE = SINGLE_SIDED
VAR             MAXORDER   = 10
VAR             PARAMETER       = PLO  Unit:dbm  INTERPOLATION:Yes
VAR             PARAMETER       = PRF  Unit:dbm  INTERPOLATION:Yes

VAR PLO = 10dbm  ! Reference LO power
VAR PRF = -20dbm ! Reference input power

BEGIN DATA
                % MIXERSPURS  MAXORDER = 10
                *0 1  2  3  4  5  6  7  8  9  10
20 18 32 40 50 60 70 80 90 100 110
30 0  40 50 60 70 80 90 100 110
40 20 50 60 70 80 90 70 110
50 30 60 70 80 70 80 110
60 40 70 70 80 90 100
70 50 80 90 100 110
80 60 90 100 110
90 70 100 100
100 80 110
110 100
120
END
END BLOCK
```

Figure 4. Mixer Spurs Table Data Example

It is always assumed that mixer spur tables are obtained at the output IF port when the RF and IF ports of a mixer are both terminated in 50Ω and a single-tone RF source (with a given frequency and available input power) is applied to the mixer's RF port.

Mixer Spurs Indices

Mixer spurs indices can be viewed at any external port or voltage/power probe to indicate the spectral origin in terms of each RF and LO multiplier through all mixing stages.

The following assumptions are made when computing these indexes:

1. Mixer spurs are calculated only for the principal carrier. If more than one mixer is present in the system, the calculation proceeds in this manner: new mixer spurs are generated using only the principal carrier at the input to each mixer. Previously generated mixer spurs (available at the input of each mixer) are translated through each mixer using the

principal integer multipliers N_{RF} and N_{LO}, (N_{RF} = 1, N_{LO} = -1), without generating any additional Mixer Spurs. This assumption is based on the fact that additional spurs are typically either filtered out or too weak compared to the spurs generated by the principal RF carrier.

2. The index calculations do not account for IMD harmonics generated by nonlinear two port components in the system such as amplifiers.

Frequency Domain Analysis References

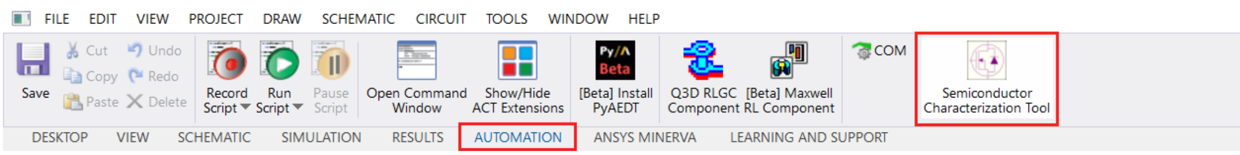
1. Guillermo Gonzalez, Microwave Transistor Amplifiers Analysis and Design, PRENTICE-HALL, Englewood Cliffs, NJ, 1984.
2. V. Rizzoli, A. Lipparini, Computer-Aided Noise Analysis of Linear Multiport Networks of Arbitrary Topology, IEEE on MTT, Vol. MTT-33, No.12, December 1985.
3. G. D. Vendelin, A. M. Pavio, and U. L. Rohde, Microwave Circuit Design Using Linear and Nonlinear Techniques, John Wiley & Sons, 1990

14 - Semiconductor Characterization Tool

The semiconductor characterization tool aims to enable users to create physics-based models of various electronics devices for operation at a desired characteristic. Currently, the tool supports modeling of SiC power MOSFET devices.

Upon first selecting the Semiconductor Characterization tool, you will be prompted with set up instructions.

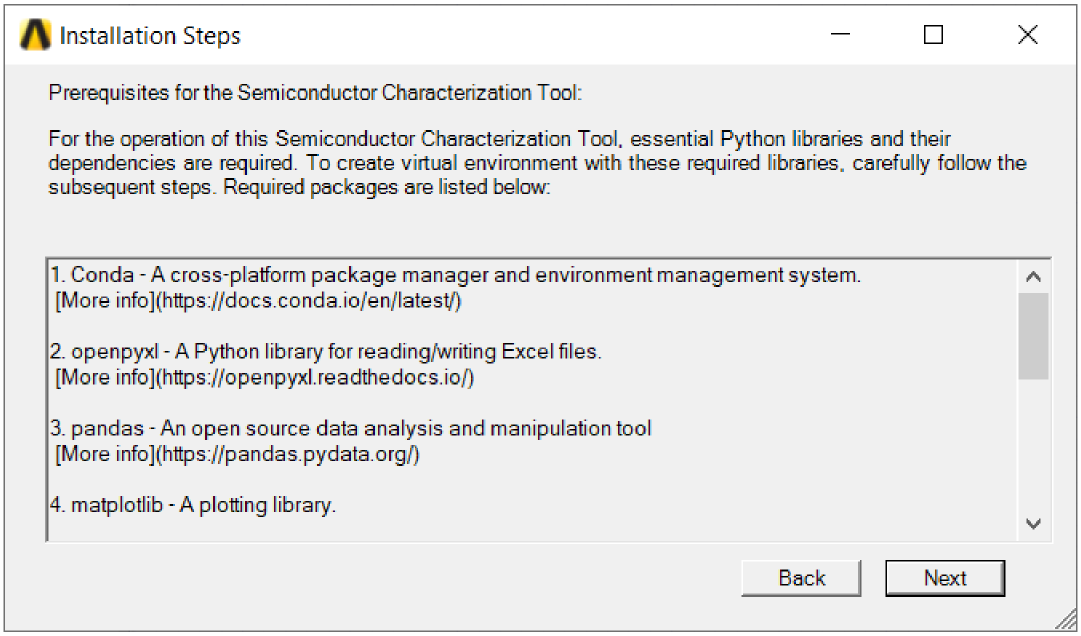
In the main ribbon, select **Automation**, then **Semiconductor Characterization Tool**.



The steps will walk you through the following processes:

- Prerequisite Information
- Conda Installation
- Creating a Virtual Environment
- Installing Essential Libraries
- Installing Essential Library "pygmo"

Prerequisites



1. Conda - A cross-platform package manager and environment management system.

   [More info](https://docs.conda.io/en/latest/)
2. openpyxl - A Python library for reading/writing Excel files.

   [More info](https://openpyxl.readthedocs.io/)
3. pandas - An open source data analysis and manipulation tool

   [More info](https://pandas.pydata.org/)
4. matplotlib - A plotting library.

   [More info](https://matplotlib.org/)
5. numpy - The fundamental package for numerical computations in Python.

   [More info](https://numpy.org/)
6. scikit-learn - Machine learning library in Python.

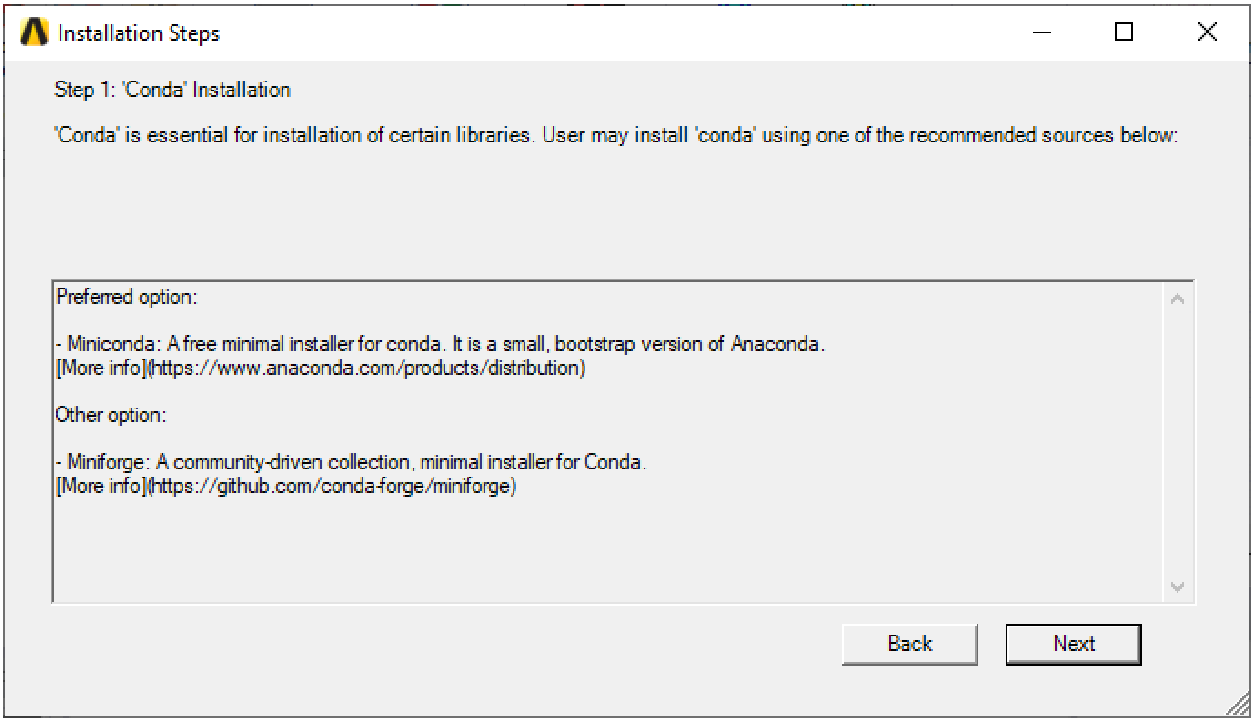
   [More info](https://scikit-learn.org/)
7. pyside6 - The official Python module from Qt for creating UIs.

   [More info](https://doc.qt.io/qtforpython/)

8. pygmo - A parallel optimization tool, useful for algorithmic research.

   [More info](https://esa.github.io/pagmo2/)

Step 1: Conda Installation

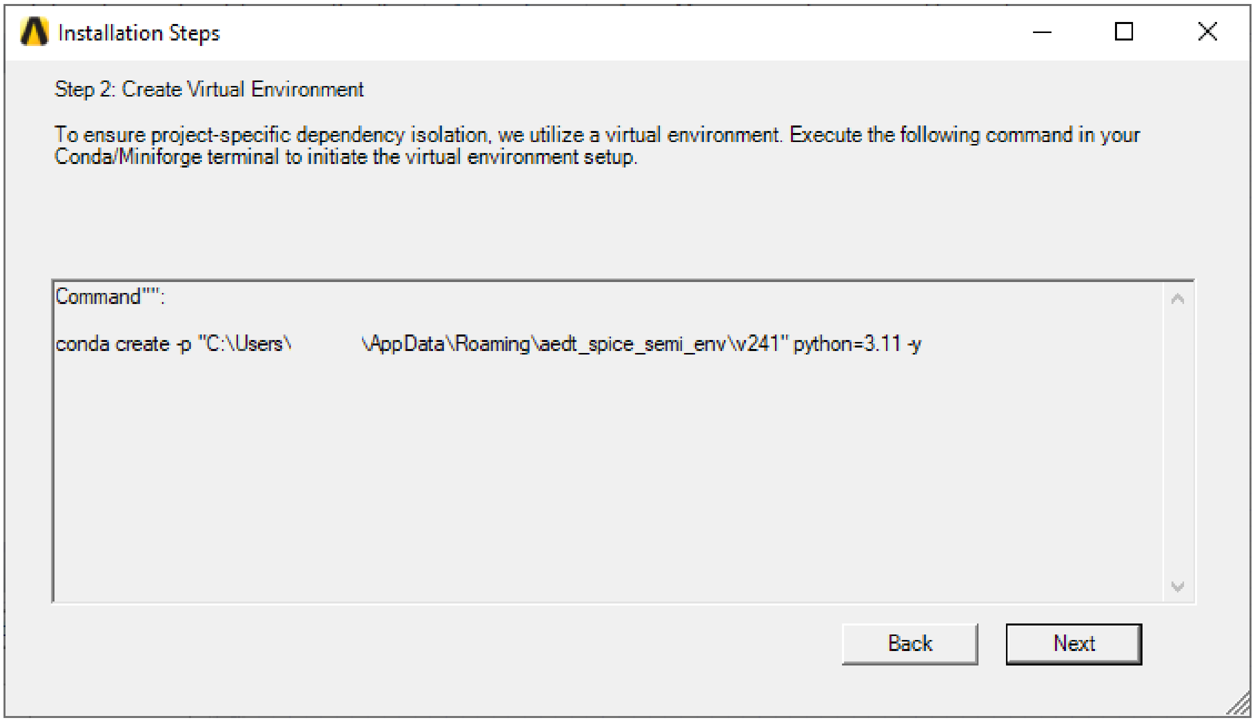


- Miniconda: A free minimal installer for conda. It is a small, bootstrap version of Anaconda.

  [More info](https://www.anaconda.com/products/distribution)
- Miniforge: A community-driven collection, minimal installer for Conda.

  [More info](https://github.com/conda-forge/miniforge)

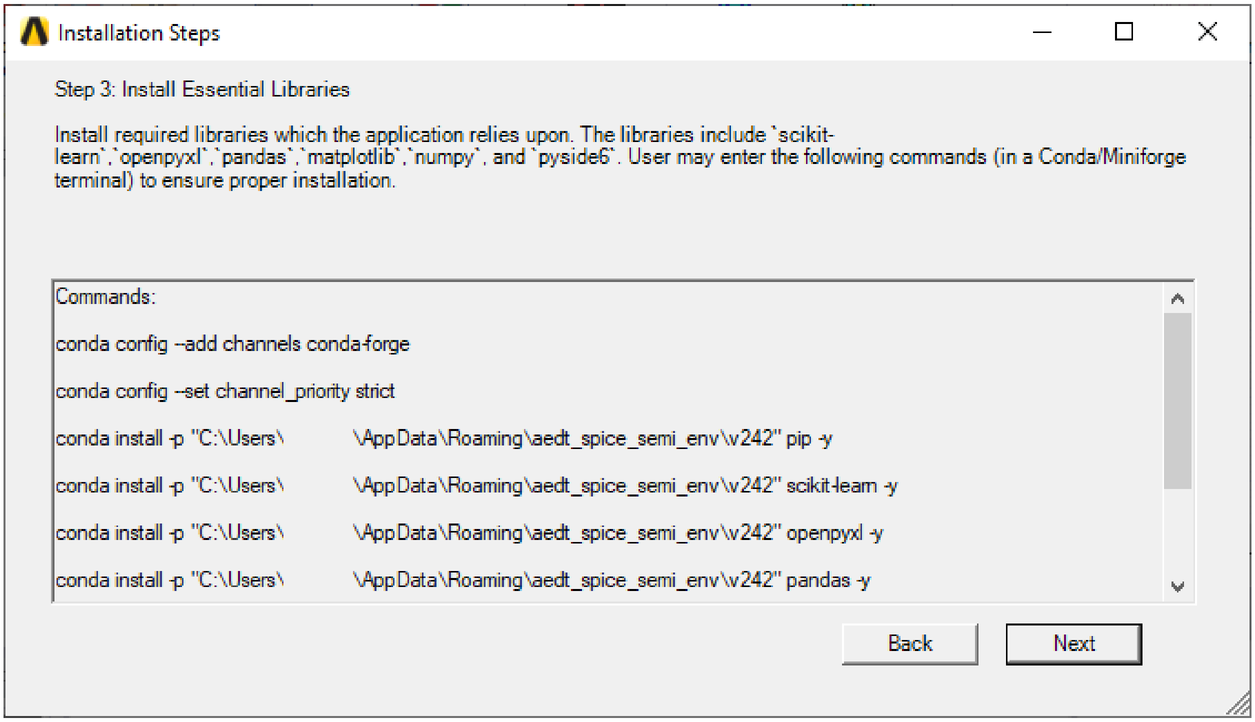
Step 2: Create Virtual Environment



Command:

```
conda create -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" -y
```

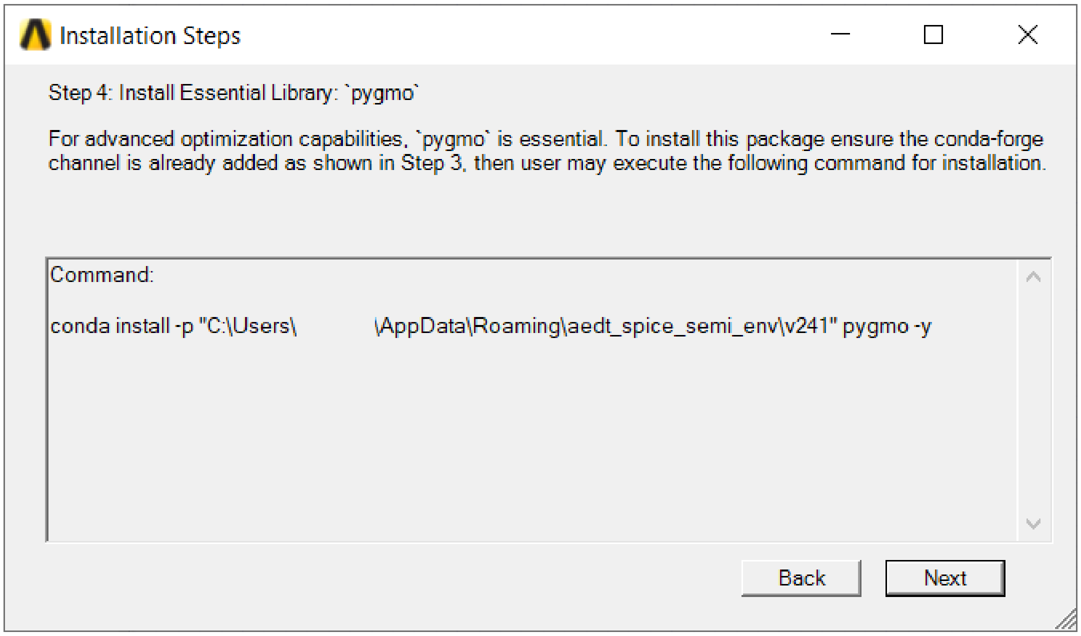
Step 3: Installing Essential Libraries.



Commands:

```
conda config --add channels conda-forge
conda config --set channel_priority strict
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" pip -y
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" scikit-learn -y
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" openpyxl -y
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" pandas -y
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" matplotlib -y
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" numpy -y
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" pyside6=6.4.3 -y
```

Step 4: Install Essential Library: "pygmo"



Command:

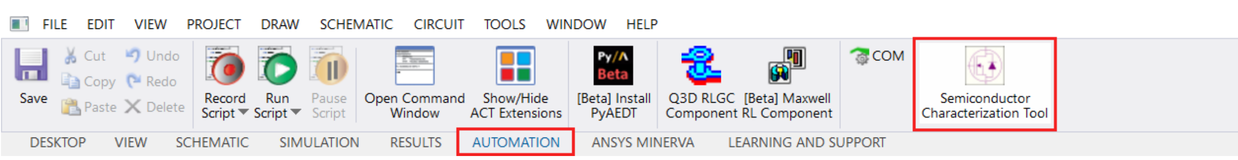
```
conda install -p "C:\Users\UserName\AppData\Roaming\aedt_spice_semi_
env\v242" pygmo -y
```

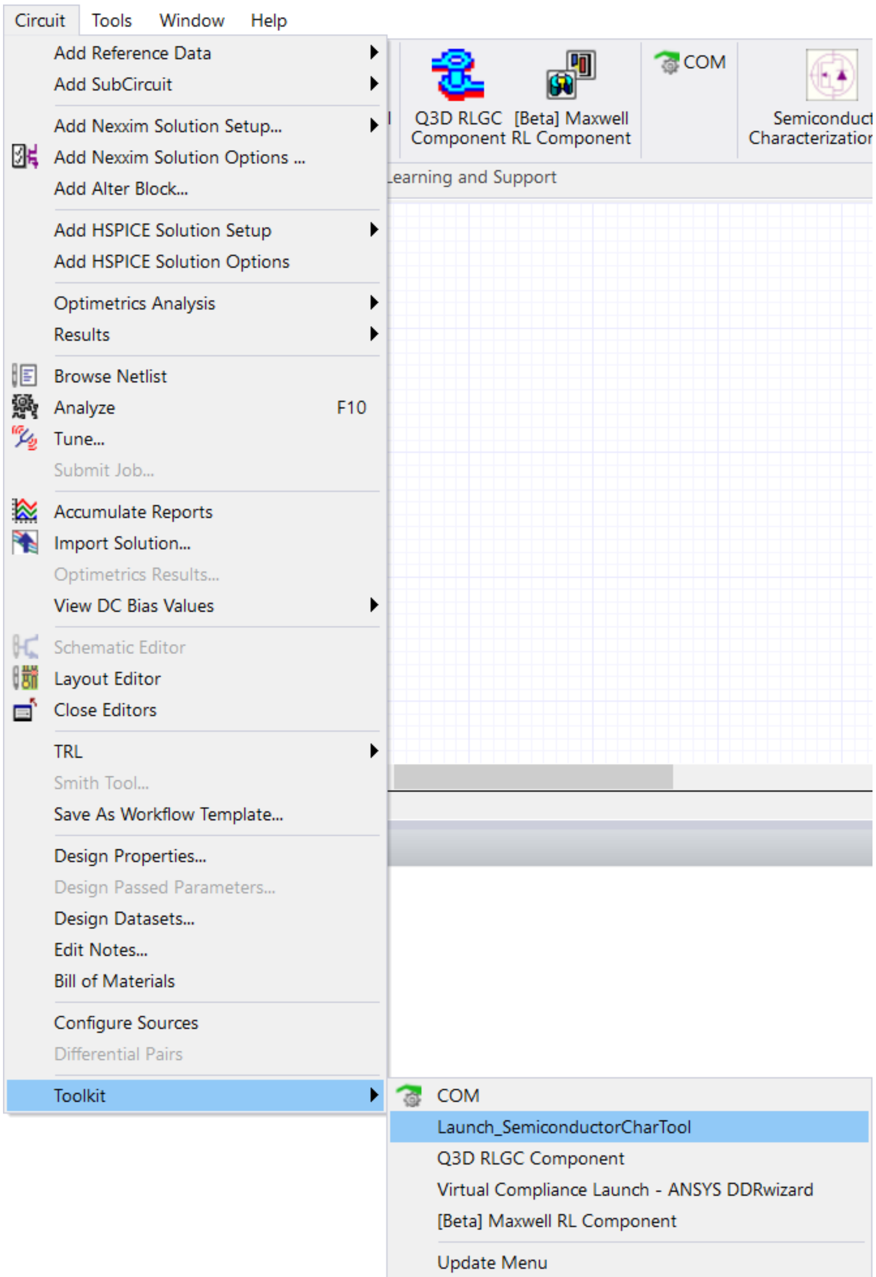
At the end of these steps, the toolkit will be ready to launch.

Semiconductor Characterization Tool

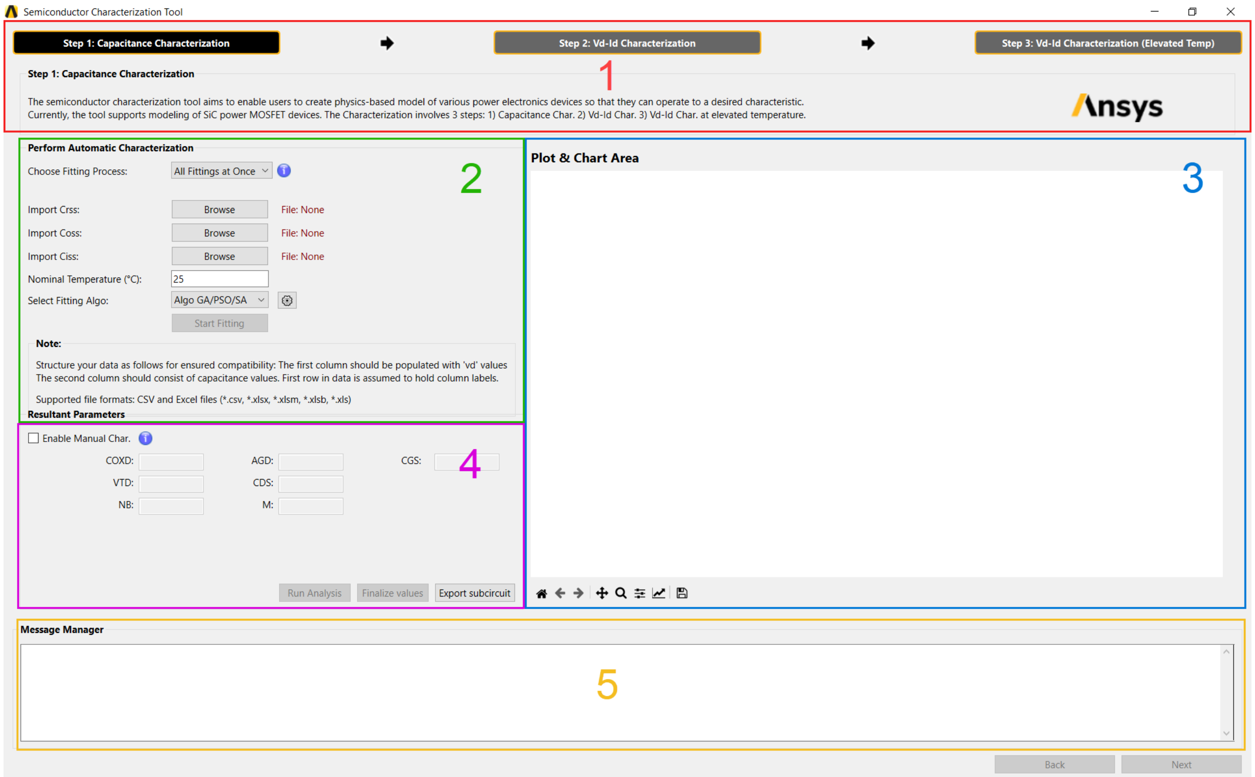
Overview

Click the **Semiconductor Characterization Tool** in the **Automation** tab of the main ribbon, or select **Circuit**>**Toolkit**> **Launch Semiconductor_CharTool**.




Circuit Tools Window Help
Add Reference Data
Add SubCircuit
Add Nexxim Solution Setup...
Add Nexxim Solution Options ...
Add Alter Block...
Add HSPICE Solution Setup
Add HSPICE Solution Options
Optimetrics Analysis
Results
Browse Netlist
Analyze F10
Tune...
Submit Job...
Accumulate Reports
Import Solution...
Optimetrics Results...
View DC Bias Values
Schematic Editor
Layout Editor
Close Editors
TRL
Smith Tool...
Save As Workflow Template...
Design Properties...
Design Passed Parameters...
Design Datasets...
Edit Notes...
Bill of Materials
Configure Sources
Differential Pairs
Toolkit
Q3D RLGC Component
[Beta] Maxwell RL Component
COM
Learning and Support
COM
Launch_SemiconductorCharTool
Q3D RLGC Component
Virtual Compliance Launch - ANSYS DDRwizard
[Beta] Maxwell RL Component
Update Menu

Overview



1. Lists the Step in the characterization process. The window will update according to the step number.

   The tool is comprised of 3 characterization steps.

   - **Step 1**: Capacitance Characterization
   - **Step 2**: Vd-Id Characterization
   - **Step 3**: Vd-Id Characterization (Elevated Temp)

2. **Perform Automatic Characterization** group box. Choose the fitting process, input the data files and temperature, and choose the fitting algorithm for analysis.

   There are two options for the fitting process:

   - **All Fittings at Once** - import / input the data for each step. After the data is loaded, select **Fit All** in Step 3.
   - **Step-by-Step Fitting** - perform each fitting one step at a time. This method ensures a good fit at each step.

3. **Plot & Chart Area**. The graph will update automatically to display the reference data and the simulated data.
4. **Enable Manual Characterization** group box. Users can input parameter values to run analysis.
5. **Message Manager**. Displays tool information such as file locations, step in analysis, and parameter values.

> **Note:** Structure your data as follows for ensured compatibility:
> The first column should be populated with 'vd' values. The second column should consist of capacitance values.
> The first row in the data file is assumed to hold column labels. Supported file formats inclue CSV and Excel files (*.csv, *.xlsx, *.xlsm, *.xlsb, *.xls).

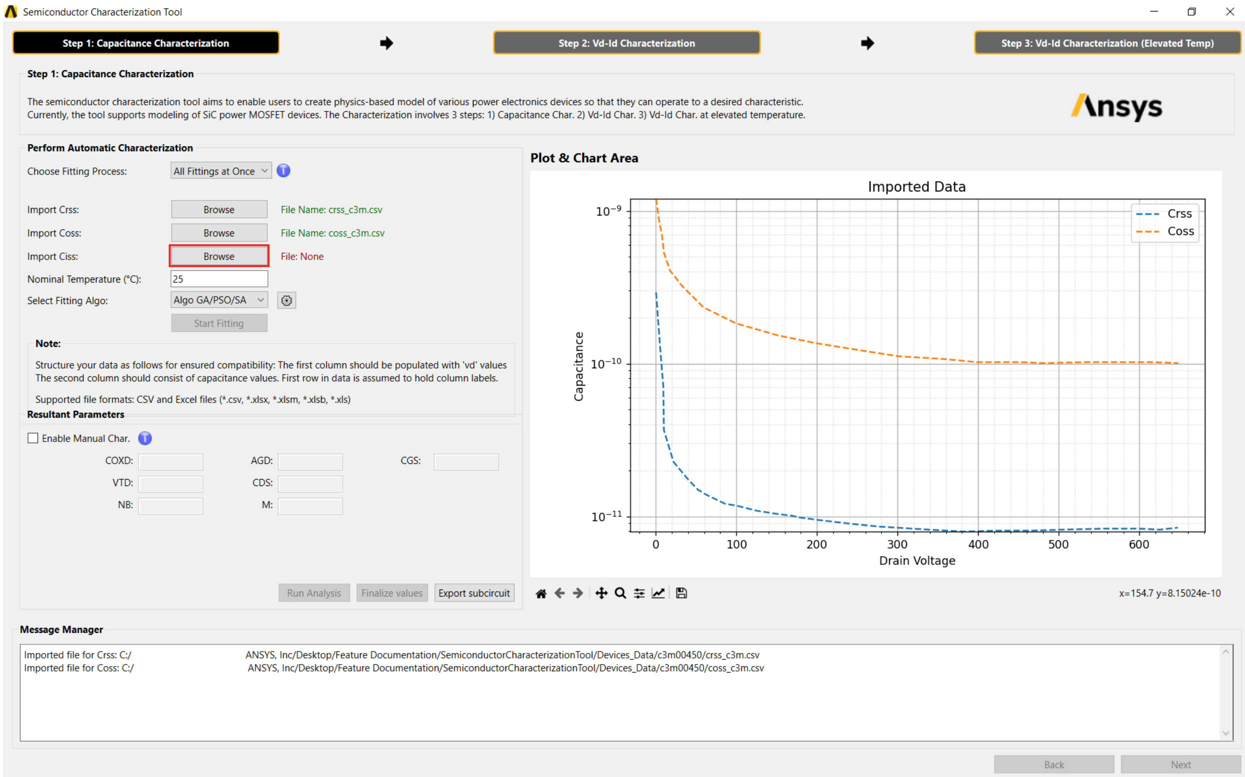
Semiconductor Characterization Tool

Characterization Process

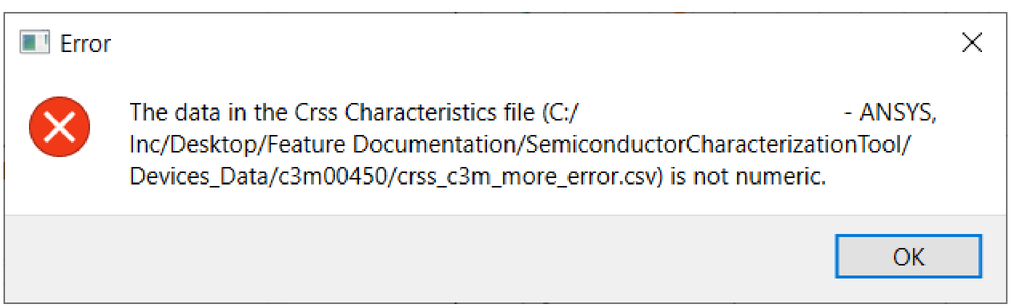
Step 1: Capacitance Characterization

Capacitance characterization involves sequential fitting of 3 curves: Crss, Coss, Ciss.

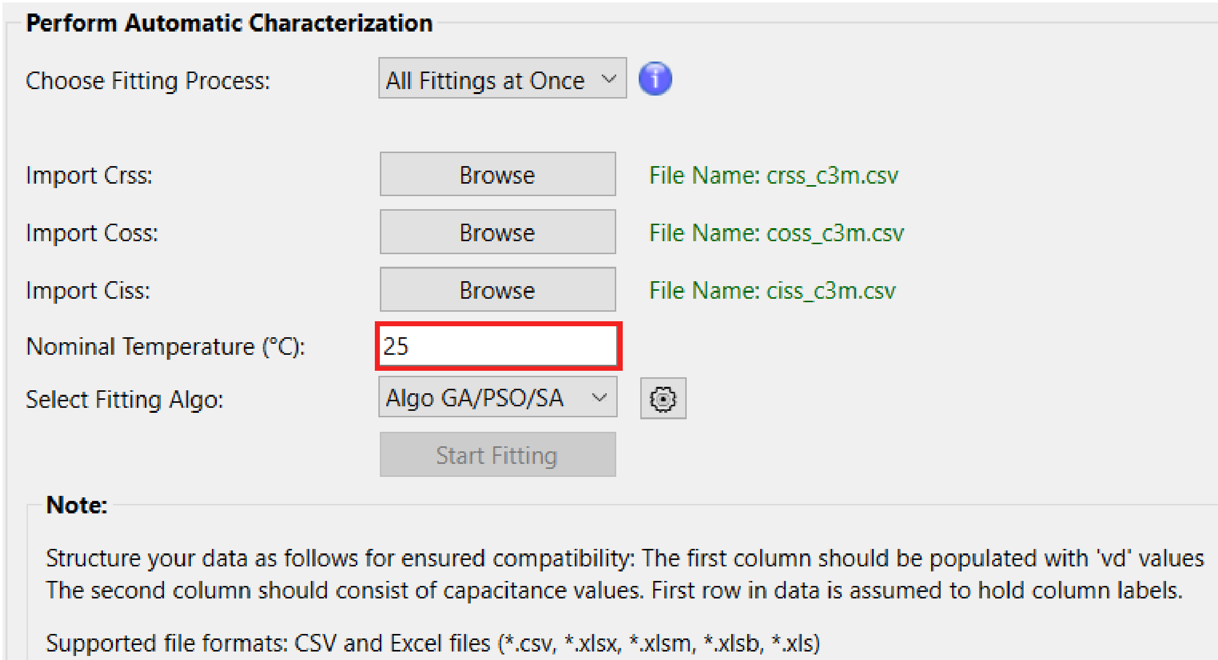
1. Use the **Browse** button for each curve type to import the appropriate file. The data will be plotted in the **Plot & Chart Area**.



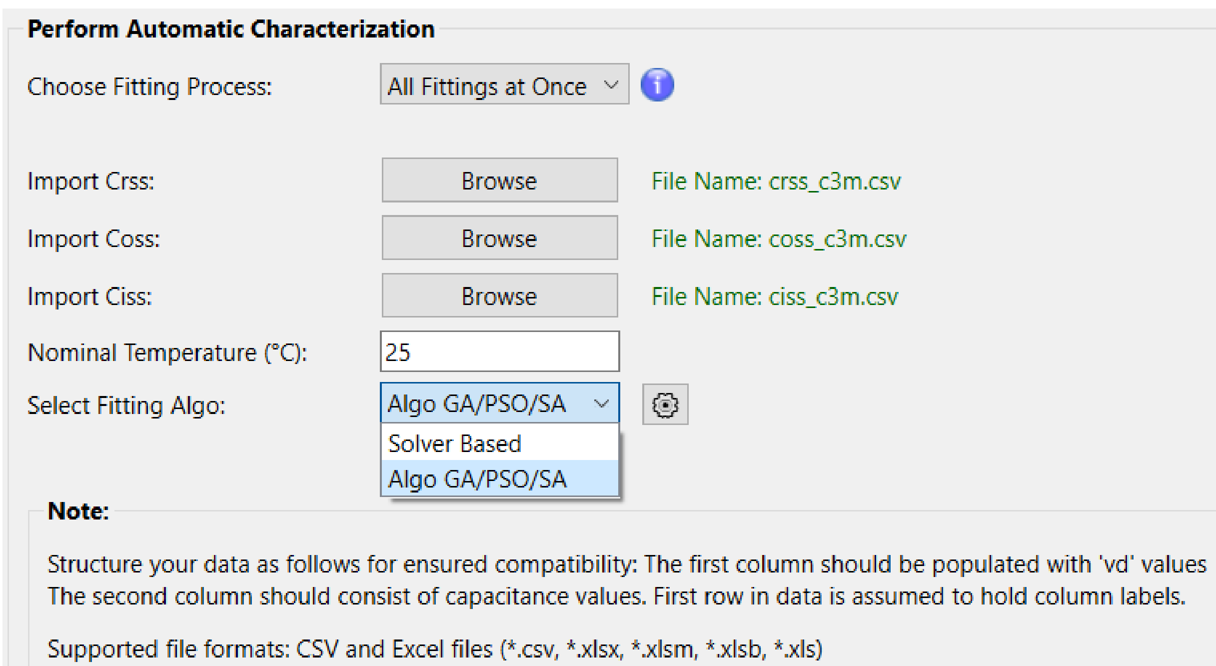
The tool will provide an error message if there is a non-numerical value in the data.



2. Enter the temperature at which the data has been obtained.



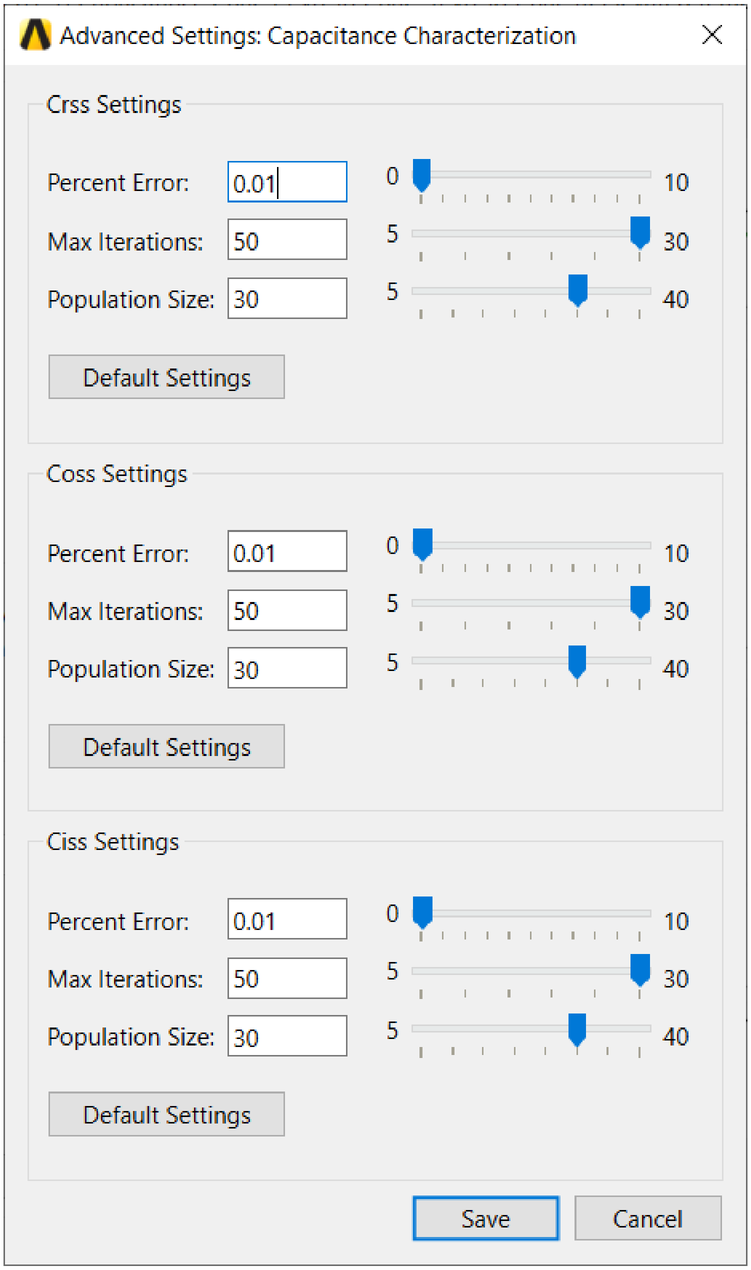
3. Select the fitting algorithm:



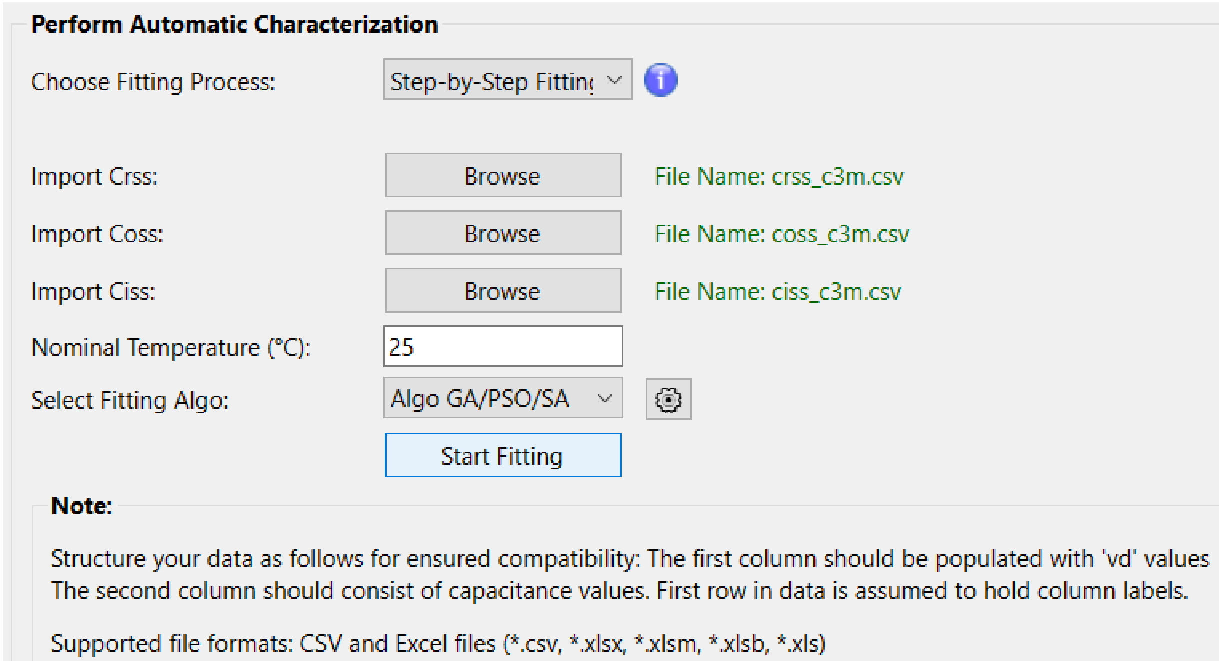
i. Solver Based- local optimization-based approach.

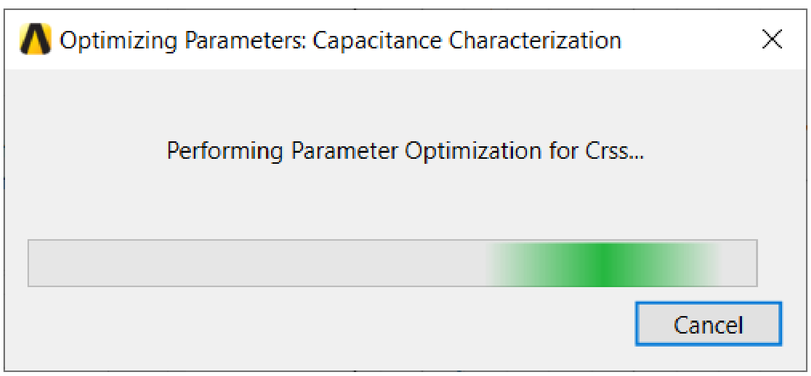
ii. Algo GA/PSO/SA

iii. Advanced Settings- User defined settings.

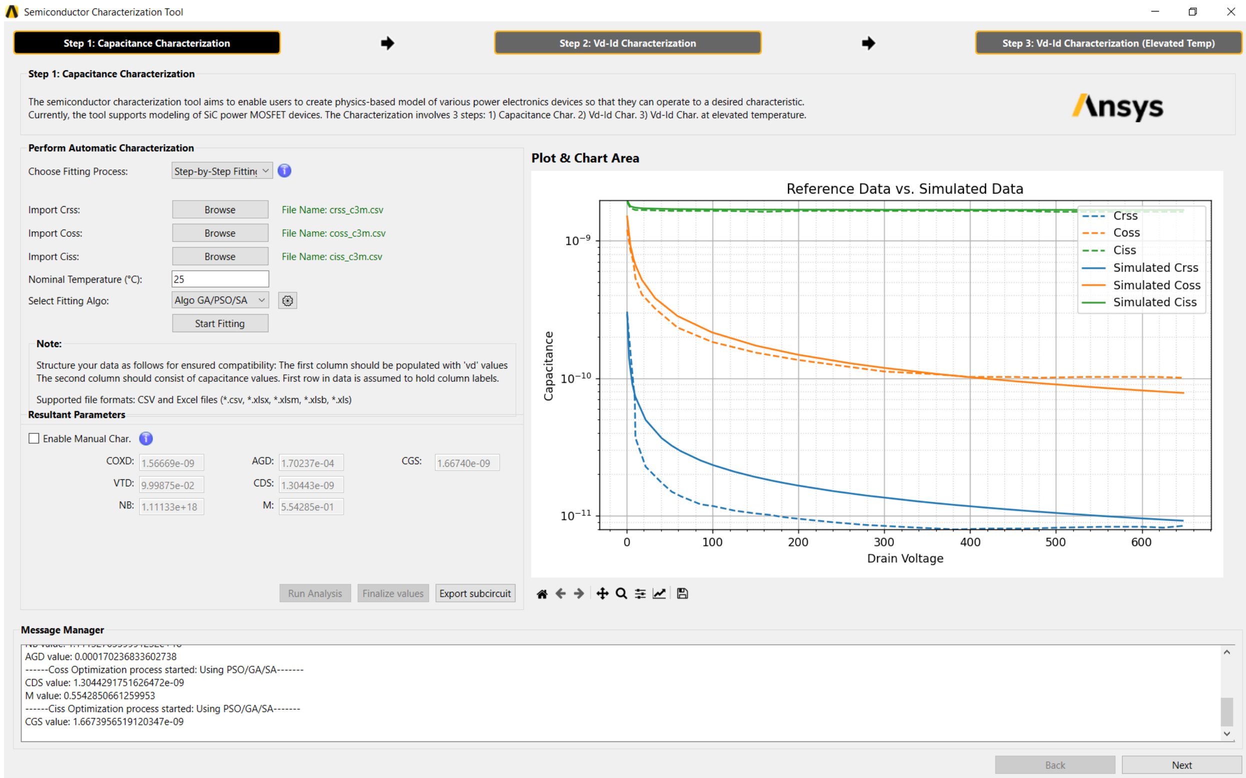


4. Select **Start Fitting**

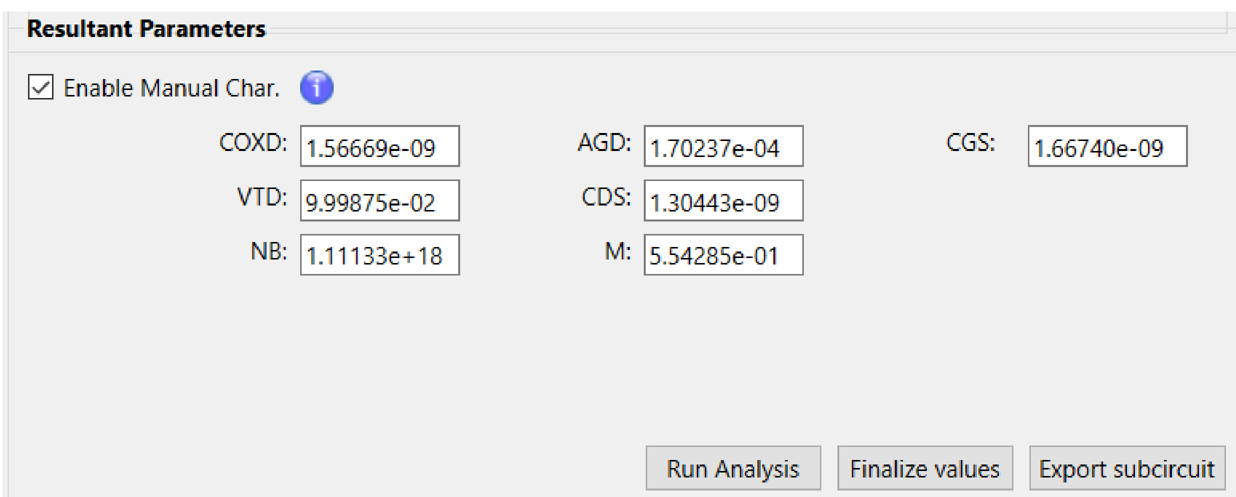




Once the fitting is complete, the **Plot & Chart Area** will update and the corresponding parameters will be populated in the **Resultant Parameters** dialog box. Note, the **Plot & Chart Area** will show the Reference Data and the Simulated Data.



5. (Optional) **Enable Manual Char.** button allows the user to enter values for the parameter fitting. Select **Run Analysis**. Once the results are satisfactory, select **Finalize Values**.

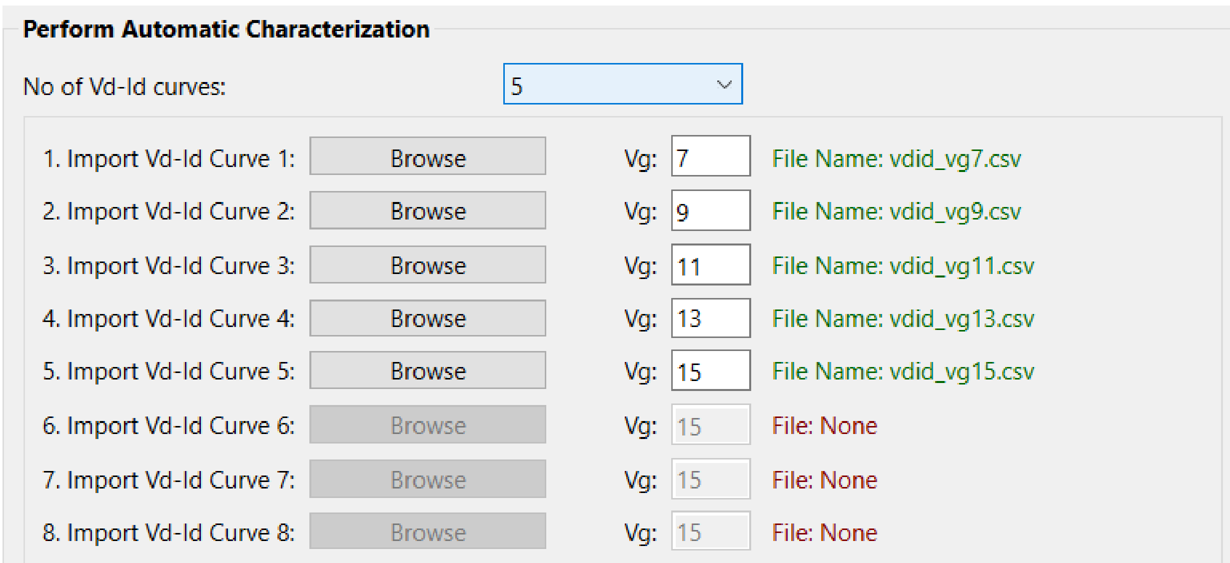


6. (Optional) The parameters may be exported using the **Export Subcircuit** button.
7. Select **Next** in the Semiconductor Characterization Tool window.

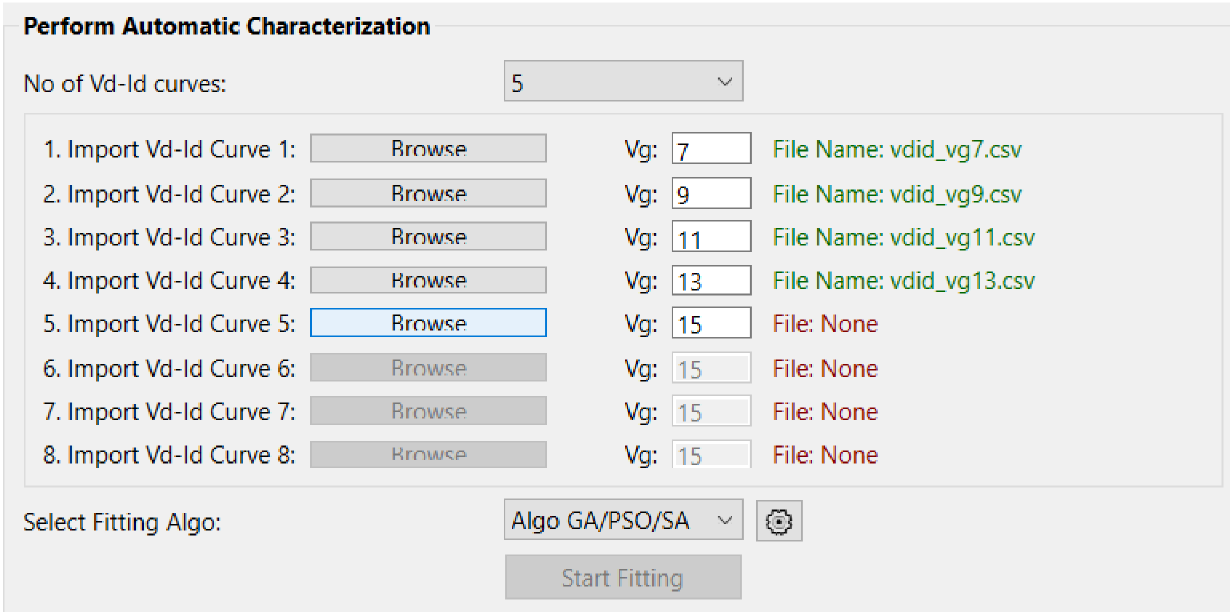
Note, the activation status of the **Next** button depends on the chosen **Fitting Type** as well as the presence of imported data and existing parameter values.

Step 2: Vd-Id Characterization

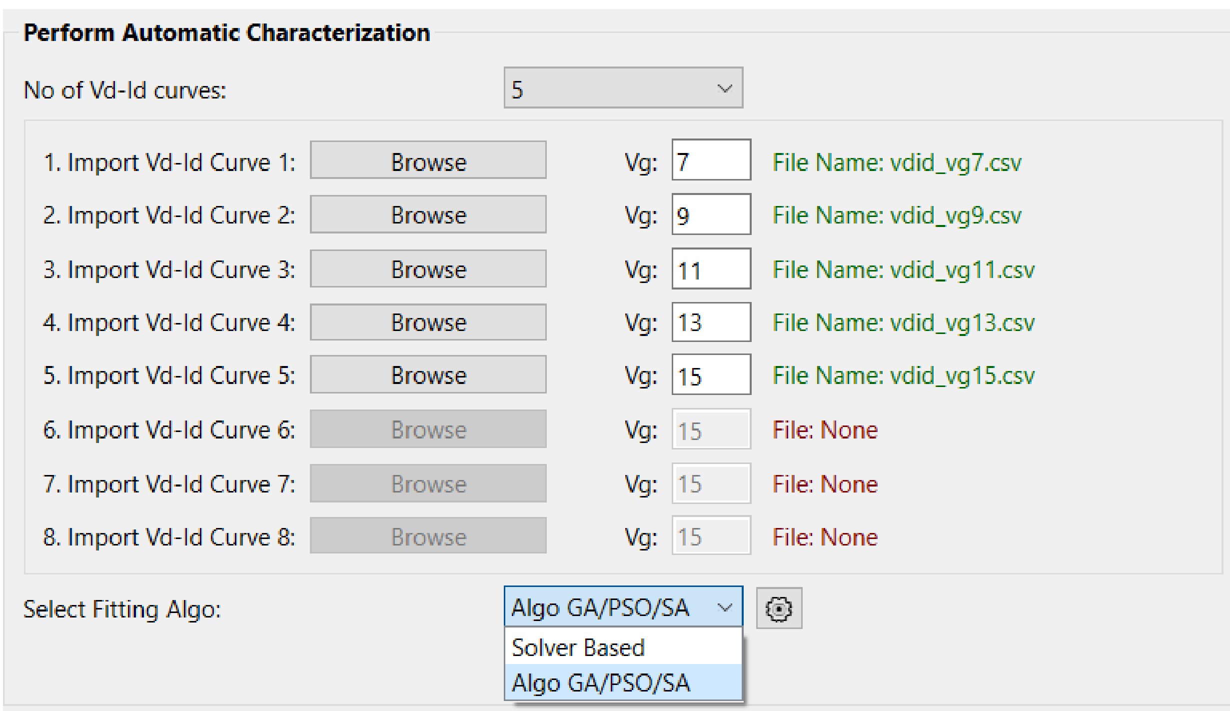
1. Select the number of **Vd-Id curves**.



2. Use the **Browse** button for each curve to import the appropriate file.



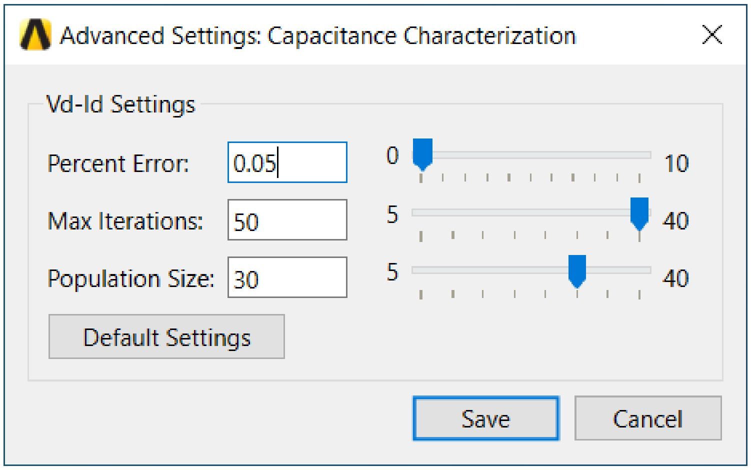
3. Select the fitting algorithm:



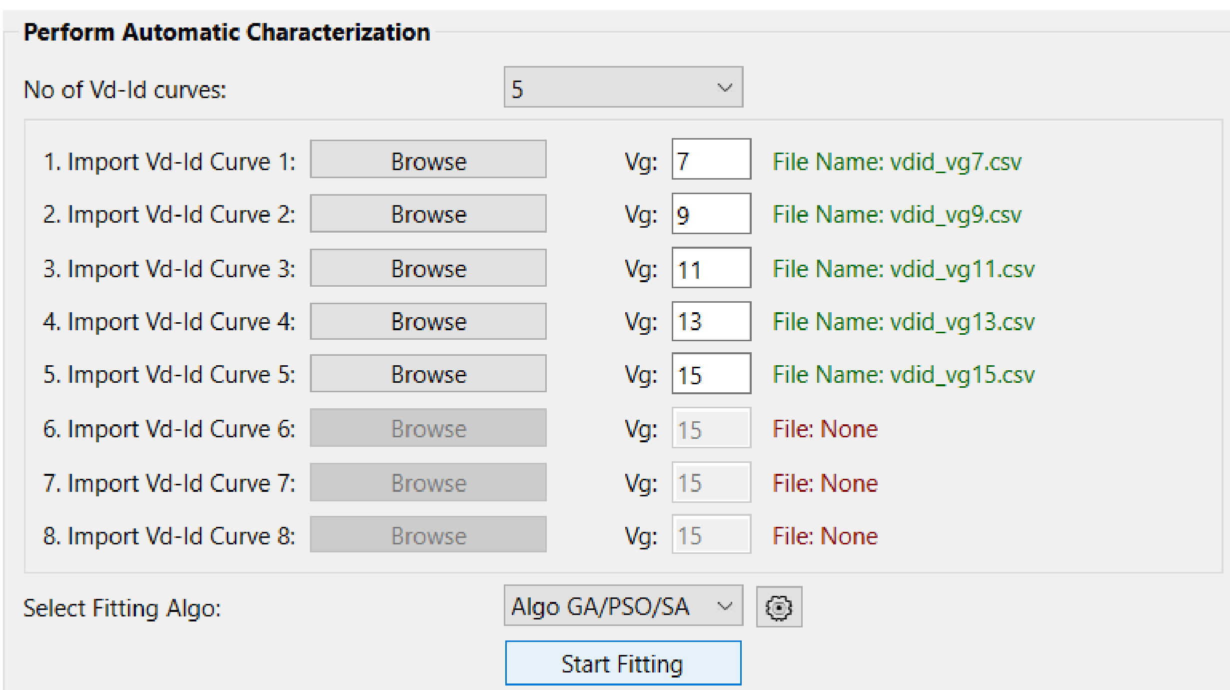
i. Solver Based- local optimization-based approach.

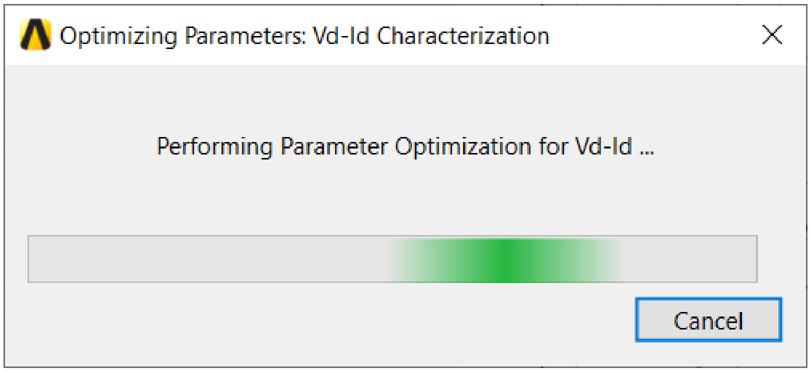
ii. Algo GA/PSO/SA

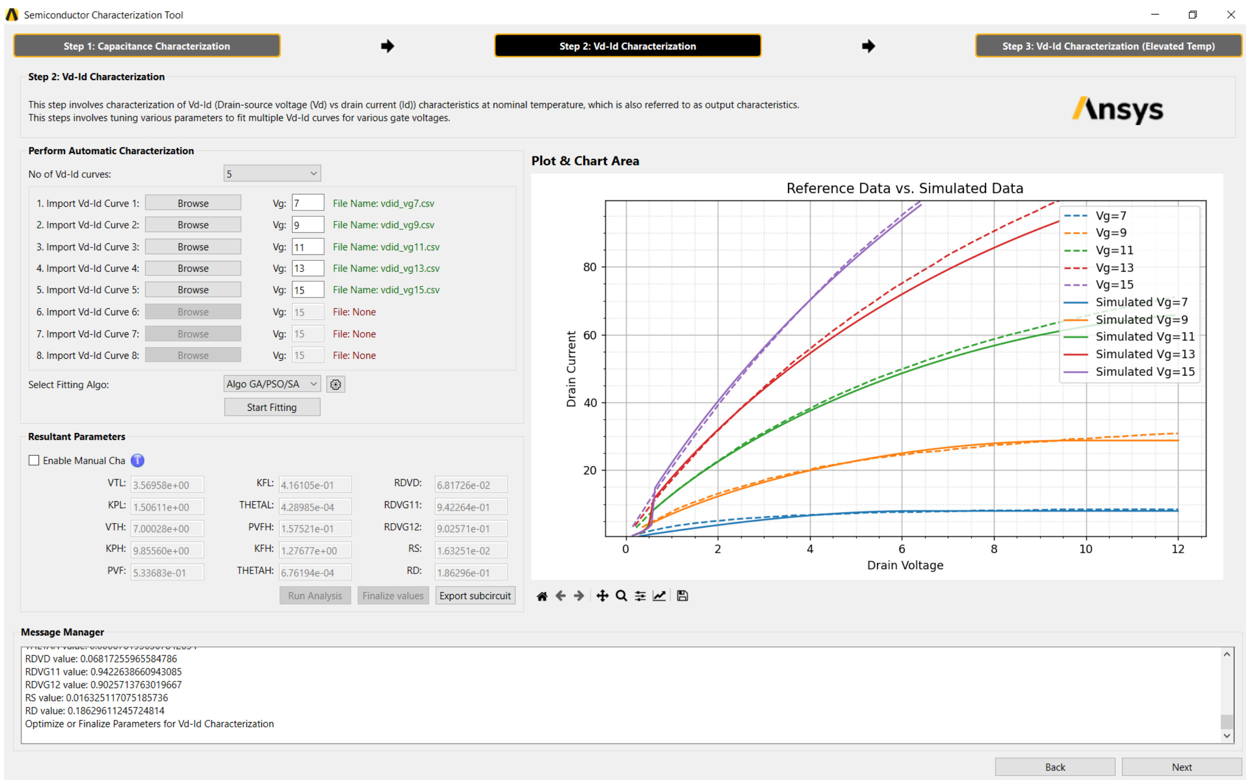
iii. Advanced Settings- User defined settings.



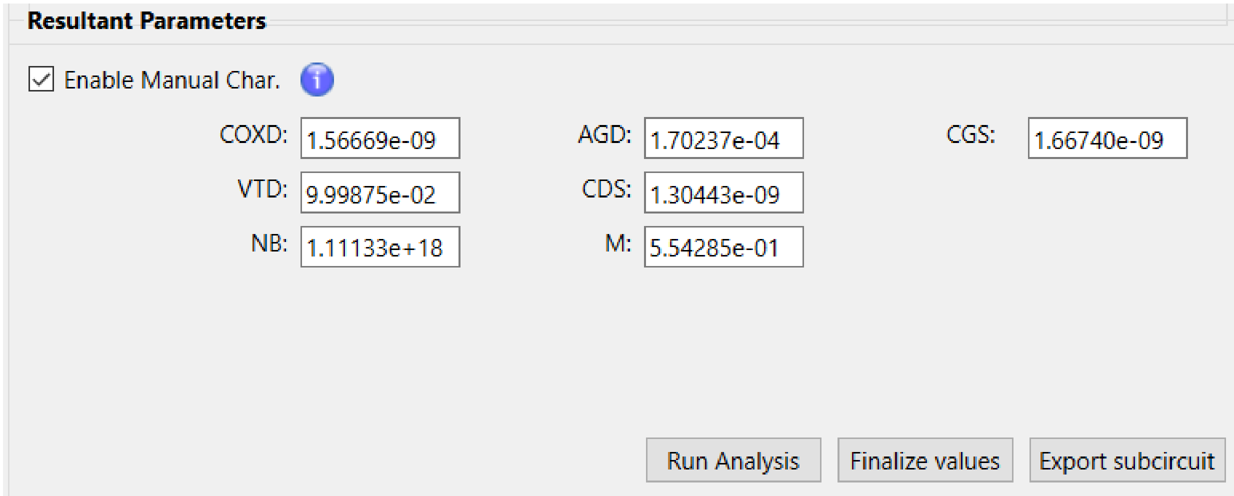
4. Select **Start Fitting**. The **Plot & Chart Area** will update and the corresponding parameters will be populated in the **Resultant Parameters** dialog box. Note, the Plot & Chart Area will show the Reference Data and the Simulated data.







5. (Optional) **Enable Manual Cha** button allows the user to enter values for the parameter fitting. Select **Run Analysis**. Once the results are satisfactory, select **Finalize Values**.

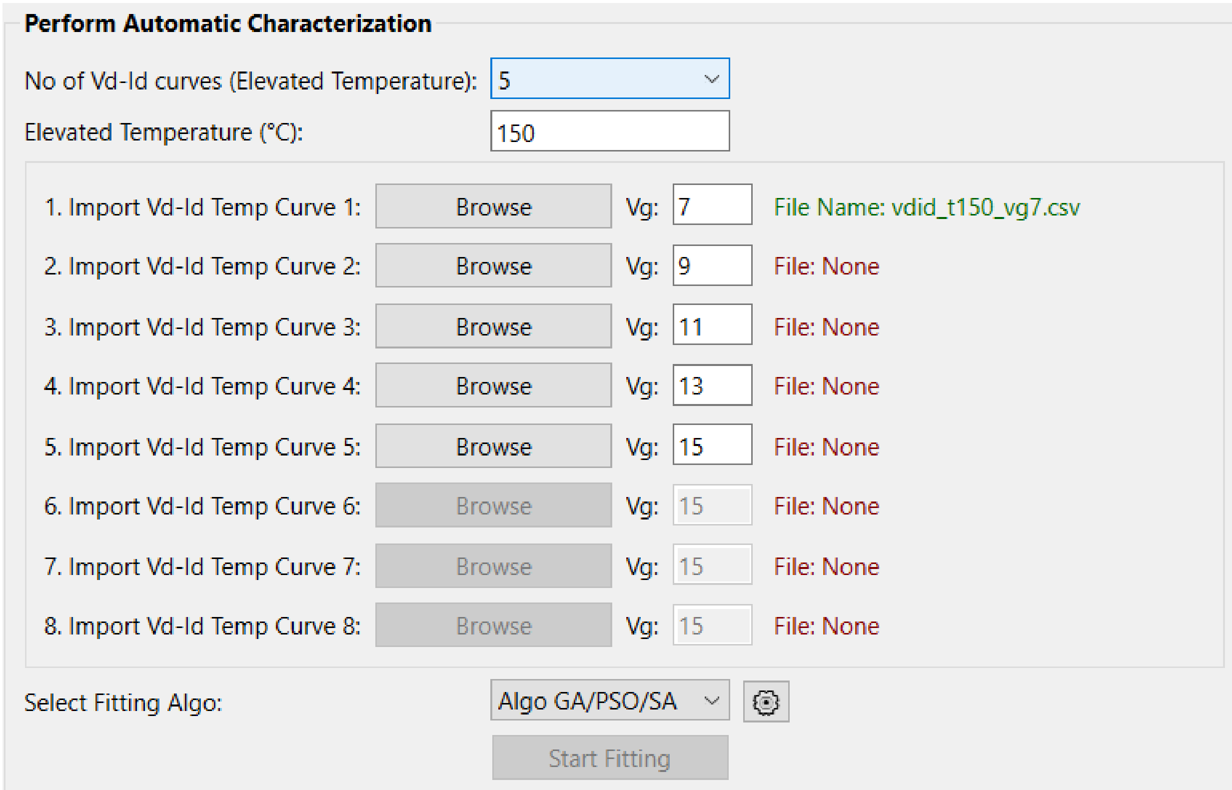


6. These parameters may be exported using the **Export Subcircuit** button.
7. Select **Next**

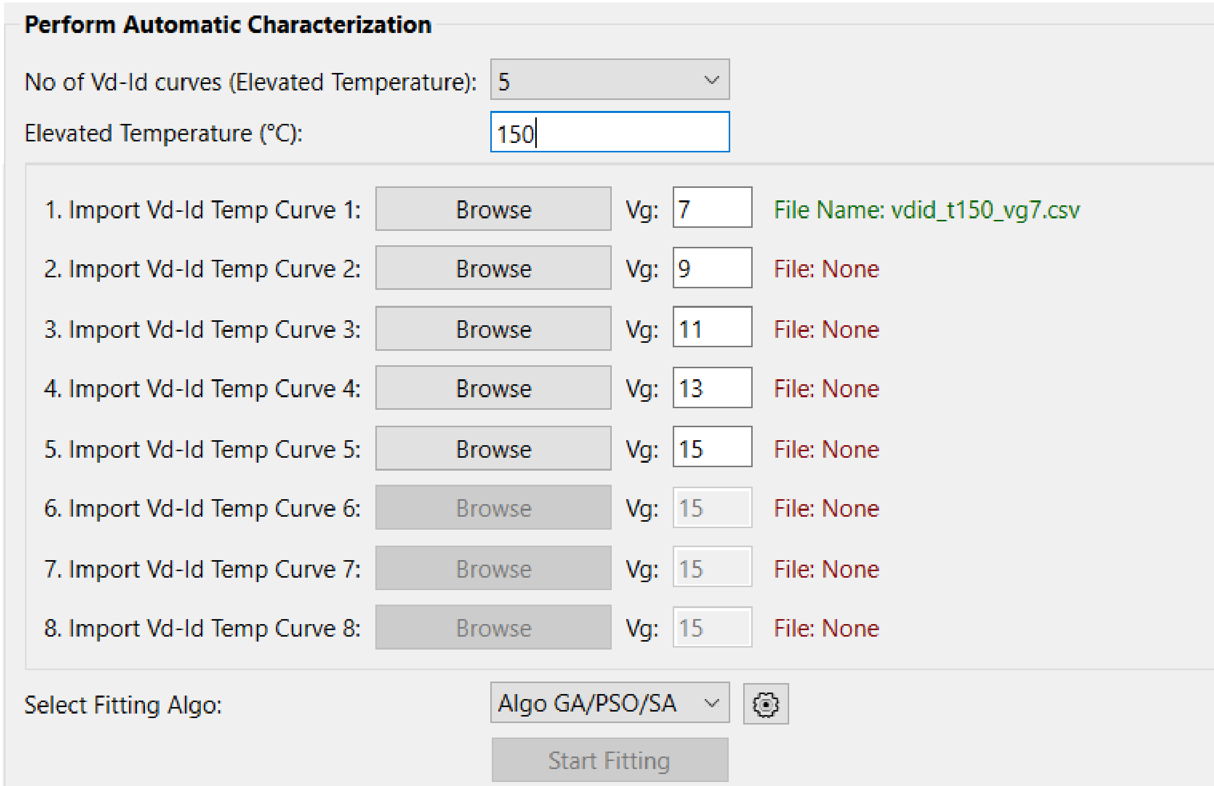
Step 3: Vd-Id Characterization (Elevated Temp)

The Vd-Id Characterization at Elevated Temperature is like the Step 2 process, with the additional parameter of the user-entered Elevated Temperature.

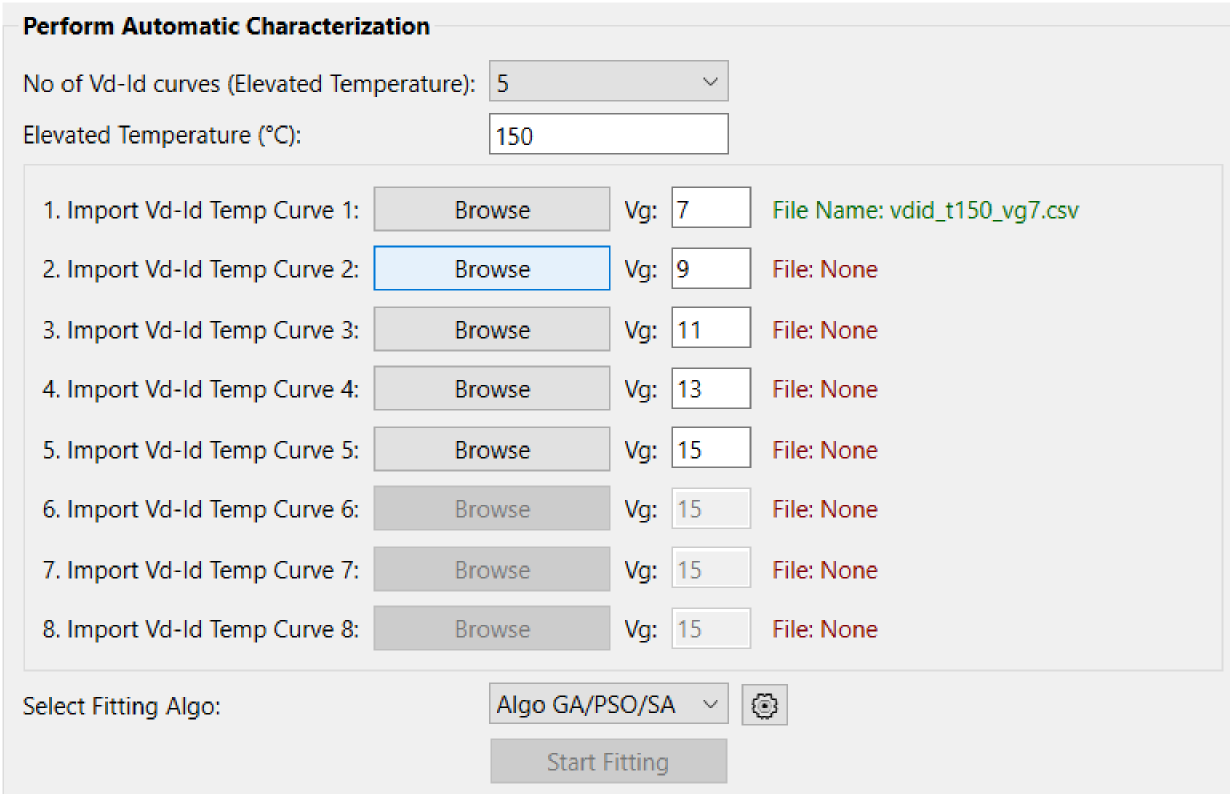
1. Select the number of **Vd-Id curves**.



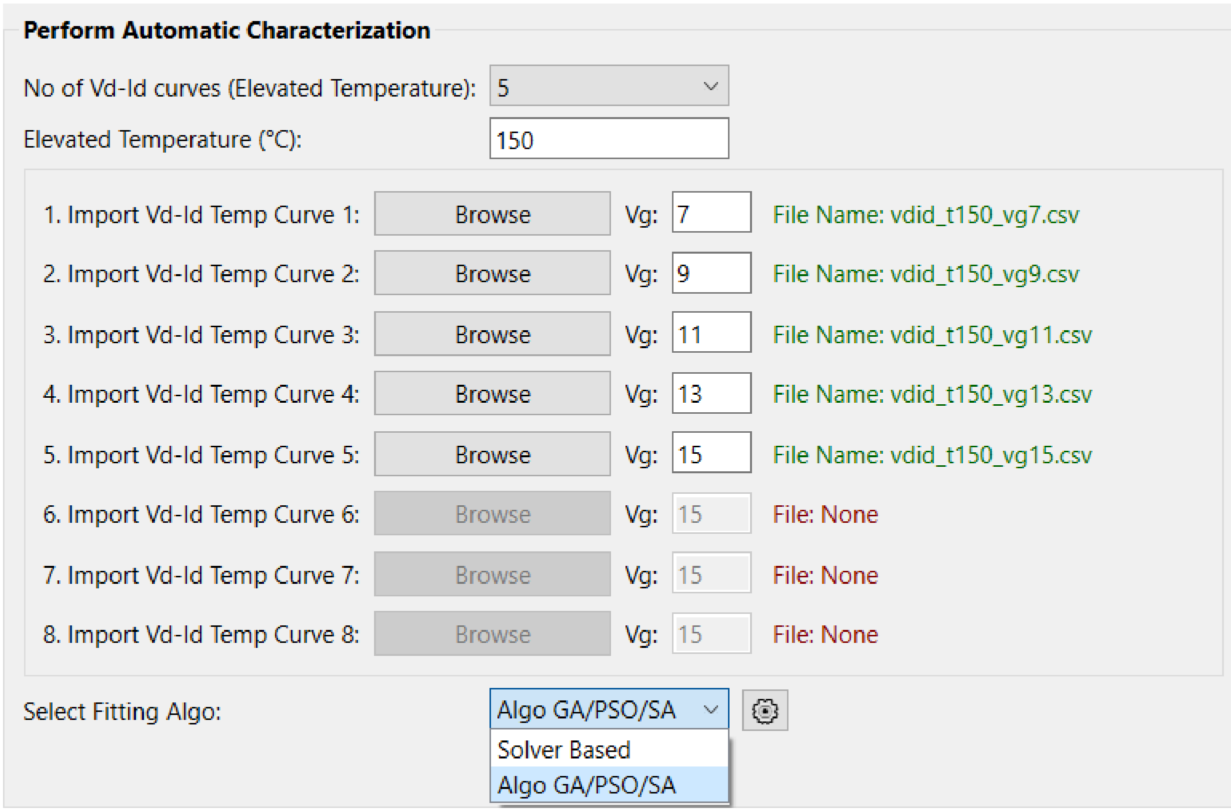
2. Enter the **Elevated Temperature** in degrees C.



3. Use the **Browse** button for each curve to import the appropriate file.



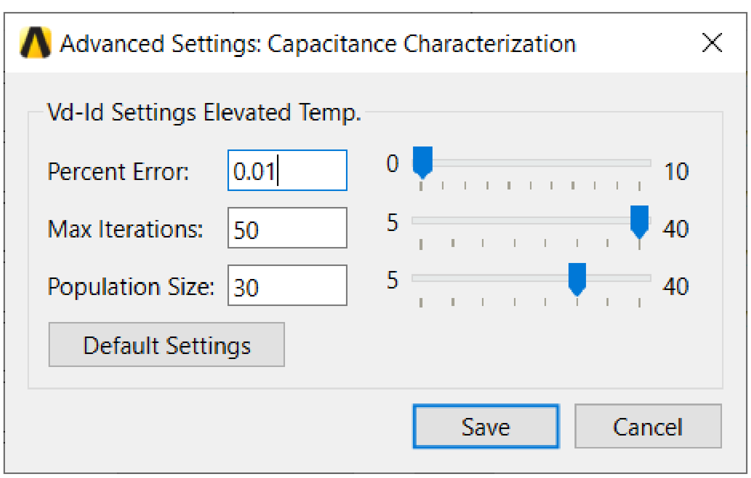
4. Select the fitting algorithm:



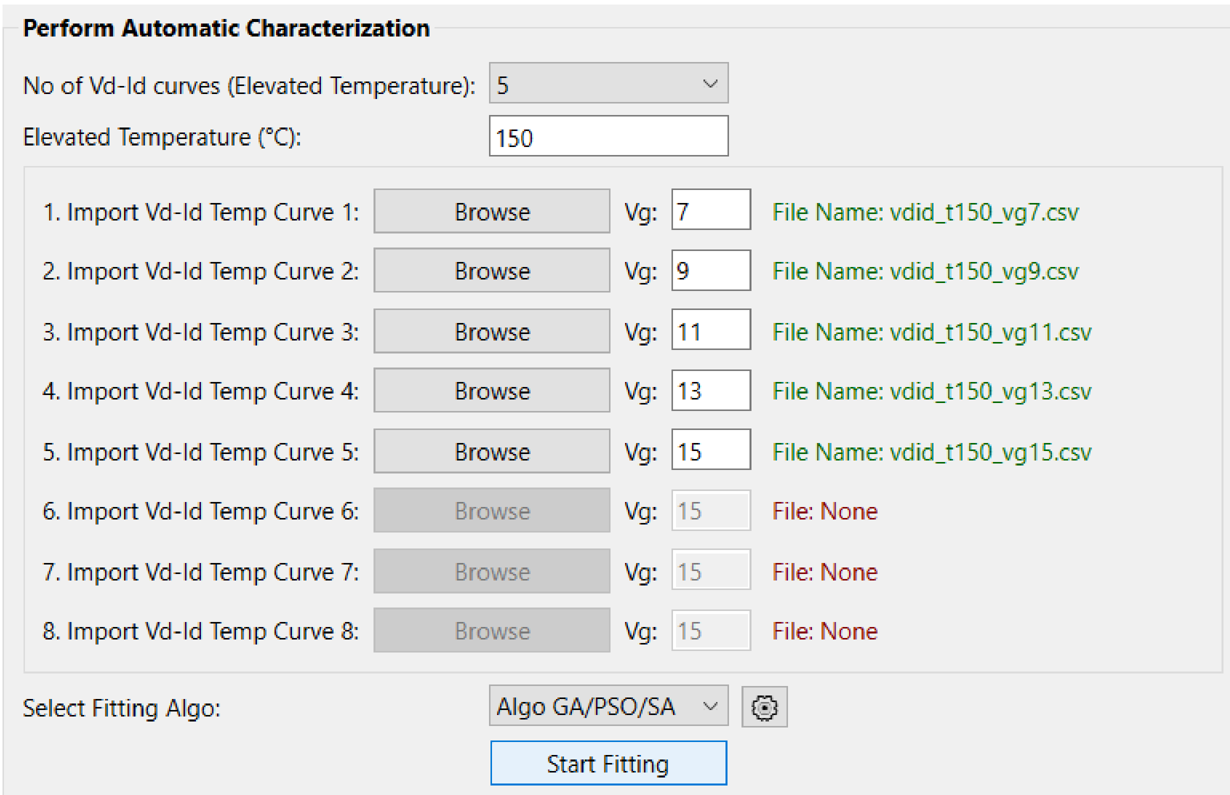
i. **Solver Based** - local optimization-based approach.

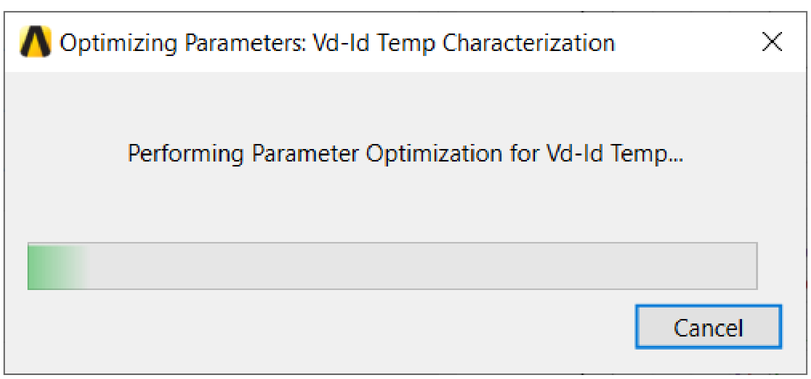
ii. **Algo GA/PSO/SA**

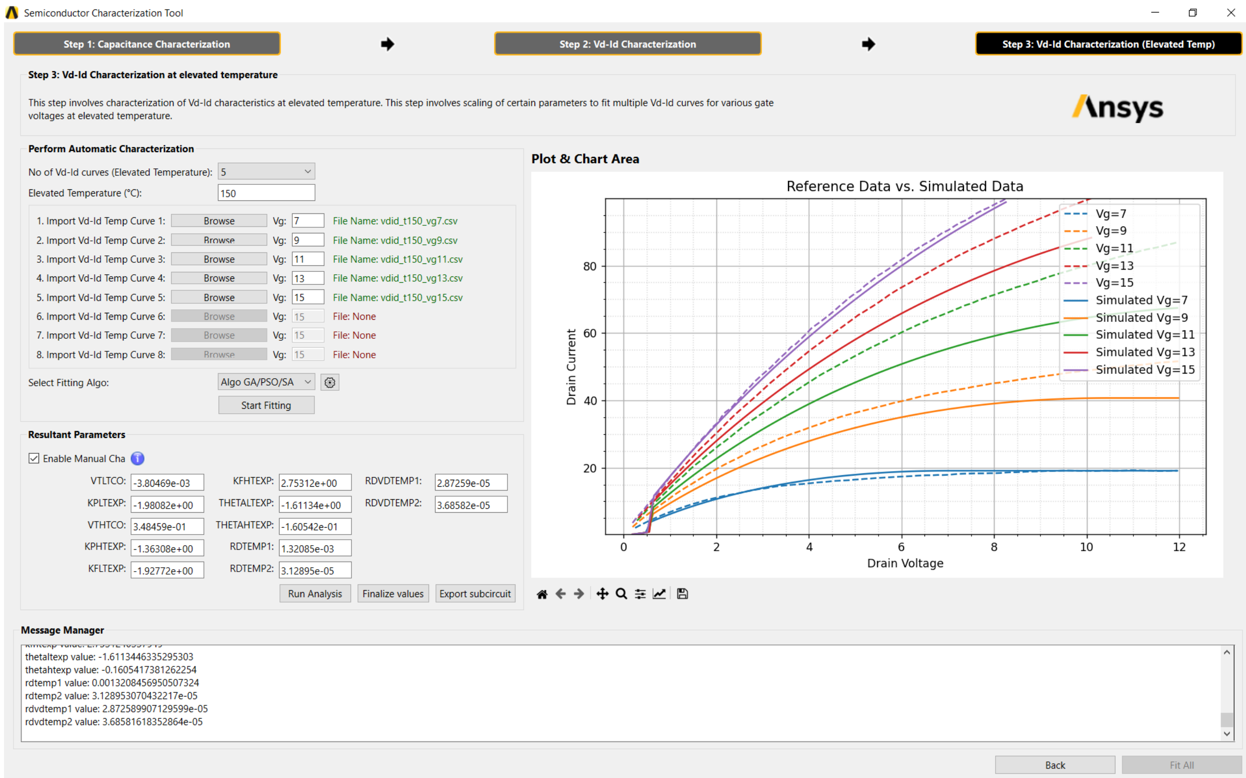
iii. **Advanced Settings icon** Launches a dialog. Default settings option.



5. Select **Start Fitting**. The **Plot & Chart Area** will update and the corresponding parameters will be populated in the **Resultant Parameters** dialog box. Note, the Plot & Chart Area will show the Reference Data and the Simulated data.







6. (Optional) **Enable Manual Cha** button allows the user to enter values for the parameter fitting. Select **Run Analysis**. Once the results are satisfactory, select **Finalize Values**.
7. (Optional)The parameters may be exported using the **Export Subcircuit** button.

15 - SPISim

Ansys SPISim is a simulation technology that performs modeling and signal integrity analysis. It features:

- IBIS Modeling Suite, a simplified workflow for generating new IBIS and IBIS-AMI models and for inspecting existing components.
- COM and USB Channel Compliance Testing, a wizard-based approach for setting up and reviewing criteria for channel performance.
- Full channel operating margin (COM) analysis summary report that can be generated through SPISim or through Electronic Desktop. Refer to Generate a COM Analysis Report for details.
- S-parameter capabilities support Ansys's proprietary State-Space Model format (i.e., extension .sss), as well as Touchstone 1.0 and Touchstone 2.0 formats (e.g., extensions .s1p, s2p, s3p, .s*n*p, etcetera).

Launch SPISim

SPISim is available with an Ansys Electronics Enterprise license. It is launched on the **Electronics Desktop**. In the menu bar, click **Tools** > **SPISim**.

You can also launch SPISim from a command prompt. In Microsoft Windows, run the executable %ProgramFiles%\AnsysEM\<version>Win64\spisim\bin\spisim64.exe.

On Linux, run the script <installdir>Linux64/spisim/bin/SPISIM. Built-in NPort and IBIS models for on-the-fly specification model generation can be accessed though the Schematic editor.

SPISim Modules

SPISim offers several modules for different functions.

- SPISim AMI, a free tool to test drive IBIS AMI modeling.
- SPISim AMI Modeling and Testing, how to generate AMI models and test the model.
  - **Features Added in 22.2**: Expanded Generate Spec. IBIS-AMI Model, PAM4 Modeling Support, PAM4 and DDR5 Analyses Options, and DC_OFFSET Keyword Options.
  - **Features Added in 23.2**: IBIS/AMI Enhancements.
  - **New Features Added in 24.1**: Adaptive CTLE IBIS-AMI Model.
- SPISIM Full AMI Modeling Flow, six demonstrations of modeling and other key features.
- SPISim BPro, an IBIS model generation and processing module.
- SPISim BPro Advanced Features, advanced modeling features in BPro.

- SPISim IBIS Hands-On Workshop, demonstrations with SPIPro.
  - **Features Added in 22.2**: IBIS V7.1 Support.
  - **Features Added in 23.2**: IBIS Figure of Merits Report, IBIS-AMI Model Expiration Selection.
  - **New Features Added in 24.1**: IBIS Modeling Flow Batch Mode Support. Clock Forwarding support. New sample models PCIeG5_Tx and PCIeG6_TxRx.
- SPISim MPro, a modeling tool.
- SPISim Proxy, an IBIS-AMI modeling proxy module.
- SPISim SPro, a S-parameter model.
  - **Features Added in 23.1**: Expanded GUI/Non-GUI Batch Command Line Compliance Analysis Report.
- SPISim TPro, a transmission line analysis and stackup model.
- SPISim VPro, a waveform viewer and analysis module.
  - Effective return loss (ERL) calculations are conducted by SPISim via the VPro waveform viewer and analysis module. Refer to Calculating Effective Return Loss for details.

See the online help for videos about these modules.

Navigate Between Modules

From the SPISim application window, select **Module** from the main ribbon to view the available modules. Select the module to open it in the current application window.

SPISim and Nexxim

SPISim uses Nexxim as the default simulator for IBIS and TLine model generation. To use another simulator, open SPISim and click **Tools** > **Options**. In the **Options** window, click **SPISim** and open the **Main** tab. Enter the new simulator path in the **Spice simulator** field and click **OK**.

SPISim AMI

This utility is an IBIS-AMI compliant model driver. It can be used by model users, developers and publishers for IBIS AMI testing and development.

See the Online Help for a video on this module.

In this Section:

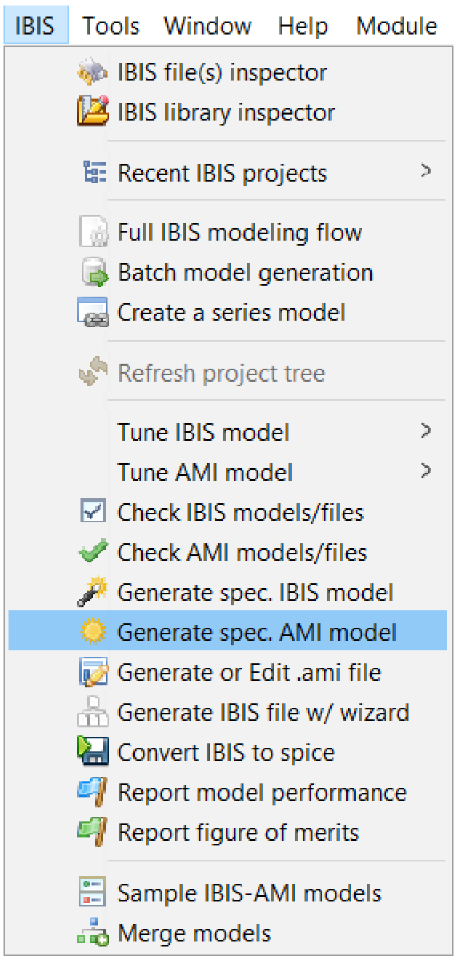
- Generate Spec. IBIS-AMI model.
- Sample IBIS-AMI Models.

Generate Spec. IBIS-AMI model: Overview

This section contains information and usage guides for Spec. IBIS-AMI model generation.

To launch the Generate Spec IBIS-AMI window:

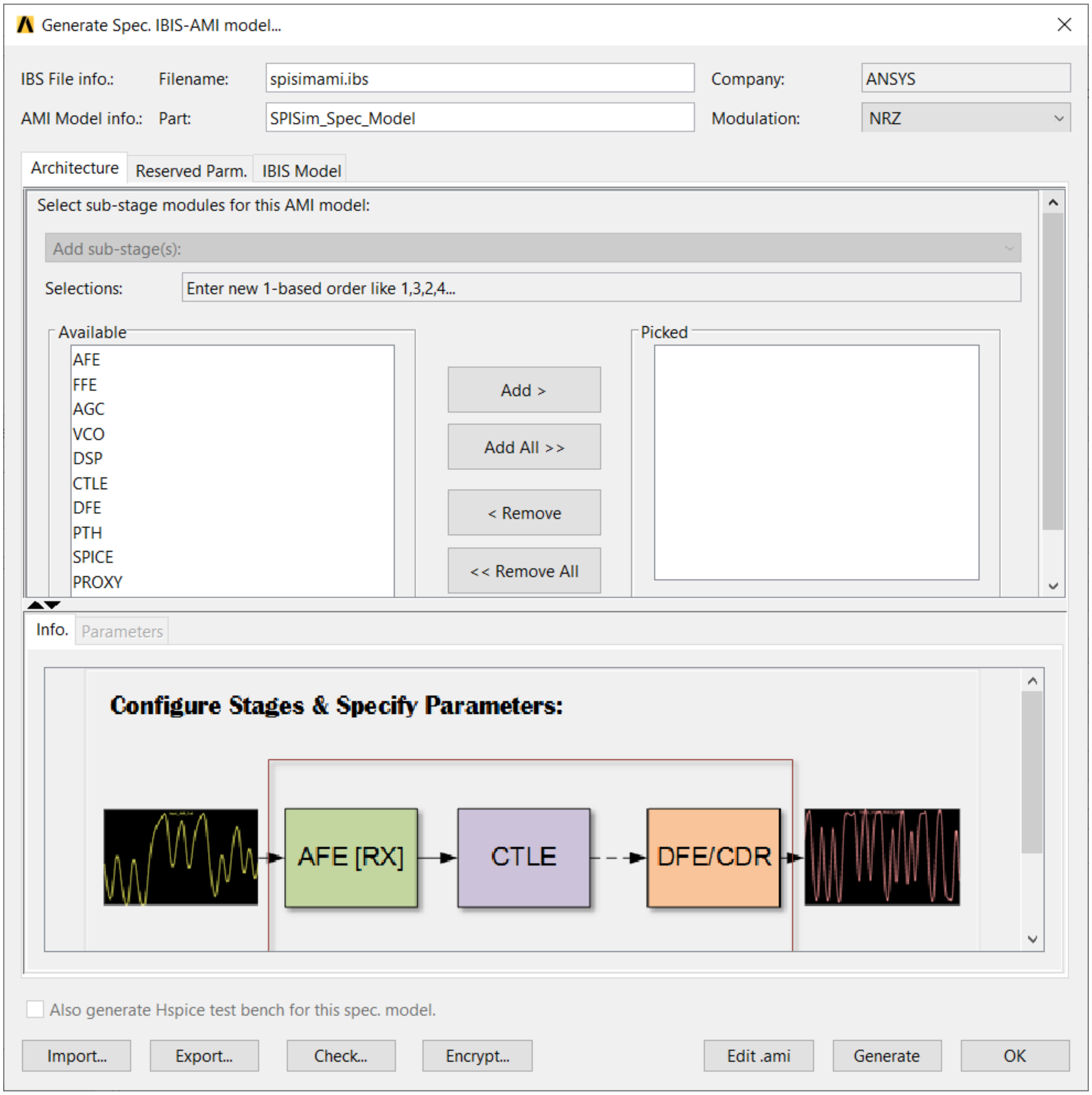
1. From the IBIS menu, select Generate Spec. AMI



The Generate Spec. IBIS-AMI window will launch. Model generation configurable parameters have been separated into three tabs:

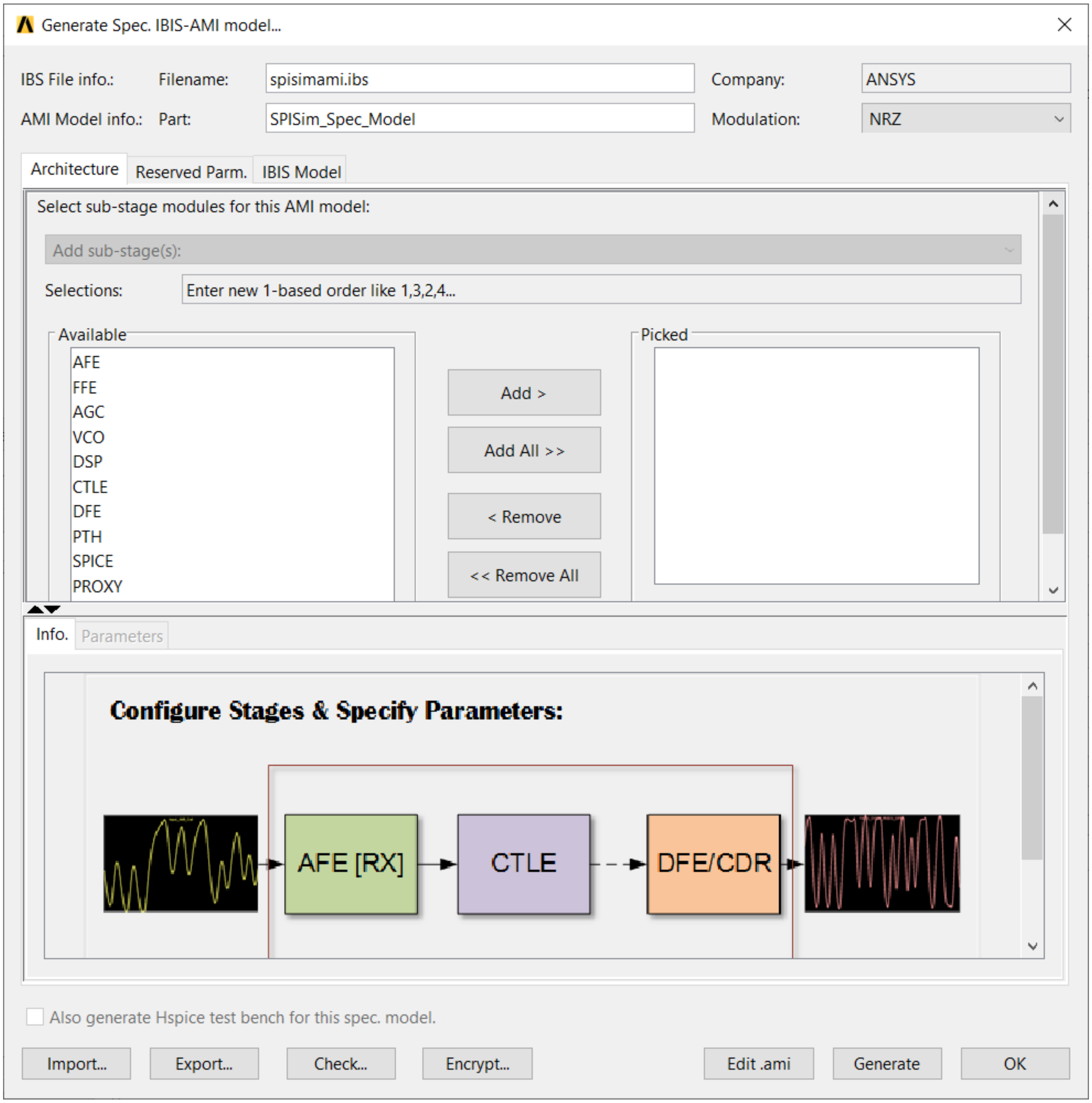
- Architecture
- Reserved Parm.

- IBIS Model



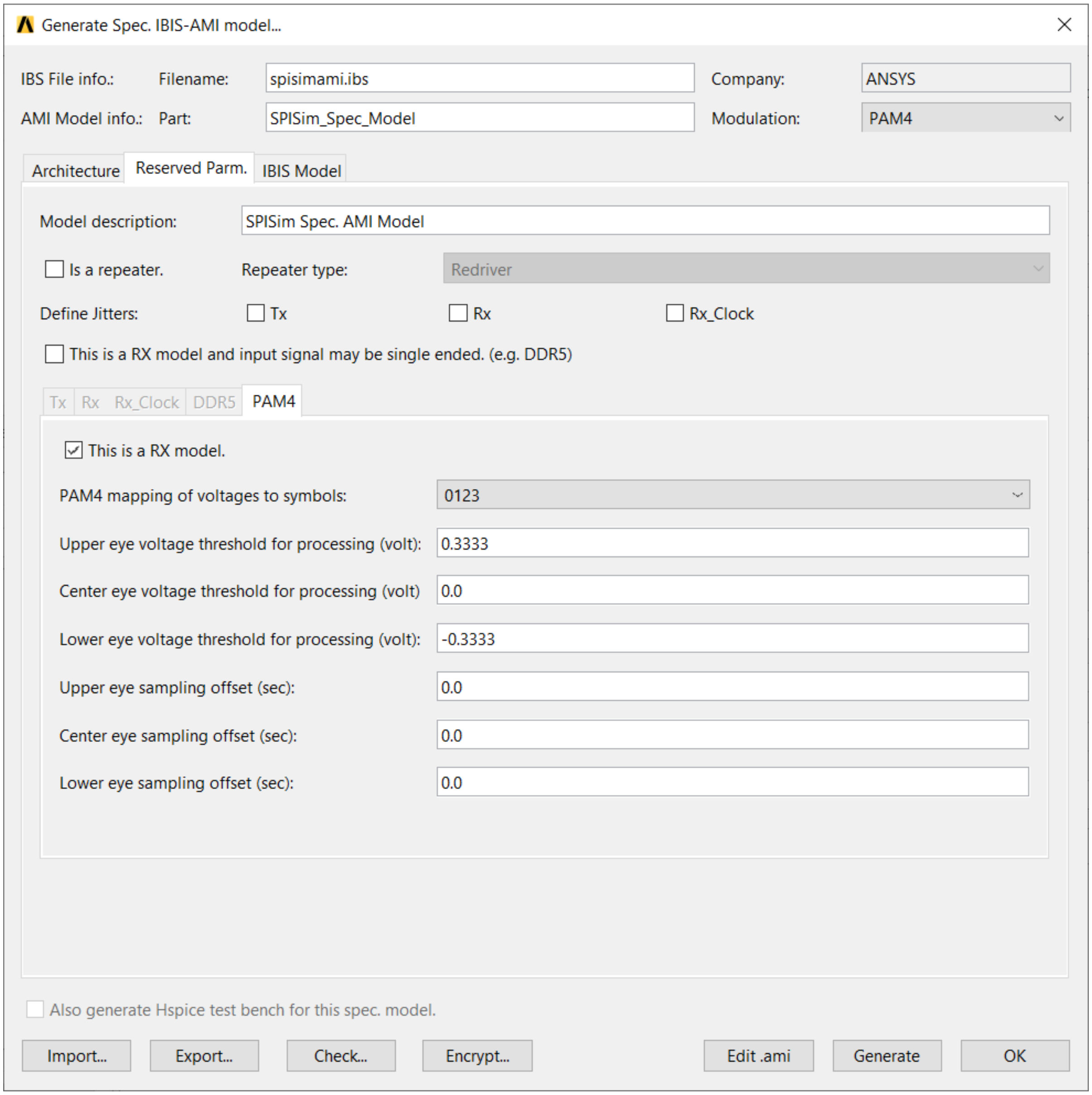
Generate Spec. IBIS-AMI model: Architecture Tab

From the **Architecture** tab, choose model generation sub-stages by adding, reordering, and/or removing sub-stages, and customizing the parameters of each sub-stage, as before.



Generate Spec. IBIS-AMI model: Reserved Parm. Tab

From the *Reserved Parm.* tab, customize the available parameters (e.g., depending on user-selection, **Modulation**, **Repeater type**, and *TX*, *Rx*, *Rx_Clock*, *DDR5*, and *PAM4* model tabs).



PAM4 Modulation and Analyses

Run a PAM4 model analysis for SerDes (Serializer/Deserializer) running 56Gbps or faster, by choosing **PAM4** on the **Modulation** drop-down menu. Select the *PAM4* tab to configure parameters. By default, only the **PAM4 mapping of voltages to symbols** field is editable. To customize RX-model parameters, check the box next to **This is a RX model and input signal may be single ended. (e.g., DDR5).**

> **Note:** Most AMI models (e.g., FFE, CTLE and PTH) are modulation agnostic. Selecting **NRZ** or **PAM4** on the **Modulation** drop-down menu will **not** change model behavior. DFE and CDR models are modulation dependent (e.g., if the AMI model is DFE and **PAM4** is the selected **Modulation**, the model's settings will automatically change to **MOD_TYPE_PAM4**.)

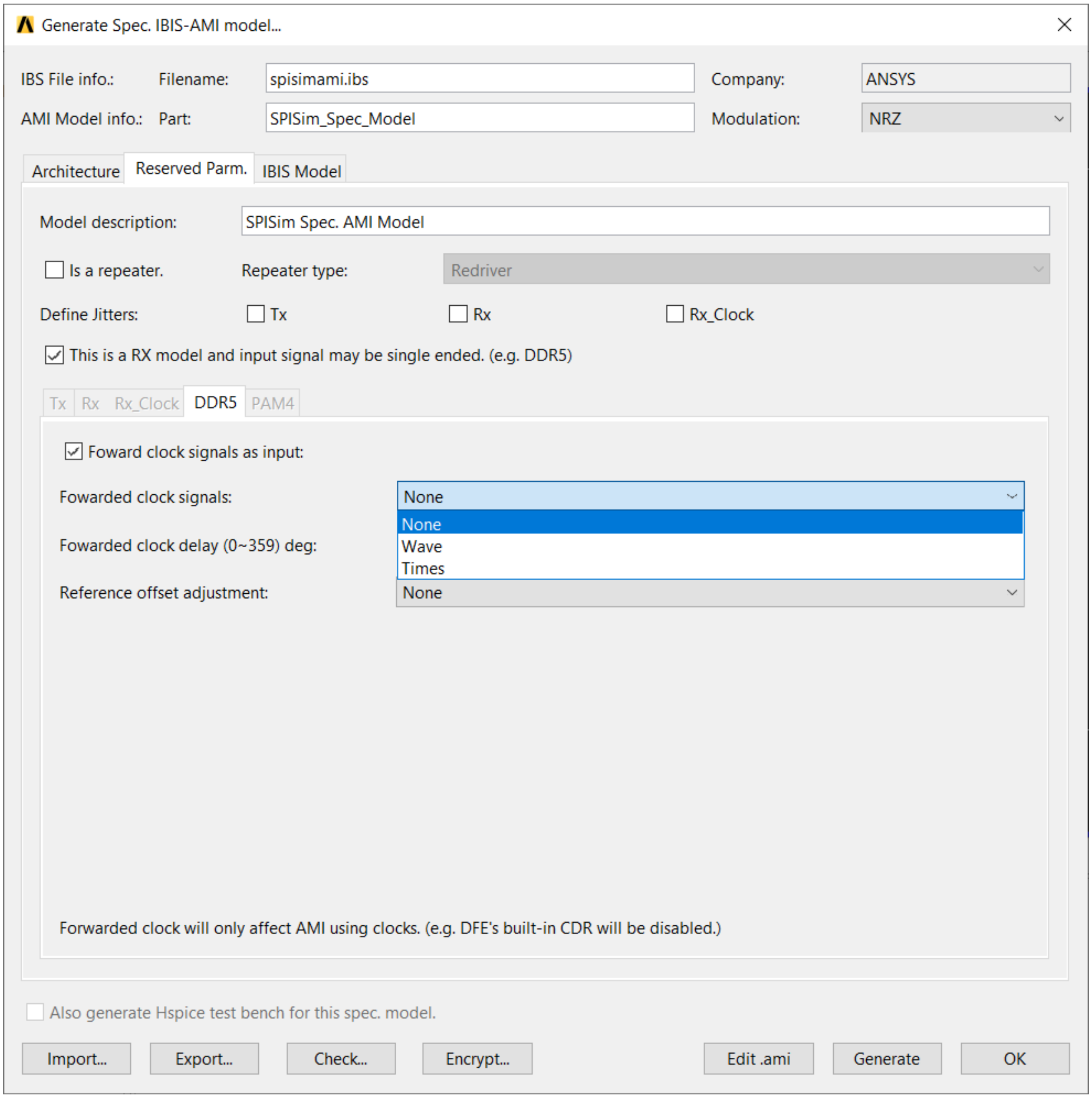
DDR5 Analyses

Run a **DDR5** model analysis for SerDes (Serializer/Deserializer) type single-ended signals (e.g., DDR5's DQ signals). The IBIS v6.1 parser is used by default. To enable , set **Modulation** to **NRZ** and check the box next to **This is a RX model and input signal may be single ended. (e.g., DDR5).** Then click the **DDR5** tab.

> **Note:** The **DDR5** tab is selectable if **Modulation** is set to **PAM4**, but will result in an error. At this time, **DDR5** modeling can only be used in conjunction with **NRZ Modulation**.

Rx Clock Forwarding

To generate a Rx model with clock forwarding:



1. Enable **Forward clock signals as input**.
2. Select **Forwarded clock signals** as either **None**, **Wave** or **Times**.
3. If needed, specify **Forwarded clock delays** with an integer value between 0 - 359.

The settings will enable the following corresponding changes in the generated .ami file:

- IBIS version will automatically be marked as “7.1”
- Rx_Use_Clock_Input keyword with specified value will appear under **Reserved parameters** session.
- MDL_FCLK_OFS keyword for specified offset value will appear under **Model_Specific** section.

Setting Descriptions:

Rx_Use_Clock_Input:

- None model will be used as if there is no clock forwarding.
- Wave model will use forwarded waveform for reference clock.
- Time model will synthesize clock signal based on forwarded time ticks.

MDL_FCLK_OFS: model will shift the forwarded waveform or time-tick by the specified degree before using it as a sample signal. The full phase is equal to 360 degrees.

Load and Test Drive an Rx Model with Clock Forwarding

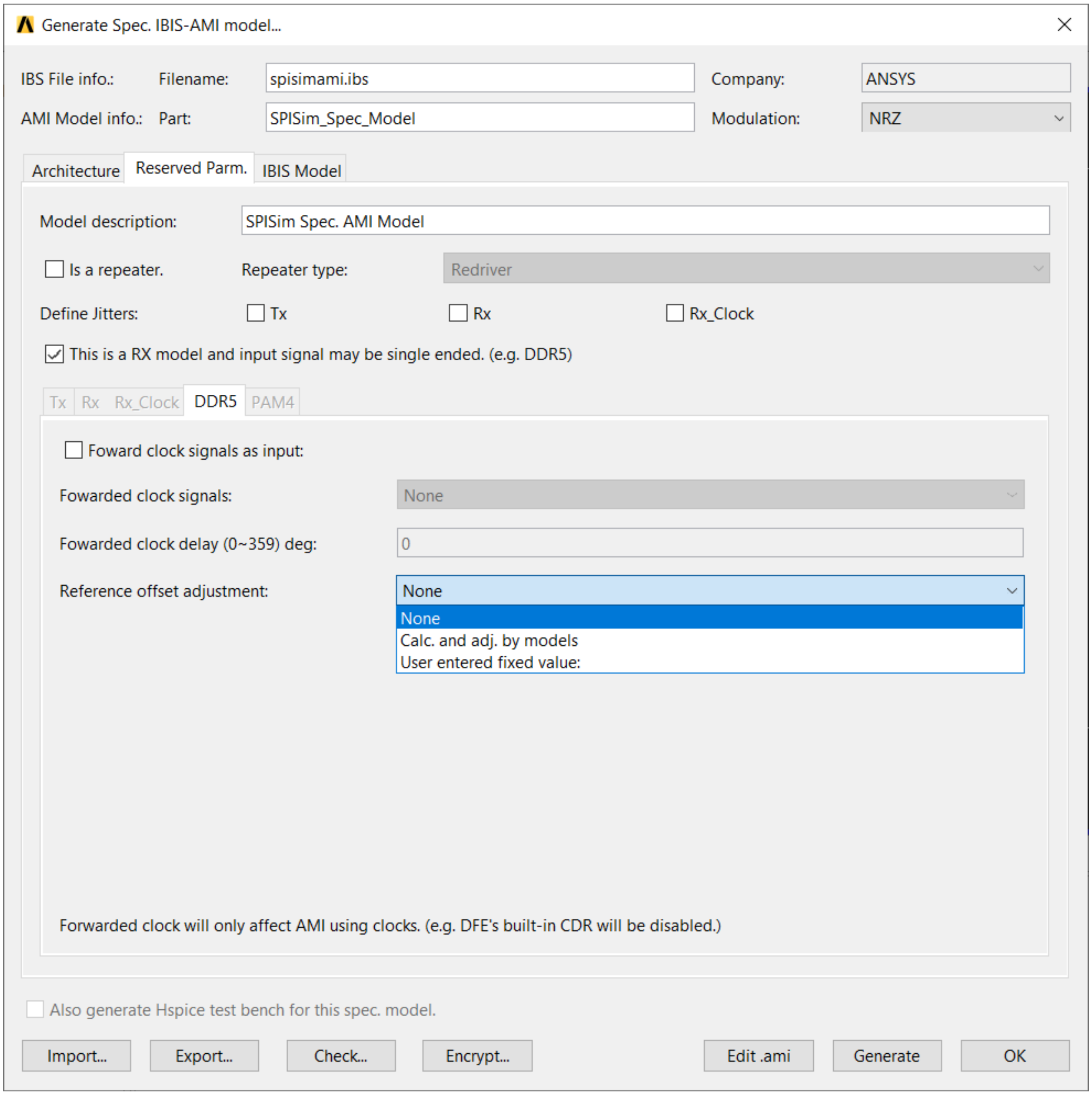
To use the Command Line Interface Tool **SPISimAMI.exe** model driver, which comes as part of AEDT/SPISim, define the following parameters:

CLK_INPT: pre-defined string signifying start of clock data.

clk_file: Clock data as an input to AMI_Getwave's clock_time.

Existing tests using SPISimAMI.exe will not be affected by newly added CLK_INPT clk_file optional parameters. When the user provides a clock file for input to drive the given models, it will be sent into the model to mimic simulator’s corresponding forwarding features.

DC_Offset Keyword (NRZ Modulation Only)

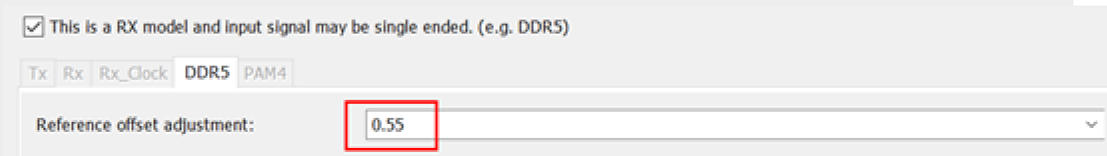
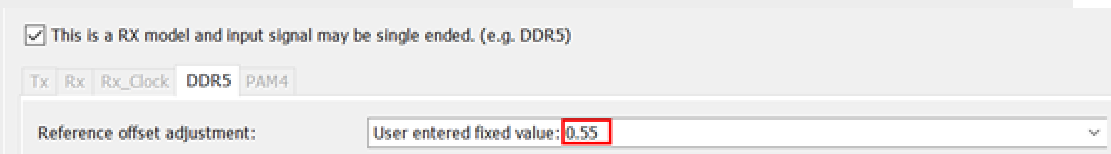


From the **DDR5** tab, select one of the following options on the **Reference offset adjustment** (i.e., **DC_OFFSET**) drop-down menu.

- **None**: for IBIS v7.1-compliant channel simulators (e.g., Nexxim). The simulator will adjust single-ended signals to differential before updating the model(s) (e.g., model(s) do not need to perform any processing, as all signals received will be differential).
- **Calc. and adj. by models**: for simulators that do not adjust the single-ended signals automatically. **Calc. and adj. by models** instructs the model(s) to calculate offset, then

apply the offset to the signal before processing begins. After processing and before data is returned to the simulator, the offset is subtracted on the output, so the returned data will be single-ended.

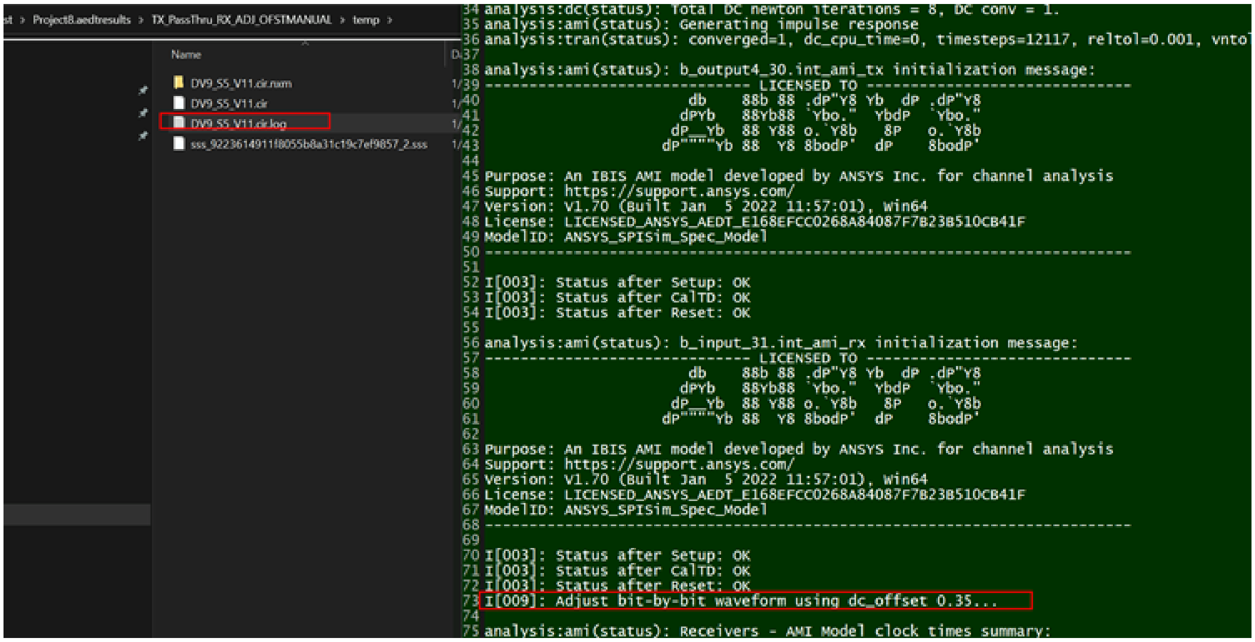
- **User entered fixed value**: enter values directly into the field to instruct models to perform an offset adjustment before and after processing.

> **Note:** Values can be entered manually in two ways:
>
> 1. Enter a value directly in the **Reference offset adjustment** field, clearing out any other text or selected choice present, if necessary (e.g., **0.55**).
>
> 
>
>
> 2. Select **User entered fixed value** on the **Reference offset adjustment** drop-down menu and enter the offset value directly after the existing text in the field (e.g., **User entered fixed value: 0.55**).
>
> 
>

DC_OFFSET Adjustment at Command Line

While all three **Reference offset adjustment** options automatically adjust the appropriate value before and after signal processing, users have additional options at the command line.

After the simulation is complete, open the log file and ensure the `DC_OFFSET` value was applied.



Change the following variables, as appropriate:

- `SPISIMAMI_NOADJ_DCOFSTINP=T`: cancel the `DC_OFFSET` adjustment before processing begins.
- `SPISIMAMI_NOADJ_DCOFSTOUT=T`: cancel the `DC_OFFSET` adjustment after processing ends.

If the `DC_OFFSET` value reported in the log file is *d*V, and the model output without either variable set is V1 ~ V2, the same variables react as follows:

- `SPISIMAMI_NOADJ_DCOFSTINP=T`: output will be V1+*d*V ~ V2+*d*V (extra *d*V applied after processing)
- `SPISIMAMI_NOADJ_DCOFSTOUT=T`: output will be V1-*d*V ~ V2-*d*V (extra *d*V applied before processing, but not restored after processing)

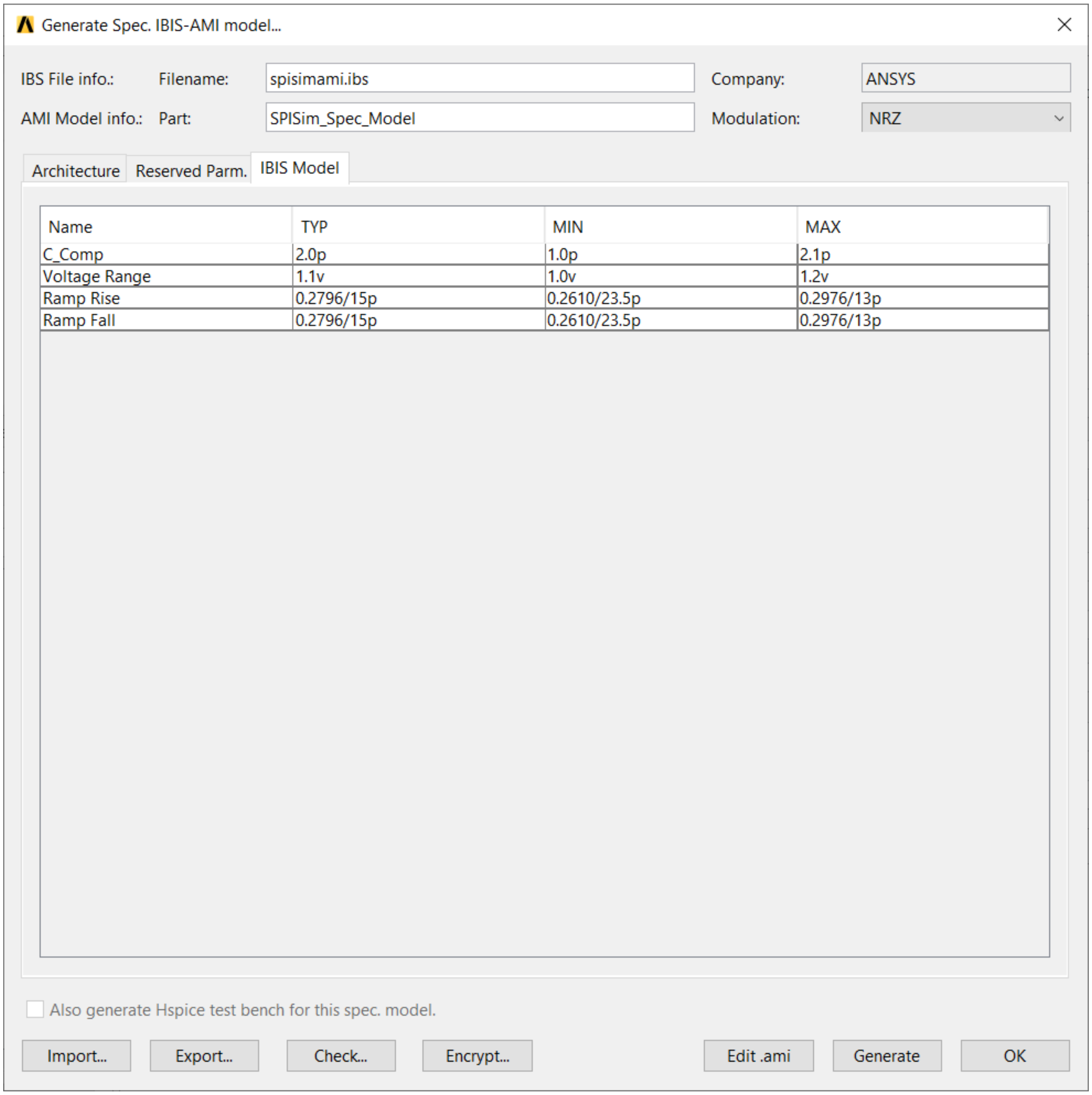
Set both variables **=T** and it will appear as if no `DC_OFFSET` adjustment has been applied, even if the log reports adjusted values.

> **Note:** If environmental variables are changed, the user needs to relaunch **Electronic Desktop** to see the new values applied.

Generate Spec. IBIS-AMI model: IBIS Model Tab

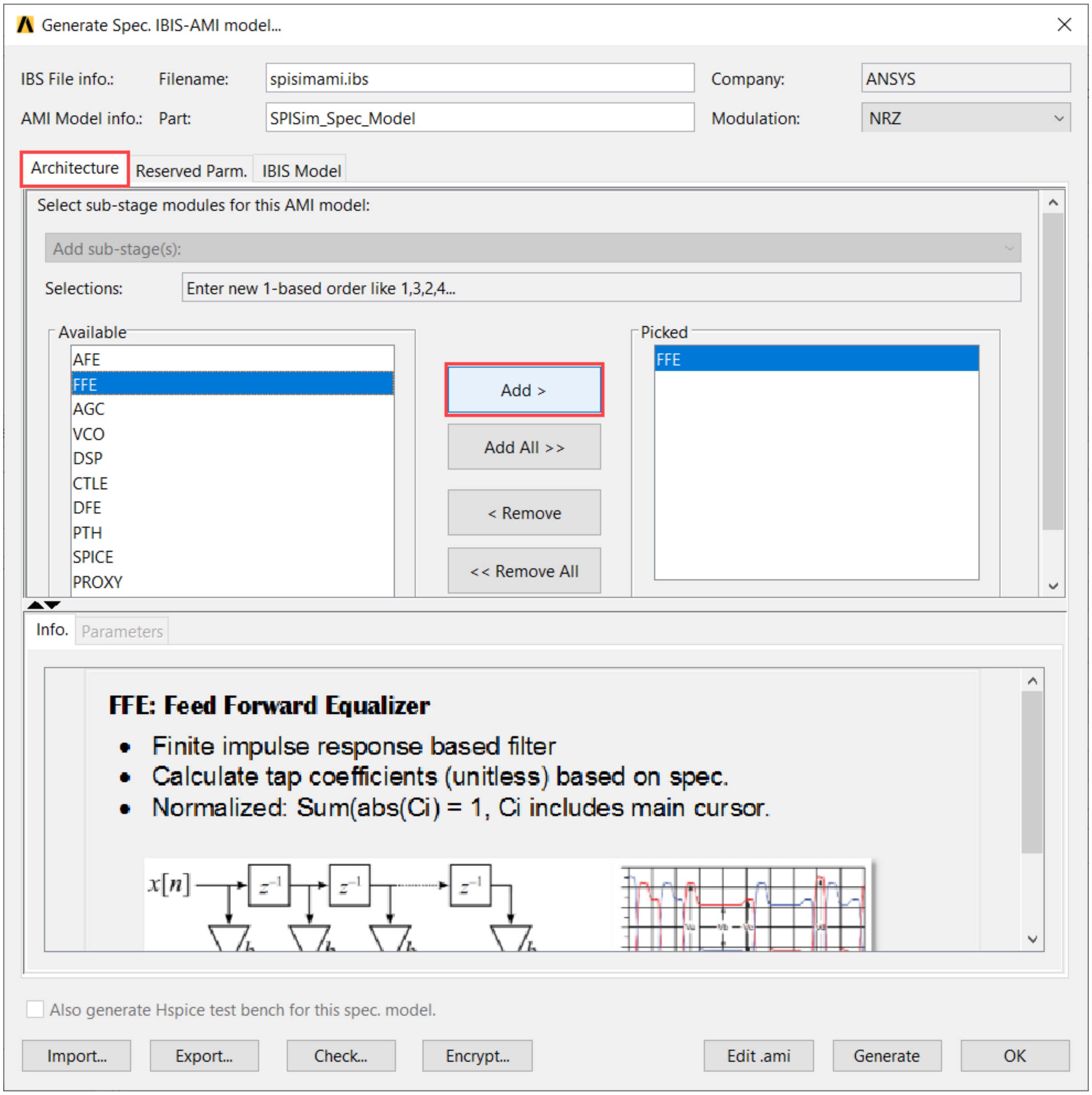
The IBIS Model tab allows configuration of major analog properties such as C_Comp, voltage range, rise, and fall time for the associated IBIS model.

The generated .ibs will use the specified properties.

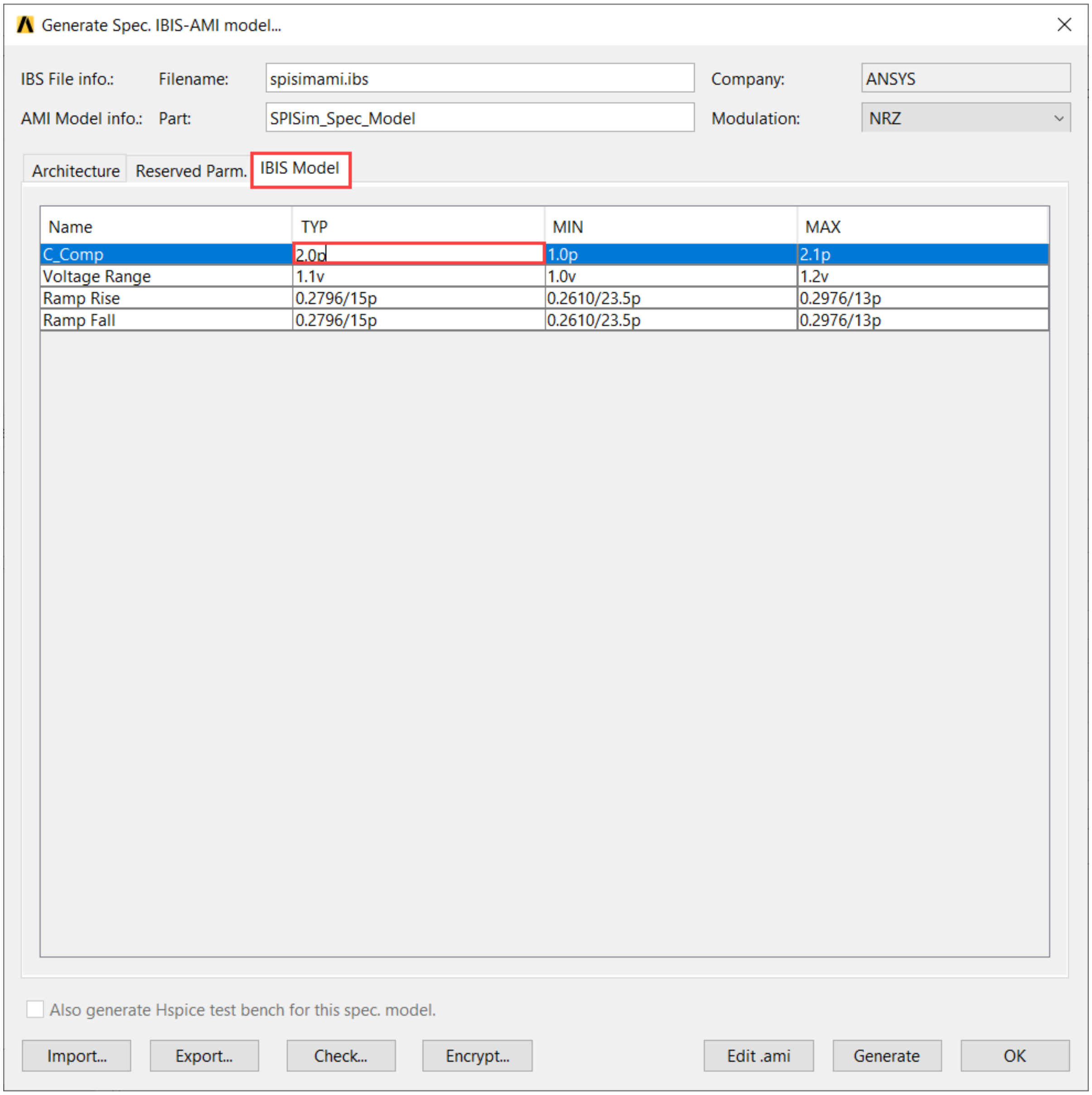


To use:

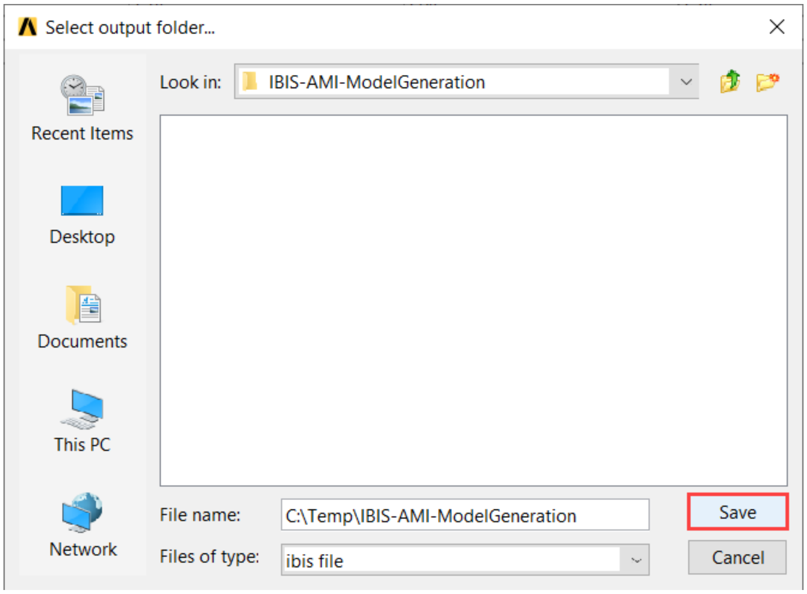
1. Select at least one sub-stage from the **Architecture Tab**.



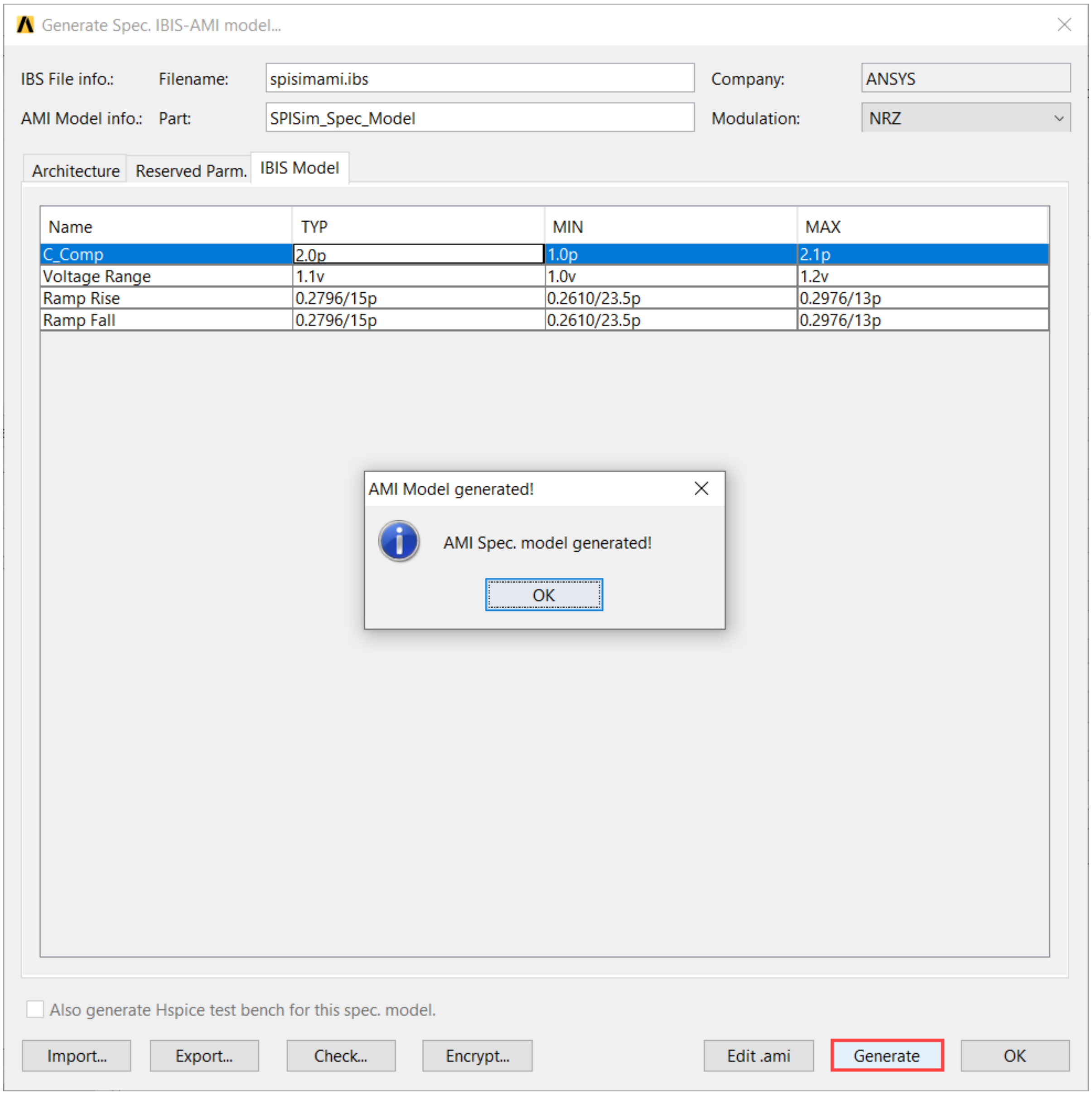
2. Select the **IBIS Model Tab** and enter **TYP**, **MIN**, and **MAX** configuration values for **C_ Comp**, **Voltage Range**, **Ramp Rise**, and **Ramp Fall**.



3. Click **Generate** and select an output folder. Click **Save**.



4. The AMI Spec. model will be generated and located in the selected output folder.



Sample IBIS-AMI Models

Example industry standard models are provided, they can be generated from SPISim or from AEDT directly.

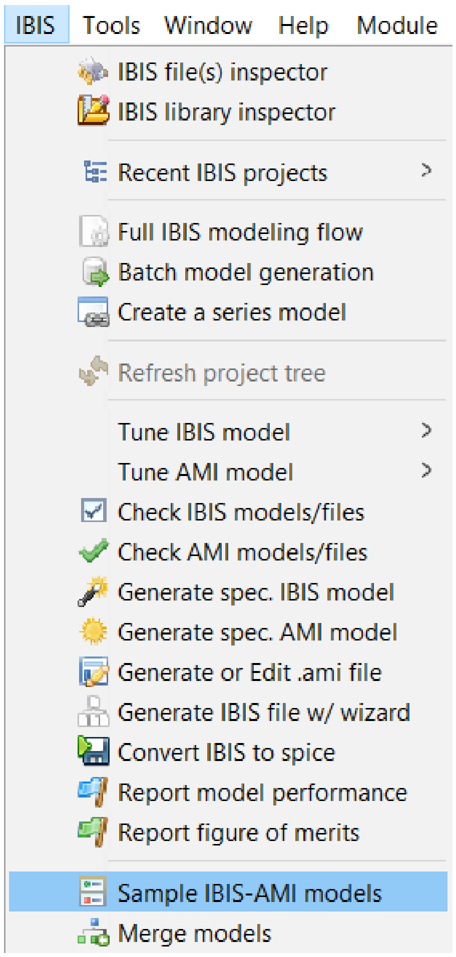
- DDR5_Tx
- PCIeG2_Tx
- PCIeG4_Rx

- PCIeG4_Tx
- PCIeG5_16GT
- PCIeG5_32GT
- PCIeG5_8GT
- PCIeG5_Tx **NEW in 24.1 Release**

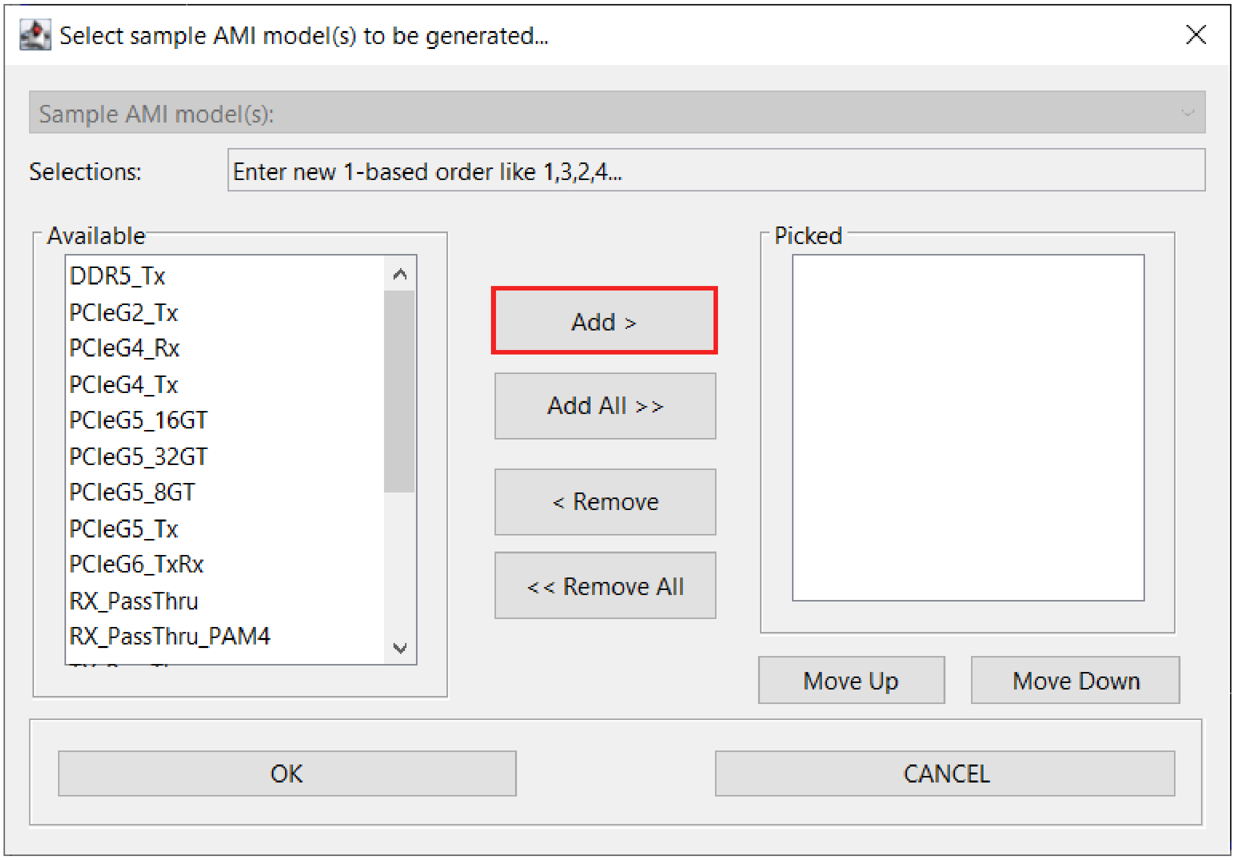
  PCIeG6_TxRx **NEW in 24.1 Release**
- RX_PassThru
- RX_PassThru_PAM4
- TX_PassThru
- TX_PassThru_PAM4
- USB3P2G1_LongCh
- USB3P2G1_ShortCh
- USB3P2G2_Rx
- USB3P2_Tx
- USB4_Rx
- USB4_Tx

To generate a sample model in SPISim:

1. Select **Sample IBIS-AMI Models** from the **IBIS** menu.



2. Select the model(s) by double-clicking the model name or by using the **Add >** button.

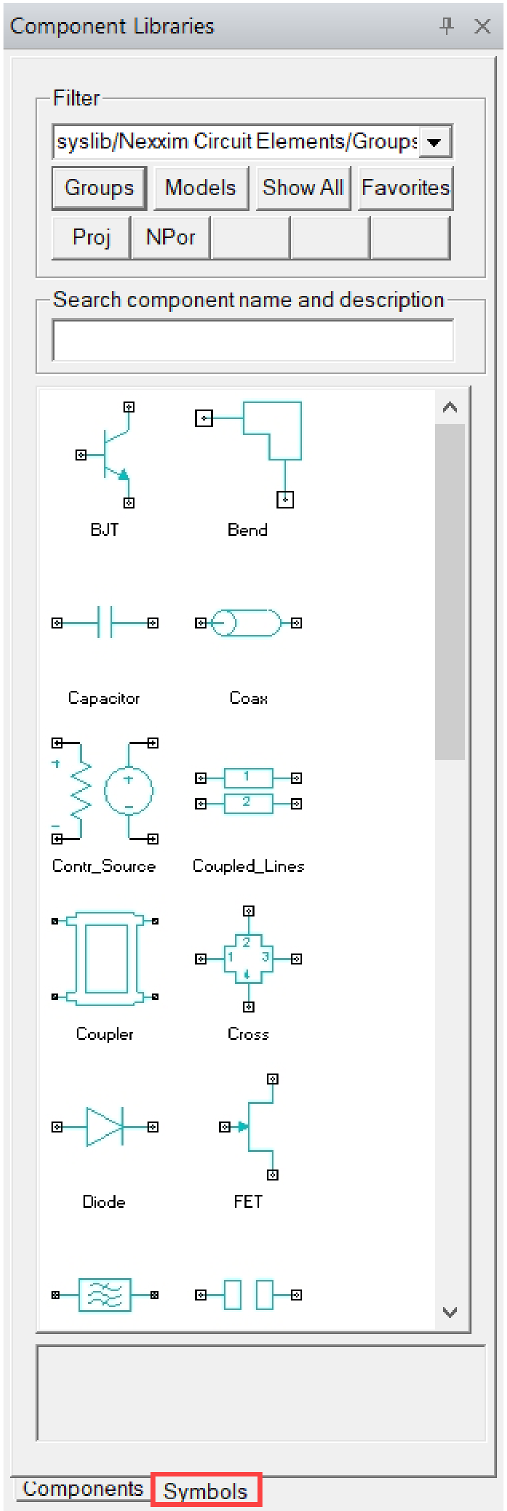


3. Click **OK**.

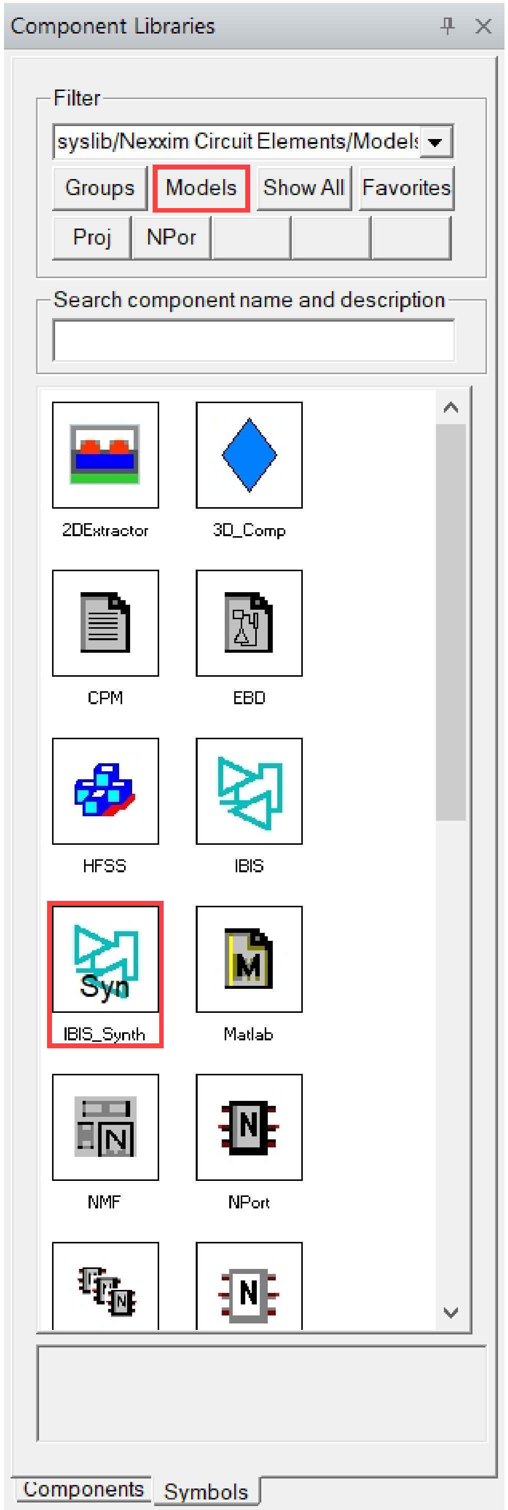
The model(s) will be generated in the working directory.

To generate the models directly from AEDT's schematic editor for channel simulation:

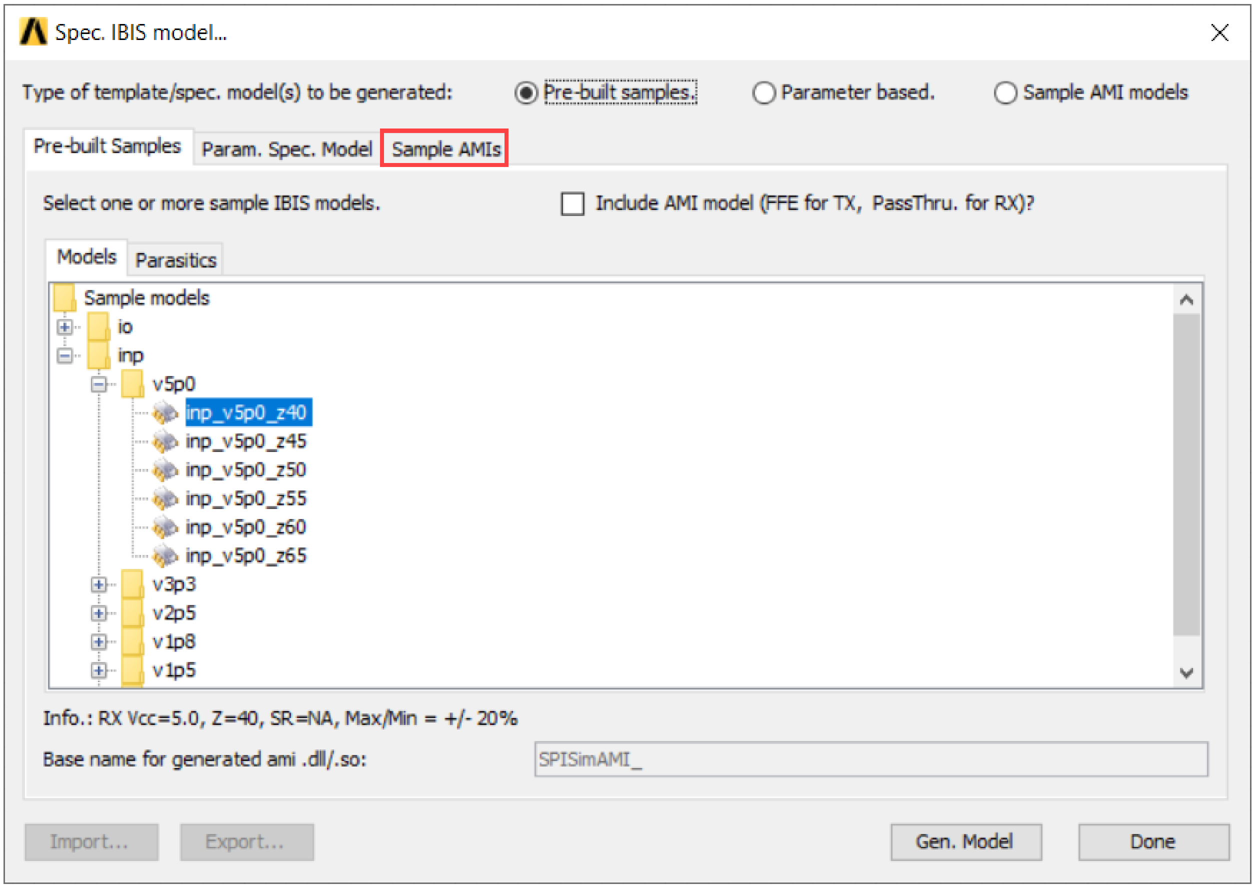
1. Select **Symbols** from the **Component Libraries** window.



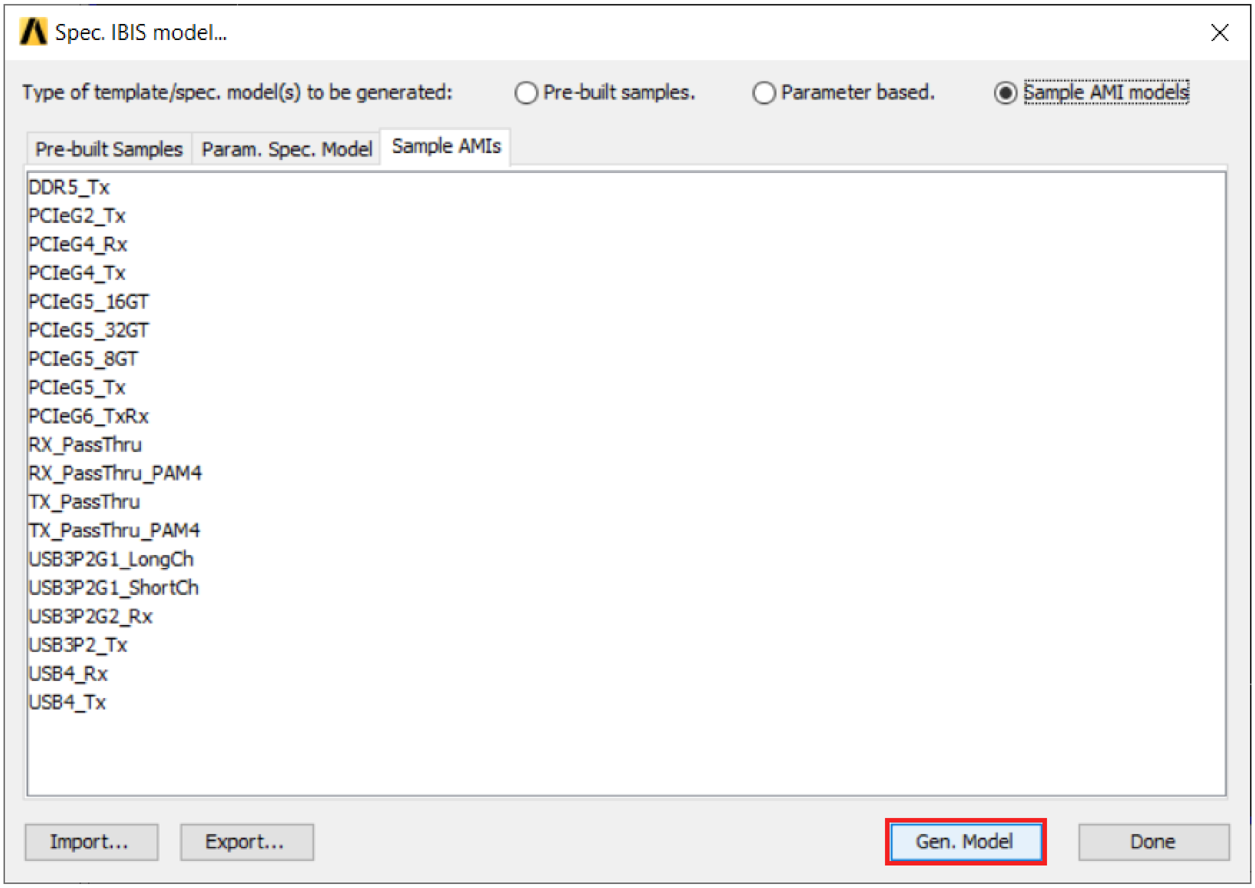
2. Select **Models** then **IBIS_Synth** to launch the **Spec. IBIS Model** window.



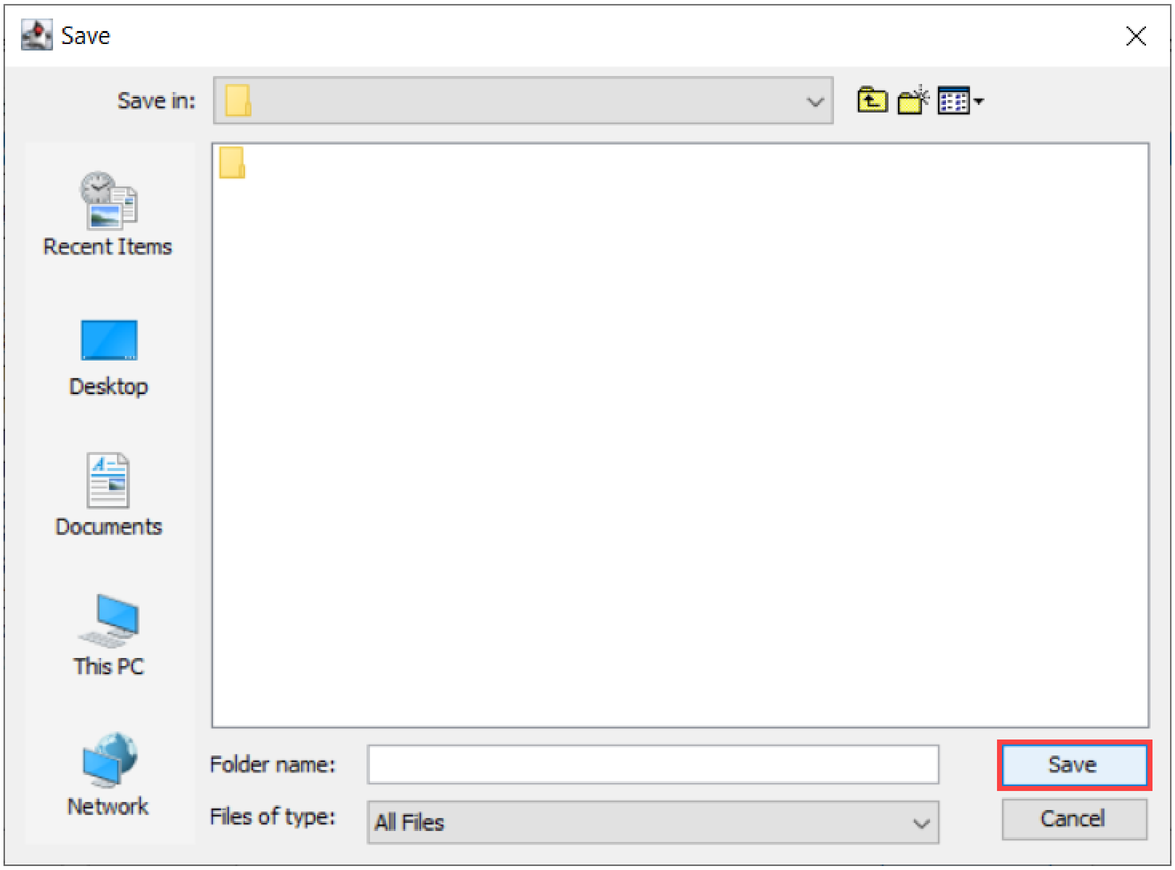
3. Select **Sample AMIs**

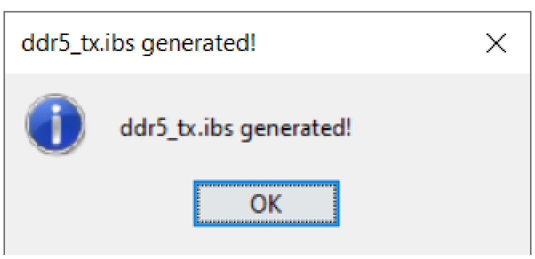


4. Select the model(s) and click Generate Model



5. Select an output folder and click **Save**.

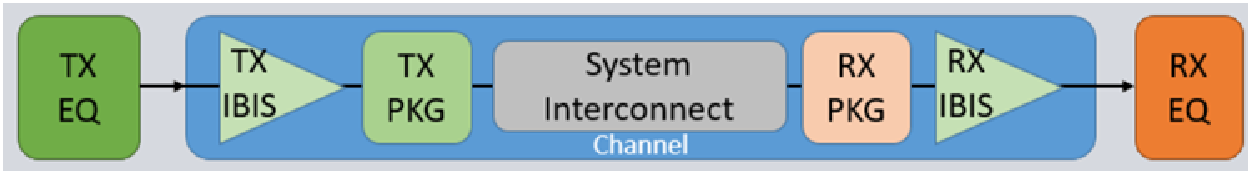




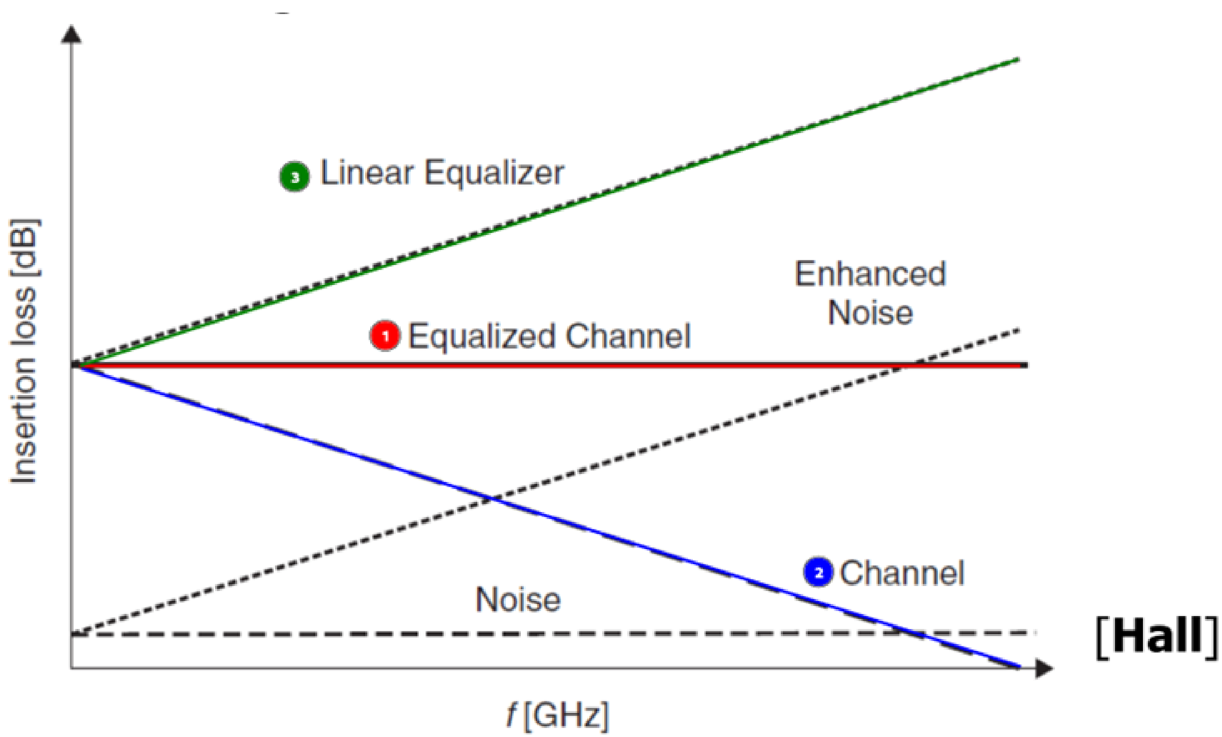
The model will be generated in the specified directory.

Adaptive CTLE IBIS-AMI Model

Background

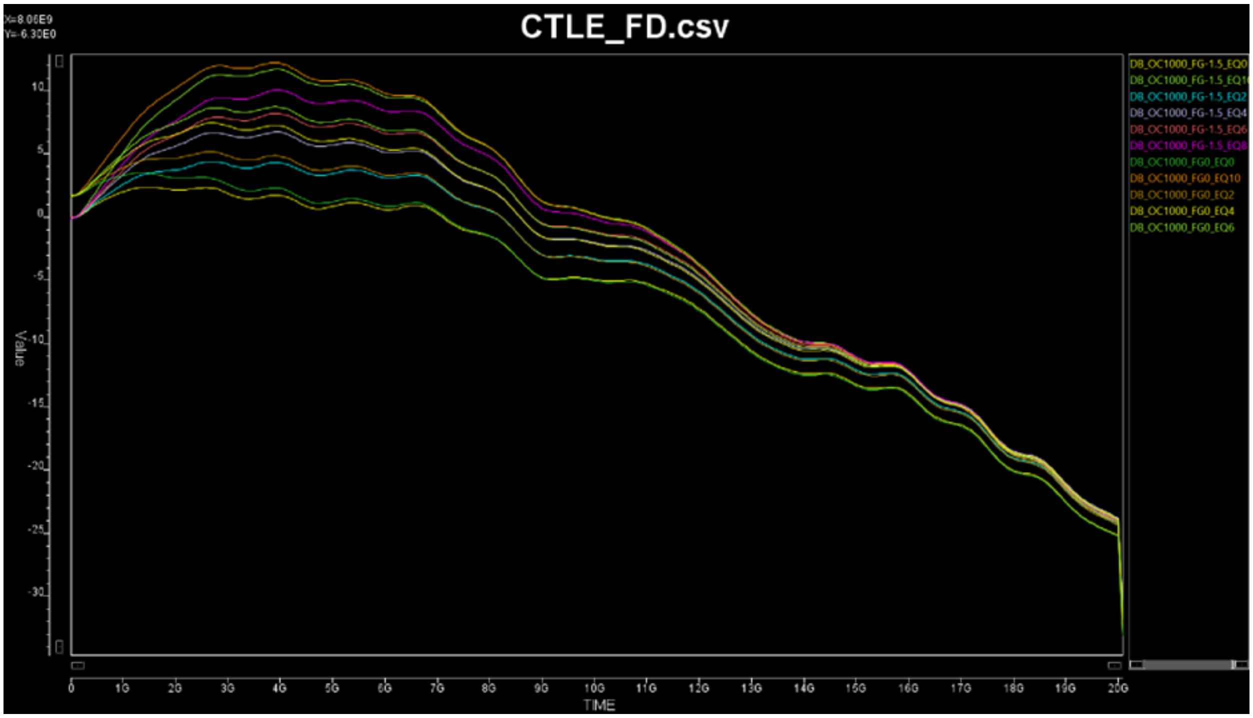


An ideal equalized channel for signal path from input of TXEQ to output of RXEQ is an all pass filter, represented by line 1, in the graph below. In an all pass filter, signals of all frequencies have the same gain, equal to 1, i.e. 0dB, so no signal distortions exist. When signals of different frequencies travel at different speeds and losses down a channel, loss or dispersion will occur resulting in distorted signal bits. This distortion is a common behavior of a channel, represented by the components in the blue box of the flow diagram above, and marked as line 2 in the graph below. To accommodate for this, linear equalizers TXEQ and RXEQ are introduced, with behaviors represented by line 3, below. The goal is that adding behaviors 2 and 3 will results in an overall channel performance similar to line 1, an equalized channel.



REFERENCE: Palermo, Sam. RX Equalization Noise Enhancement. 2023. Digital, ECEN720: High-Speed Links Circuits and Systems Spring 2023. https://people.engr.tamu.edu/spalermo/ecen689/lecture8_ee720_rx_adaptive_eq.pdf

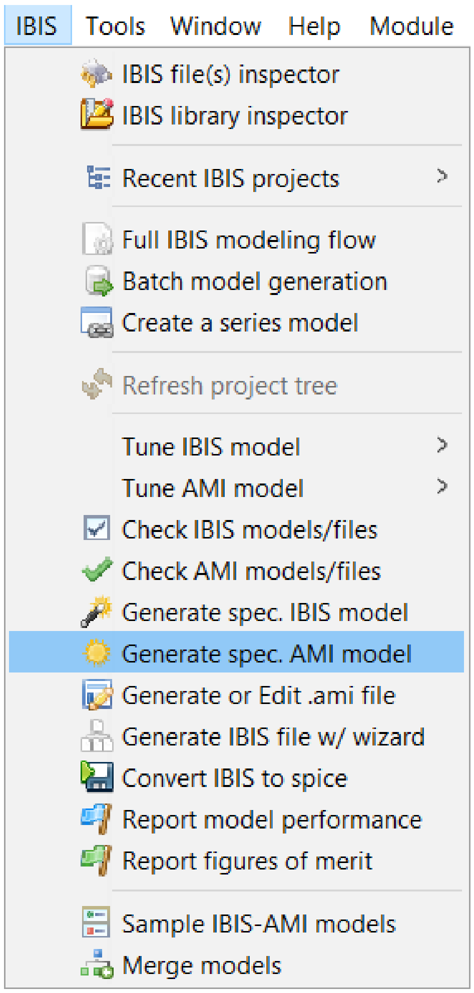
The linear behavior of line 3, above, is an ideal case. In reality, the behavior of CTLE (RXEQ) is better represented by one of the curves below. It's common that a CTLE has many curves to chose from and the user must find a matching one for a given channel so that the overall performance is acceptable.



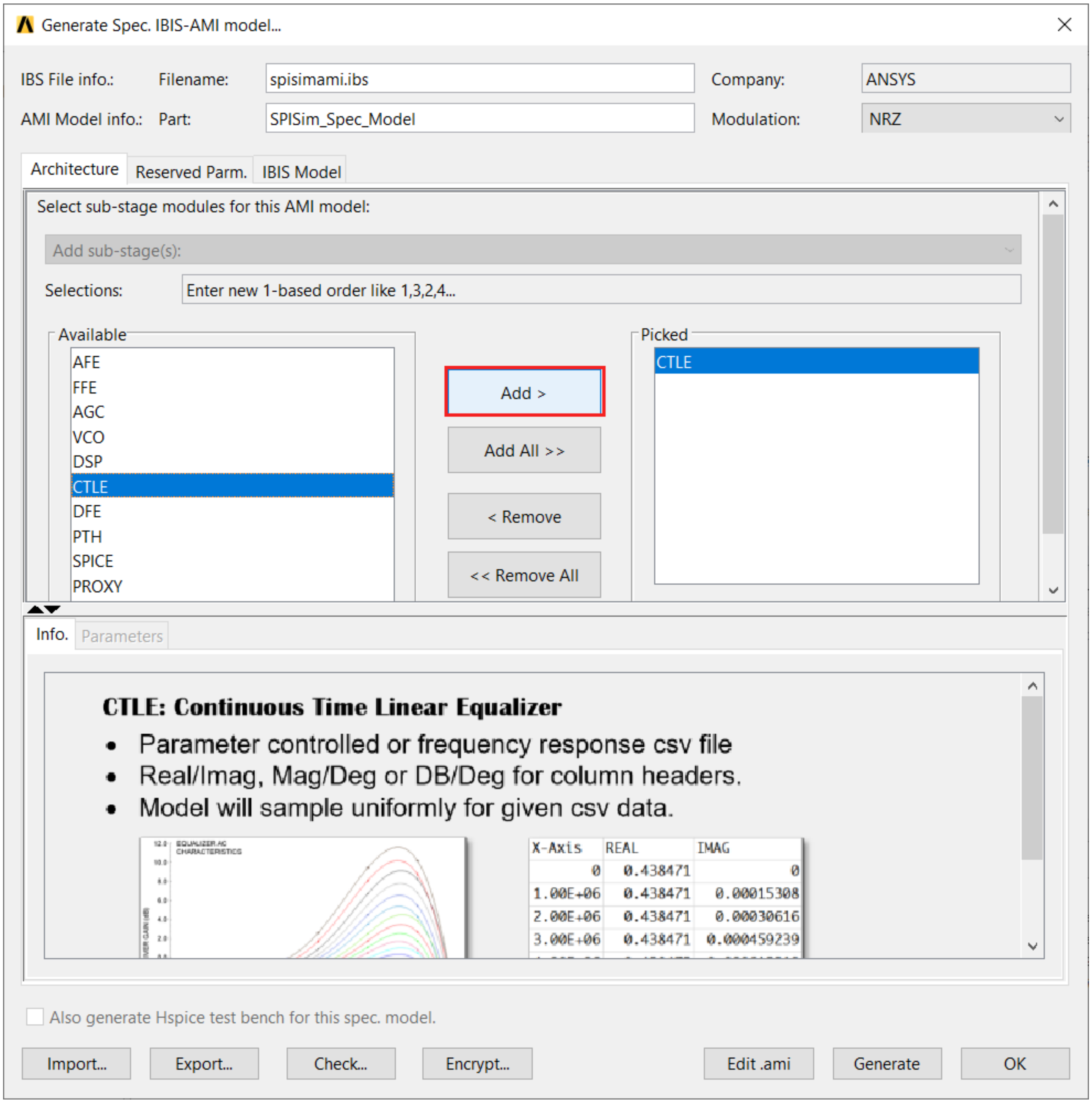
The adaptive CTLE IBIS-AMI feature provides a “self-adapting” capability in the CTLE model to find a good curve among those available to produce good analysis results

Steps

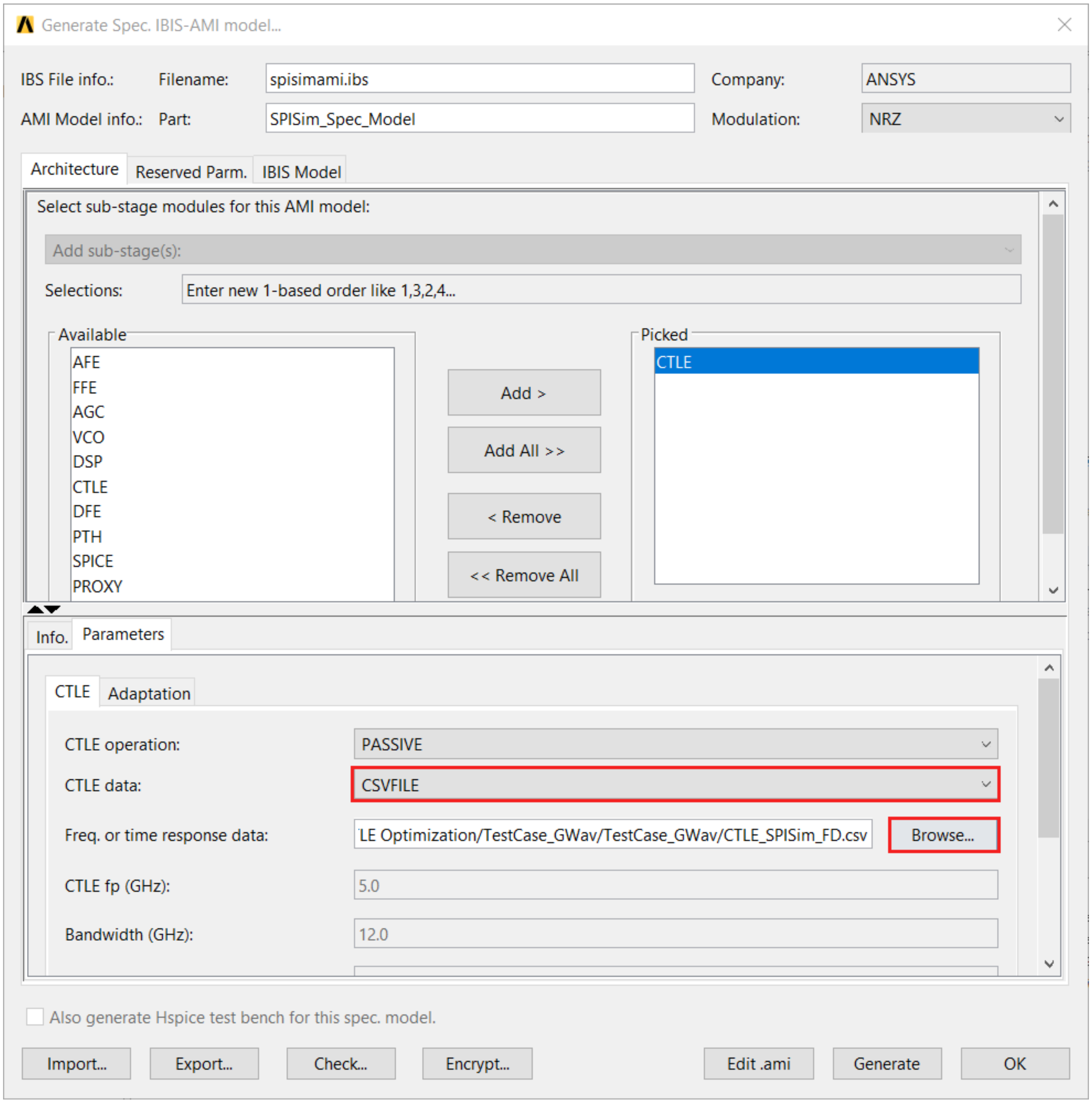
1. Launch SPISim, choose IBIS from the top ribbon and select **Generate spec. AMI Model**.



2. Select **CTLE** from the **Available** group box and click **Add**.



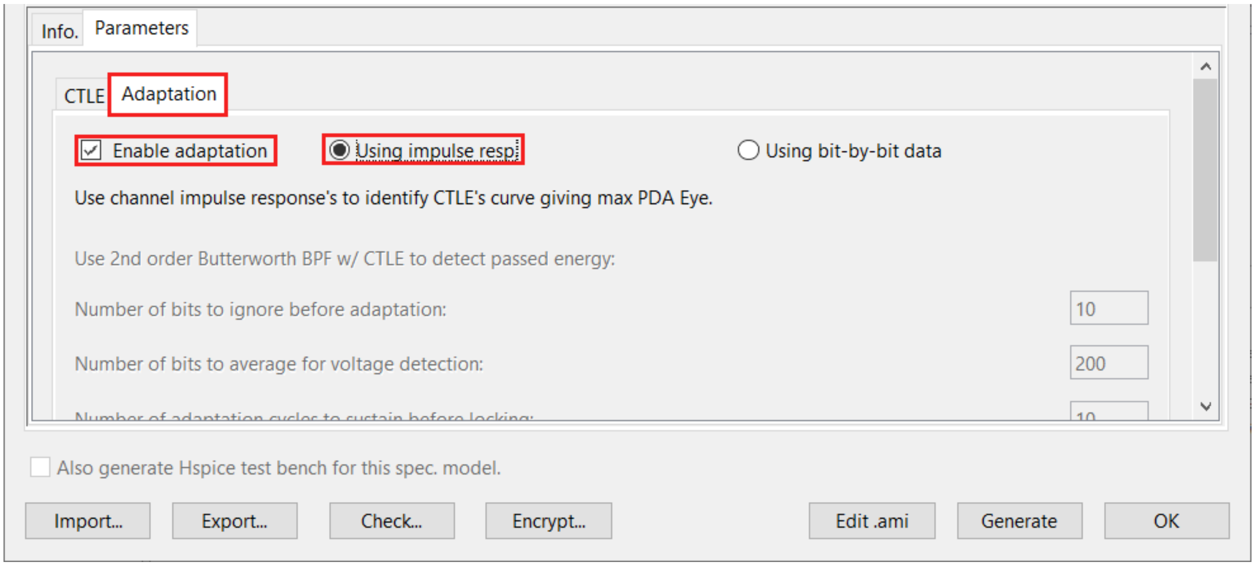
3. From the **CTLE data** drop down menu, select CSVFILE and use the **Browse** button to choose your CSV file.



The **Adaptation** tab will be enabled.

4. In the **Adaptation tab**, check Enable Adaptation to choose between **Using impulse response** or **Using bit-by-bit data**.

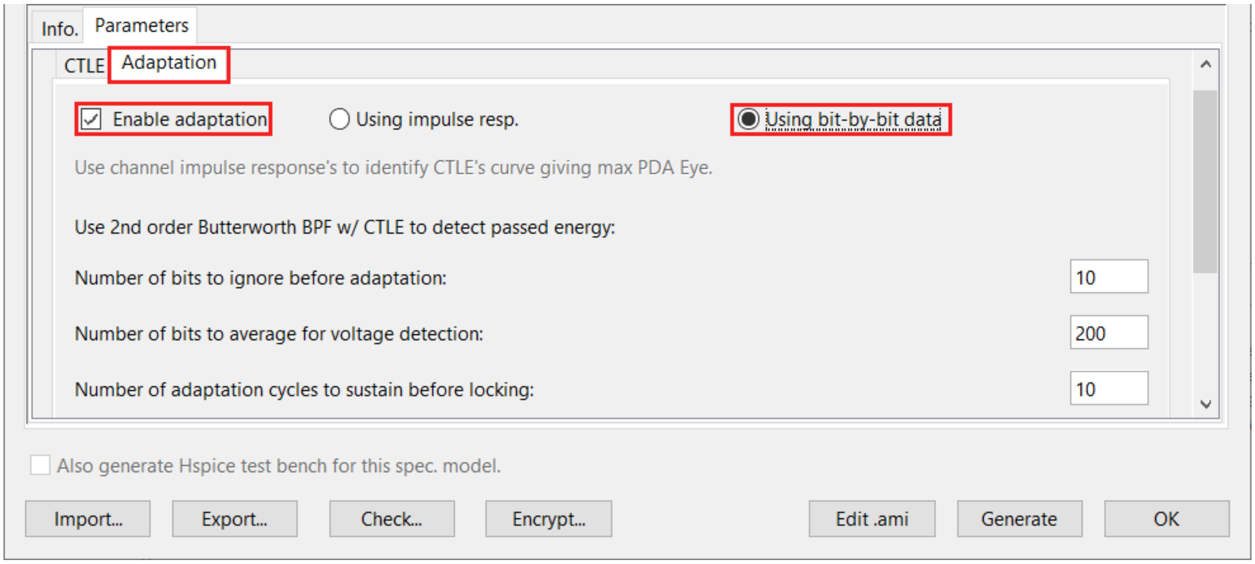
   a. **Using Impulse Response**: During AmiInit (e.g. Nexxim's VerifEye analysis and AMI analysis), impulse response is input to the model. PDA based approach will be used to calculate eye height. CTLE curve resulting maximum PDA eye-height will be selected for subsequent analysis. All curves will be “scanned” in this approach.



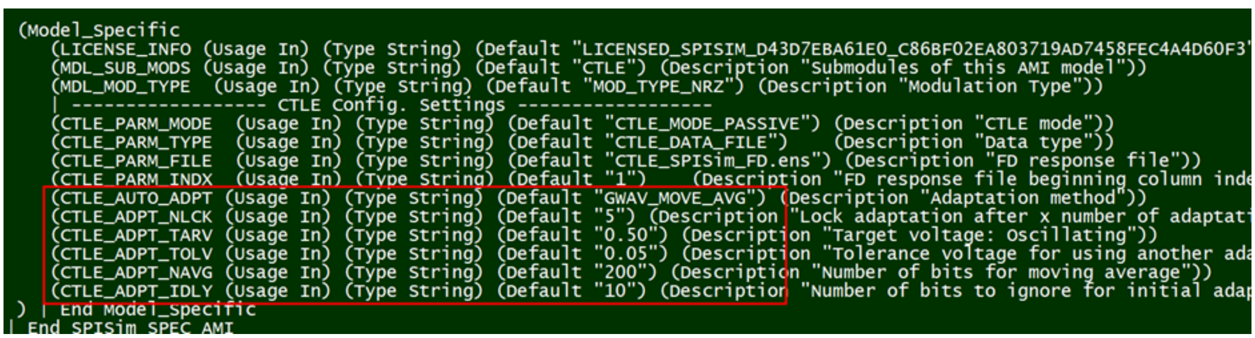
The generated model's .ami file will contain the new corresponding AMI parameter:

```
) | End Reserved_Parameters

|***************************************************************************
| Remove or tamper LICENSE_INFO values will cause simulation being aborted!
|***************************************************************************
(Model_Specific
   (LICENSE_INFO (Usage In) (Type String) (Default "LICENSED_ANSYS_AEDT_E168EFCC0268A84087F7B23B510CB41F")
        | ------------------ MAIN Settings ------------------
   (MDL_SUB_MODS (Usage In) (Type String) (Default "CTLE") (Description "Cascaded stages"))
   | ------------------ CTLE Settings ------------------
   (CTLE_PARM_TYPE (Usage In) (Type String) (Default "CTLE_DATA_FILE") (Description "Data type"))
   (CTLE_PARM_FILE (Usage In) (Type String) (Default "C:/Temp/00_TestPlan/20230831_F822807_Optimization/Te
   (CTLE_PARM_INDX (Usage In) (Type String) (Default "1") (Description "FD response file beginning id"))
   (CTLE_AUTO_ADPT (Usage In) (Type String) (Default "INIT_PDA_EYEH") (Description "Adaptation method"))
```

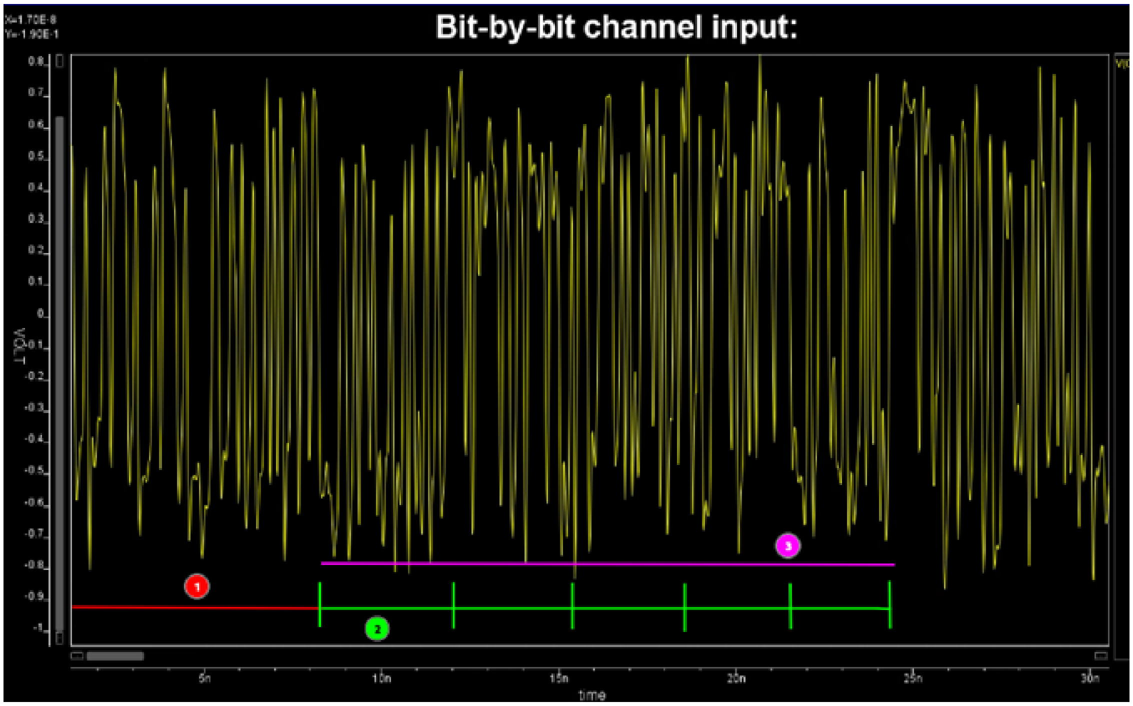
b. **Using bit-by-bit data**: During getWave (e.g. Nexxim’s AMI analysis), bit-by-bit waveform input will be used for the model. Moving average after band pass filter (BPF) and various CTLE curves will be calculated to find best one giving passing energy around the targeted value.



The specified values will also be reflected in the new AMI parameters.



Using Bit-by-Bit Data Parameter Descriptions



Let the yellow curve above be the bit-by-bit waveform simulator provided for RX AMI models using **GetWave** simulation. Simulator will provide sampling interval **dt** and bit time **UI** to the model. Thus, the number of samples there are in one bit: nSampBit = UI / dt.

i. **Number of bits to ignored**: Marked as 1, red above, this value times nSampBit will be ignored from input data. Data provided for the model to adapt will start from the ones above green lines.

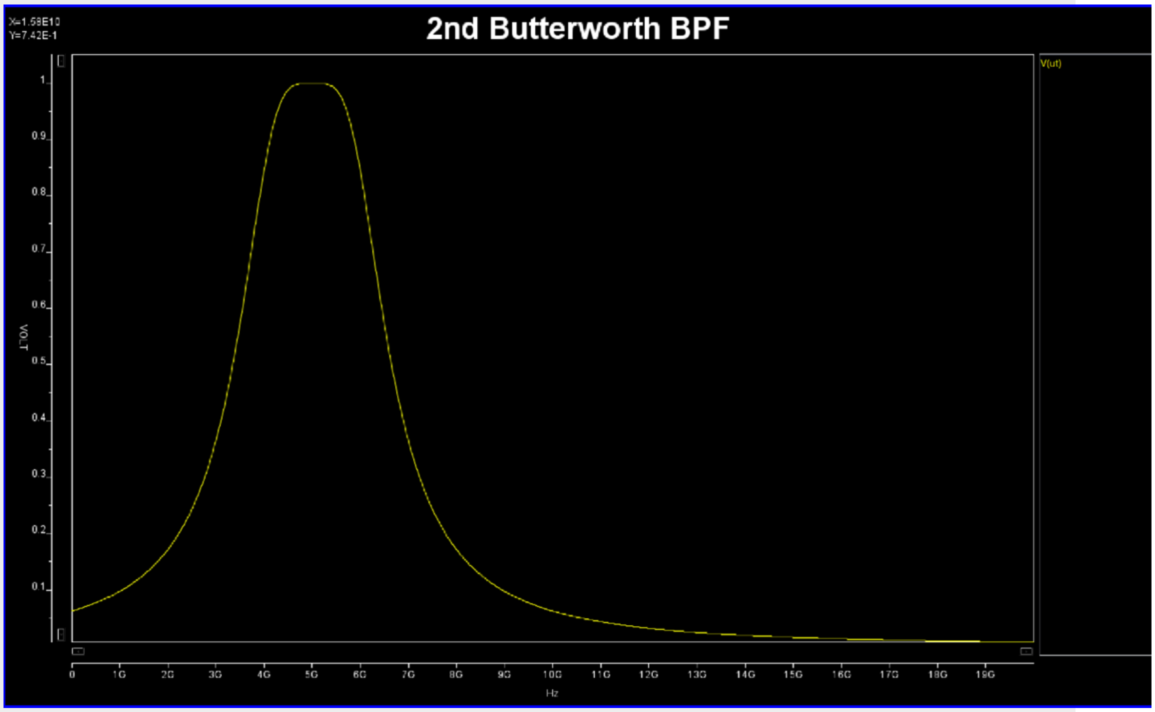
ii. **Number of bits to average**: each average take one chunk of data marked as 2, green, above. At the end of the "chunk", judgement will be made to see whether a better CTLE curves should be chosen before moving to next "chunk".

iii. **Number of sustaining cycles**: If the selected curve produces average energy within V+-dV, then this curve will be reused for the next "chunk" of data. It needs to stay within range for this number of cycles/chunks to be declared "locked" and selected, marked as 3, magenta, above

iv. **Target voltage**: V mentioned in (iii).

v. **Tolerance**: dV mentioned in (iii).

Note: In this BPF/Energy based adapting algorithm, not all curves will be scanned. This is the algorithm used:
1. Based on the UI, a second-order Butterworth BPF will be generated:



2. A single bit passing this BPF will be used as input data. Each curve will be convolved with this input to calculate output energy. The curves are sorted by final value (boost level) from low to high. As only a single bit is used here, this process is very quick.
3. During actual AMI analysis where many bits are provided, CTLE curve of lowest boost level is first used. If the resulting passed average energy is not within the V +- dV range, the selection will move up and down (using next chunk of data) until either adaptation is successful or failed.
4. For the adaptation process to proceed, the input data needs to be least this long: `nSampBit * (num_bit_ignored + num_bits_to_average * num_sustain_cycle)`. The three variables used in the parentheses are parameters defined in (i-iii) mentioned above.
5. All the processes are "non-blocking". That is, even if the adaptation failed, the model will still use the last best value to continue the AMI analysis. User can see model's output messages for adaptation details.

5. Select **Generate**. The model will be generated.

SPISim IBIS Support

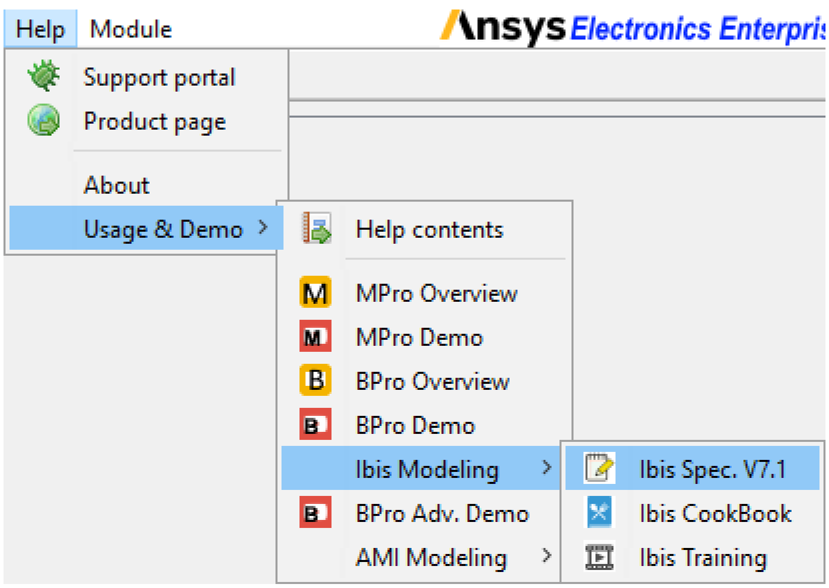
In this Section:

- IBIS Version 7.1
- IBIS Figure of Merit
- IBIS Non-perpetual AMI Expiration

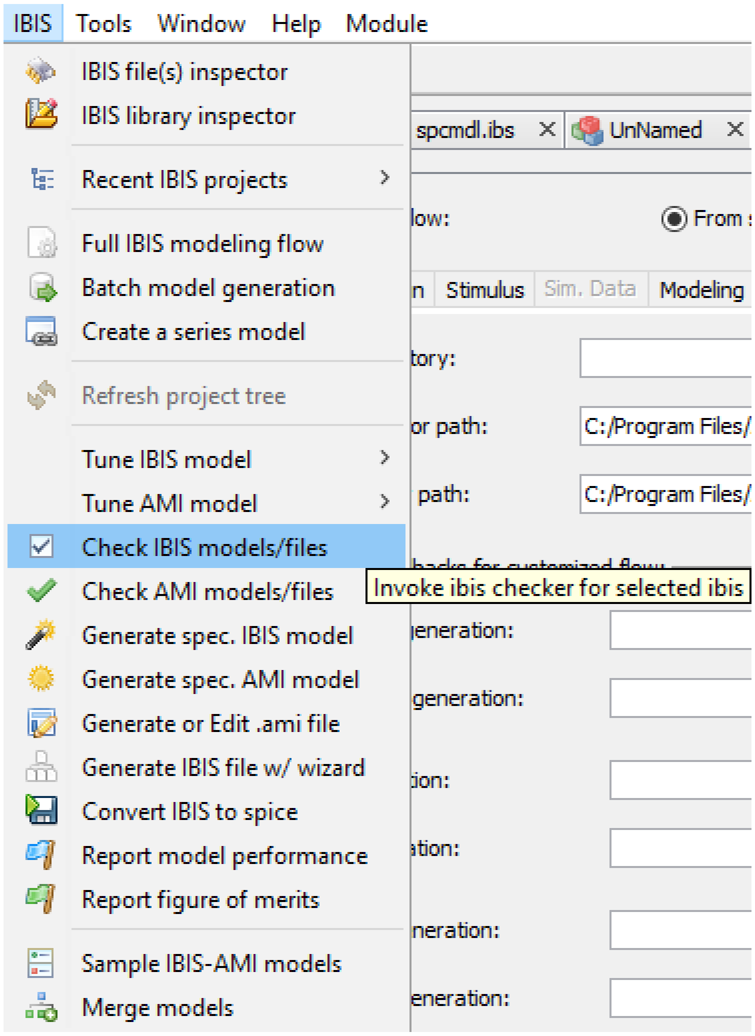
IBIS Version 7.1

The SPISim internal parser, golden parser, and embedded spec. have been upgraded to the most recent version of IBIS. Refer to the IBIS Version 7.1 specification document for details. To ensure SPISim is running the latest version of IBIS, do one of the following:

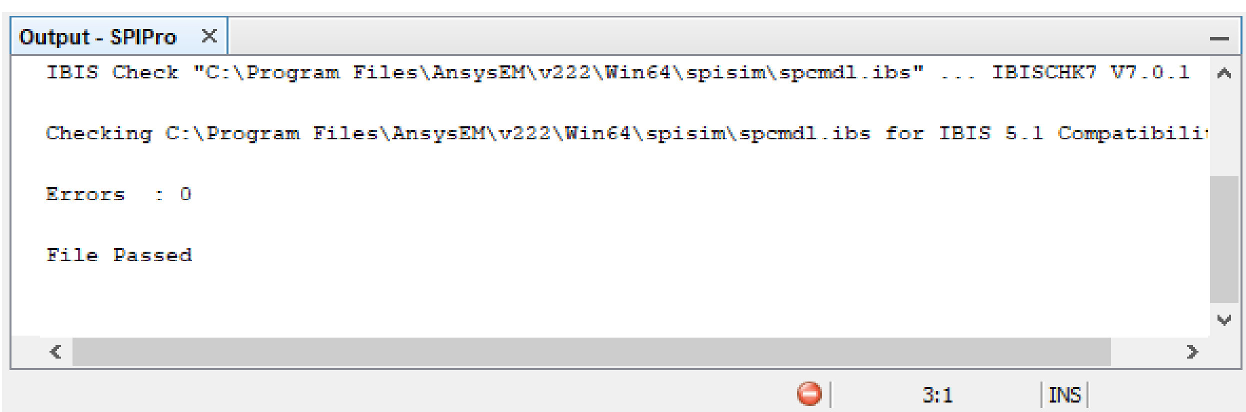
- Navigate to **Help** > **Ibis Modeling** and view the **Ibis Spec. V7.1** menu item.



- Test the software by making a relevant request (e.g., **Check IBIS models/files**).



Complete any actions necessary to generate a result and view the **Output - SPIPro** window. The **IBIS Check** line will identify the parser version used during the request.



IBIS Figure of Merits

The Figure of Merits (FOM) report is used to provide a quantitative measure for correlating between transistors design and its IBIS model's performance. Two FOM correlation methods are implemented based on sections 4.3 and 4.4 of the IBIS Accuracy Specification.

FOM1: Average of absolute value of relative errors, defined in IBIS handbook.

$$FOM1 = 100 * \left[1 - \frac{\sum_{i-1}^{N} | \; Y1_i - Y2_i |}{YW * N}\right]$$

FOM2: Maximum relative error.

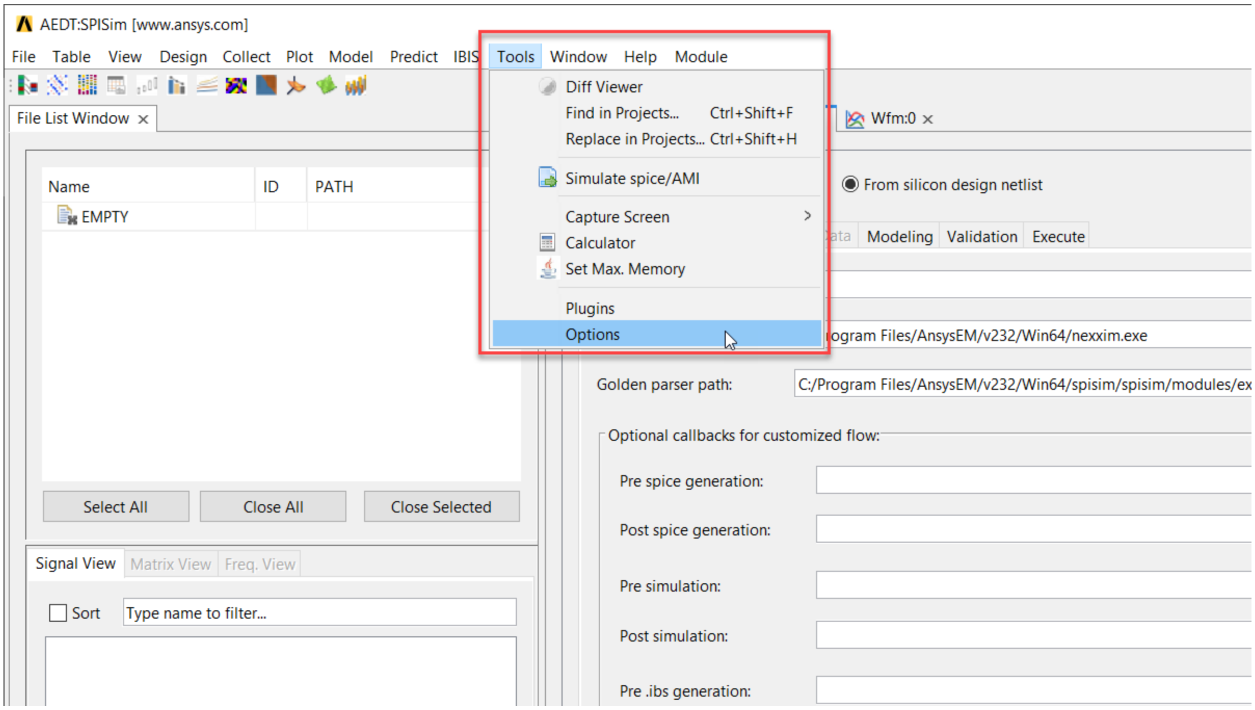
$$FOM2 = 100 * \left[\; \frac{max| \; Y1_i - Y2_i |}{YW}\right]$$

An IBIS model that correlates to the original transistor design will result in approximately equal simulation results from the transistor (Y1) and the IBIS model (Y2). In this case, FOM1 will be high, or approximately 100%, and FOM2 will be low, or approximately 0%.

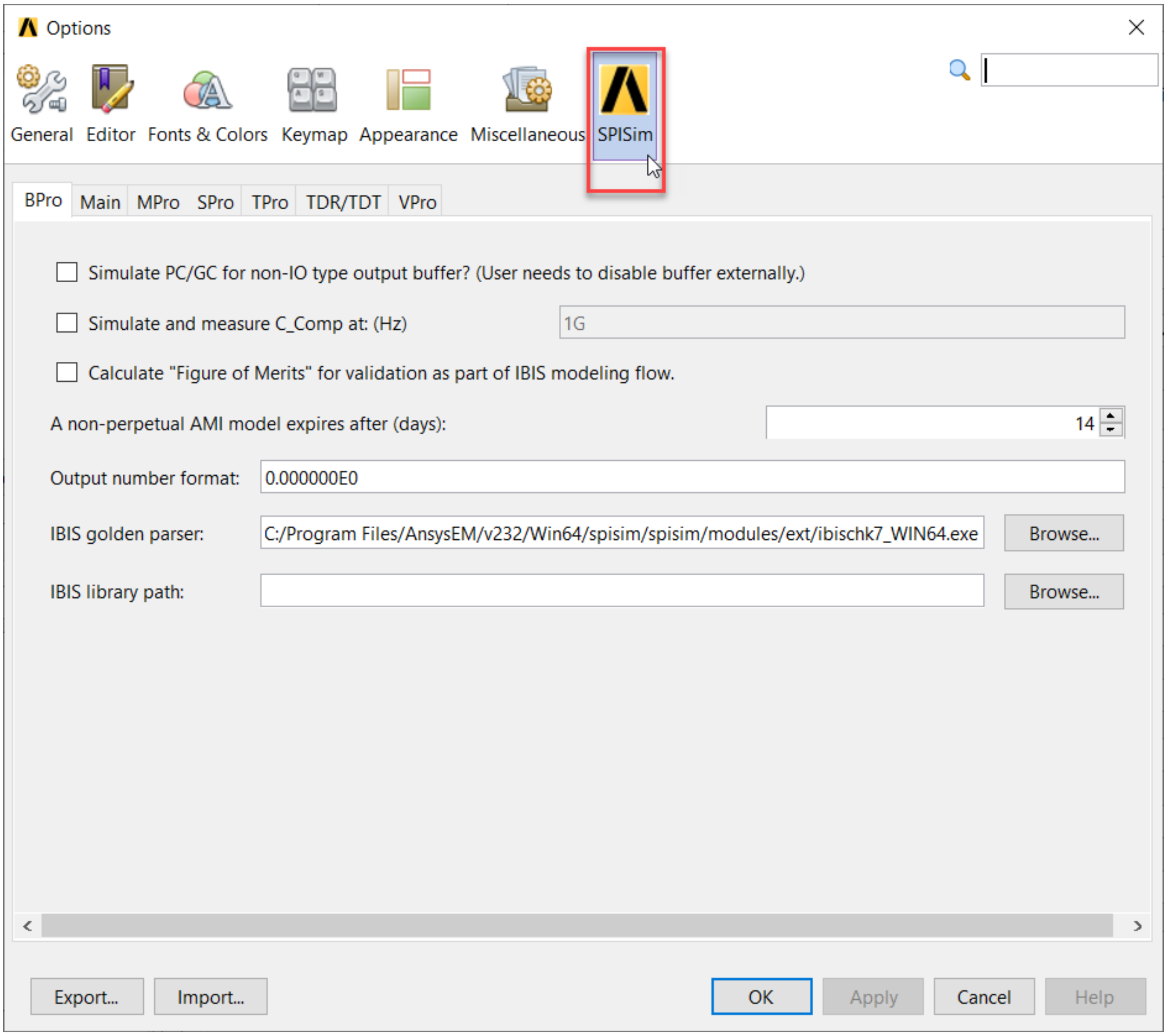
> **Note:** The Figure of Merits calculation is OFF by default as the added FOM calculation will increase the number of simulations required for modeling.

To configure the FOM option in SPISim:

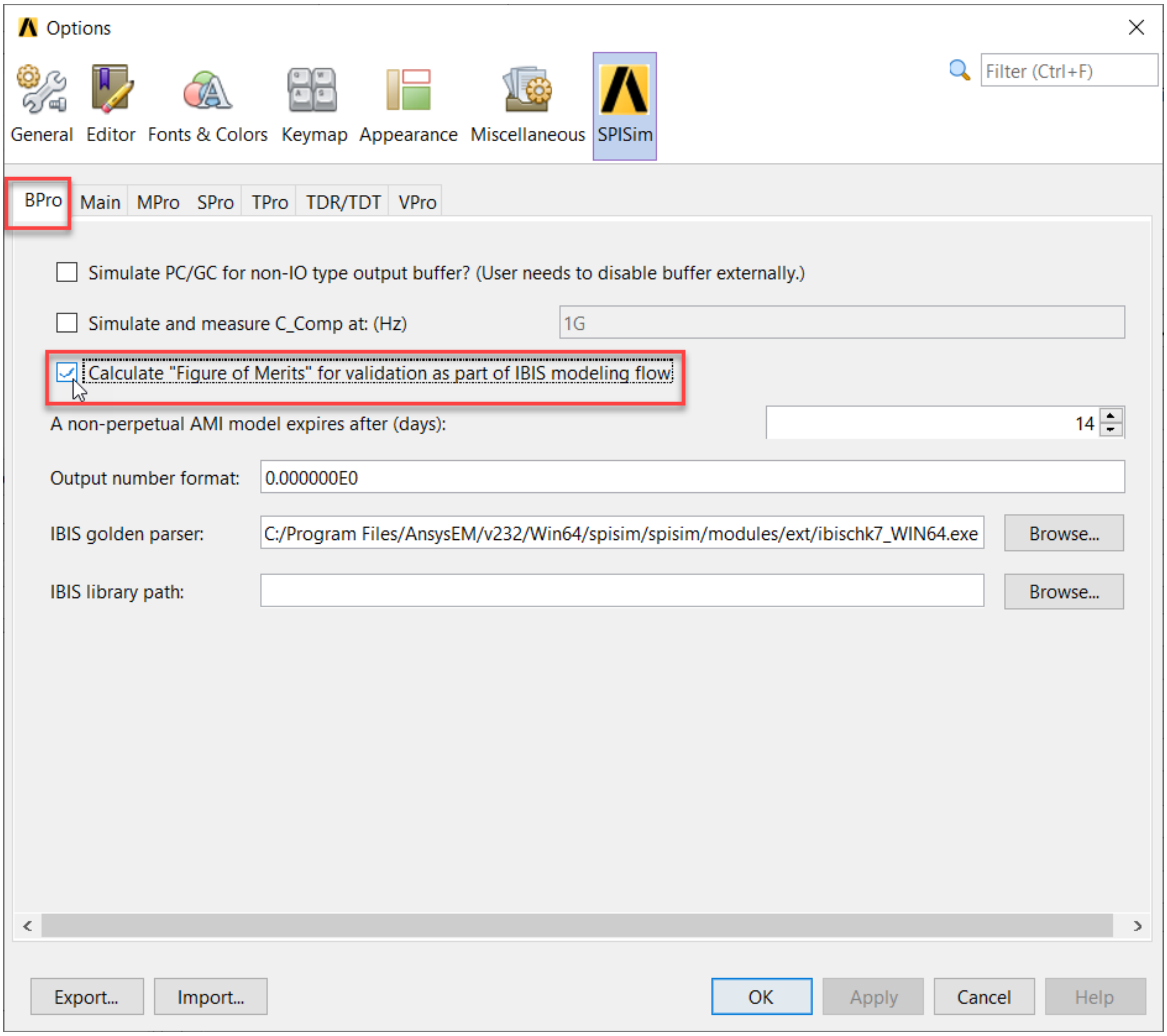
1. Select the **Tools** menu and click **Options**.



2. Select the **SPISim** tab.



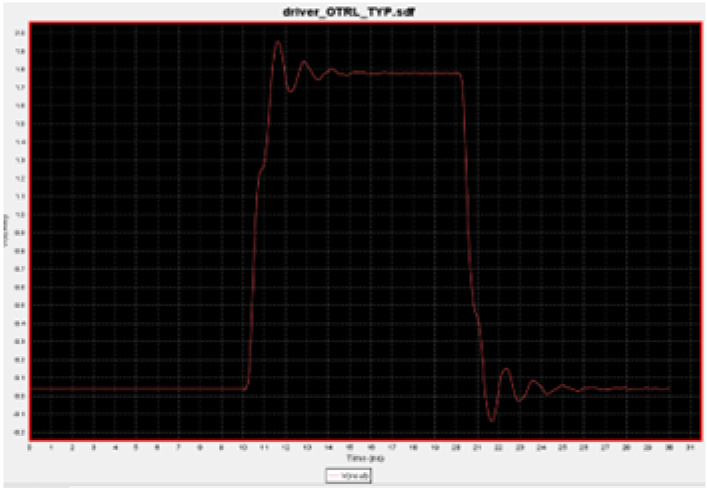
3. In the **BPro** tab, select **Calculate "Figure of Merits" for validation as part of IBIS modeling flow.** and select **OK**.



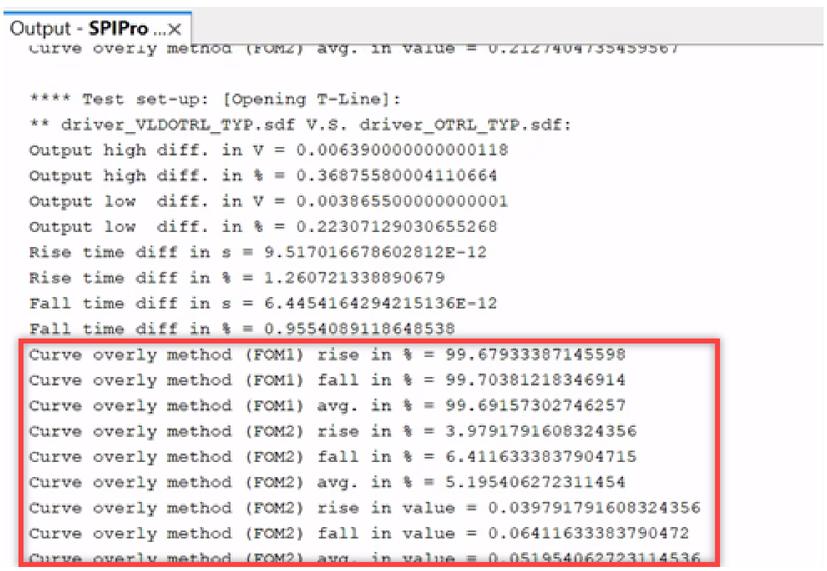
4. In the SPISim IBIS modeling flow **Execute** tab, click through the IBIS model generation flow as normal for Steps 0 - 5. An additional open-ended transmission test load will be created during steps 2, 3, 6, and 7. In step 2 and 3, a transistor buffer connecting to the test load will be generated and simulated together with other regular buffer modeling setup (IVPU, IVPD etc).

5. Click **Run** in Step 6 of the IBIS model generation flow. The generated IBIS model will be connected to the OTRL load and simulated.

> **Important:** Users should verify the simulated results (either IBIS or transistor) have typical transmission line ringing/reflection effect.



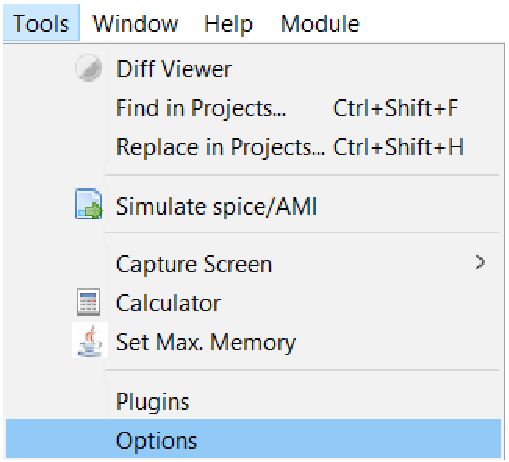
6. Click **Run** in Step 7 of the modeling flow. The FOM1 and FOM2 results will be printed in the output window.



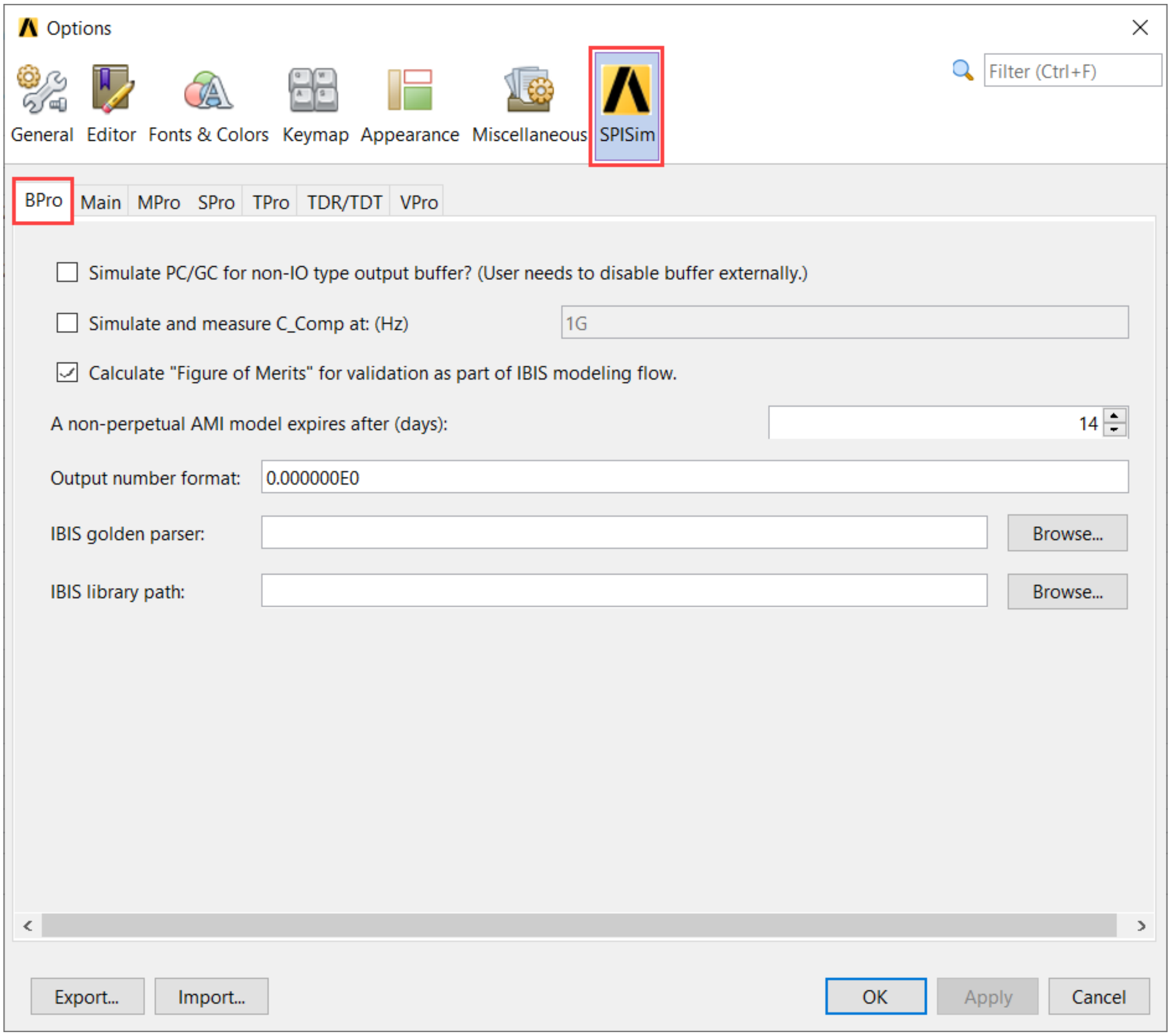
Non-perpetual IBIS-AMI Model Expiration

To specify the number of days before a non-perpetual IBIS-AMI model will expire:

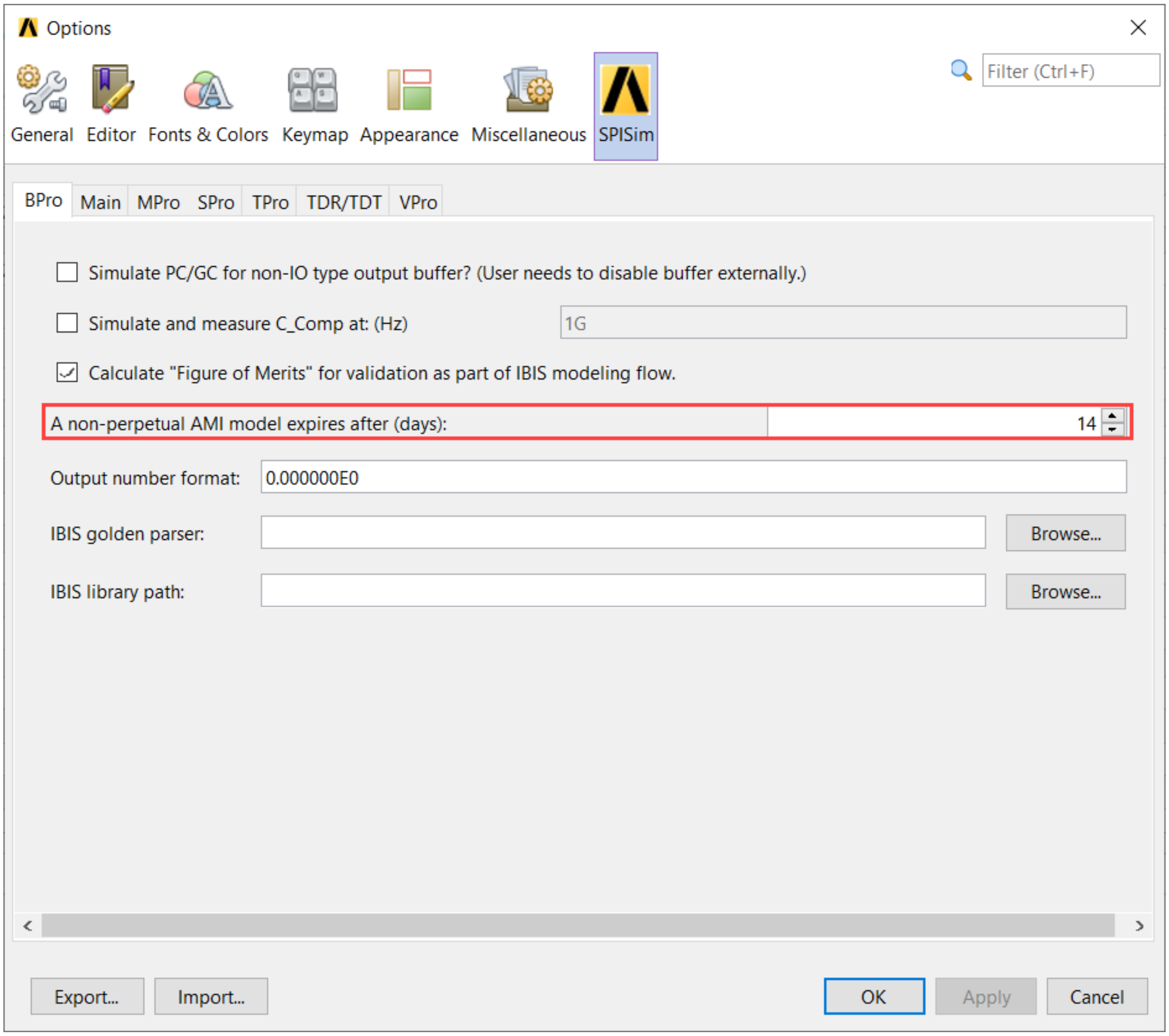
1. Navigate to **Tools > Options**



2. Select **SPISim > BPro**



3. Select the desired number of days before the model expires. Use the arrows or enter a number between 7 and 30.



IBIS Modeling Flow Batch Mode Support

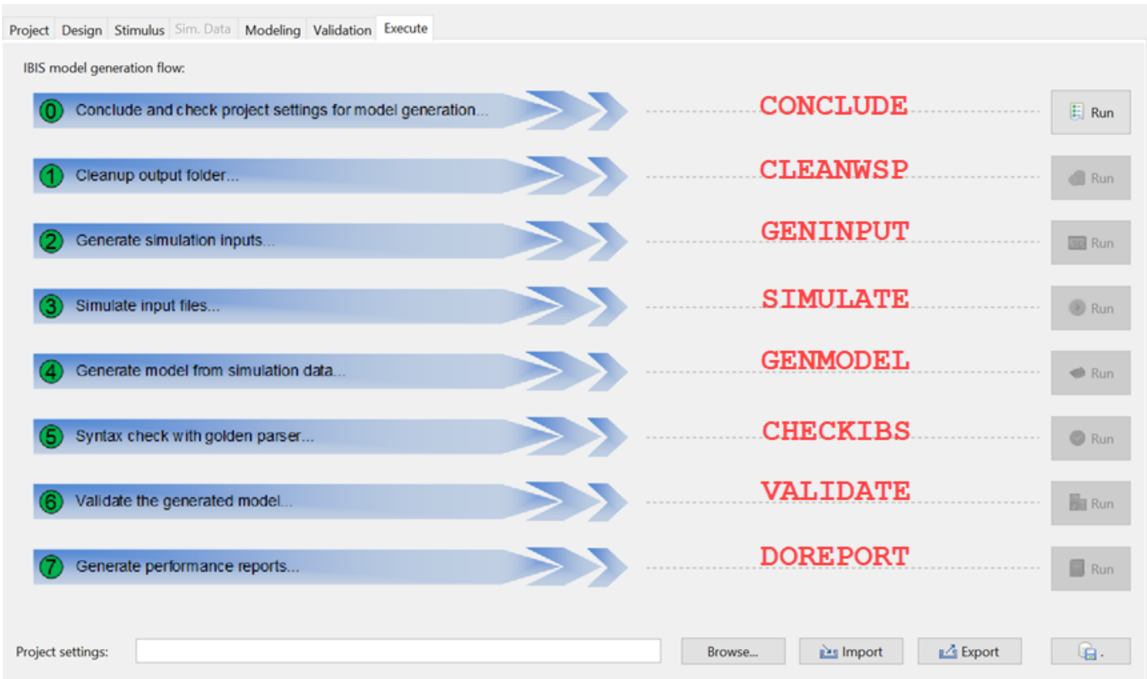
Batch mode support for IBIS modeling flow fully streamlines the process for multiple model generation.

To use the batch model generation tool, a config file is required. The file should contain all the settings necessary for a good set up- that is, the file should be ready to be imported and used to generate a valid IBIS model using the GUI mode.

Each step of the IBIS modeling flow has a corresponding command:

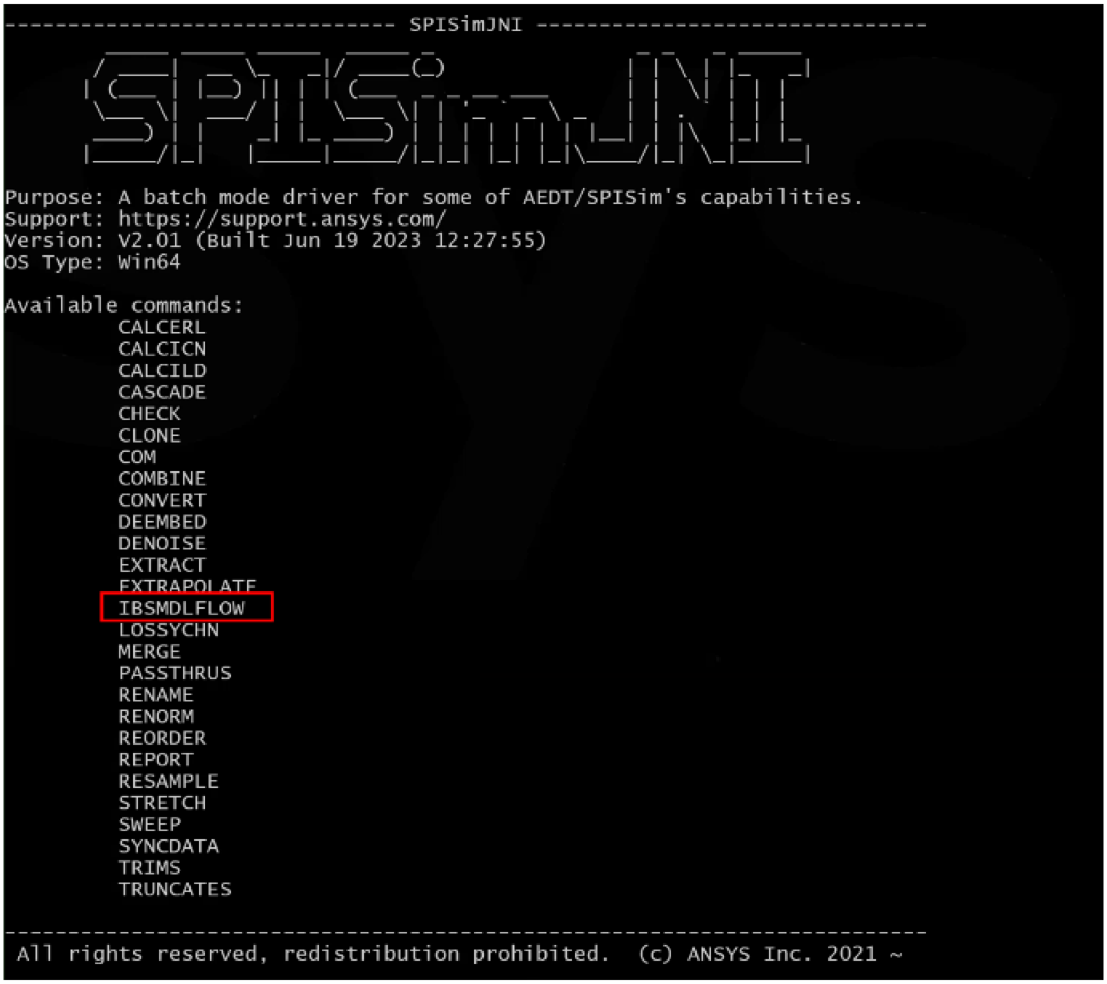
`CONCLUDE, CLEANWSP, GENINPUT, SIMULATE, GENMODEL, CHECKIBS, VALIDATE, DOREPORT.`

The commands are not case-sensitive and the order is not important when being issued to run batch mode. The commands will automatically be rearranged to the correct execution order internally. Additionally, `CONCLUDE` will always be executed so users can skip issuing this step as well.

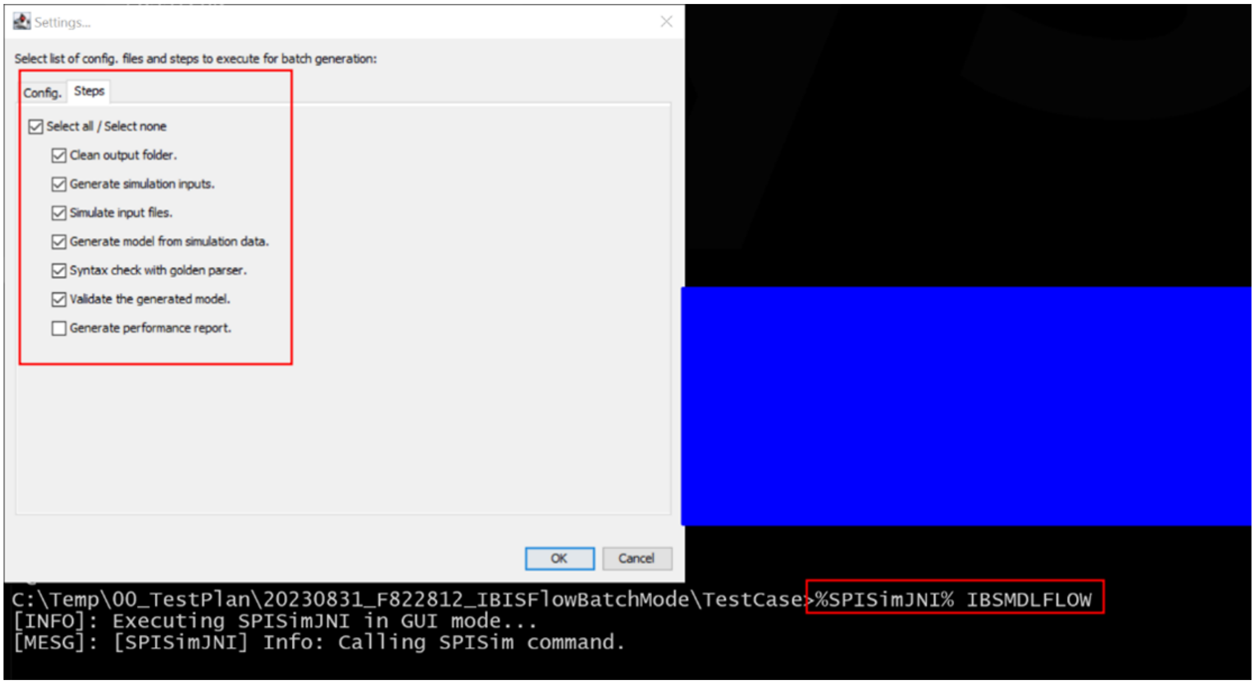


The batch mode executable is **SPISimJNI_Win64/Lx64.exe**, which is same as all the other batch mode commands' entry point.

Running the executable will show the list of commands supported, including newly added **IBISMDLFLOW** command:

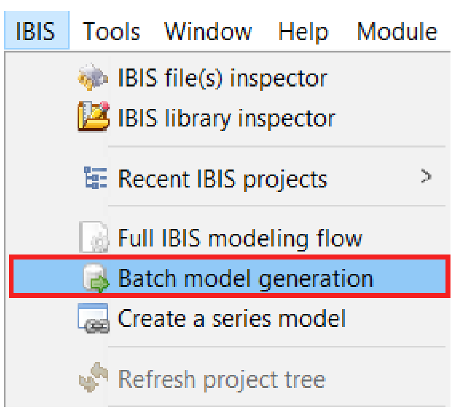


To run batch mode GUI mode, do SPISimJNI_WIN64.exe IBISMDLFLOW:

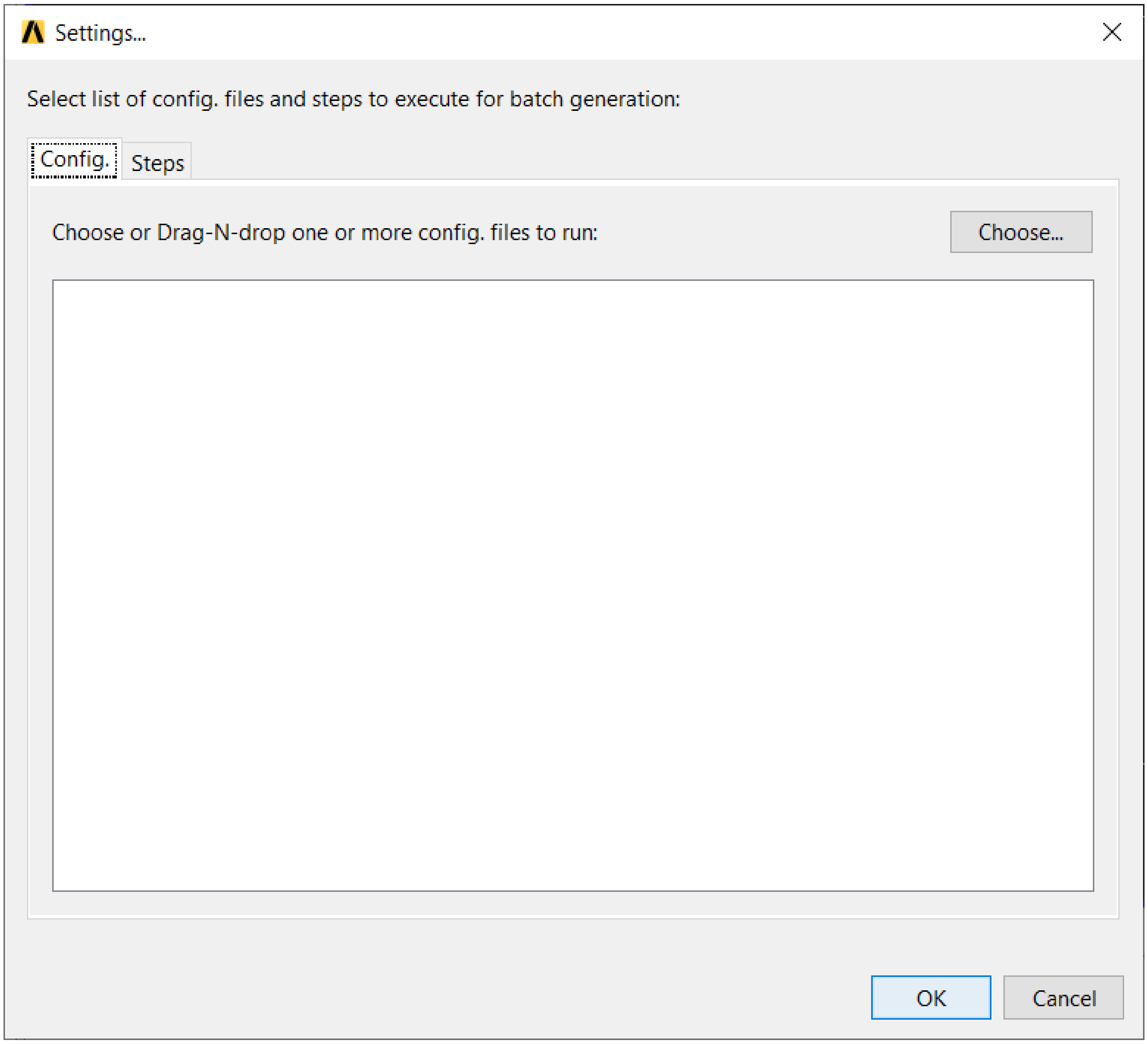


To run batch modeling from the SPISim user interface:

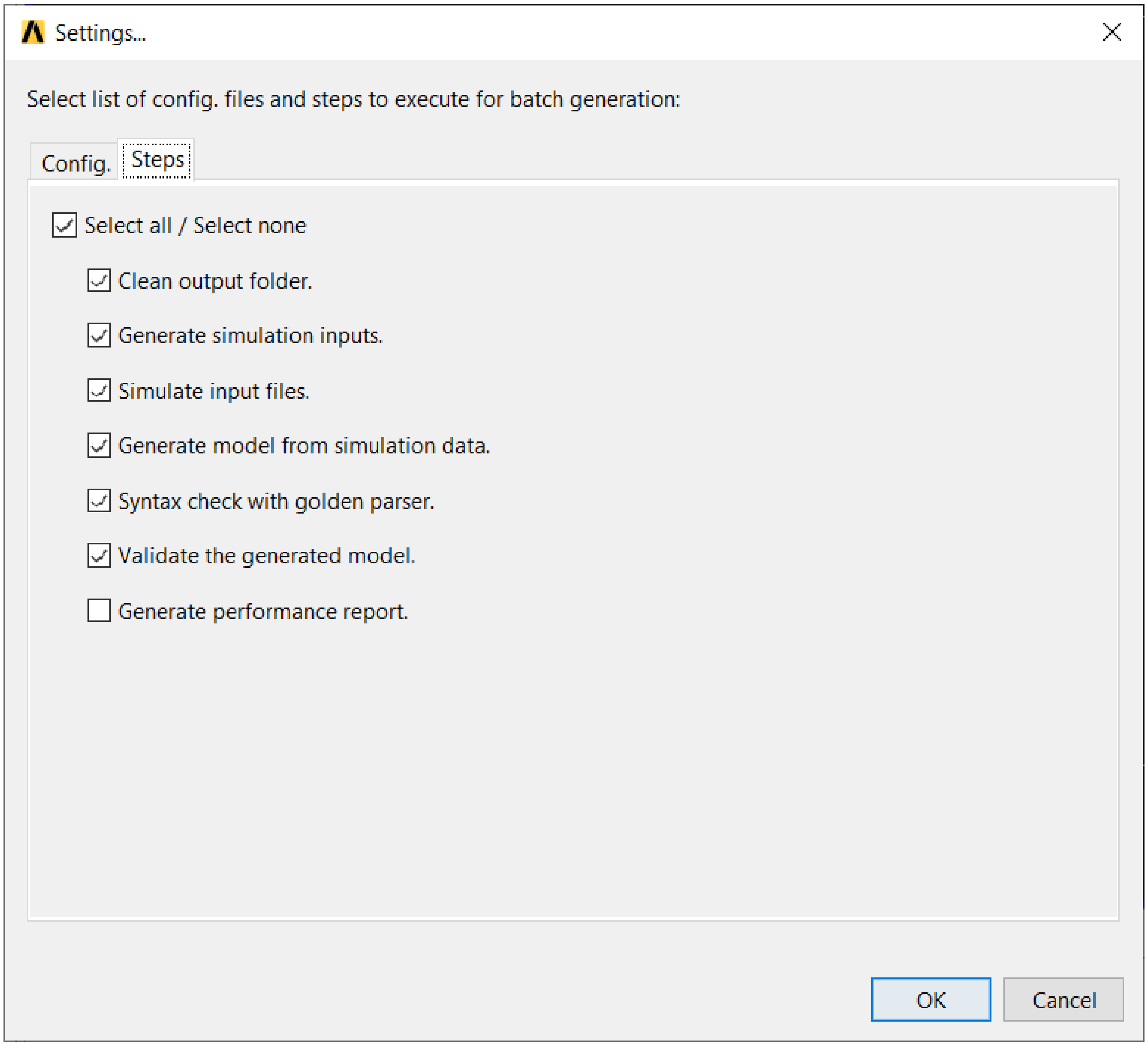
1. Select IBIS > Batch Model Generation



2. Use the **Choose** button, or Drag-N-drop one or more config files to run.



3. Select the steps to be run and click **OK**.



For headless mode, the following three command syntaxes are supported:

1. `SPISimJNI_WIN64.exe IBISFLOW -v CFGFILE=XXX.cfg,GENSTEP=ALL`
2. `SPISimJNI_WIN64.exe IBISFLOW -v CFGFILE=XXX.cfg,GENSTEP=UNTIL:SIMULATE`
3. `SPISimJNI_WIN64.exe IBISFLOW -v CFGFILE=XXX.cfg,GENSTEP="CLEANWSP;GENINPUT"`

`GENSTEP` is the parameter name for the steps to be executed.

`ALL` means run the full flow from begin to end.

`UNTIL:XXXX` means run up to and include the `XXXX` step.

`XXXX` is one of the defined commands mentioned above.

The last syntax list various commands to be run separated with semi-column ";" One example of the last syntax's application is that user may choose to run until all inputs are generated, then simulate outside the flow (e.g. using parallel simulation workgroup), then continue the flow to start `GENMODEL` and finish up the modeling process.

SPISim SPro

SPISim SPro provided S-parameter analysis and viewing of waveforms, SPro can generate configurable time-domain and frequency-domain plots.

> **Note:** A full channel operating margin (COM) analysis summary report can now be generated either through **SPISim** (on the **S-Param** menu, select **Generate Report** > **Channel operating margin**) or through **Electronic Desktop** (on the **Circuit** menu, select **Toolkit** > **[Beta] COM**). Refer to **Generate a COM Analysis Report** for details.

See the Online Help for a video on this module.

In this section:

Generate a Compliance Analysis Report

Generate a Compliance Analysis Report Using Only a Non-GUI Batch Command Line

Generate a Compliance Analysis Report Using a Batch Command Line to Invoke the User Interface

Generate a Compliance Analysis Report

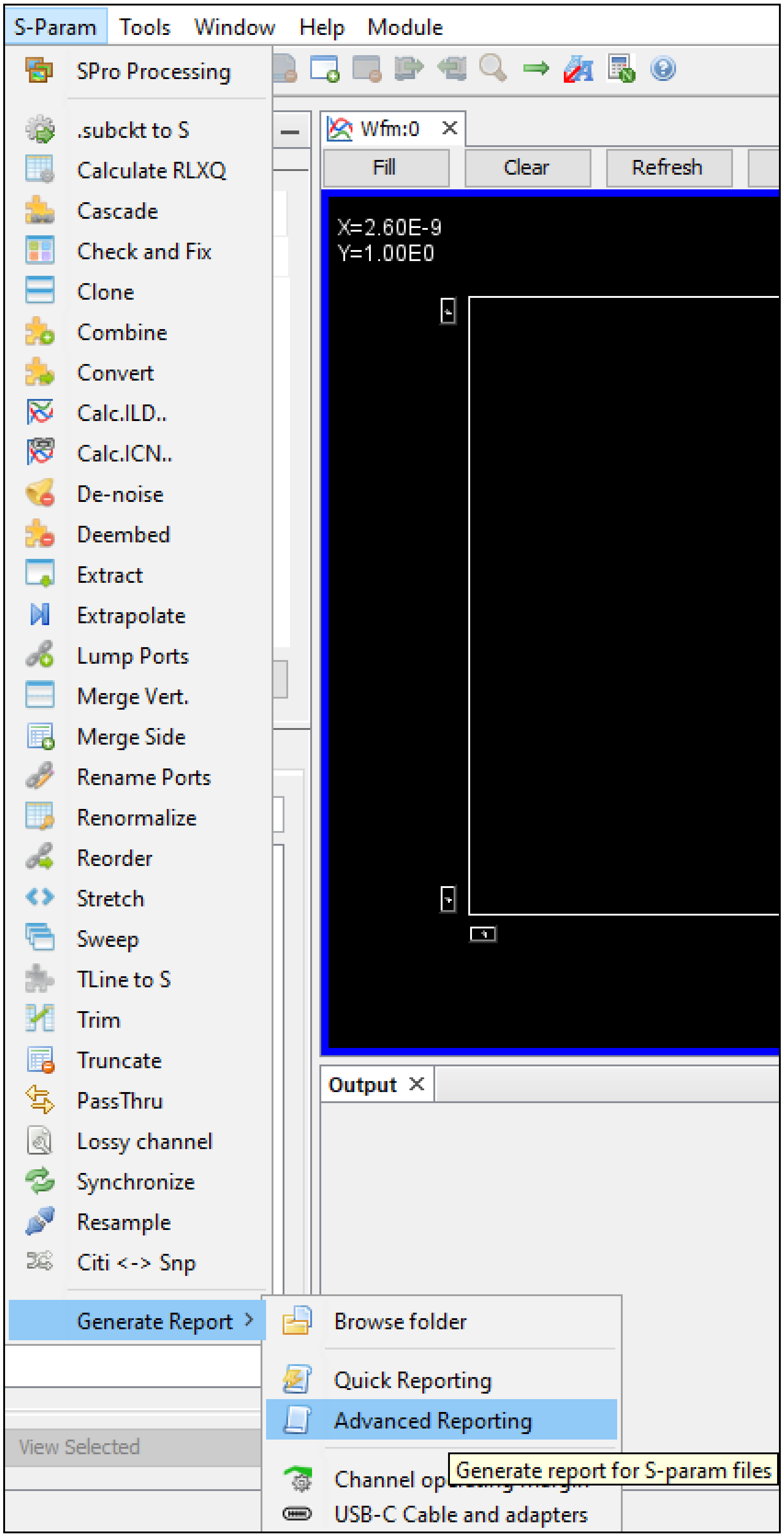
SPISim can generate a compliance report for the following standards:

IEEE 802.3 25G AUI C2M	**IEEE 802.3 50GAUI-1 C2M**
IEEE 802.3 25GAUI C2C	**IEEE 802.3 50GAUI-1 C2C**
IEEE 802.3 2G5BASE CR	**IEEE 802.3 50GBASE CR**
OIF CEI 28G CEI-25-LR	**OIF CEI 28G CEI-28G-MR**
OIF CEI 56G CEI-56G-LR-PAM4	**IEEE 100_1000_BaseT1_CMC**
IEEE 100_1000_BaseT1_MDI	**MIPI**

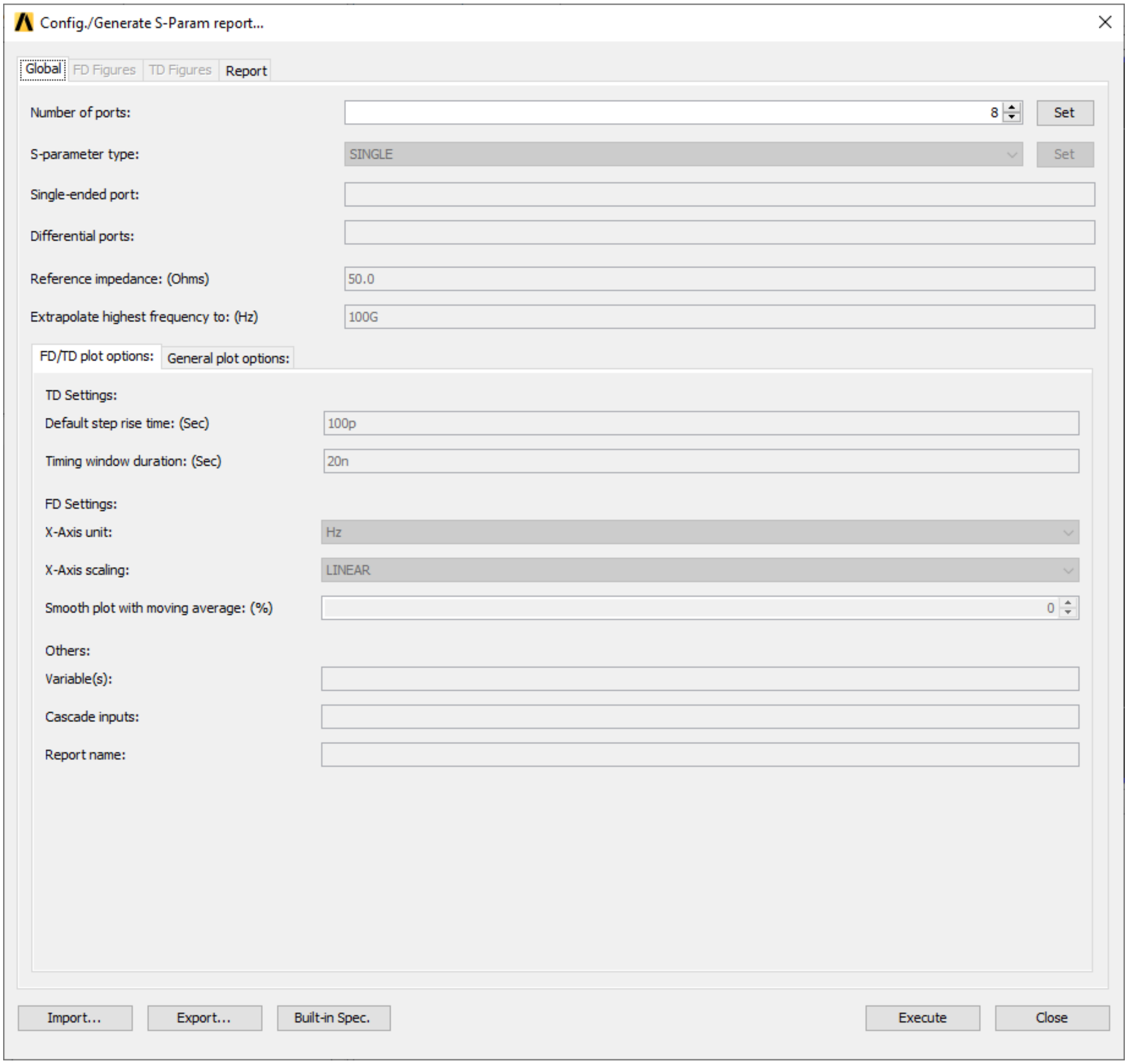
Complete the following steps to create a compliance analysis report.

> **Note:** A report can also be generated using batch command line. Refer to Generate a Compliance Analysis Report Using Online a Non-GUI Batch Command Line or Generate a Compliance Analysis Report Using a Batch Command Line to Invoke the User Interface for details.

1. From **S-Param**, select **Generate Report** > **Advanced Reporting** to open the **Config./Generate S-Param report...** window.


S-Param
Tools
Window
Help
Module
SPro Processing
.subckt to S
Calculate RLXQ
Cascade
Check and Fix
Clone
Combine
Convert
Calc.ILD..
Calc.ICN..
De-noise
Deembed
Extract
Extrapolate
Lump Ports
Merge Vert.
Merge Side
Rename Ports
Renormalize
Reorder
Stretch
Sweep
TLine to S
Trim
Truncate
PassThru
Lossy channel
Synchronize
Resample
Citi <-> Snp
Generate Report
View Selected
Wfm:0
Fill
Clear
Refresh
X=2.60E-9
Y=1.00E0
Output
Browse folder
Quick Reporting
Advanced Reporting
Channel op
Generate report for S-param files
USB-C Cable and adapters

2. Click **Built-in Spec.** to open the **Load built-in spec. template...** window.



Note:

The **Built-in Spec.** option will populate the **Config./Generate S-Param report...** window with the appropriate parameters. If the user selects either an **OIF** or **MIPI** built-in spec template, follow these additional steps.

From the **Global** tab, select the correct **Number of ports** (i.e., *.s12p* for **OIF**, *.s6p* for **MIPI**). Then click **Set**.

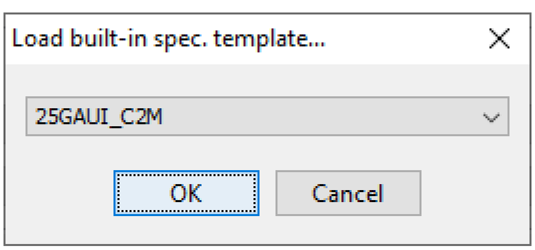
The **OIF** built-in spec templates assume input will be a 12-port file (i.e., *s12p*), with first and third pair for aggressors and second (i.e., central) pair for signal traces. At least four ports are required, or the user will need to import a modified built-in spec. template file (i.e., *.tpl*) to continue processing regardless. With the *.s4p* files arranged in "Even-Odd" mode, it will not be necessary to import a modified *.tpl* file.

The **MIPI** built-in spec template assumes input will be a 6-port file (i.e., *.s6p*), and individually targets each pair. Users may need to extract one or more *.s4p* files before processing (e.g., AB, from port (1, 2, 4, 5), AC from port (1, 3, 4, 6), or BC from port (2, 3, 5, 6). With the *.s4p* files arranged in "Even-Odd" mode, it will not be necessary to import a modified *.tpl* file.

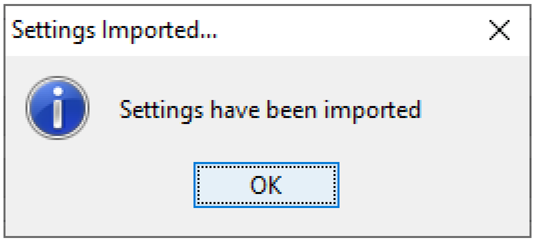
Choosing **8** (i.e., *.s8p*) ports will not work for **OIF** or **MIPI** built-in spec templates. A modified *.tpl* file will need to be imported.

For all advanced template modification, please contact **Ansys Support**.

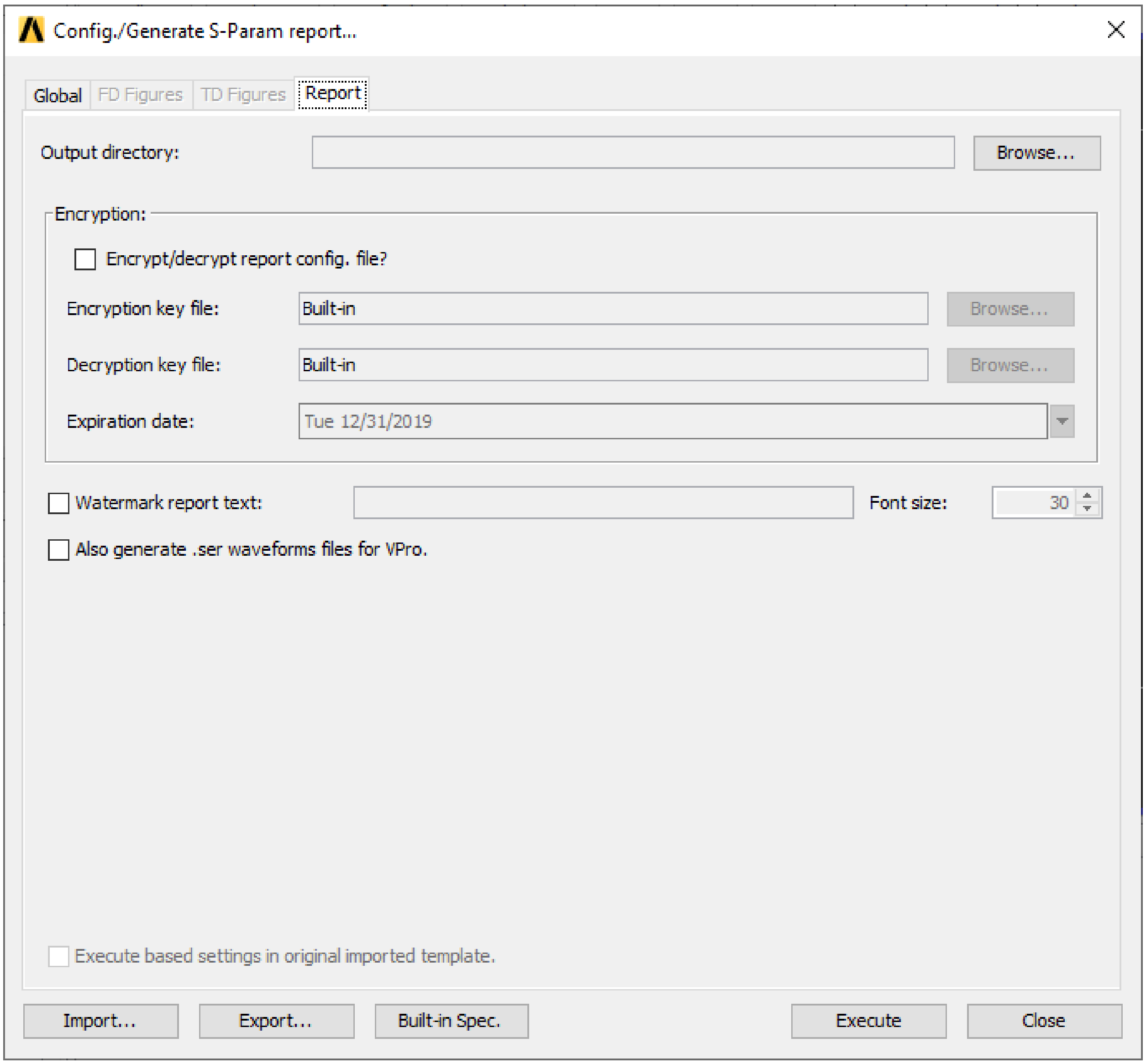
3. Select a built-in spec template of the appropriate compliance protocol from the drop-down menu. Then click **OK** to close the **Load built-in spec. template...** window and populate the **Config./Generate S-Param report...** window with the template's parameters. When the process completes, a new window appears confirming the import.



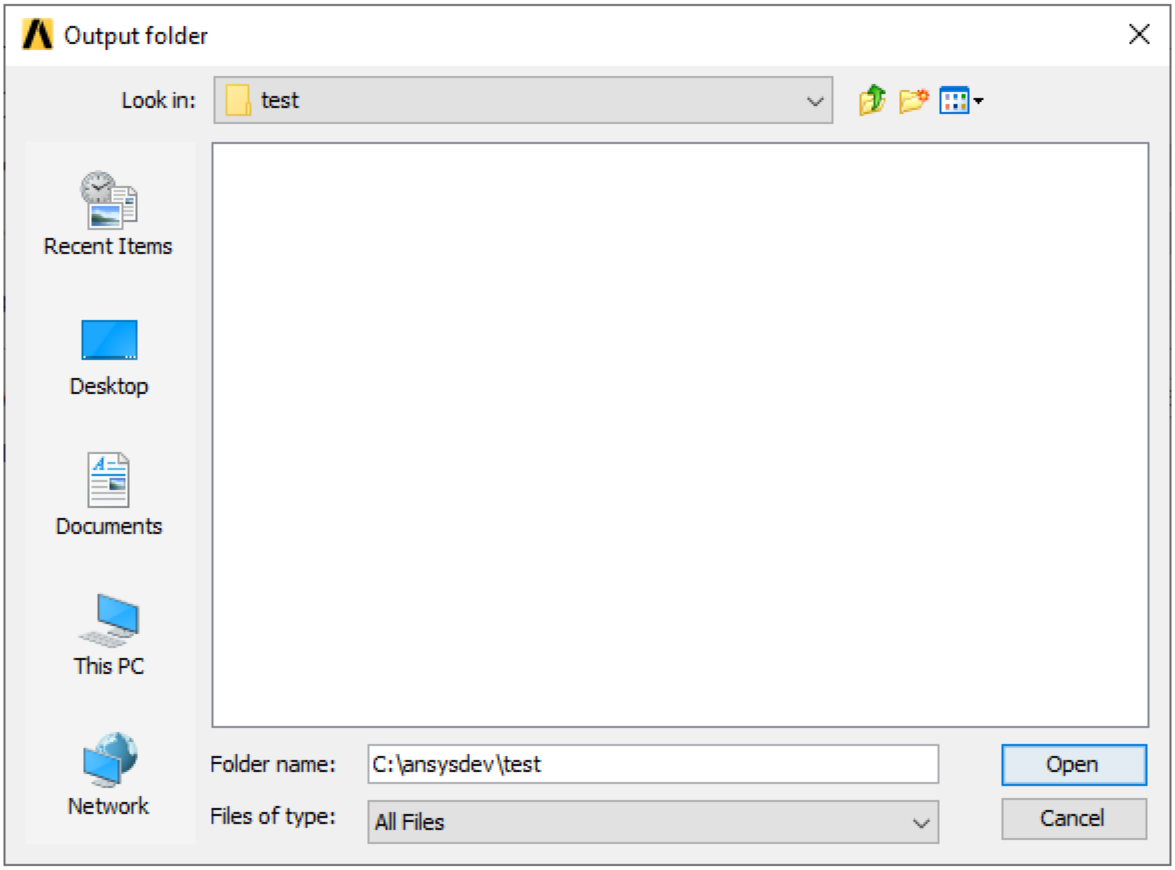
4. Click **OK**.



5. Click the **Report** tab.

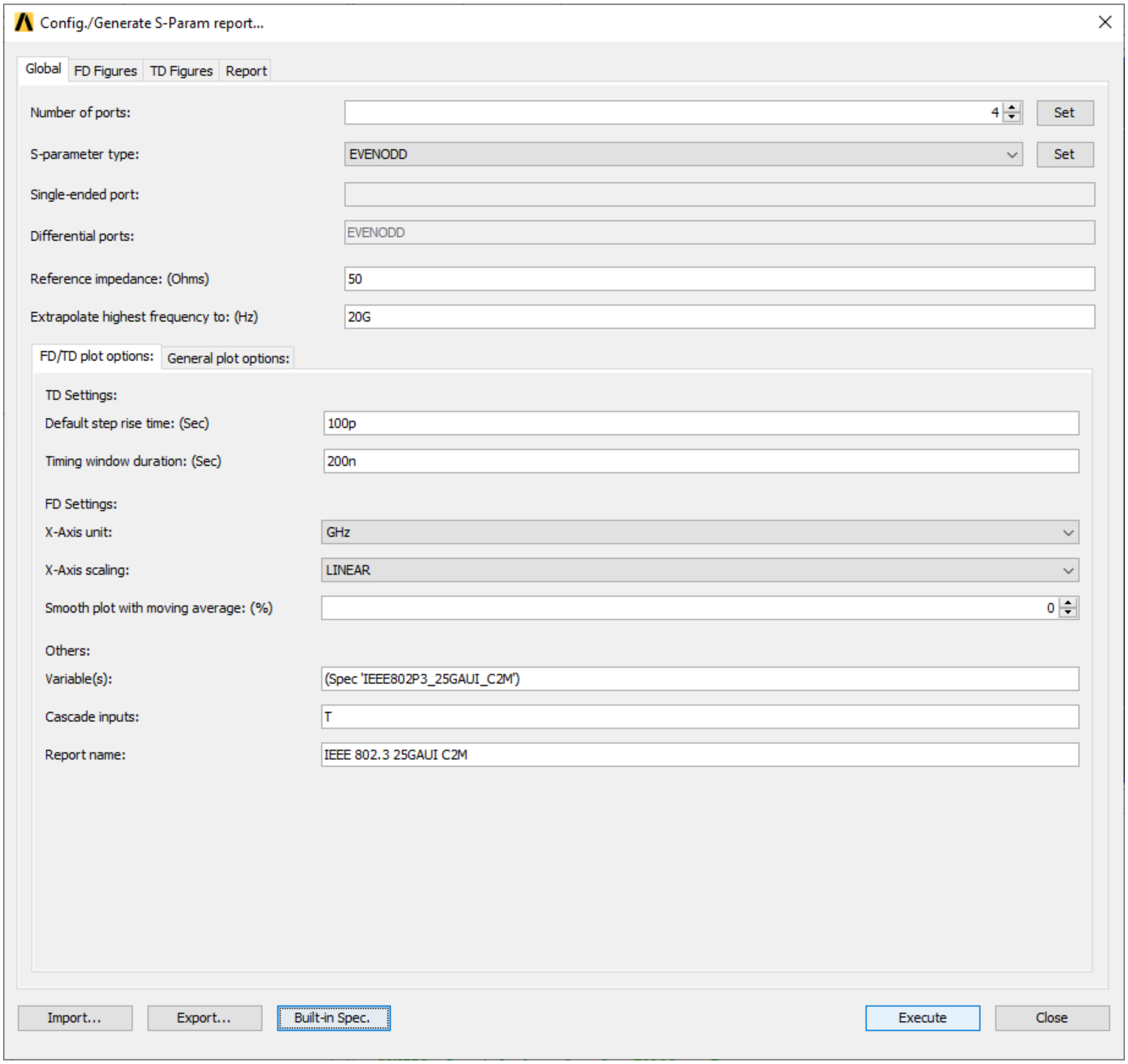


6. Click **Browse...** adjacent to **Output directory** to open an explorer window. Navigate to the appropriate output folder. Then click **Open** to select the folder and close the explorer window.



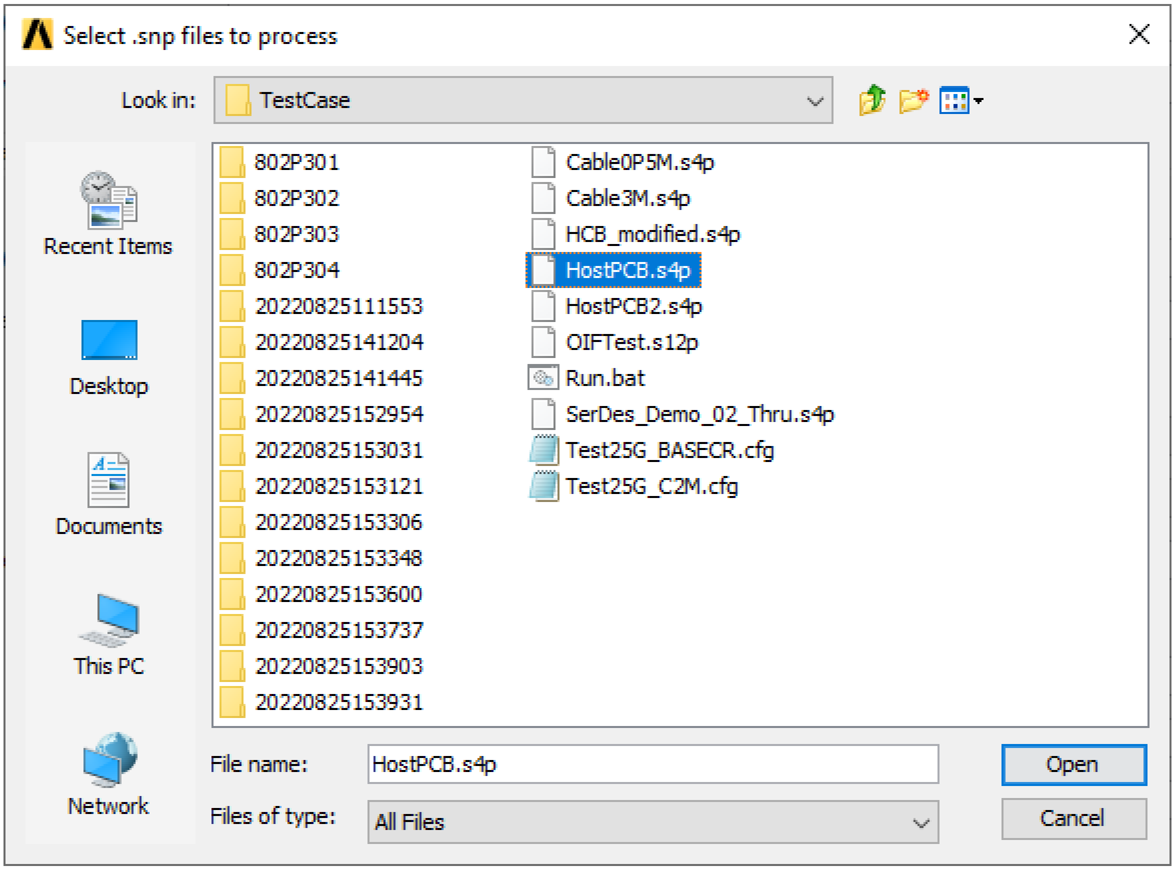
> **Note:** If no **Output directory** is chosen, the *.snp* file directory will be used as the **Output directory**.

7. Click **Execute**.

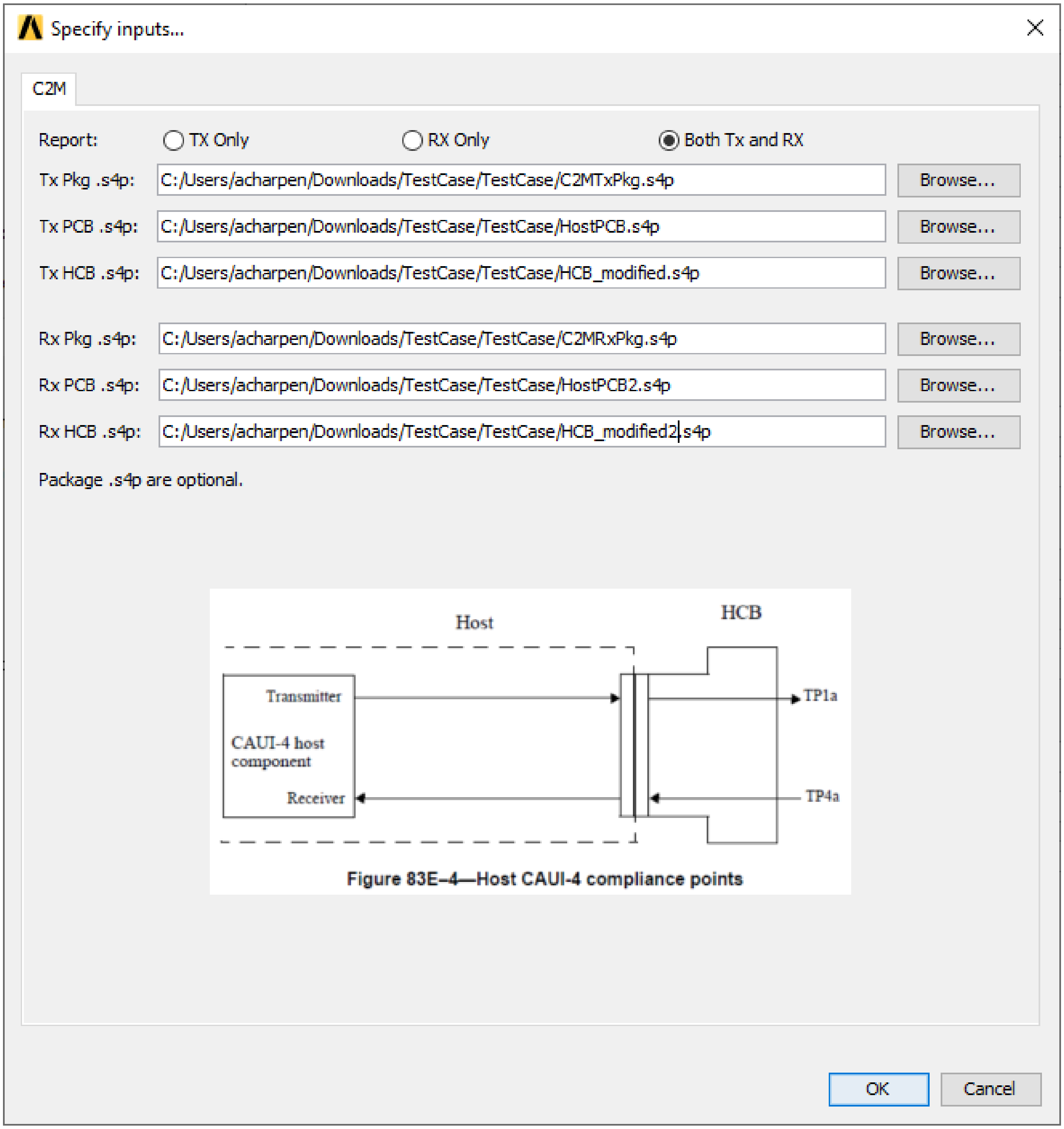


8. Depending on the template chosen in step 3., one of three screens will appear:

- **All OIF, IEEE 802.3 25G/50G C2C, 100BT/1000BT Base-T1 CMC/MDI, and MIPI built-in spec templates -** an explorer window will appear. Navigate to the appropriate .sN*p* file (s) (e.g., *.s12p*). Select one or more to input, then click **Open** to begin generating a report. Every .sN*p* file will be treated equally and processed in order.



- **IEEE 802.3 25G/50G C2M built-in spec templates** - a **Specify inputs...** window will appear. Select a **Report** type (i.e., **TX Only**, **RX Only**, or **Both Tx and RX**) to activate and deactivate the appropriate rows. Then use the adjacent **Browse...** buttons in the enabled columns to select the appropriate *.s4p* files (e.g., **Tx Pkg *.s4p***, **Tx PCB *.s4p***, **Tx HCB *.s4p***, **Rx Pkg *.s4p***, **Rx PCB *.s4p***, and/or **Rx HCB *.s4p***). **PCB** and **HCB *.s4p*** files **must** be provided. If no **Pkg *.s4p*** is provided, only **PCB** and **HCB*.s4p*** files will be used to generate a cascaded file and form channel *.s4p*. If a **Pkg *.s4p*** is provided, all three files will be used to process disparate sections of the compliance report. The generated report will only contain data relevant to the **Report** selection made in this window (i.e., **TX Only**, **RX Only**, or **Both Tx and RX**)

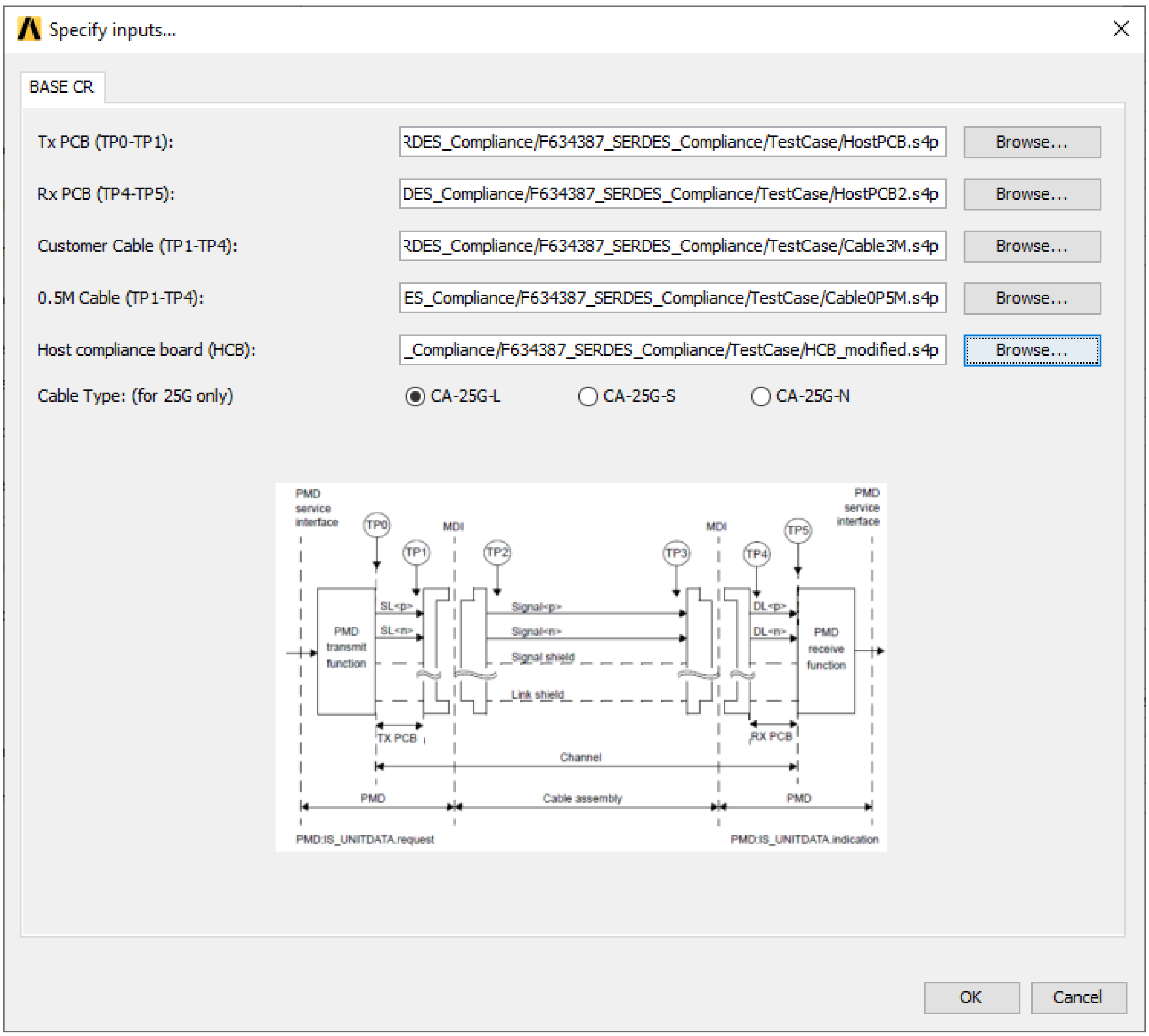


Note:

The **Host TF** *.s4p* file must be valid. If an **HCB** *.s4p* file is unavailable, no cascaded file can be generated and the **Host TF** *.s4p* will be used to process the entire compliance report.

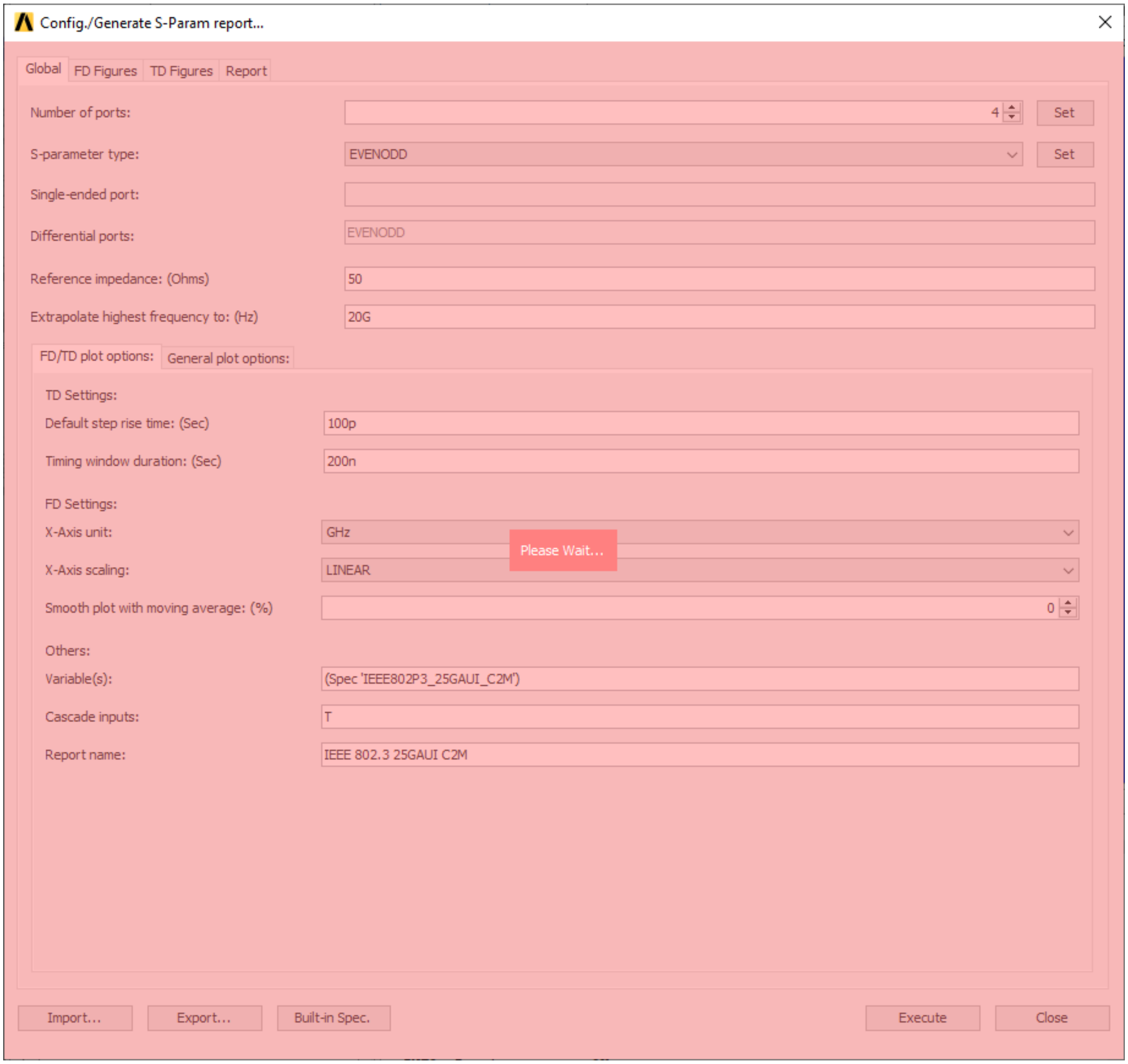
- **IEEE 802.3 25G/50G Base CR built-in spec templates** - a **Specify inputs. . ..** window will appear. Use the adjacent **Browse...** buttons to select the appropriate **Tx PCB**, **Rx PCB**, **Customer Cable**, **0.5M Cable**, and **Host compliance board** *.s4p* files. Then select the **Cable Type** (i.e., **CA-25G-L**, **CA-25G-S**, or **CA-25G-N**).

> **Note:**
>
> The order of the inputs becomes important when the compliance report is completed via non-GUI batch command line. Refer to **"Generate a Compliance Analysis Report" on page 15-51**.

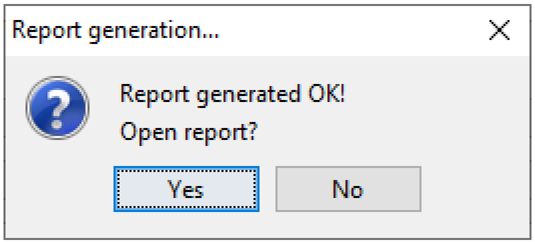


9. Click **Open**/**OK**, depending on the window that opened in step 8, to close the window and begin calculation.

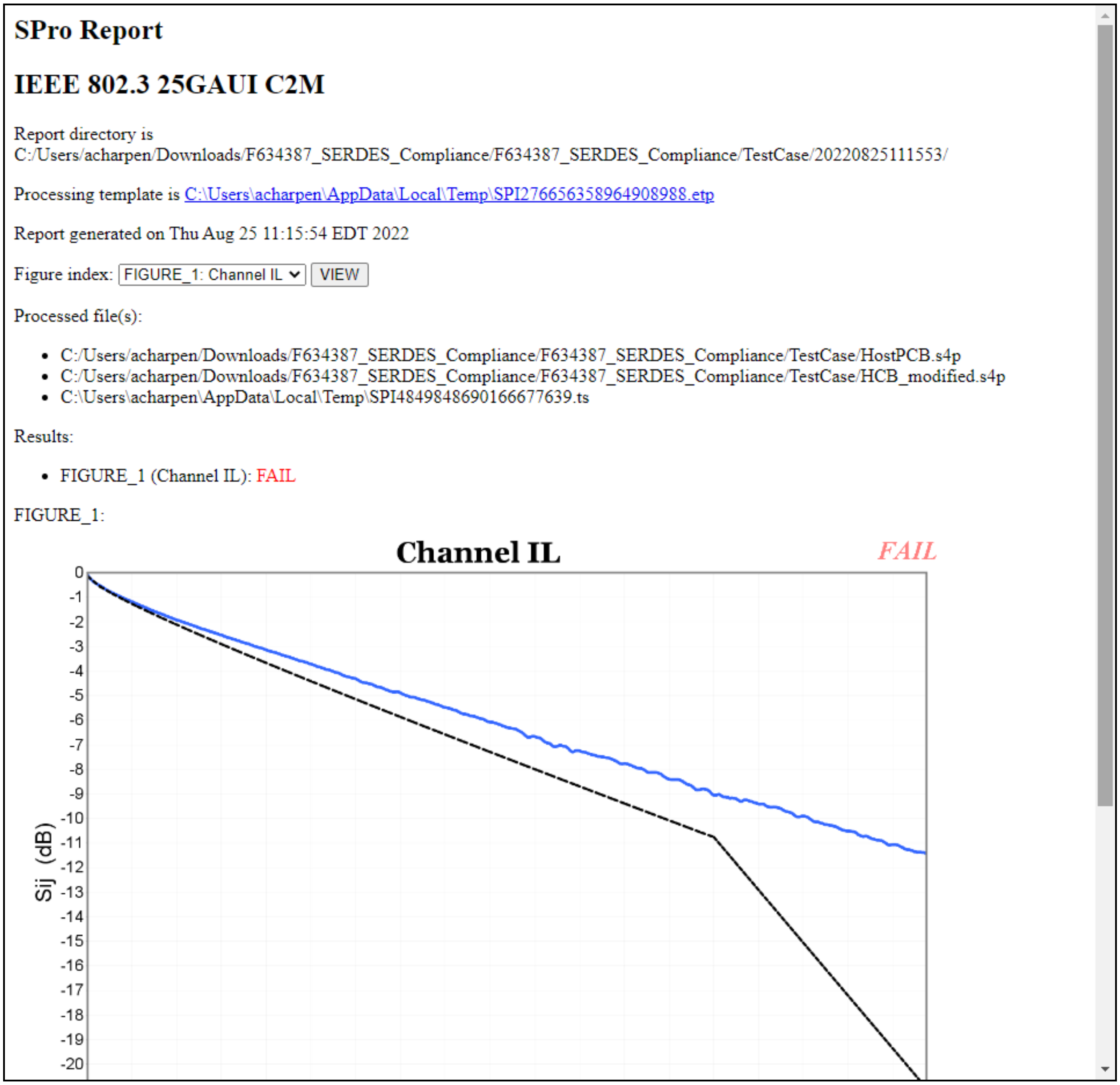
10. The **Config./Generate S-Param report...** window will turn red and the words **Please Wait...** will appear momentarily.



11. When the report is ready, a **Report generation...** window appears.



Click **Yes** to open the report in the system's default browser. The second line in the report heading states the compliance protocol that was tested. The report contents will vary depending on the chosen template and the number of input files.

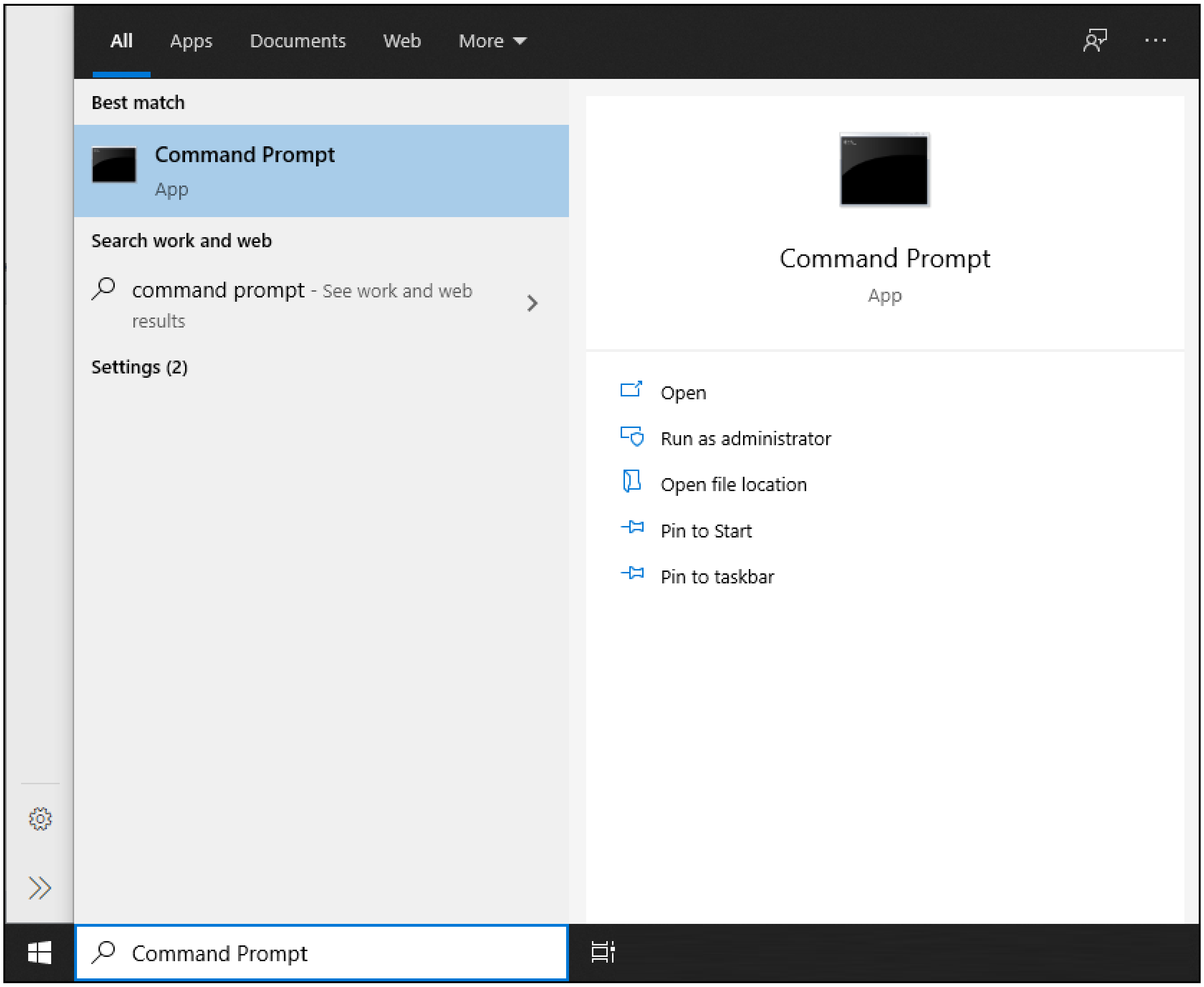


12. The **Config./Generate S-Param report...** window remains open beneath the browser window. To generate a report with different specs, simply repeat steps 2-10.

Generate a Compliance Analysis Report Using Only a Non-GUI Batch Command Line

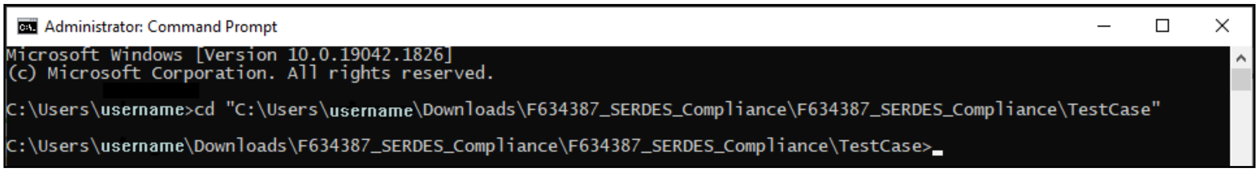
Complete the following steps to run a compliance analysis report using only a non-GUI batch command line.

1. Run a command prompt (e.g., in a Windows environment, type *Command Prompt* into the **Search Bar** and press **Enter** to open a **Command Prompt** window).

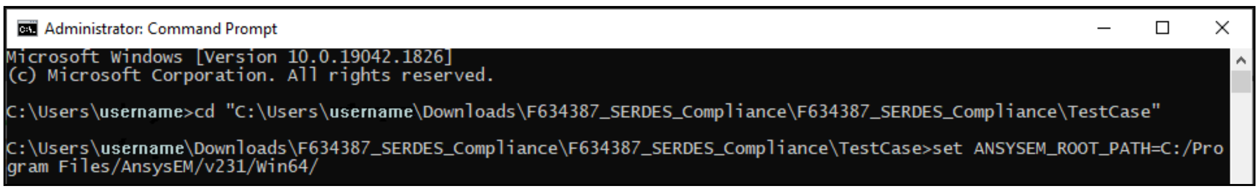


Note: The following example assumes **Electronics Desktop** is installed in the default directory in a Windows operating system environment. Structure the command appropriately for a different directory, path, and/or operating system, as necessary. Do **not** move the ***SPISimJNI_WIN64.exe*** executable file. The executable file relies on the relative path to the **Electronics Desktop** installation folder.

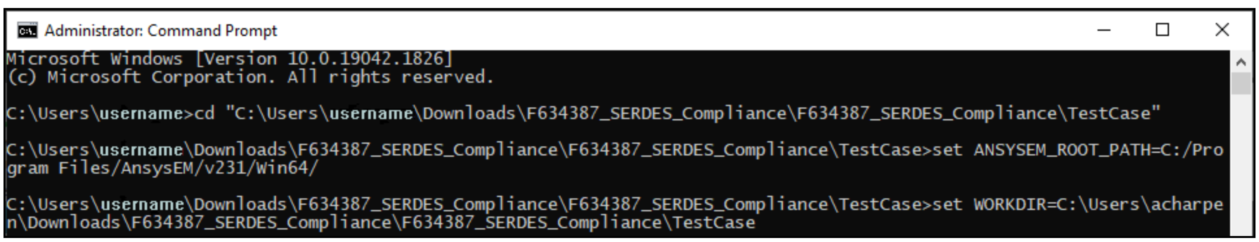
2. Change the current directory by entering the following command: `cd "[The direct path to the appropriate .sNp file(s).]"`. Then press **Enter**.



3. Set the Ansys installation root directory by entering the following command: `set ANSYSEM_ROOT_PATH=C:/Program Files/AnsysEM/v242/Win64/`. Then press **Enter**.



4. Set the report output directory by entering the following command: `set WORKDIR=[The direct path to an appropriate directory where a report will be output.]`. Then press **Enter**.



Note: Depending on the type of compliance analysis report requested, adhere to the following additional instructions.

All OIF built-in spec templates - assume a *.s12p* file is used (first and third pair for aggressors, second pair for signal) in even-odd ordering scheme.

IEEE 802.3 25G/50G C2M built-in spec templates - specify up six input files. If six input files are not used, the missing inputs should be indicated by `NA` or `THRU`.

```
Administrator: Command Prompt
C:\Users\username \Downloads\F634387_SERDES_Compliance\F634387_SERDES_Compliance\TestCase>"%ANSYSEM_ROOT_
PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe" REPORT --inp %WORKDIR%/HostPCB.s4p,NA,NA,NA,NA,NA --
val BUILTIN=OIF_CEI_25G_LR,OUTPATH=%WORKDIR%/OUTPUT_OIF1
BUILTIN=OIF_CEI_25G_LR,OUTPATH=C:\Users\username \Downloads\F634387_SERDES_Compliance\F634387_SERDES_Comp
liance\TestCase/OUTPUT_OIF1, [INFO]: Executing SPISimJNI in NON-GUI mode...
[MESG]: [INFO] Checking out PPE mode licenses...
[MESG]: [SPISimJNI] Info: Calling SPISim command.
[MESG]: CONFIG: Set default value for DURAT to 200n...
[MESG]: INFO: Parsing C:\Users\username \Downloads\F634387_SERDES_Compliance\F634387_SERDES_Compliance\Te
stCase/HostPCB.s4p...
[MESG]: INFO: Generating figure Figure_2
[MESG]: UDP_COMM_CONCLUDE
[INFO]: Concluded.
```

IEEE 802.3 25G/50G Base CR built-in spec templates - specify one of three cable types (i.e., `CBTLTYPE`): `CA_25G_L`, `CA_25G_S`, `CA_25G_N`.

```
"%ANSYSEM_ROOT_PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe"
REPORT --inp %WORKDIR%/HostPCB.s4p,%WORKDIR%/HostPCB2.s4p,%WORKDIR
%/Cable3M.s4p,%WORKDIR%/Cable0P5M.s4p,%WORKDIR%/HCB_modified.s4p
--val BUILTIN=25GBASE_CR,OUTPATH=%WORKDIR%/802P304,CBLTYPE=CA_25G_L
```

Batch Mode

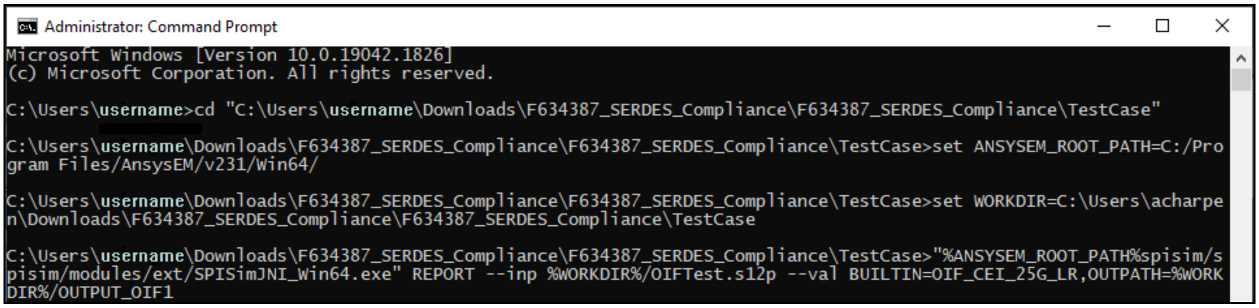
Users can input several .*s*N*p* files for analysis (i.e., batch mode).

```
"%ANSYSEM_ROOT_PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe"
REPORT --inp %WORKDIR%/OIFTest.s12p --val BUILTIN=OIF,OUTPATH=
%WORKDIR%/OUTPUT_OIF0
"%ANSYSEM_ROOT_PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe"
REPORT --inp %WORKDIR%/OIFTest.s12p --val
BUILTIN=OIF_CEI_25G_LR,OUTPATH=%WORKDIR%/OUTPUT_OIF1
"%ANSYSEM_ROOT_PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe"
REPORT --inp %WORKDIR%/OIFTest.s12p --val
BUILTIN=OIF_CEI_28G_MR,OUTPATH=%WORKDIR%/OUTPUT_OIF2
"%ANSYSEM_ROOT_PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe"
REPORT --inp %WORKDIR%/OIFTest.s12p --val
BUILTIN=OIF_CEI_56G_LR_PAM4,OUTPATH=%WORKDIR%/OUTPUT_OIF3
```

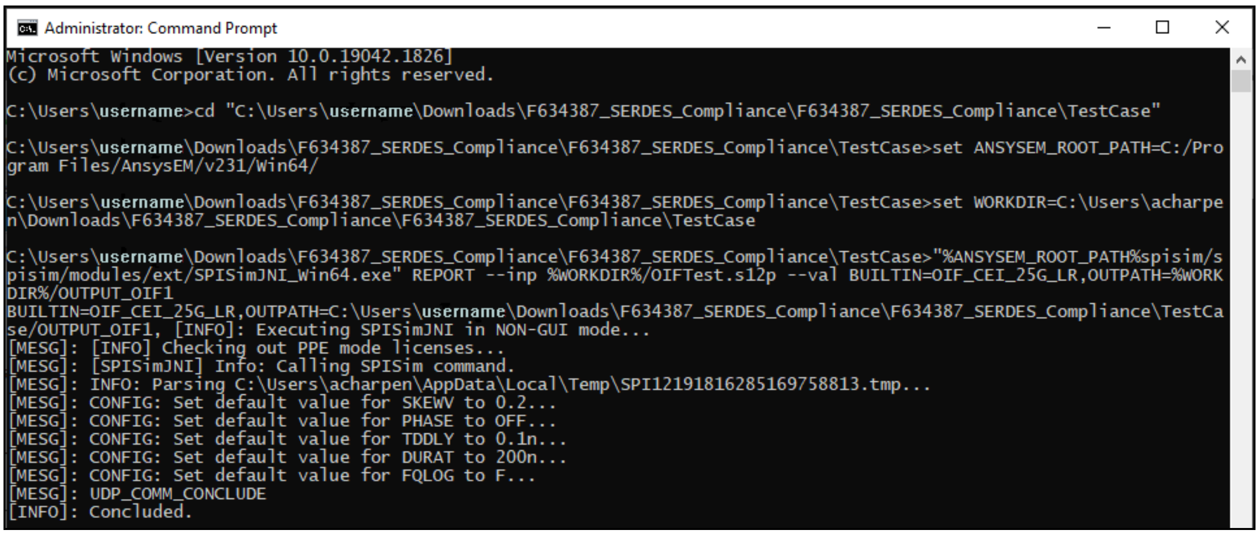
When creating batch analysis reports of certain types, adhere to the following additional instructions.

- **IEEE 802.3 25G/50G C2M built-in spec templates** - input two .*s4p* files, ordered as follows:
  0: TP0-TP1 (PCB)
  1: TP1-TP2 (HCB)
- **IEEE 802.3 25G/50G Base CR built-in spec templates** - input up to five *s4p*files, ordered as follows:
  0: TP0-TP1 (Tx PCB)
  1: TP4-TP5 (Rx PCB)
  2: TP1-TP4 (Customer Cable)
  3: TP1-TP4 (0.5M Cable)
  4: TP1-TP2 (HCB)

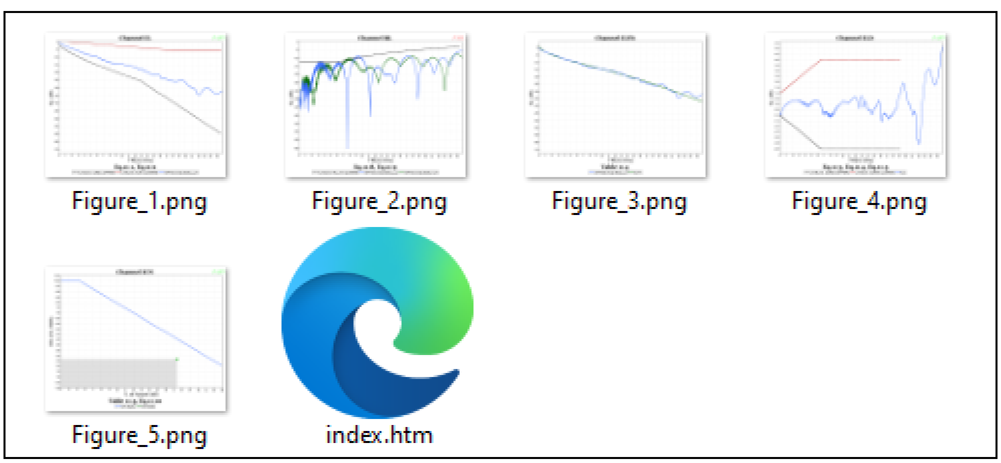
5. Order a compliance analysis using a built-in spec template. by entering the following command: `"%ANSYSEM_ROOT_PATH%spisim/spisim/modules/ext/SPISimJNI_Win64.exe" REPORT --inp %WORKDIR%/[Name and extension of the .sNp file to analyze (e.g., OIFTest.s12p).] --val BUILTIN=[The template to use for analysis (e.g., OIF_CEI_25G_LR)],OUTPATH=%WORKDIR%/OUTPUT_OIF1`. Then press **Enter**.



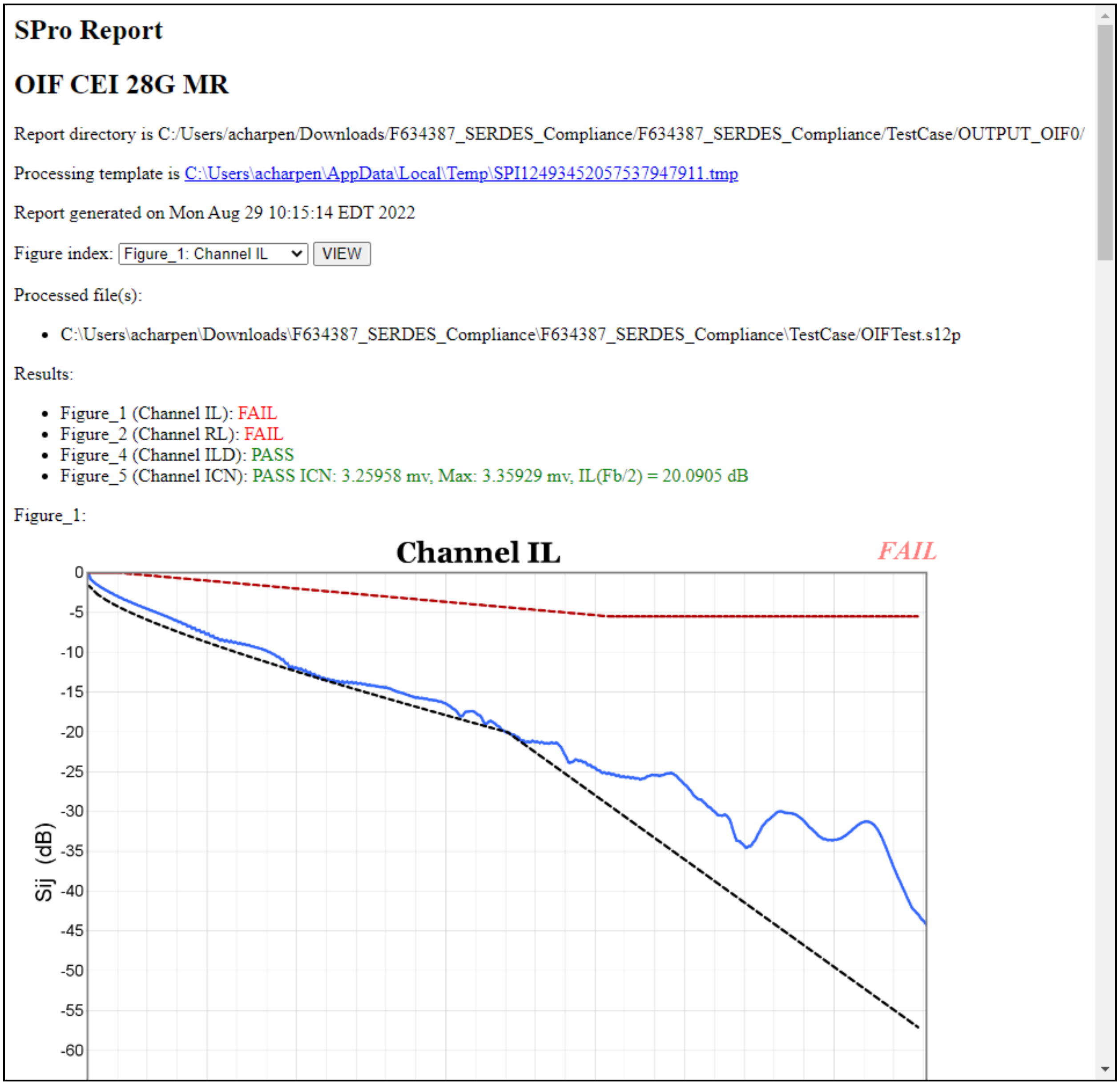
SPISim analyzes the input file using the chosen template's parameters.



SPISim outputs the report as *.png* files and an *.htm* file in the directory chosen in step 4.



6. Open the report in the system's default browser. The second line in the report heading states the compliance protocol that was tested. The report contents will vary depending on the chosen built-in spec template and the number of input files.



SPro Report

OIF CEI 28G MR

Report directory is C:/Users/acharpen/Downloads/F634387_SERDES_Compliance/F634387_SERDES_Compliance/TestCase/OUTPUT_OIF0/

Processing template is C:\Users\acharpen\AppData\Local\Temp\SPI1249345205753794791l.tmp

Report generated on Mon Aug 29 10:15:14 EDT 2022

Figure index: Figure_1: Channel IL VIEW

Processed file(s):

- C:\Users\acharpen\Downloads\F634387_SERDES_Compliance\F634387_SERDES_Compliance\TestCase/OIFTest.s12p

Results:

- Figure_1 (Channel IL): FAIL
- Figure_2 (Channel RL): FAIL
- Figure_4 (Channel ILD): PASS
- Figure_5 (Channel ICN): PASS ICN: 3.25958 mv, Max: 3.35929 mv, IL(Fb/2) = 20.0905 dB

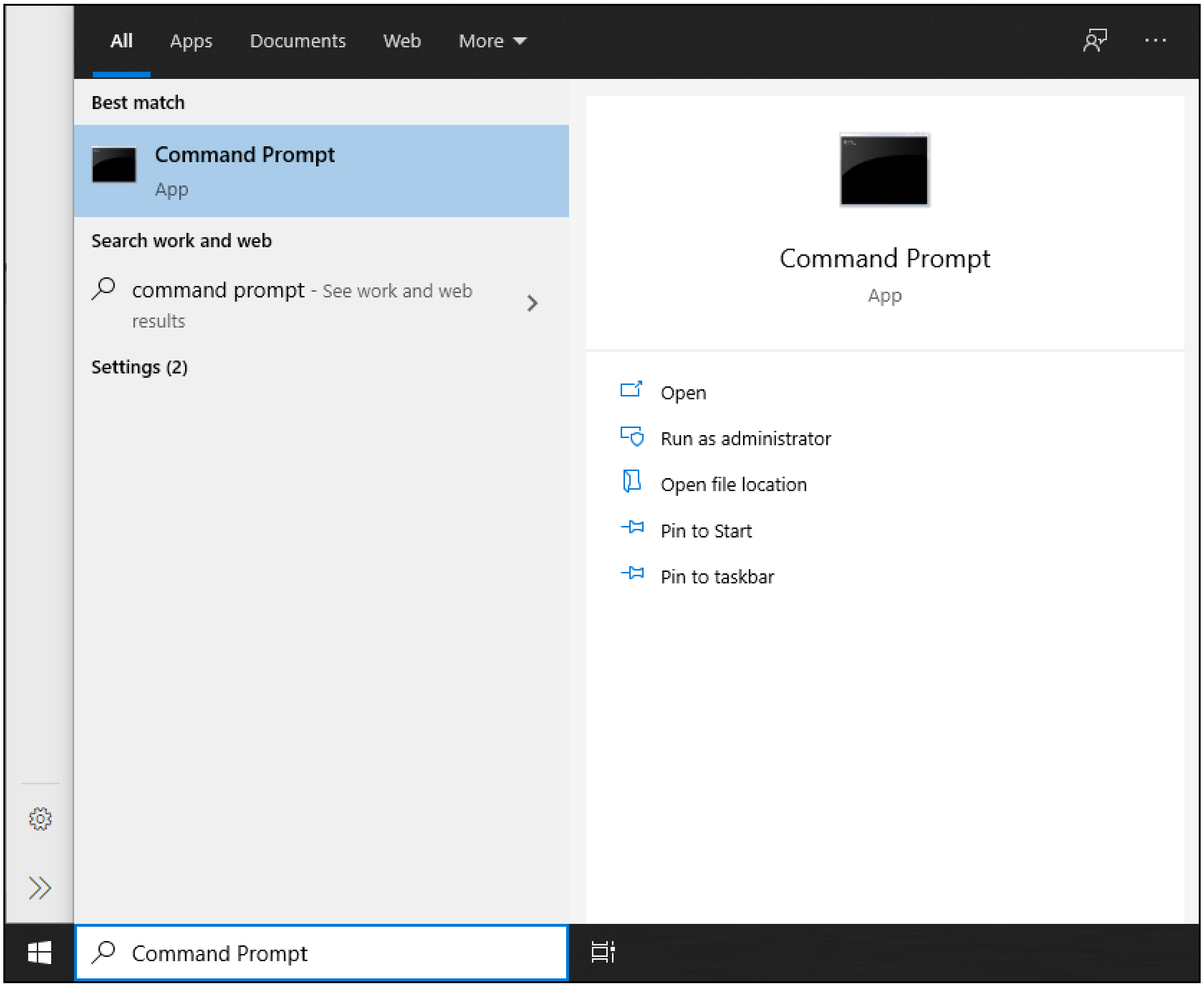
Figure_1:

The analysis can be further customized by changing the configuration settings (e.g., exporting a custom *.cfg* file and editing it in a text editor) or adding further parameters via the command line. If necessary, enter the following command for a complete list of supported commands, then press **Enter**: `"C:\Program Files\AnsysEm\v242\Win64\spisim\spisim\modules\ext\SPISimJNI_WIN64.exe" --list`.

Generate a Compliance Analysis Report Using a Batch Command Line to Invoke the User Interface

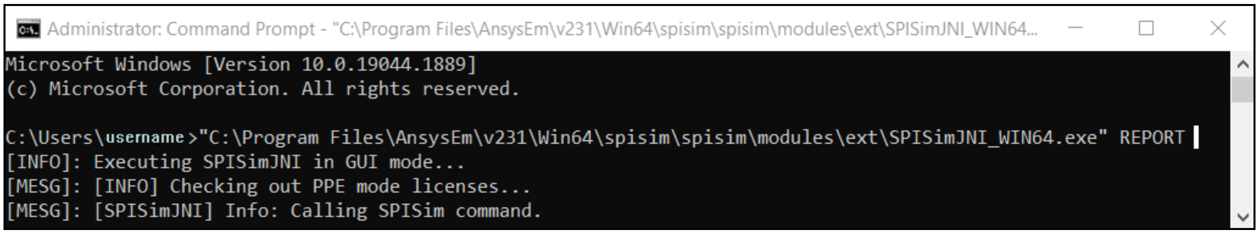
Complete the following steps to run a compliance analysis report using a batch command line to open the **SPISim** user interface.

1. Run a command prompt (e.g., in a Windows environment, type *Command Prompt* into the **Search Bar** and press **Enter** to open a **Command Prompt** window).

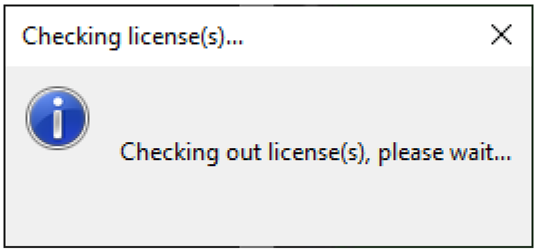


> **Note:** The following example assumes **Electronics Desktop** is installed in the default directory in a Windows operating system environment. Structure the command appropriately for a different directory, path, and/or operating system, as necessary. Do **not** move the ***SPISimJNI_WIN64.exe*** executable file. The executable file relies on the relative path to the **Electronics Desktop** installation folder.

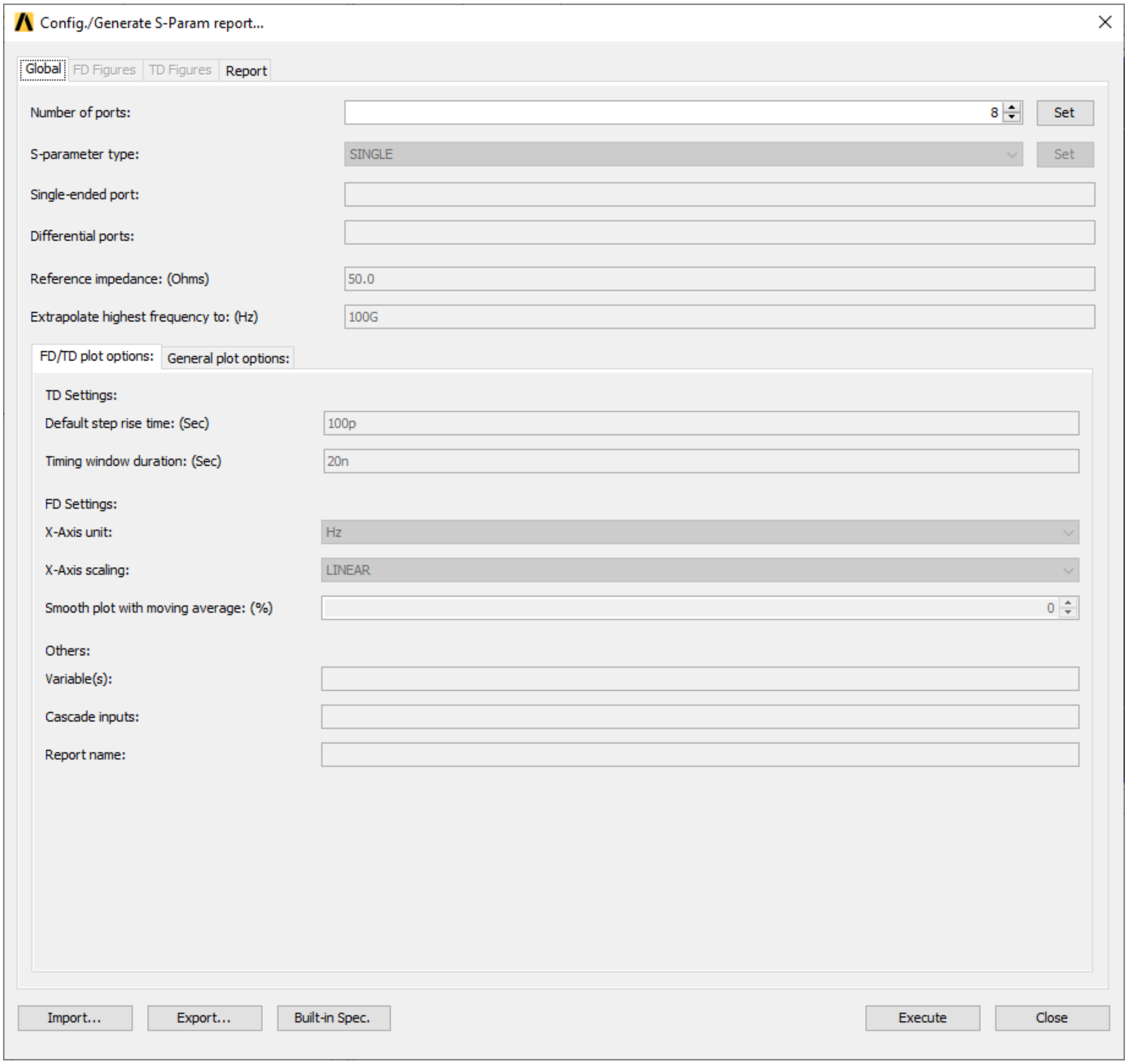
2. From the **Command Prompt** window, enter the following command, then press **Enter**:
```
"C:\Program
Files\AnsysEm\v242\Win64\spisim\spisim\modules\ext\SPISimJNI_
WIN64.exe" REPORT
```



A **Checking License(s) ...** window opens.



Once the software license is verified, the **Config./Generate S-Param report...** window opens.

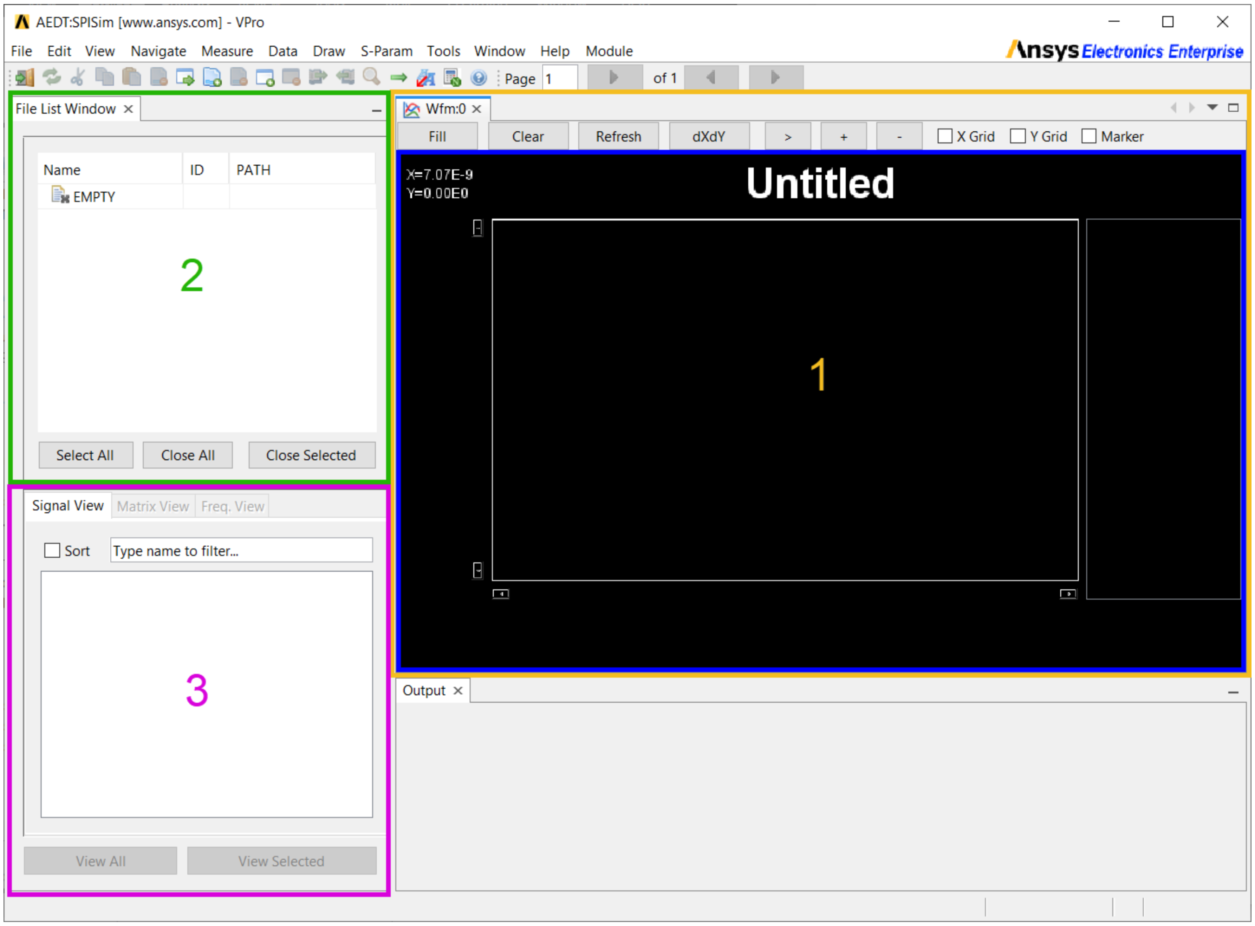


3. Complete steps 2-12 in Generate a Compliance Analysis Report.

SPISim VPro Overview

SPISim VPro allows its users to view and analyze waveforms in both time and frequency domains. VPro offers signal integrity focused measurements, data markers, waveform calculated, and script processing.

Waveform Viewing Interface and Navigation

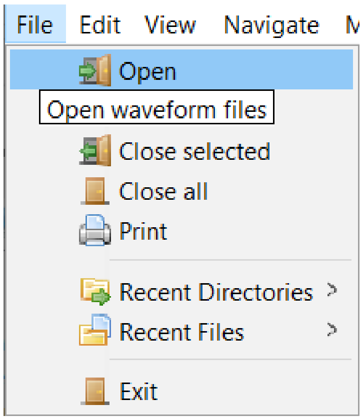


1. Waveform viewing window- open more waveform viewer windows with **Ctrl+t** or click **Window> Waveform Tab**
2. File List Window- main file index.
3. Probe and Signal Index - the probe and signal index window will automatically populate according to the selected file type. It contains different views: **Signal View**, **Matrix View**, and **Freq. View**. The available views are dependent on the file type selected.

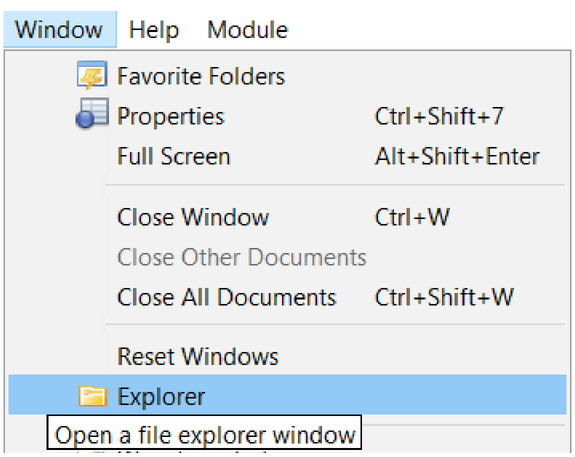
Open Waveform Files:

To open a waveform, use any of the following four methods:

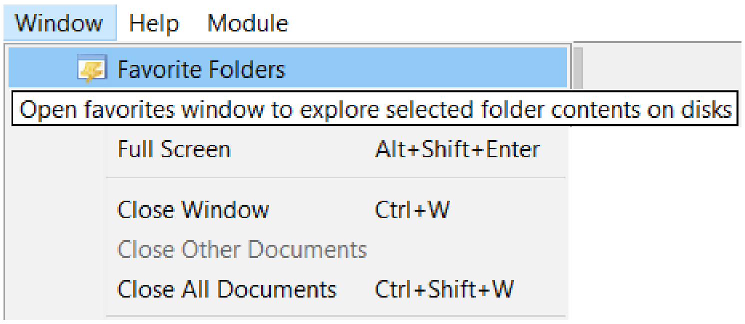
1. From the main ribbon, click **File> Open**, navigate to the folder containing the files. Select the files to open. Different file types will automatically be placed in separate file folders in the **File List Window**.



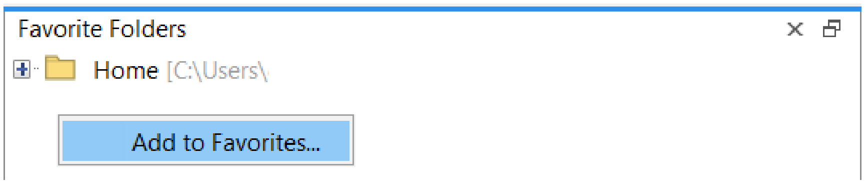
2. Use the operating system's **File Explorer** and drag-and-drop selected files into the **VPro File List Window**.
3. From the main ribbon, select **Window>Explorer** to open VPro's file browser. Check the **Waveform Only** box to view only the waveform files within a folder. Alternatively, in the VPro file explorer, click the **Launch File Manager** button to open the operating system's **File Explorer**, then drag-and-drop selected file to the **File List Window**.



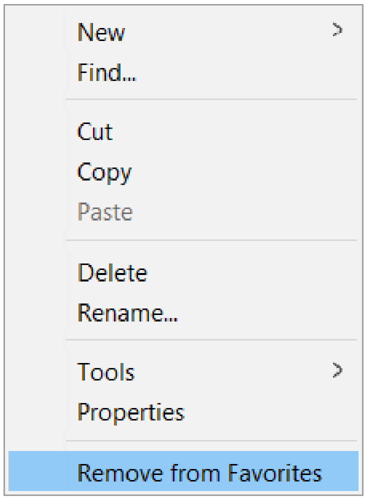
4. Use the **Favorites Folder**. From the main ribbon, select **Window> Favorites Folder** to open the **Favorites Folder**.



Right-click > Add to Favorites to open the application explorer and select the folder which contains the waveform data. The folders added to the **Favorites Folder** will remain listed through all sessions until it is removed from Favorites.



To remove the folder from Favorites, **right-click** the folder, select **Remove from Favorites**.

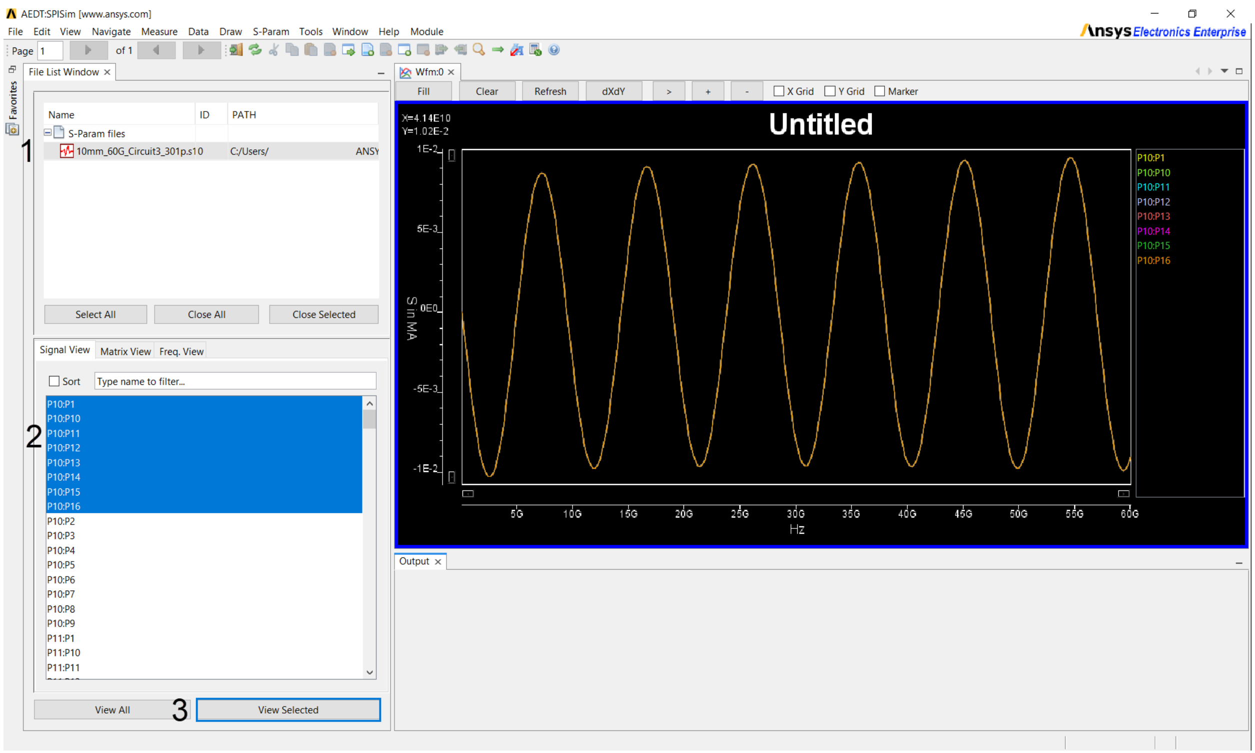


Important: If **Delete** is selected, the data will be removed from the disc.

SPISim VPro Waveform Viewer

Viewing A Waveform

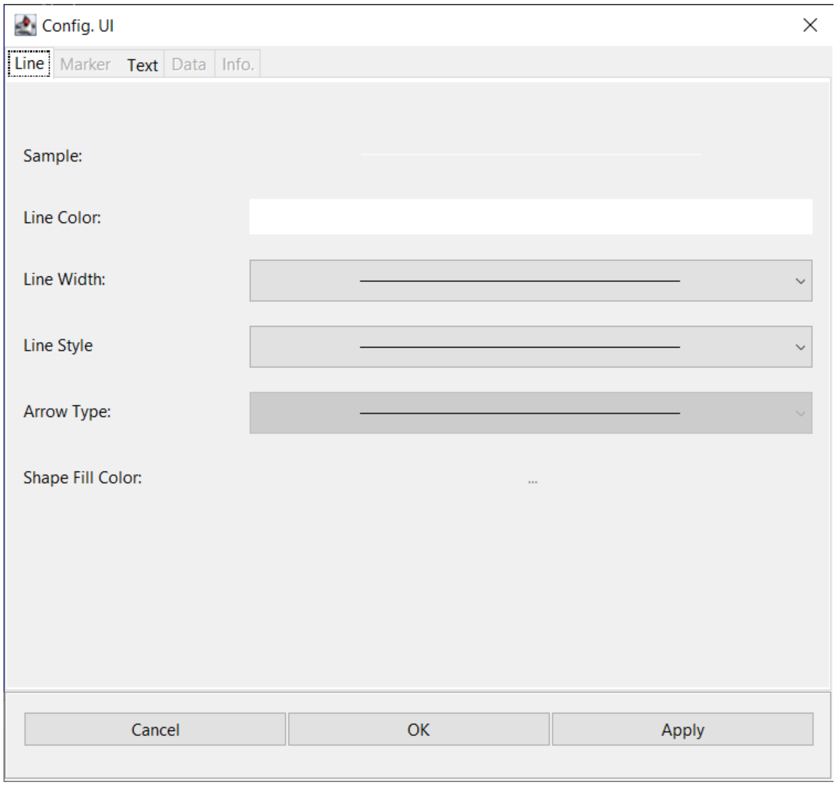
1. Select a waveform file from the **File List Window** to populate the probe.
2. Select the probe of interest.
3. Click the **View Selected** button.

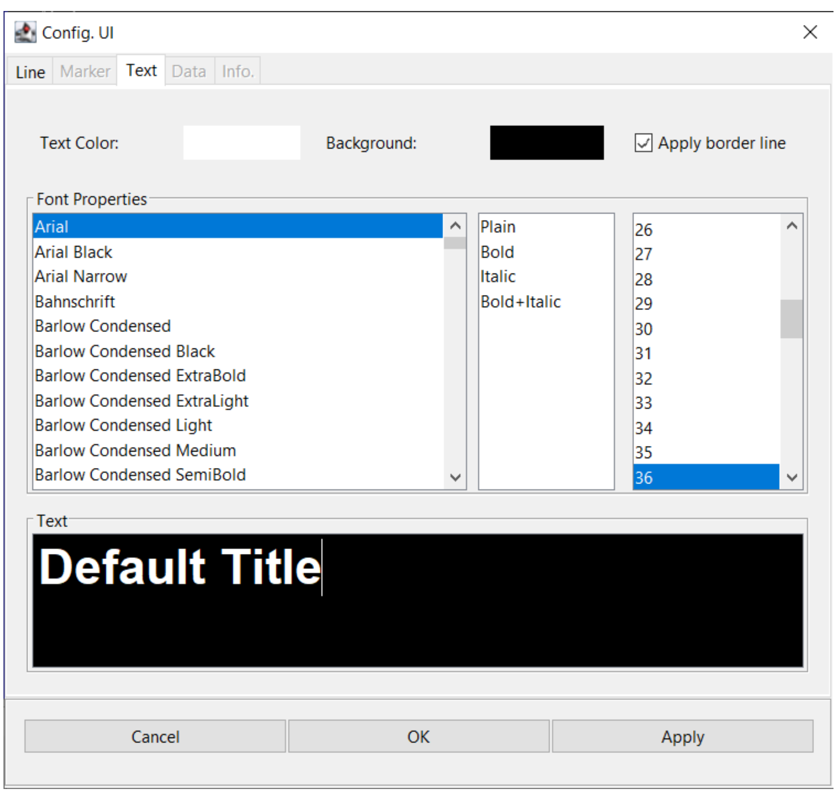


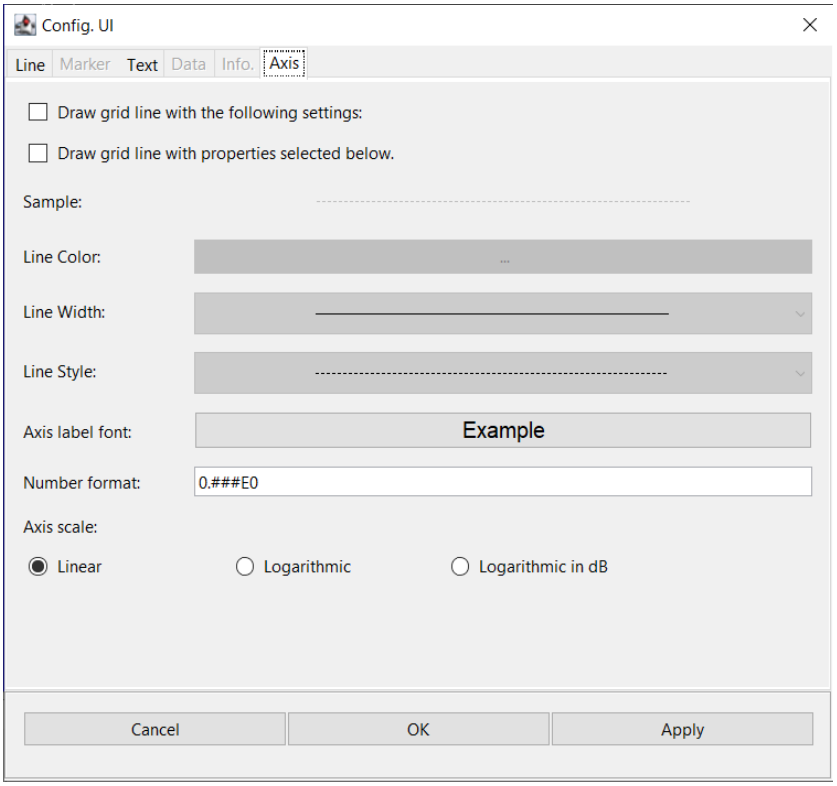
Alternatively, drag-and-drop the selected probes into the waveform viewer pane.

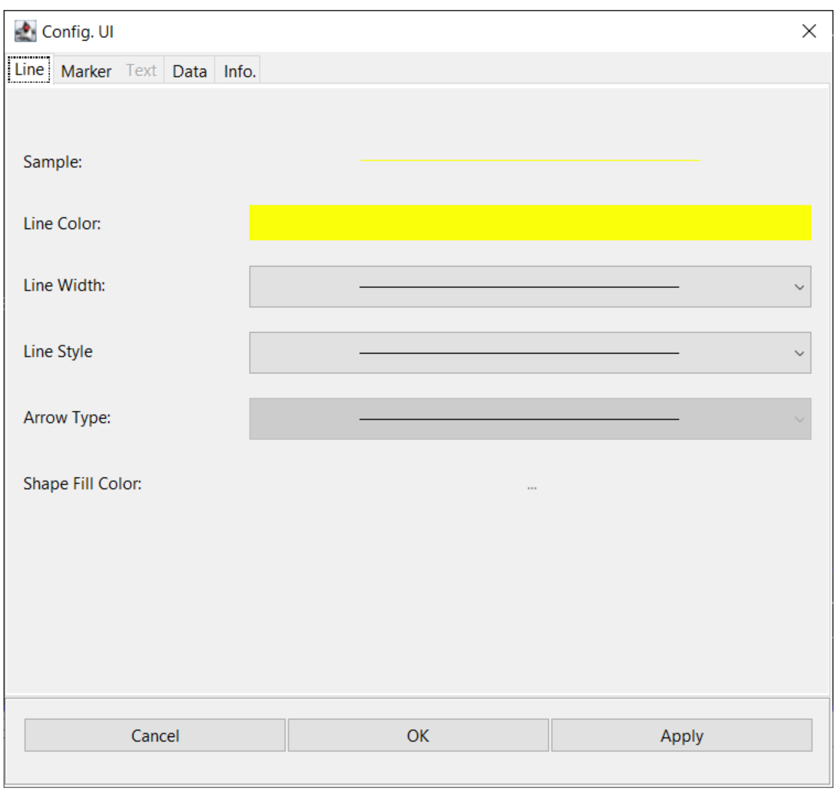
Customize the Waveform Pane

Right-click on the default title, axis, or curve to open the **Config UI** dialog. It will display the related adjustable settings.


Config. UI
Line
Marker
Text
Data
Info.
Sample:
Line Color:
Line Width:
Line Style
Arrow Type:
Shape Fill Color:
Cancel
OK
Apply

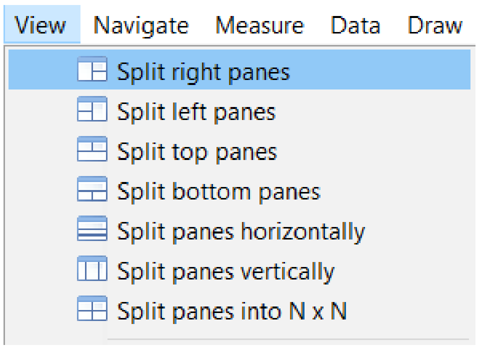

Config. UI
Line
Marker
Text
Data
Info.
Text Color:
Background:
Apply border line
Font Properties
Arial
Arial Black
Arial Narrow
Bahnschrift
Barlow Condensed
Barlow Condensed Black
Barlow Condensed ExtraBold
Barlow Condensed ExtraLight
Barlow Condensed Light
Barlow Condensed Medium
Barlow Condensed SemiBold
Plain
Bold
Italic
Bold+Italic
26
27
28
29
30
31
32
33
34
35
36
Text
Default Title
Cancel
OK
Apply


Config. UI
Line
Marker
Text
Data
Info.
Axis
Draw grid line with the following settings:
Draw grid line with properties selected below.
Sample:
Line Color:
Line Width:
Line Style:
Axis label font:
Example
Number format:
0.###E0
Axis scale:
Linear
Logarithmic
Logarithmic in dB
Cancel
OK
Apply

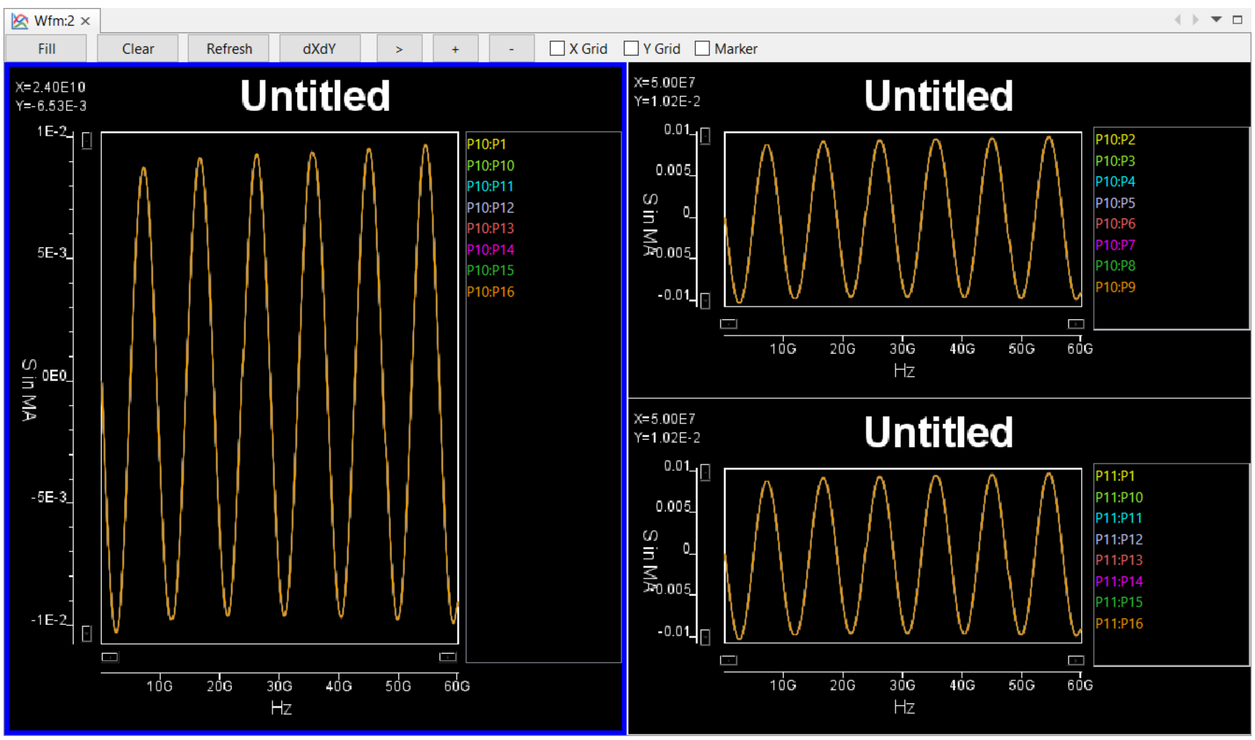

Config. UI
Line
Marker
Text
Data
Info.
Sample:
Line Color:
Line Width:
Line Style
Arrow Type:
Shape Fill Color:
Cancel
OK
Apply

Synchronize Panes

View multiple panes in a single tab at the same time. From the main ribbon, click **View** to see the pane viewing options.

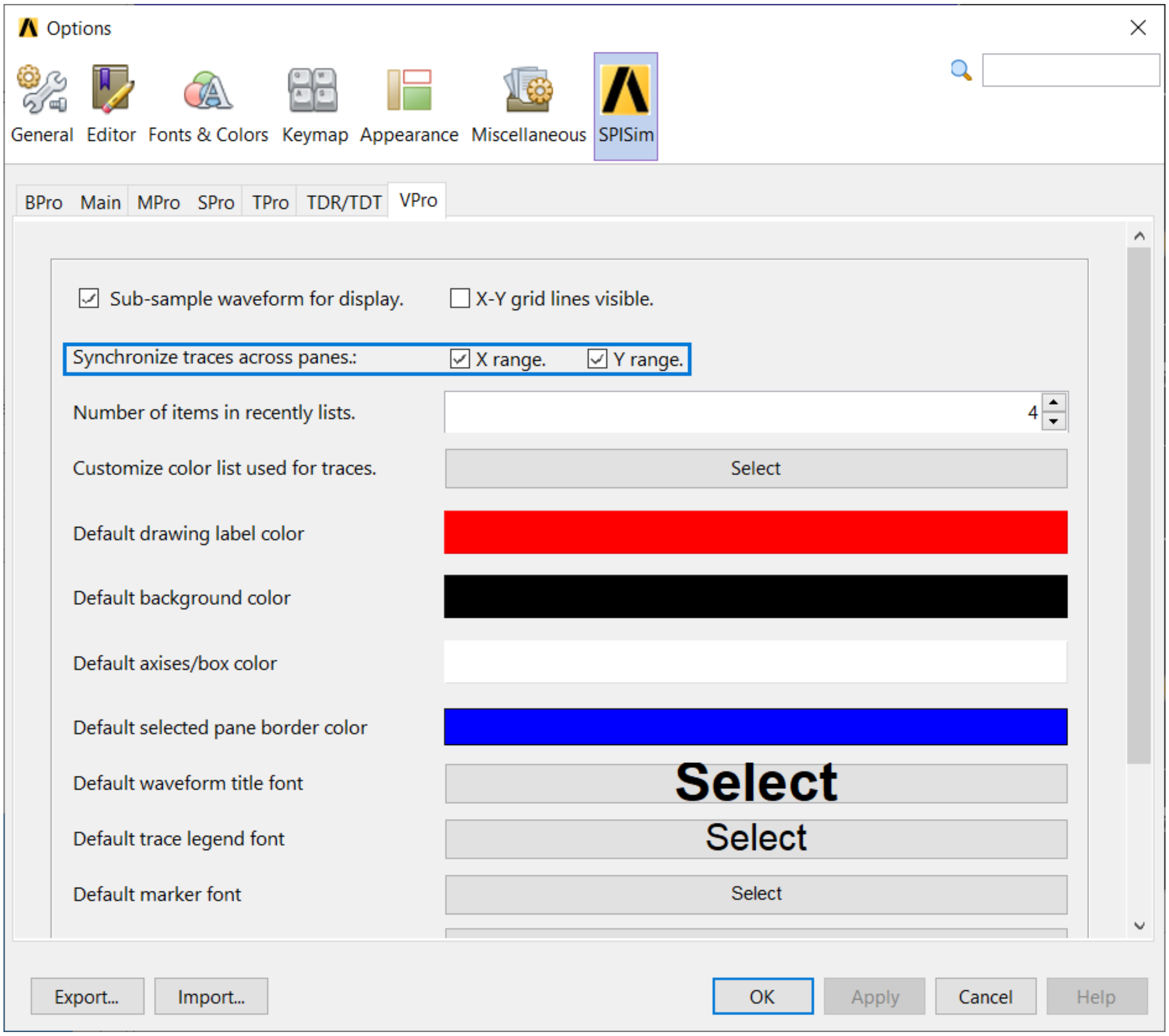


The current working pane will be highlighted in blue.

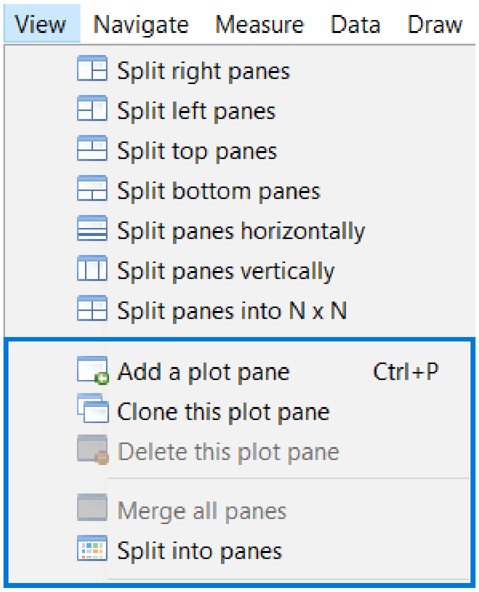


Synchronize trace across different panes. For example, if you zoom-in on one waveform pane, all other synchronized panes will be zoomed-in. Click **Tools>Options** then **VPro** to see synchronization X and Y options.

Data can be dragged and dropped between panes.

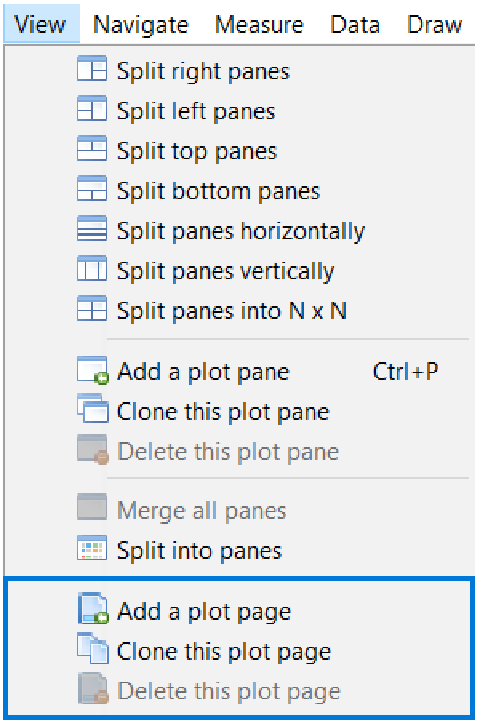


Additional Pane Options

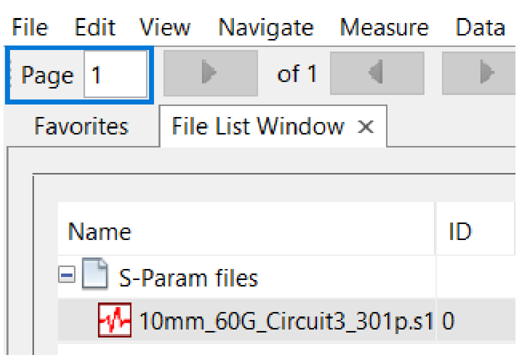


Option	Description
Add plot pane	Adds a plot pane in the current working tab.
Clone the plot pane	Using the signal from current working pane, opens a new pane with the same data.
Delete this plot pane	Closes the current working pane.
Merge all panes	Merges the data in the open panes and displays them in a single pane.
Split into panes	Splits the signal into different panes.

Pages



A tab can also contain multiple pages. The additional pages may not be immediately visible but underneath the displayed panes. Use the **Page** option to toggle between the pages.



Comparing Waveforms

Select the desired waveform files in the **File List Window**.

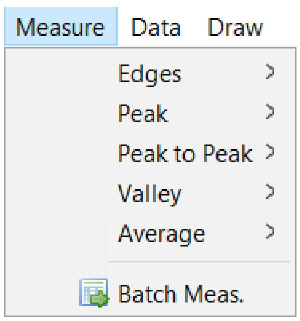
In the **Signal View Window**, only the common signal from the selected waveforms will be displayed.

Select the signal to view. The common signal from all the waveforms will be displayed.

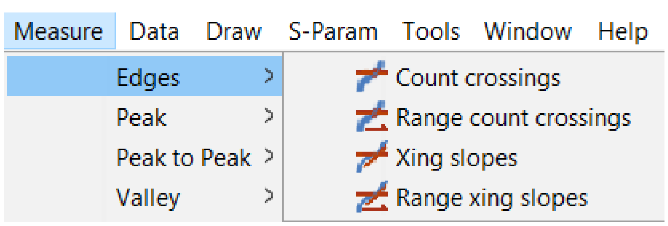
SPISim VPro Measurements

Measurements

From the main ribbon, select **Measure** to view the available calculations.

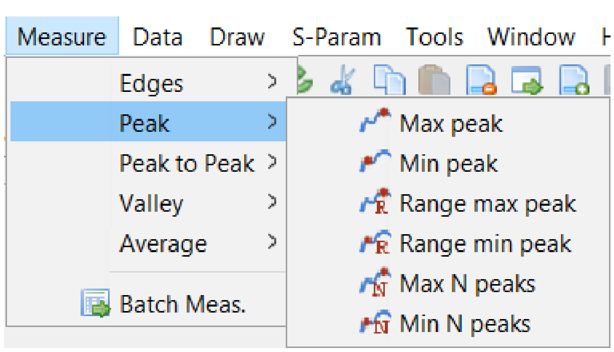


Edges



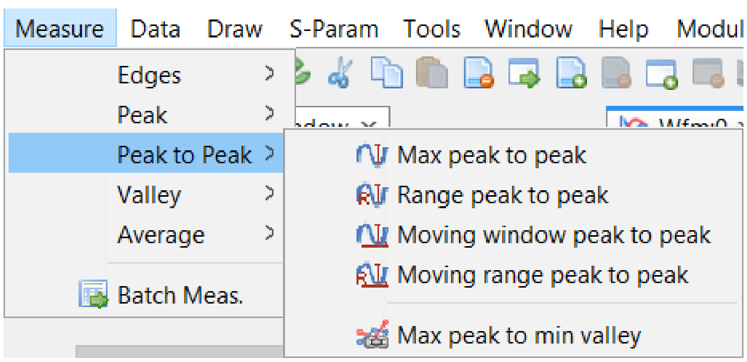
- Count Crossings- detect selected rise or fall (or both) crossings.
- Range count crossings- detect rise or fall crossings in a selected range.
- Xing slopes- calculate rise or fall slopes at crossings.
- Range xing slopes- calculate crossing slops in selected range.

Peak



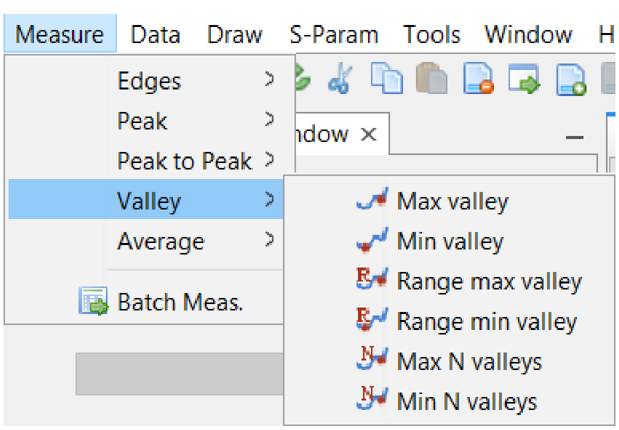
- Max peak- identify the maximum peak.
- Min peak- identify the minimum peak.
- Range max peak- identify the maximum peak within a given range.
- Range min peak- identify the minimum peak within a given range.
- Max N peaks- identify N number of maximum peaks.
- Min N peaks- identify N number of minimum peaks.

Peak to Peak



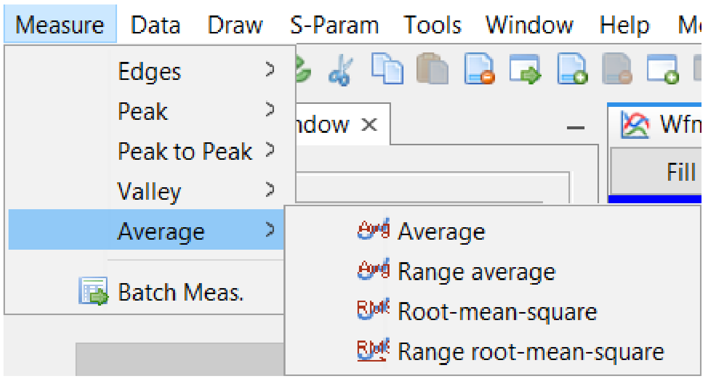
- Max peak to peak - mark peak to peak value.
- Range peak to peak - mark peak to peak in a selected range.
- Moving window peak to peak - report sliding window based peak to peak.
- Moving rage peak to peak - select range to do sliding window peak to peak.
- Max peak to min valley - peak to peak based with slope consideration.

Valley



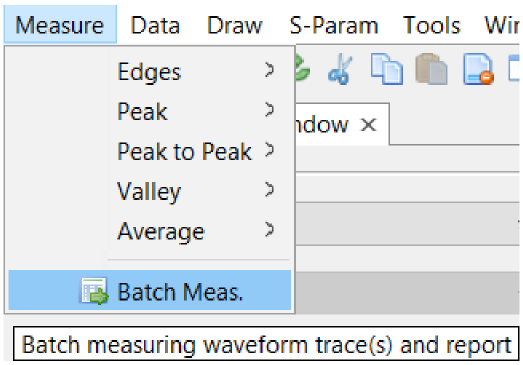
- Max valley - identify max valley.
- Min valley - identify min valley.
- Range max valley - identify max valley within a selected range.
- Range min valley - identify min valley within a selected range.
- Max N valleys - identify N number of max valleys.
- Min N valleys - identify N number of min valleys.

Average



- Average - report average value of selected traces.
- Range average - report average value in selected range.
- Root-mean-square - report RMS value of selected traces.
- Range root-mean-square - report RMS value of traces in a selected range.

Batch Measure



Launches the **Batch Measurement and Report** window. Provides batch measurements for waveform traces and reports as a csv file.

This function works in two modes:

1. Measure all traces within one waveform file, or
2. Measure common probes across different waveform file.

Add one or more measurements at specified time point/ranges with "," as the delimiter.

Select cells and apply with same value using the **Set Cells** button.

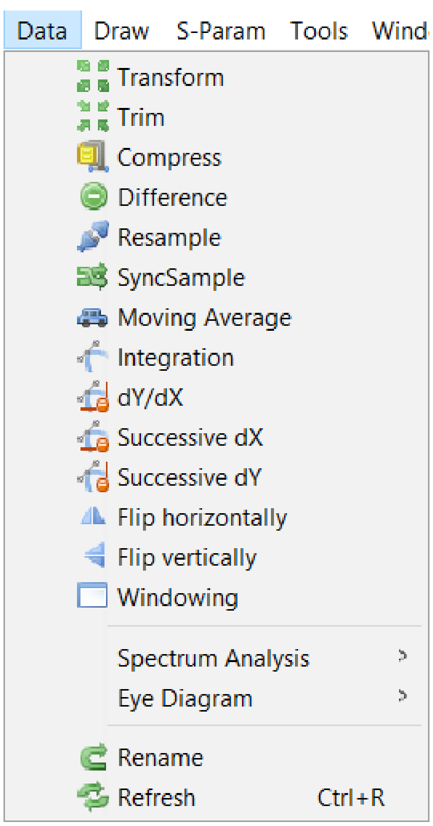
Remove unwanted measurements using "-" button.

Settings may be exported for later use.

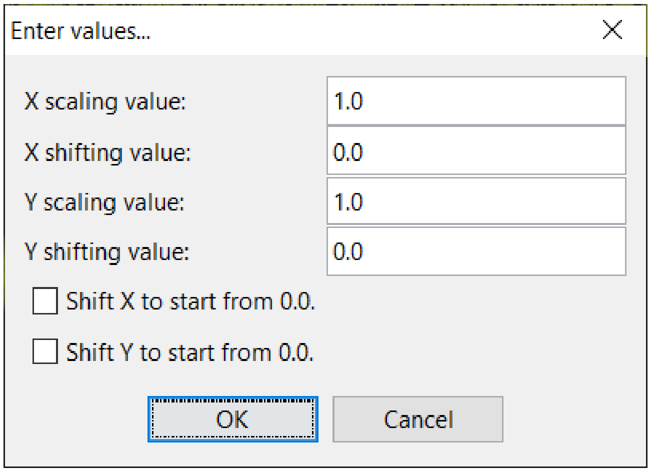
SPISim VPro Data

Data Processing

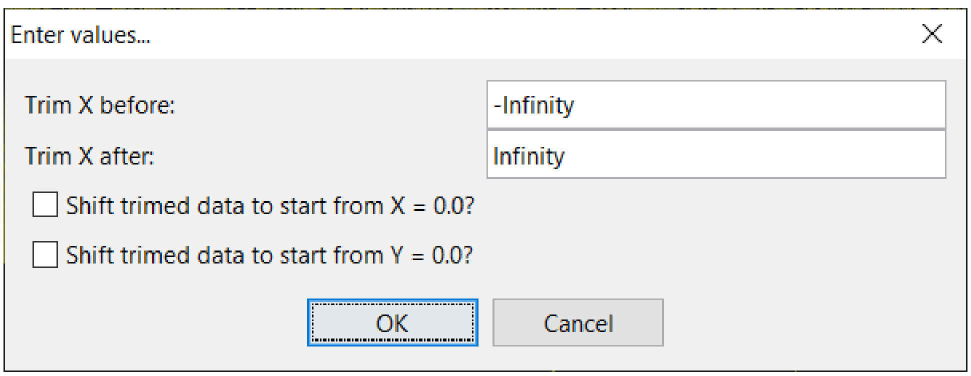
From the main ribbon, select **Data** to view the available options.



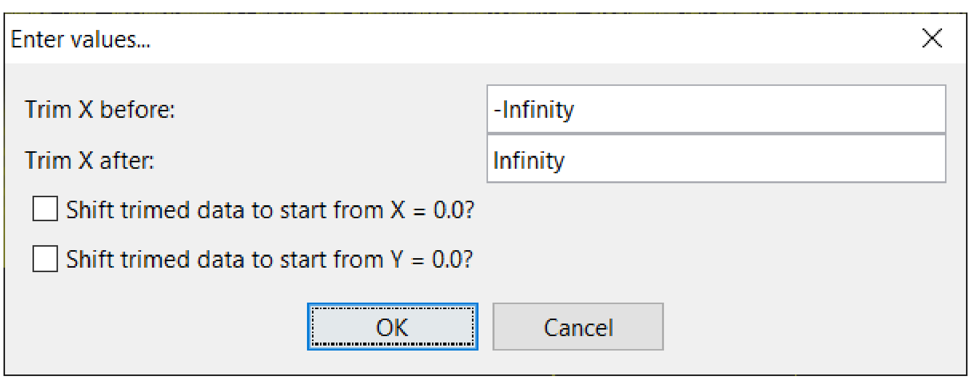
Transform: Scale/Shift selected traces.



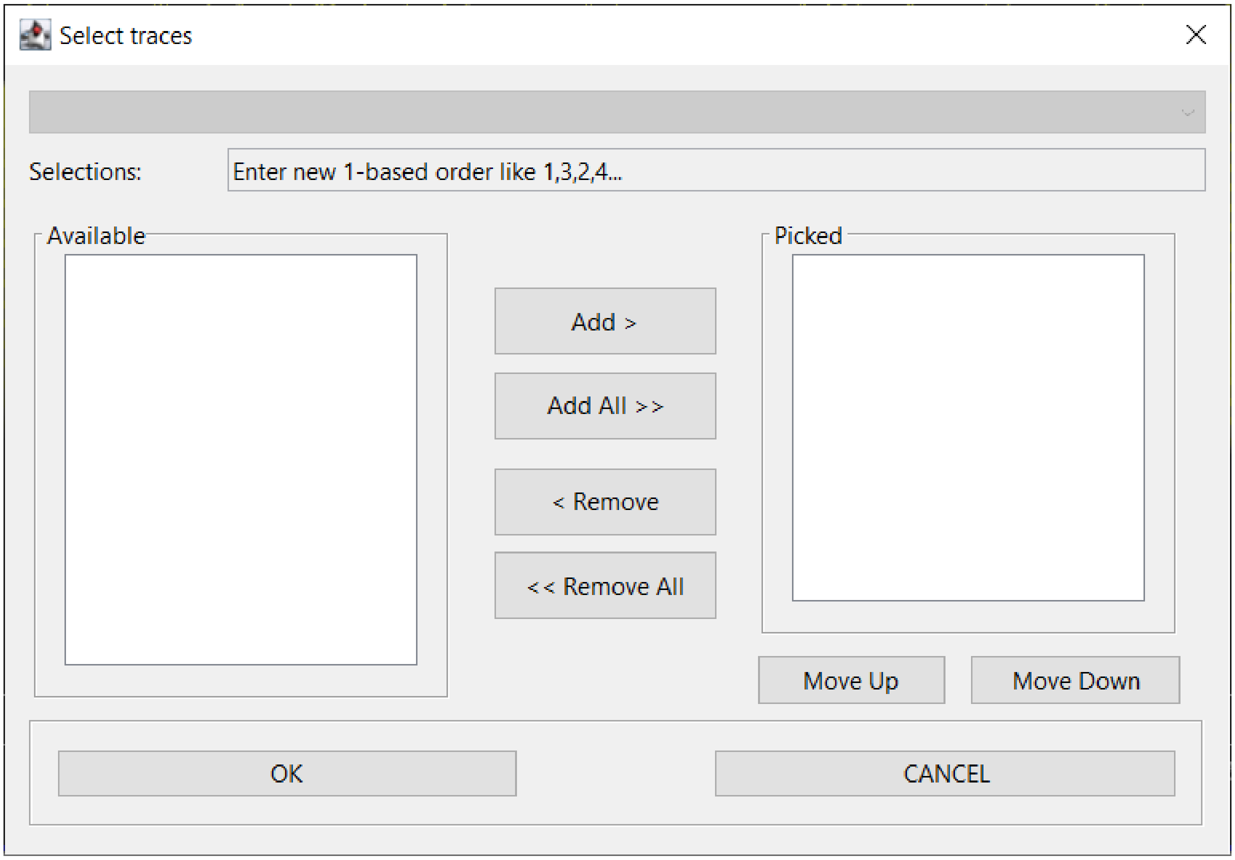
Trim: Trim selected traces.



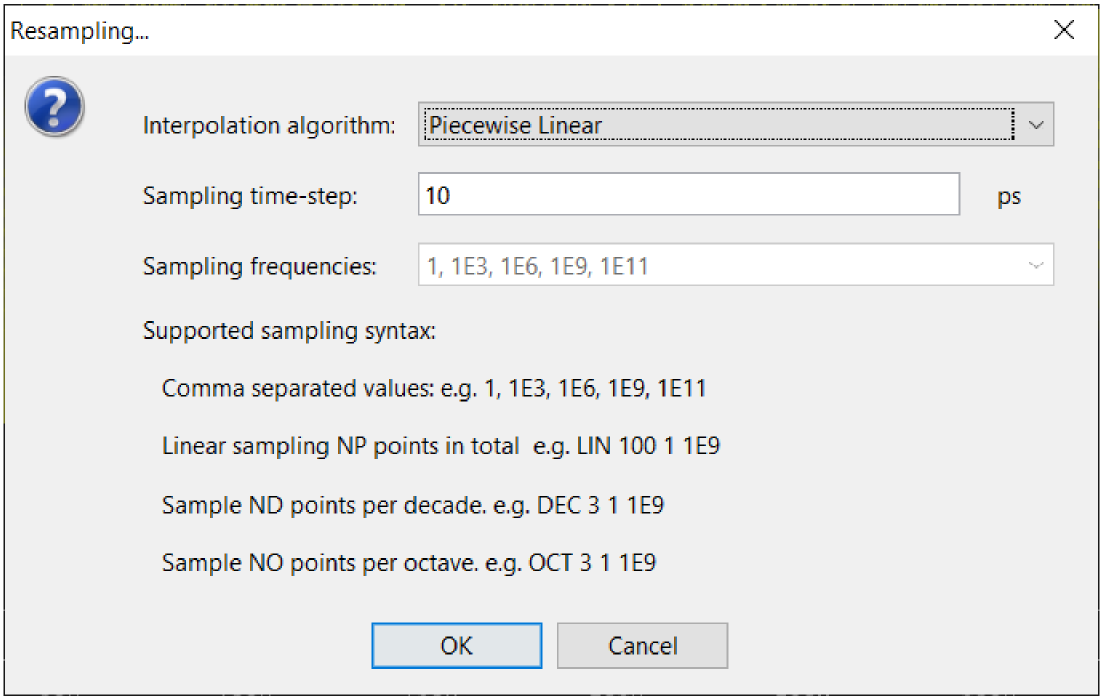
Compress: Compress data points of selected traces.



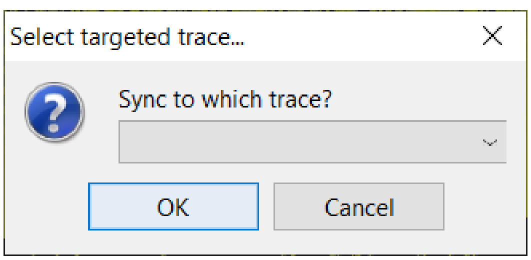
Difference: Select traces to do difference comparison.



Resample: Resample data for selected traces.



SyncSample: Synchronize sampling points of selected traces.



Moving Average: Calculate moving average to smooth data.

Integration: Do integration for selected traces.

dY/dX: Calculate dY/dX for selected traces.

Successive dX: Calculate successive dX for traces.

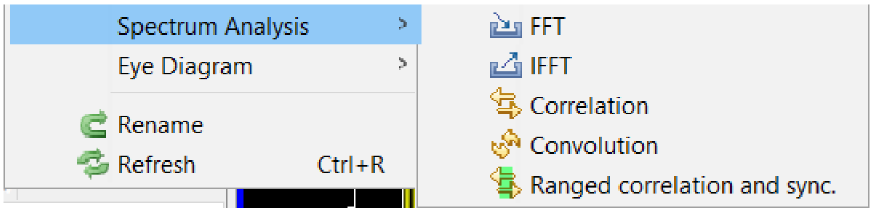
Successive dY: Calculate successive dY for traces.

Flip horizontally: Flip trace between left and right.

Flip vertically: Flip trace between top and bottom.

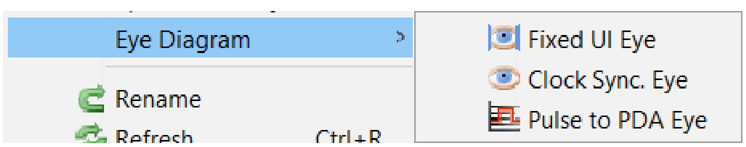
Windowing: Windowing functions available for frequency domain.

Spectrum Analysis:



- **FFT**: Perform fast Fourier transform.
- **IFFT**: Perform Inverse fast Fourier transform.
- **Correlation**: Calculate cros correlations between traces.
- **Convolution**: Convolve selected traces.
- **Ranged correlation and sync.**:Calculate correlation within selected range.

Eye Diagram:



- **Fixed UI Eye**: Do fixed UI eye plot.
- **Clock Sync Eye**: Do clock synchronous eye plot.
- **Pulse to PDA Eye**: Do peak distortion analysis (worst case) eye plot.

Rename: Rename trace label.

Refresh: Reread original data from disk.

Calculating Effective Return Loss

Follow the instructions on the following pages to calculate the effective return loss (ERL) of .sN*p* files on the **SPISim: VPro Analysis Module**.

" Calculating ERL From the SPISim User Interface" on the facing page

"Creating an ERL Configuration File" on page 15-114

" Importing an ERL Configuration File" on page 15-115

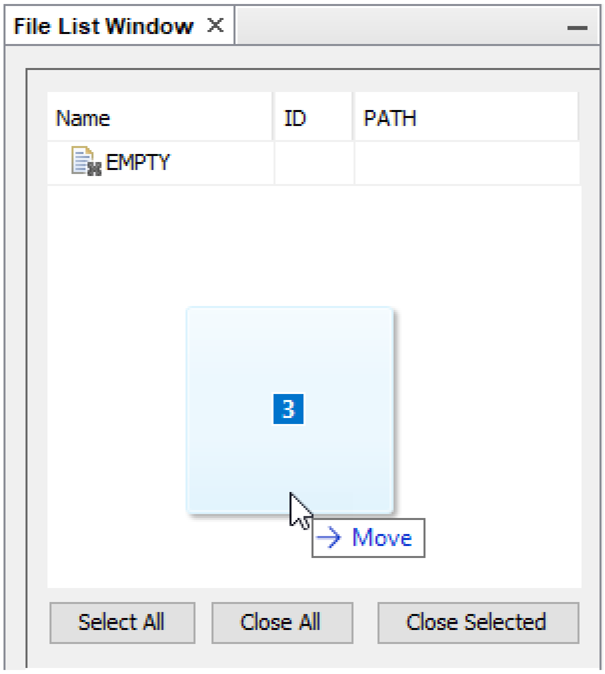
"Overriding ERL Configuration File Parameters" on page 15-131

"Generate an ERL Calculation Report with a User Defined Solution" on page 15-133

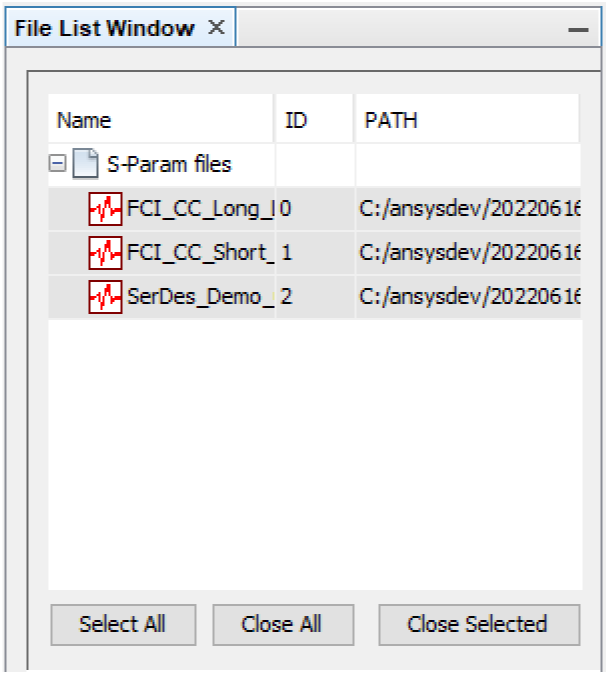
Calculating ERL From the SPISim User Interface

Follow these steps to calculate the effective return loss (ERL) of one or several *.s4p* files.

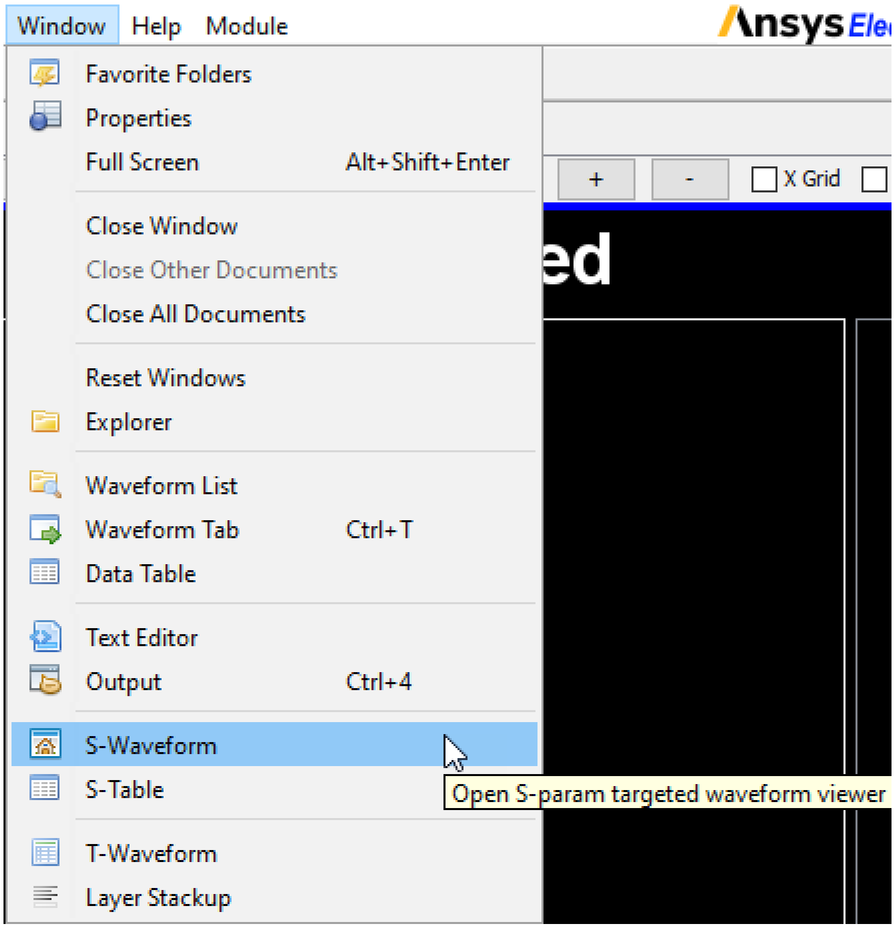
1. **Drag+drop** the relevant *.s4p* file(s) into the **File List Window** to populate the **Signal**, **Matrix**, and **Freq. View** tabs.



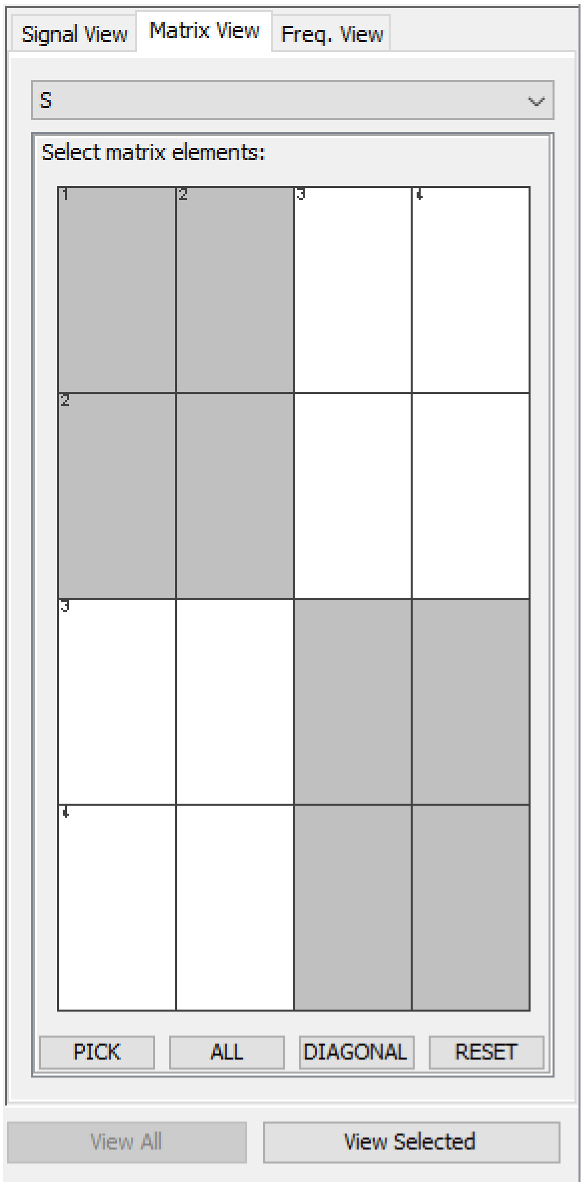
2. From the **File List Window**, click **Select All**.



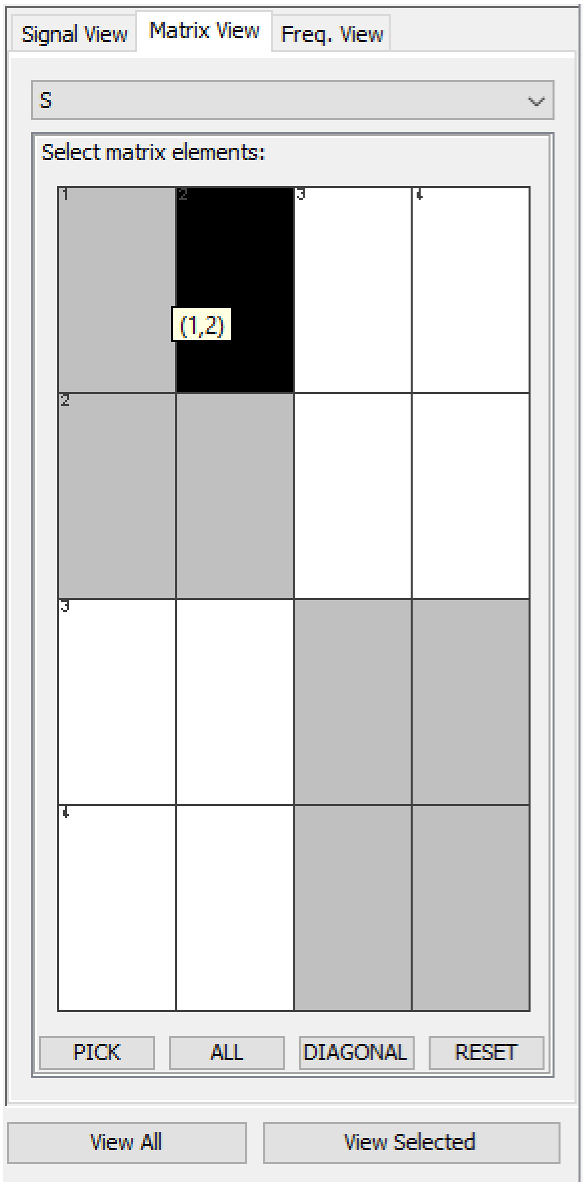
3. If necessary, navigate to **Window** > **S-Waveform** to open an S-waveform output view tab (i.e., **S-Wfm:1** in step 5).


Window
Help
Module
Ansys Ele
Favorite Folders
Properties
Full Screen
Alt+Shift+Enter
+
-
X Grid
Close Window
Close Other Documents
Close All Documents
ed
Reset Windows
Explorer
Waveform List
Waveform Tab
Ctrl+T
Data Table
Text Editor
Output
Ctrl+4
S-Waveform
S-Table
Open S-param targeted waveform viewer
T-Waveform
Layer Stackup

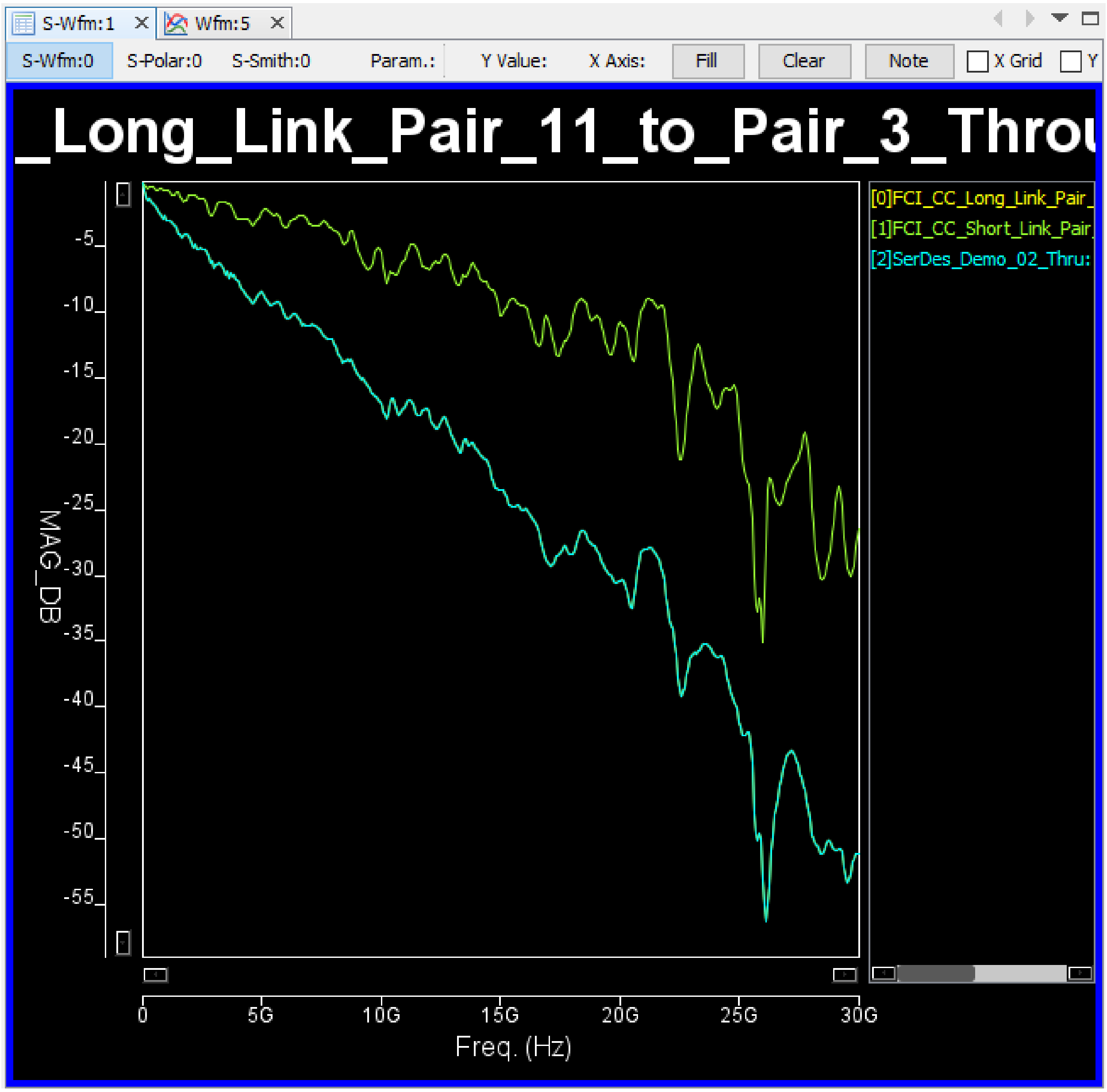
4. Navigate to the **Matrix View** tab.



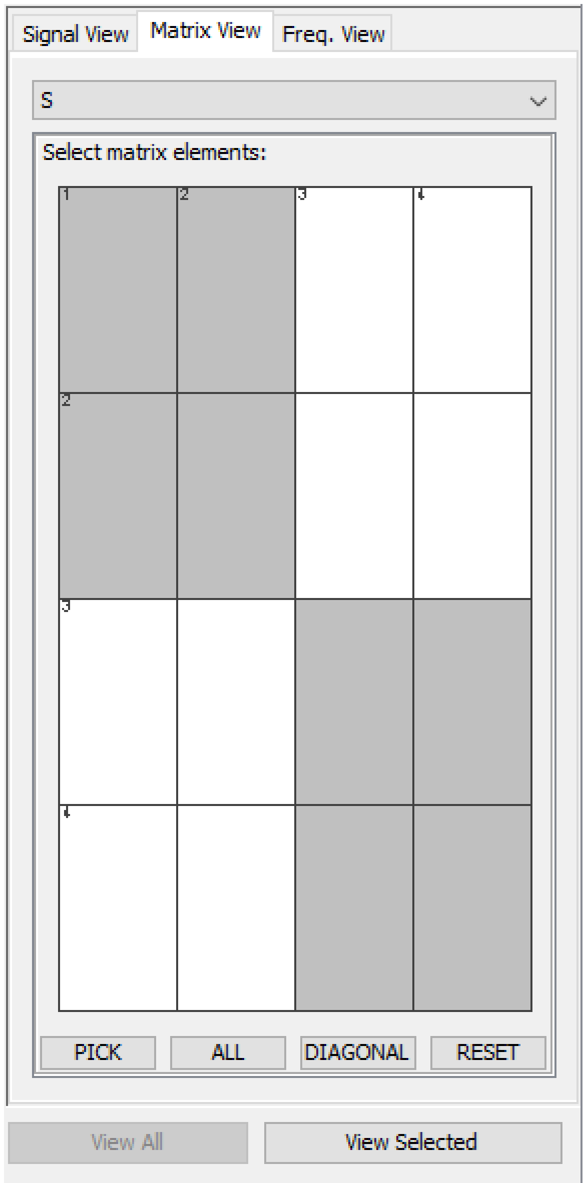
To verify port order, select a port (i.e., a rectangle in the matrix). Each rectangle is assigned a matrix coordinate, indicating the port it represents (e.g., according to both the X,Y coordinate grid and the pop-up **Tool Tip**, the selected rectangle in the following screenshot is port 1,2).



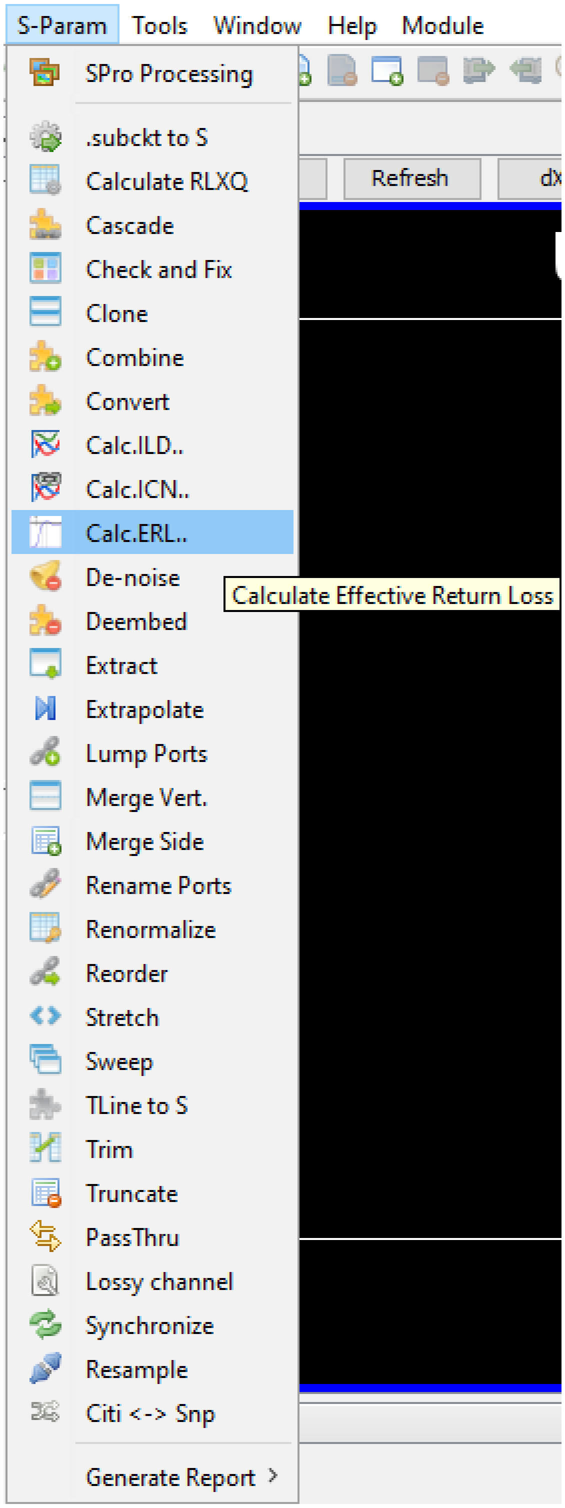
5. Click **View Selected** to view the selected port in the waveform view window (i.e., **S-Wfm:1**).



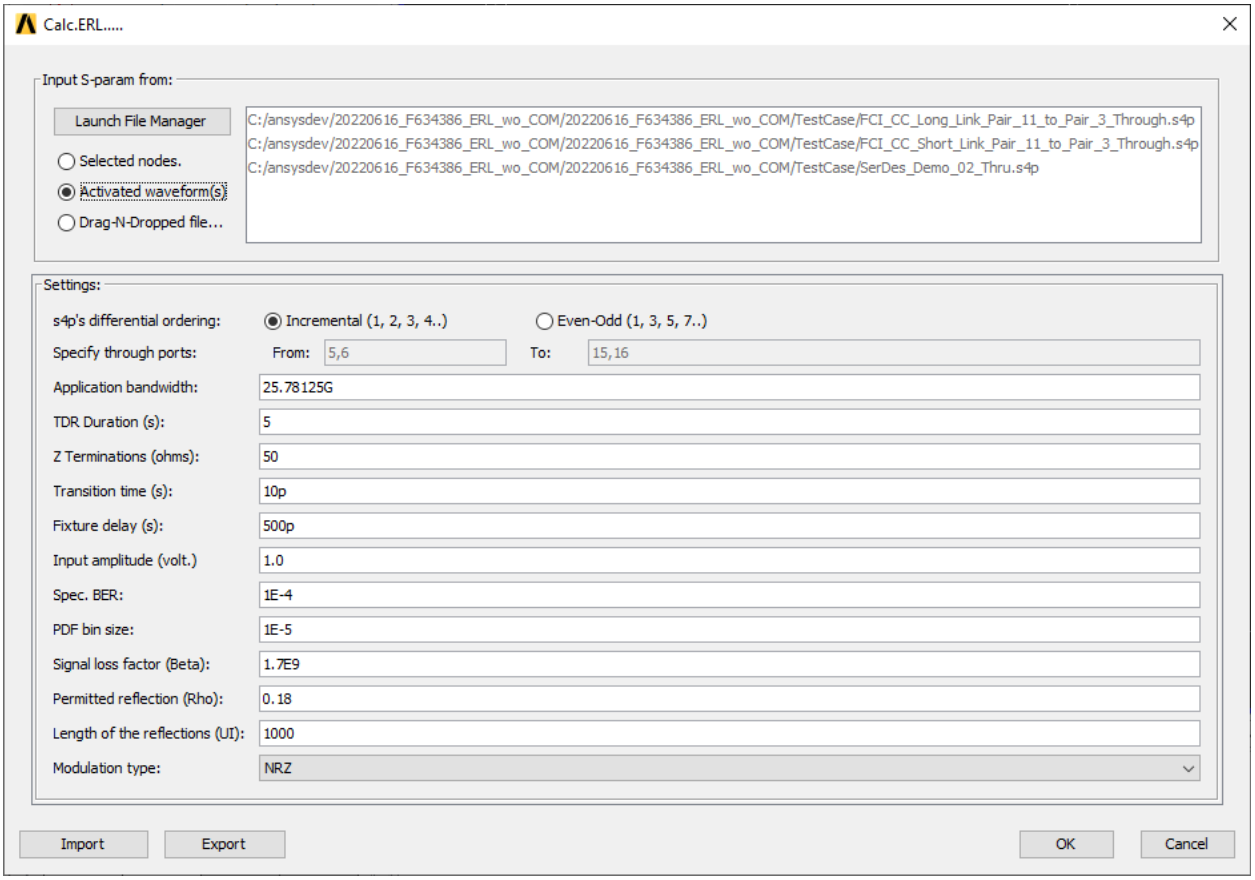
6. Once the port order is identified, select **RESET** on the **Matrix View** tab, to clear the selection.


Signal View
Matrix View
Freq. View
S
Select matrix elements:
1
2
3
4
2
3
4
PICK
ALL
DIAGONAL
RESET
View All
View Selected

From **S-Param**, select **Calc.Erl..** to open the **Calc.ERL.....** window.



The **Input S-param from** group box is pre-populated with the *.s4p* files that were dropped in the **File List Window** in step 1.



7. From the **Settings** group box, select **Even-Odd (1, 3, 5, 7..)** from **s4p's differential ordering**. Then make any other appropriate changes. The **Settings** group box also contains the following configurable parameters:

   Application bandwidth: inverse of one unit interval (UI)

   TDR Duration (s): length time domain reflectometry (TDR) data should be applied, in seconds

   Z Terminations (ohms): termination (Z11 and Z22) when TDR is calculated

   Transition time (s): how fast input pulse transitions from 0 to Vcc voltage (Tr)

   Fixture delay (s): lapse between when input starts transition from 0 to Vcc voltage (Tfx)

   Input amplitude (volt.): Vcc volt of step input

   Spec. BER: threshold at which ERL is calculated

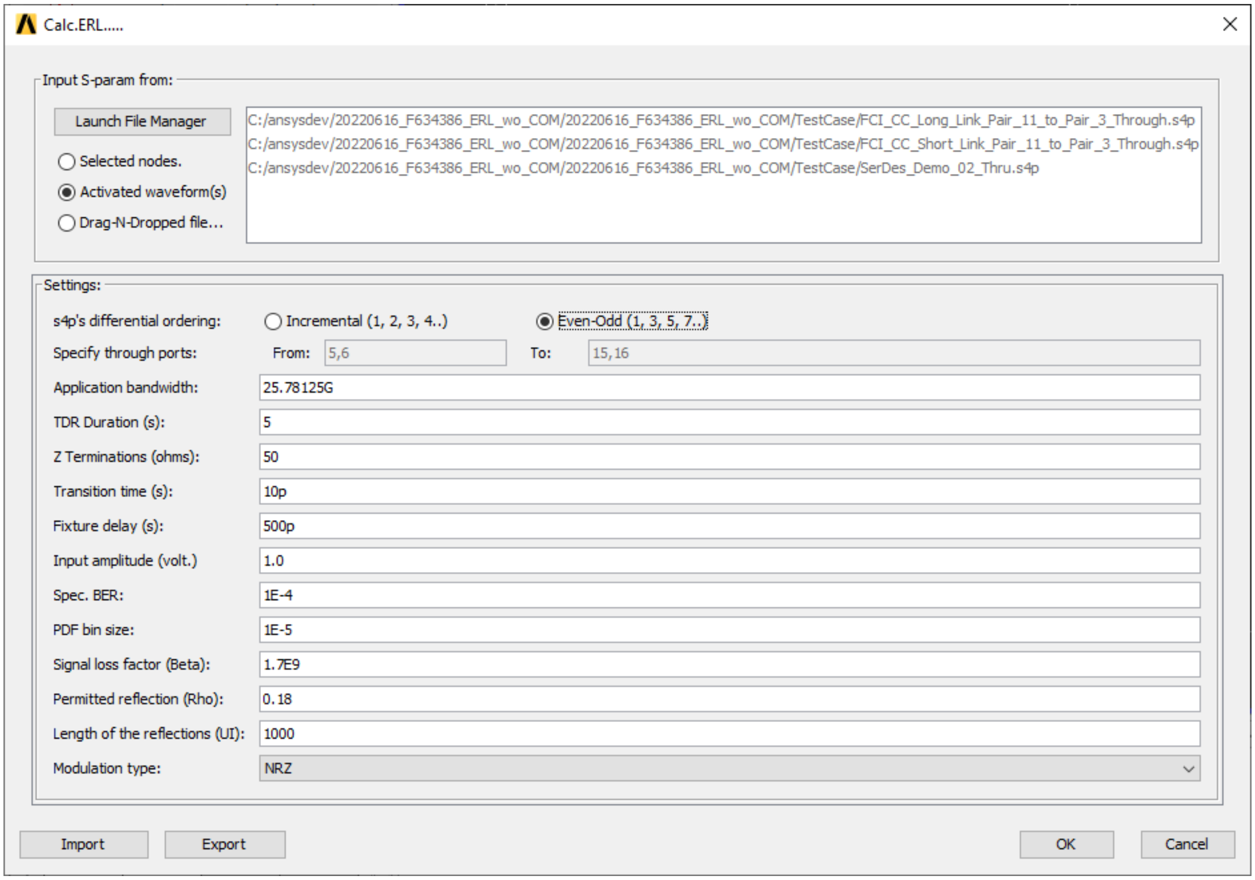
   PDF bin size: superimposed value (i.e., quantized number of bins)

Signal loss factor (Beta): exponent by which the signal loses strength as it propagates

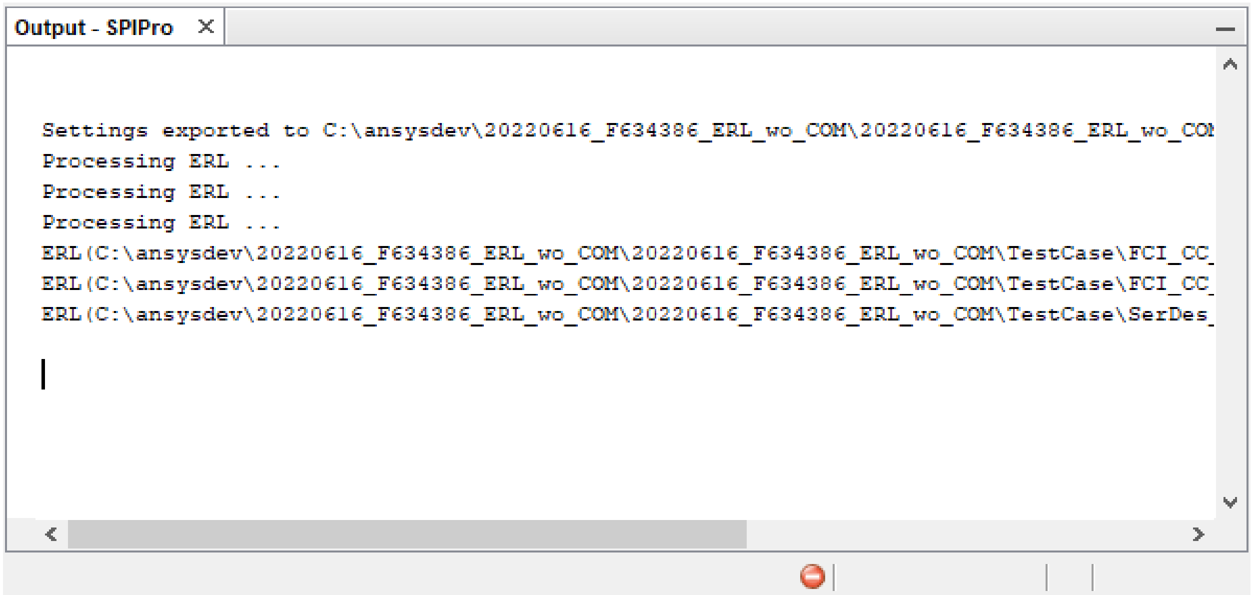
Permitted reflection (Rho): reflective coefficient

Length of the reflections (UI): number of unit intervals used to calculate ERL

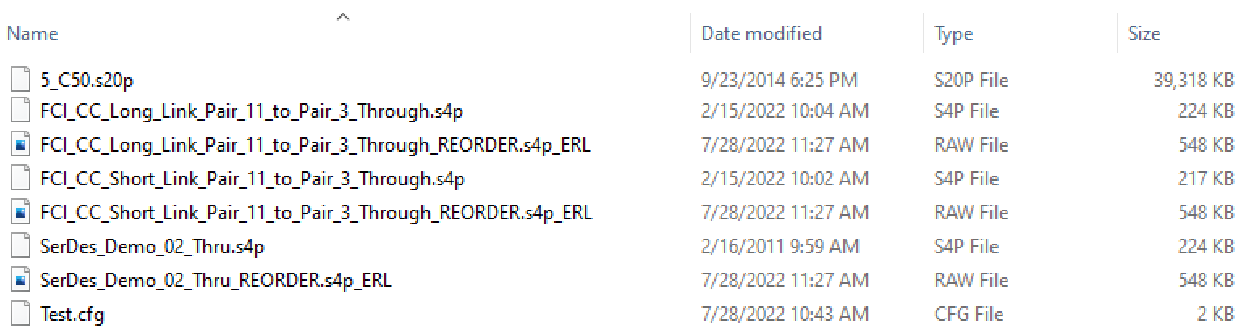
Modulation type: signal modulation type (NRZ or PAM4)



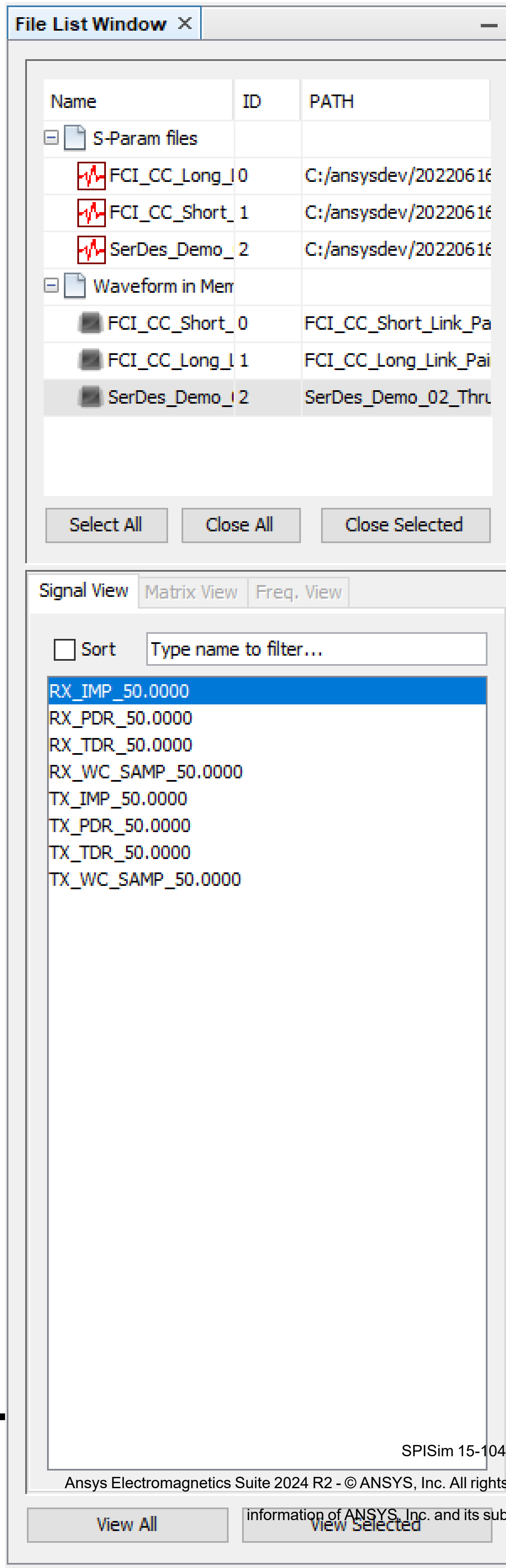
8. Click **OK** to close the **Calc.ERL.....** window and begin ERL calculation. View the process and result in the **Output - SPIPro** window.



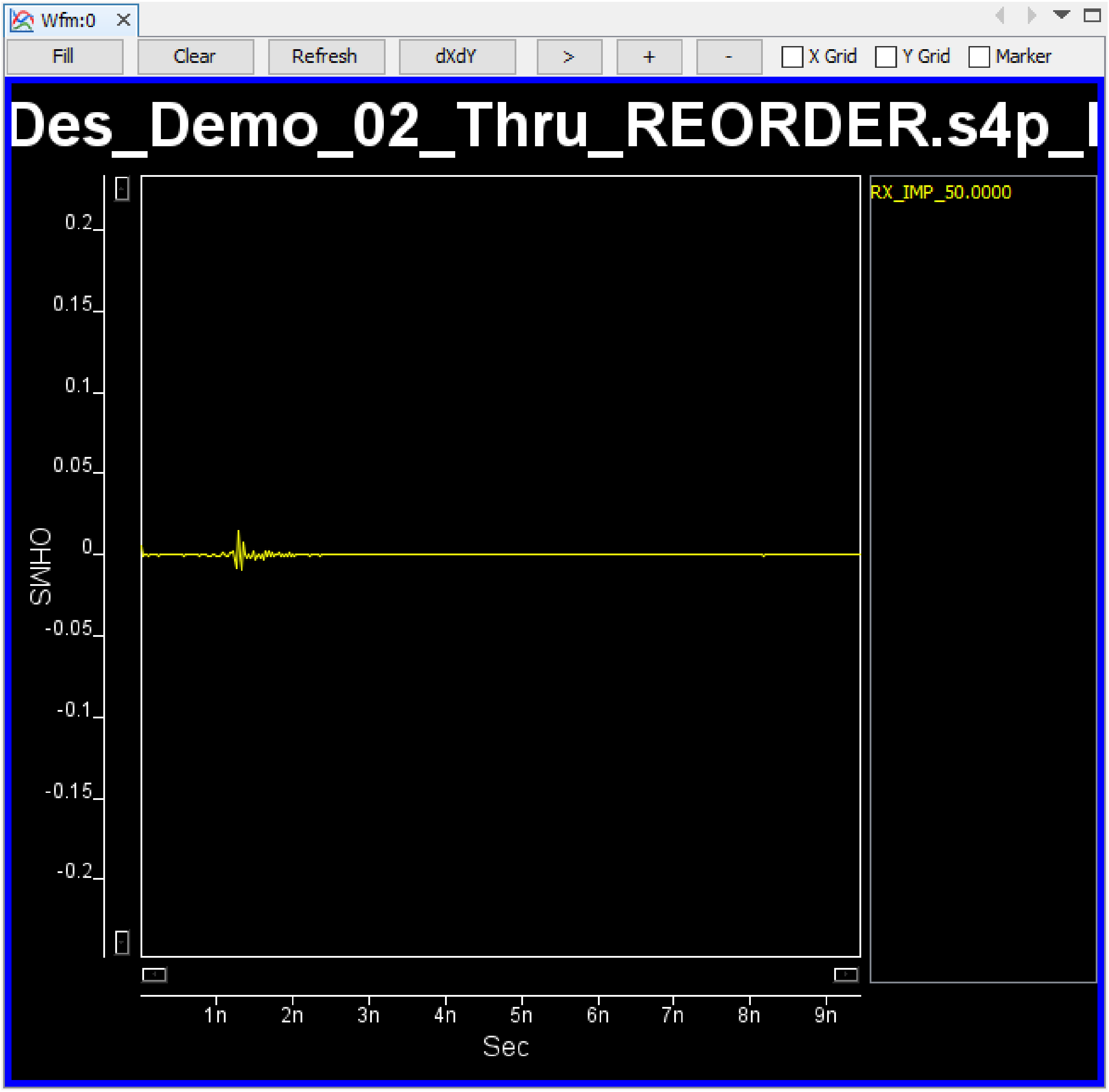
When the process is finished, new *.RAW* waveform files appear in the directory from which the original *.s4p* files were dragged.



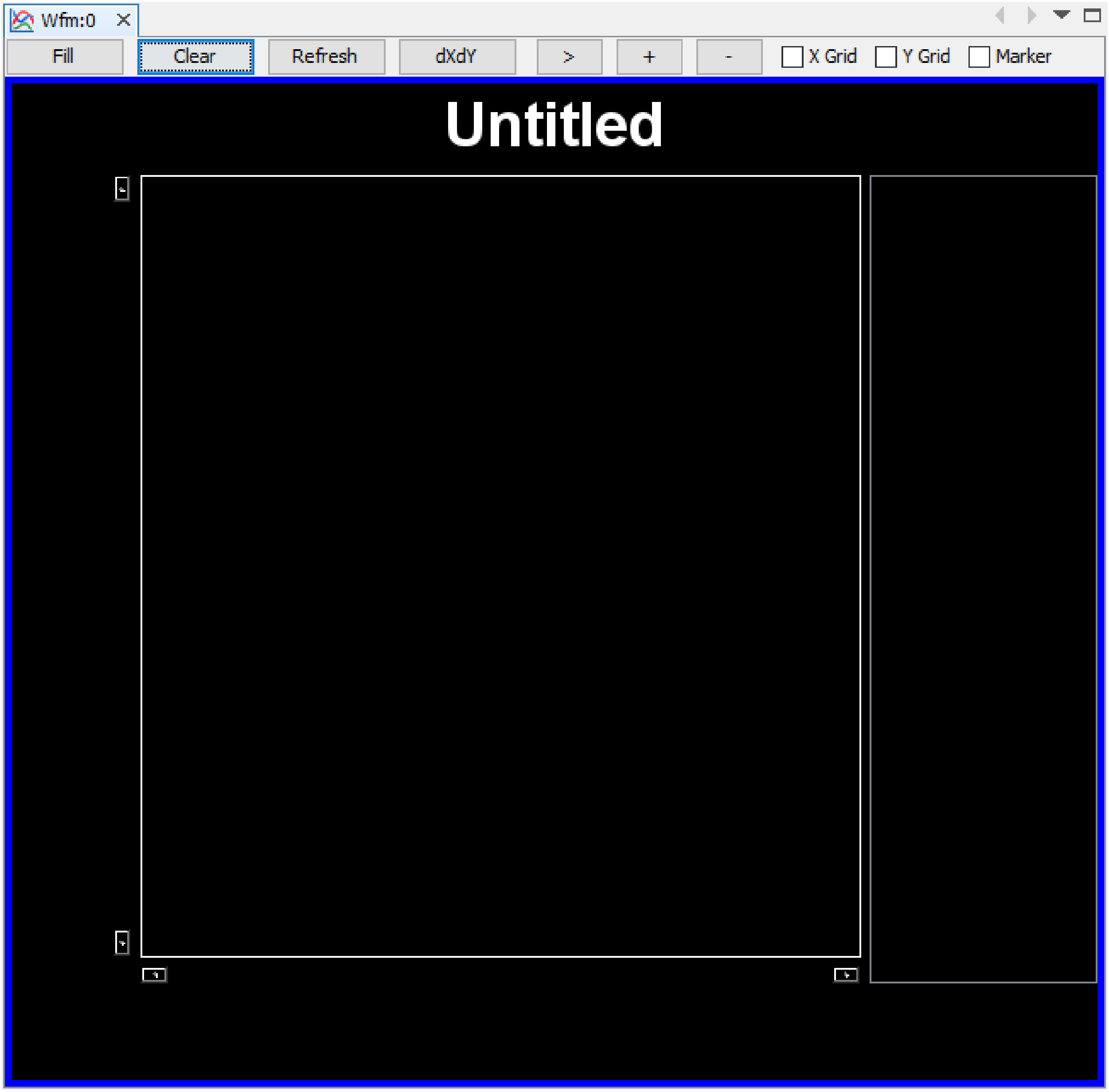
The **File List Window** and **View** tabs populate with relevant information (.e.g., the **Signal View** window is now populated with Pulse Time Domain Reflectometry (PTDR) waveforms).


File List Window
Name
ID
PATH
S-Param files
FCI_CC_Long_l 0 C:/ansysdev/2022061
FCI_CC_Short_ 1 C:/ansysdev/2022061
SerDes_Demo_ 2 C:/ansysdev/2022061
Waveform in Men
FCI_CC_Short_ 0 FCI_CC_Short_Link_Pa
FCI_CC_Long_l 1 FCI_CC_Long_Link_Pai
SerDes_Demo_ 2 SerDes_Demo_02_Thru
Select All
Close All
Close Selected
Signal View
Matrix View
Freq. View
Sort
Type name to filter...
RX_IMP_50.0000
RX_PDR_50.0000
RX_TDR_50.0000
RX_WC_SAMP_50.0000
TX_IMP_50.0000
TX_PDR_50.0000
TX_TDR_50.0000
TX_WC_SAMP_50.0000
View All
View Selected

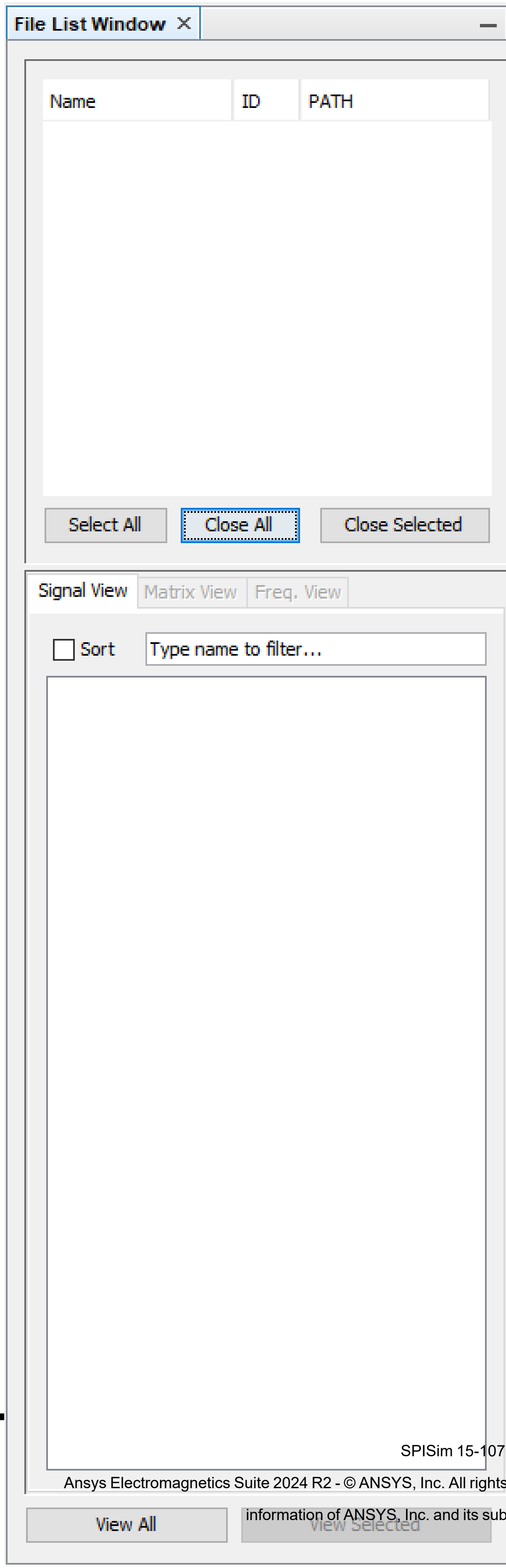
9. Select a PTDR waveform on the **Signal View** tab. Then click **View Selected** to view the PTDR waveform in the waveform view window (i.e., **Wfm:0**).



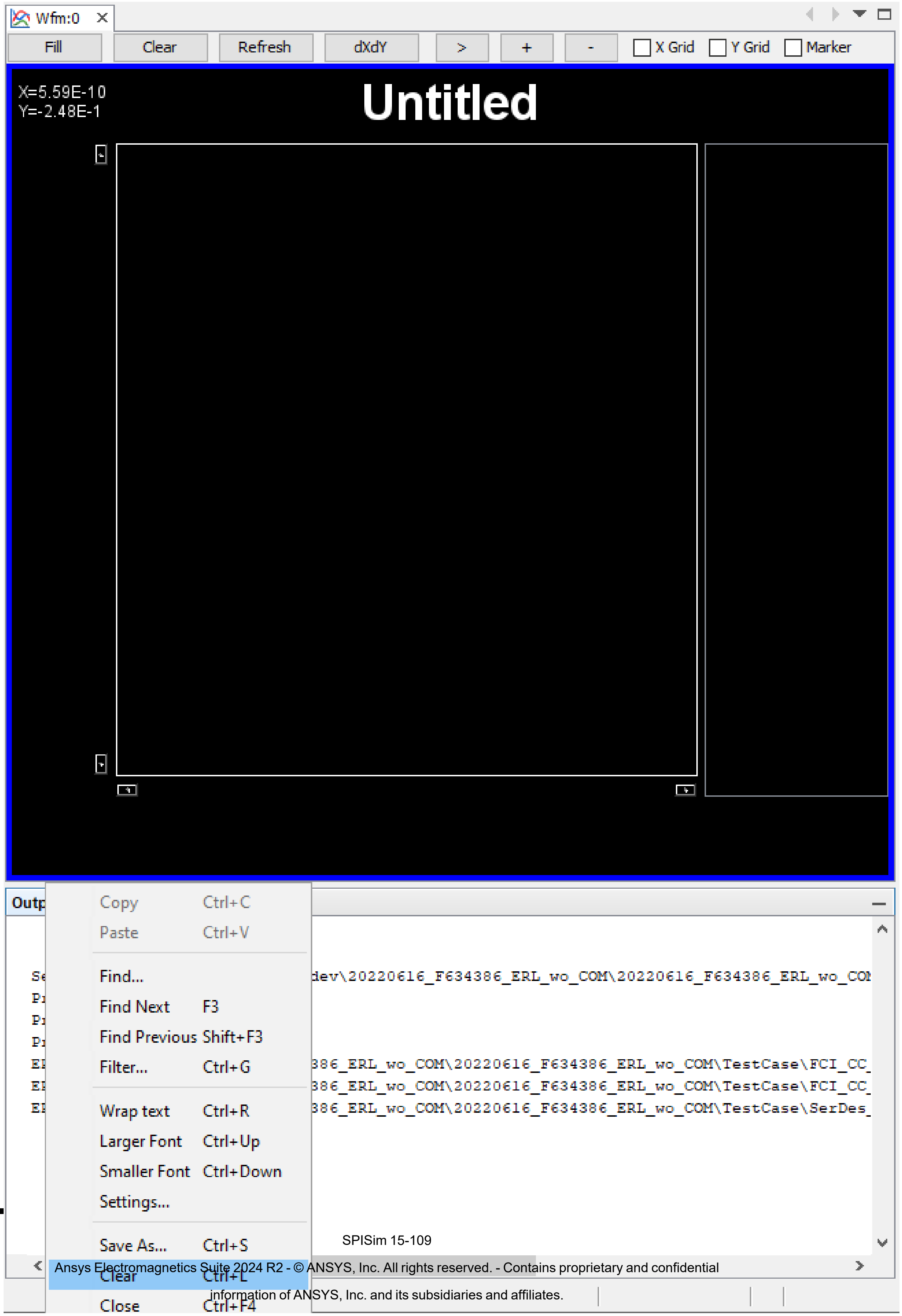
10. Click **Clear** to stop viewing the waveform.



11. From the **File List Window**, click **Close All** to clear the *.s4p* files and waveforms on the **File List Window** and the supplemental data on the **View** tabs.


File List Window
Name
ID
PATH
Select All
Close All
Close Selected
Signal View
Matrix View
Freq. View
Sort
Type name to filter...
View All
View Selected

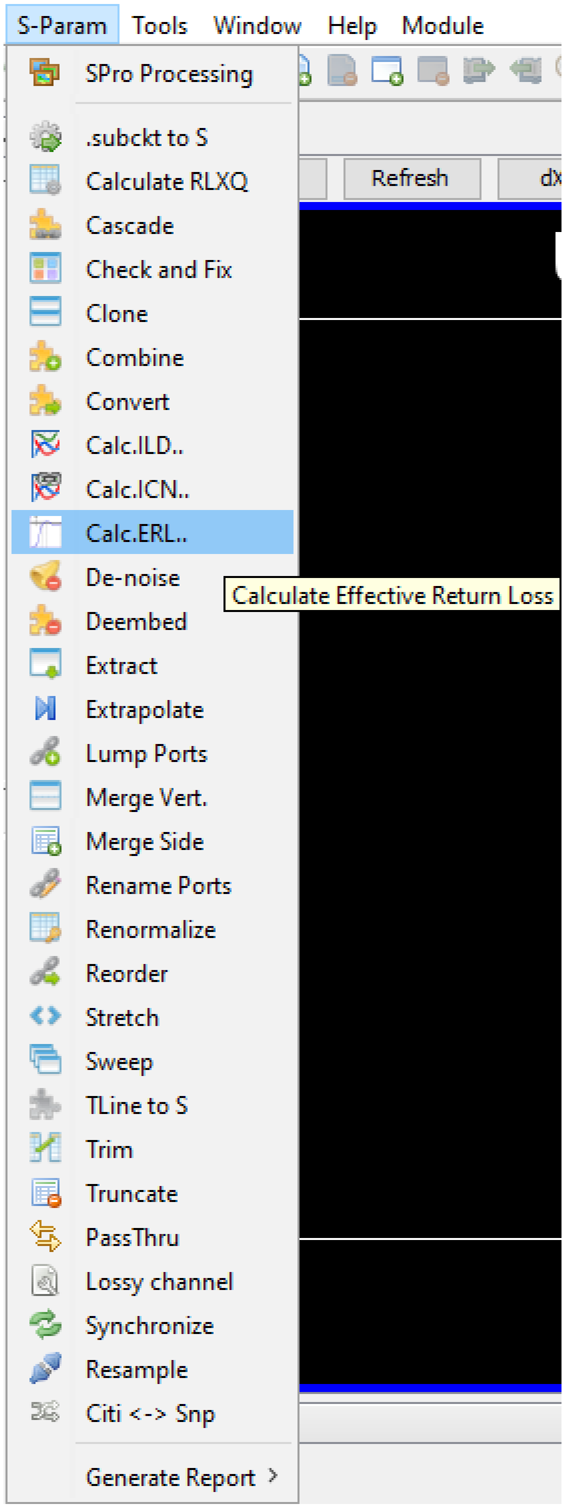
12. To clear the contents of the **Output** window, right-click within the window and select **Clear**.


Wfm:0
Fill
Clear
Refresh
dXdY
>
+
-
X Grid
Y Grid
Marker
X=5.59E-10
Y=-2.48E-1
Untitled
Copy Ctrl+C
Paste Ctrl+V
Find...
Find Next F3
Find Previous Shift+F3
Filter... Ctrl+G
Wrap text Ctrl+R
Larger Font Ctrl+Up
Smaller Font Ctrl+Down
Settings...
Save As... Ctrl+S
Clear Ctrl+L
Close Ctrl+F4

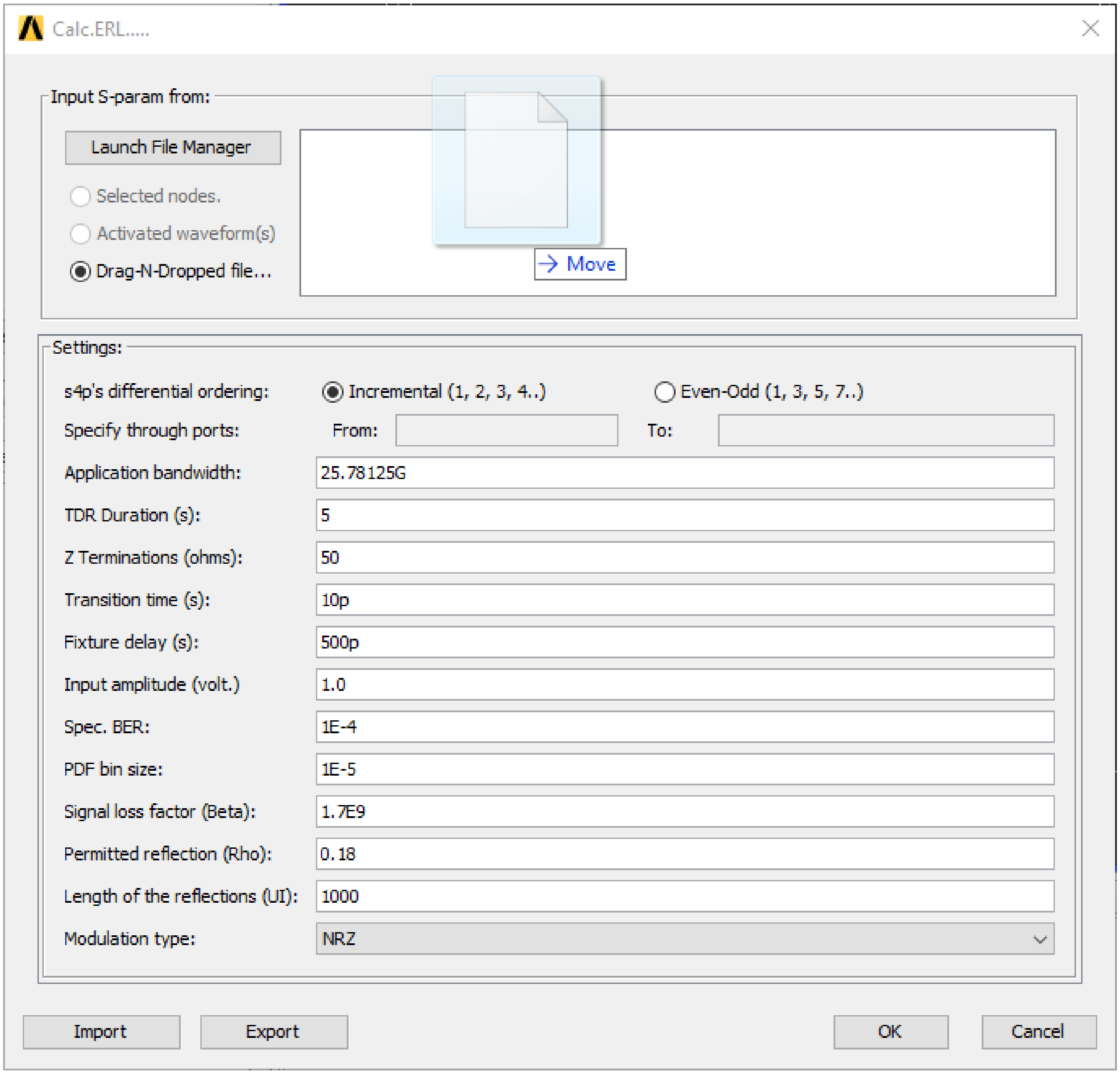
Calculating ERL of Non-.*S*4*P* .*S*N*P* Files

Follow these steps to calculate the ERL of .*sNp* files of any other variant, where ***N*** is not equal to **4**.

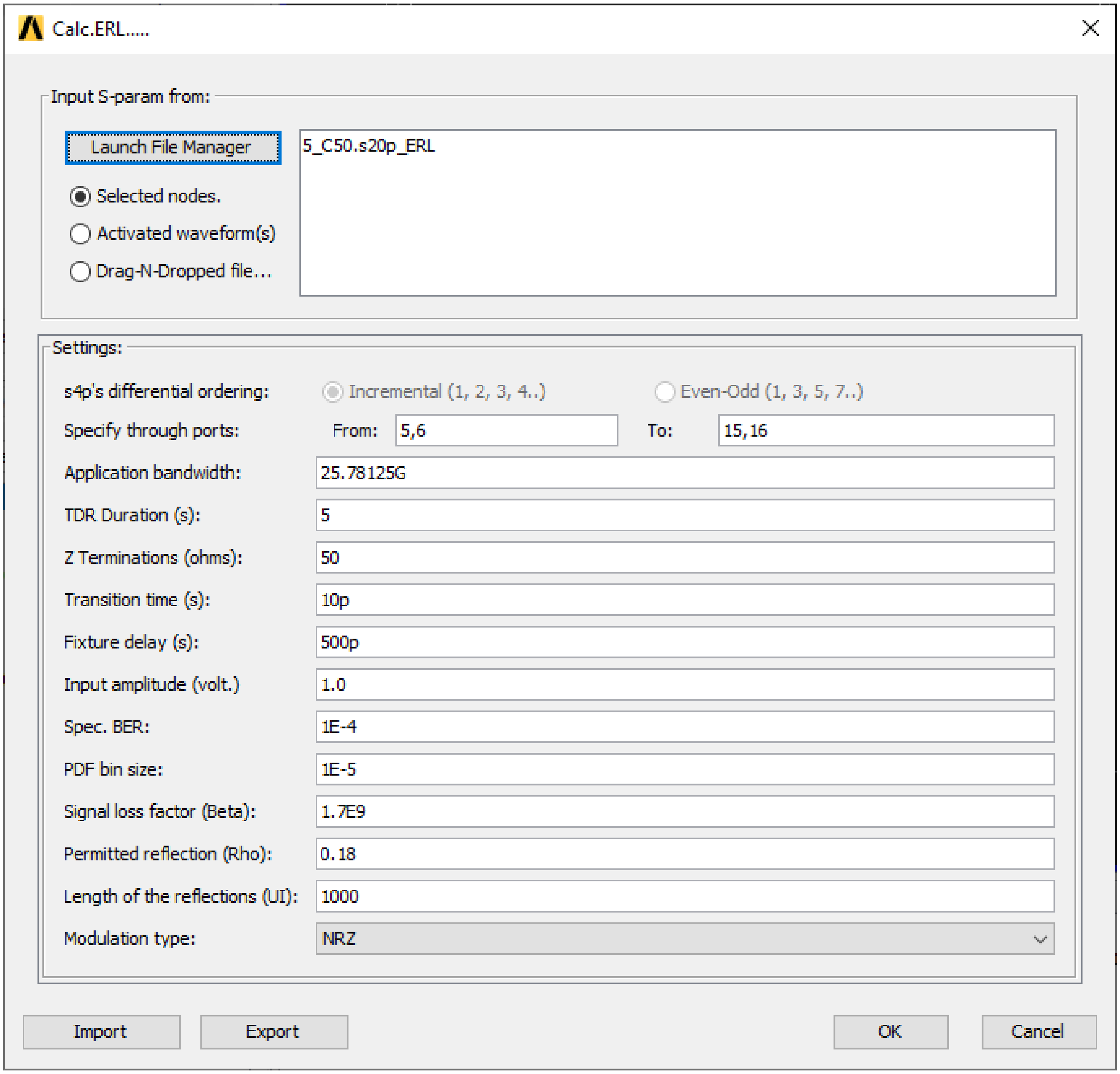
1. Navigate to **S-Param** > **Calc.Erl..** to open the **Calc.ERL.....** window.



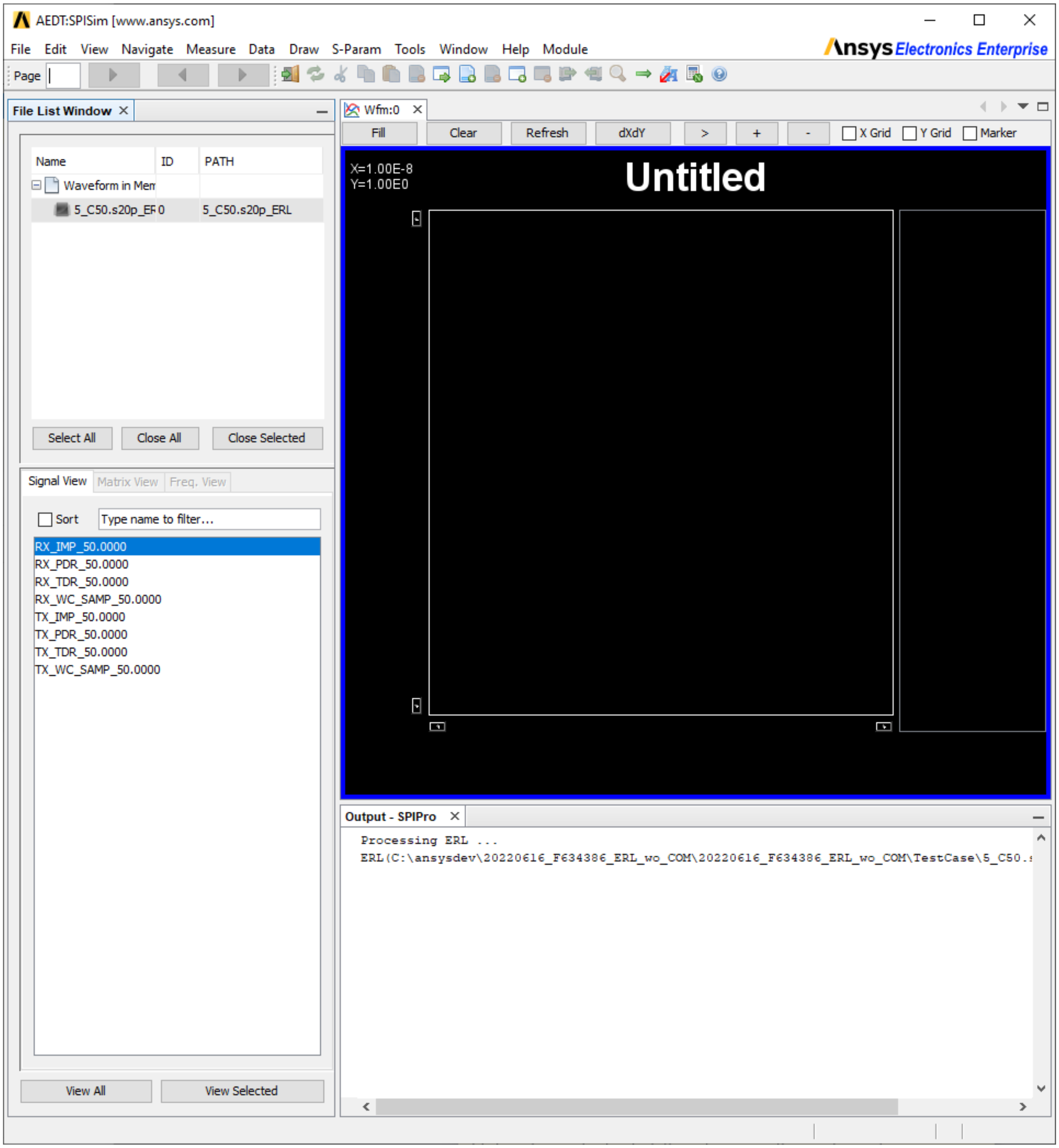
2. **Drag+drop** the relevant file(s) (e.g., ***s20p***) into the **Input S-param from** group box. Multiple .s*N*p files can be input, as long as they are of the same *N* value.



3. If necessary, enter two differential inputs (i.e., in the **From** field) and two outputs (i.e., in the **To** field).



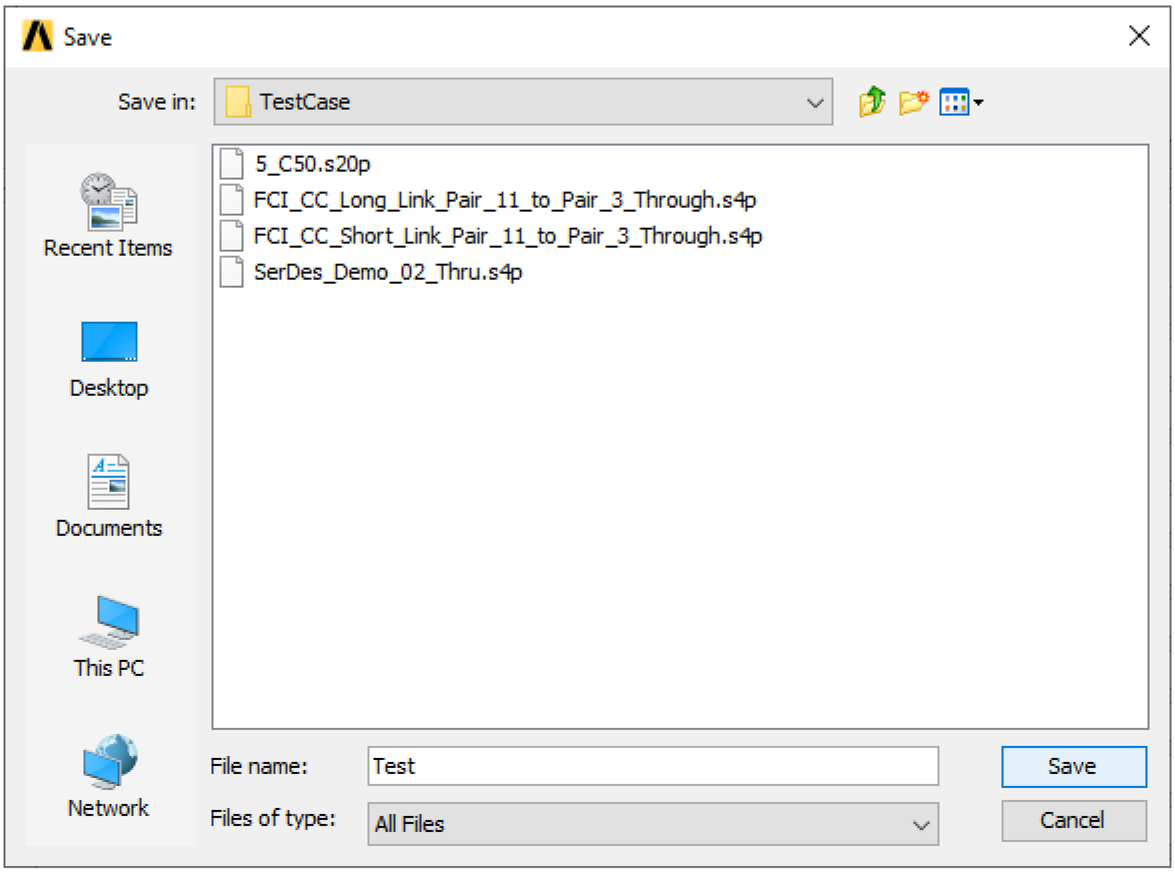
4. Click **OK** to close the **Calc.ERL.....** window and begin ERL calculation. View the process and see the result in **SPISim**.



Creating an ERL Configuration File

Complete the following steps to create (i.e., **Export**) a *.cfg* (i.e., configuration) file.

1. After populating **SPISim** with the appropriate *.s4p* files and making any necessary changes in the **Calc.ERL.....** window > **Settings** group box (i.e., complete steps 1-7 in "Calculating ERL From the SPISim User Interface" on page 15-93), click **Export** to open an explorer window.
2. Navigate to the appropriate directory.
3. Enter a **File name** for the *.cfg* file (i.e., **Test**).
4. Click **Save** to close the explorer window.



Importing an ERL Configuration File

Follow the instructions on the following pages to import an ERL configuration file using three methods.

"Importing an ERL Configuration File From the SPISIM User Interface" on the next page

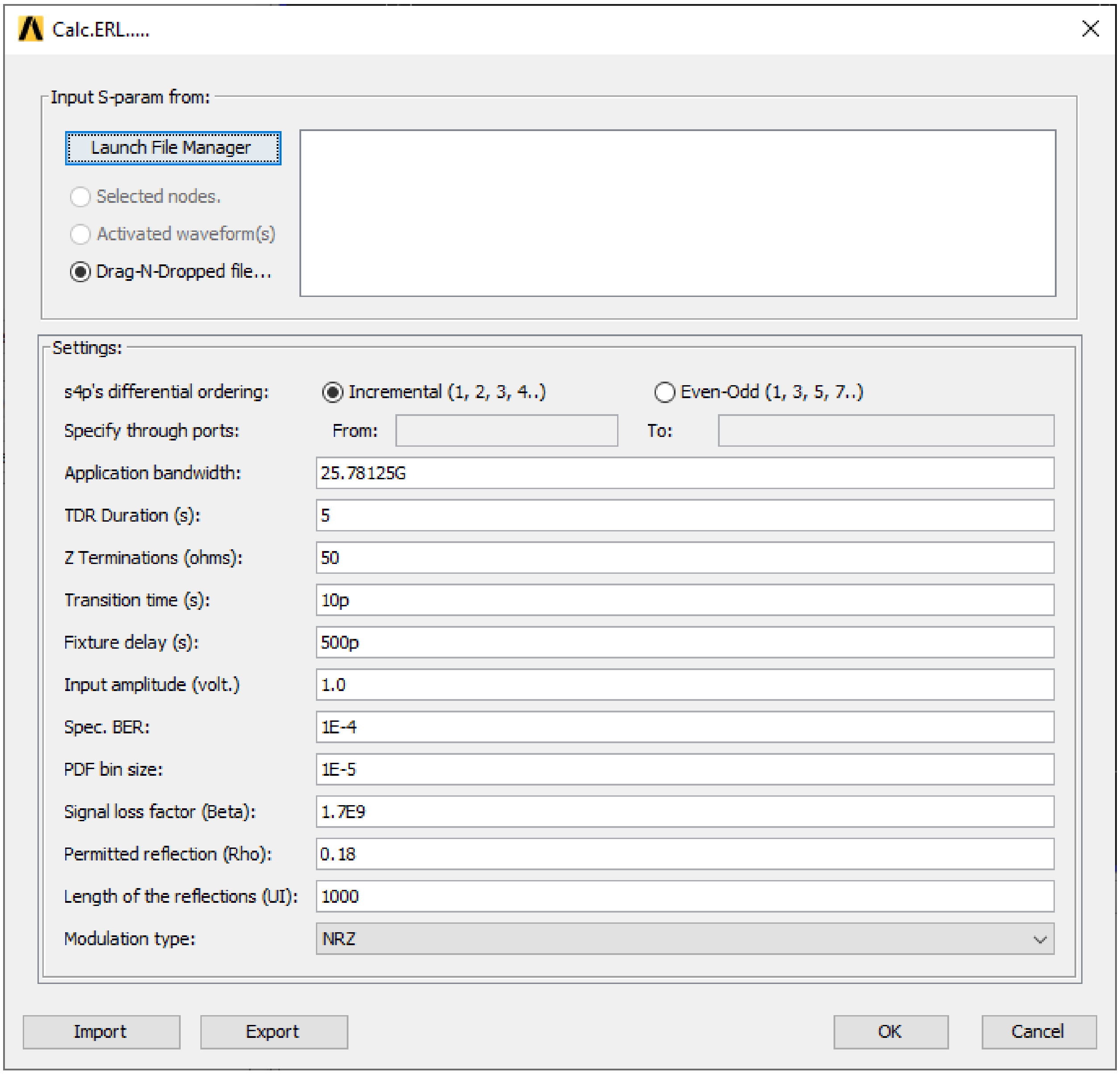
"Importing an ERL Configuration File Using a Batch Command Line to Invoke the User Interface" on page 15-121

"Importing an ERL Configuration File Using Only a Non-GUI Batch Command Line" on page 15-127

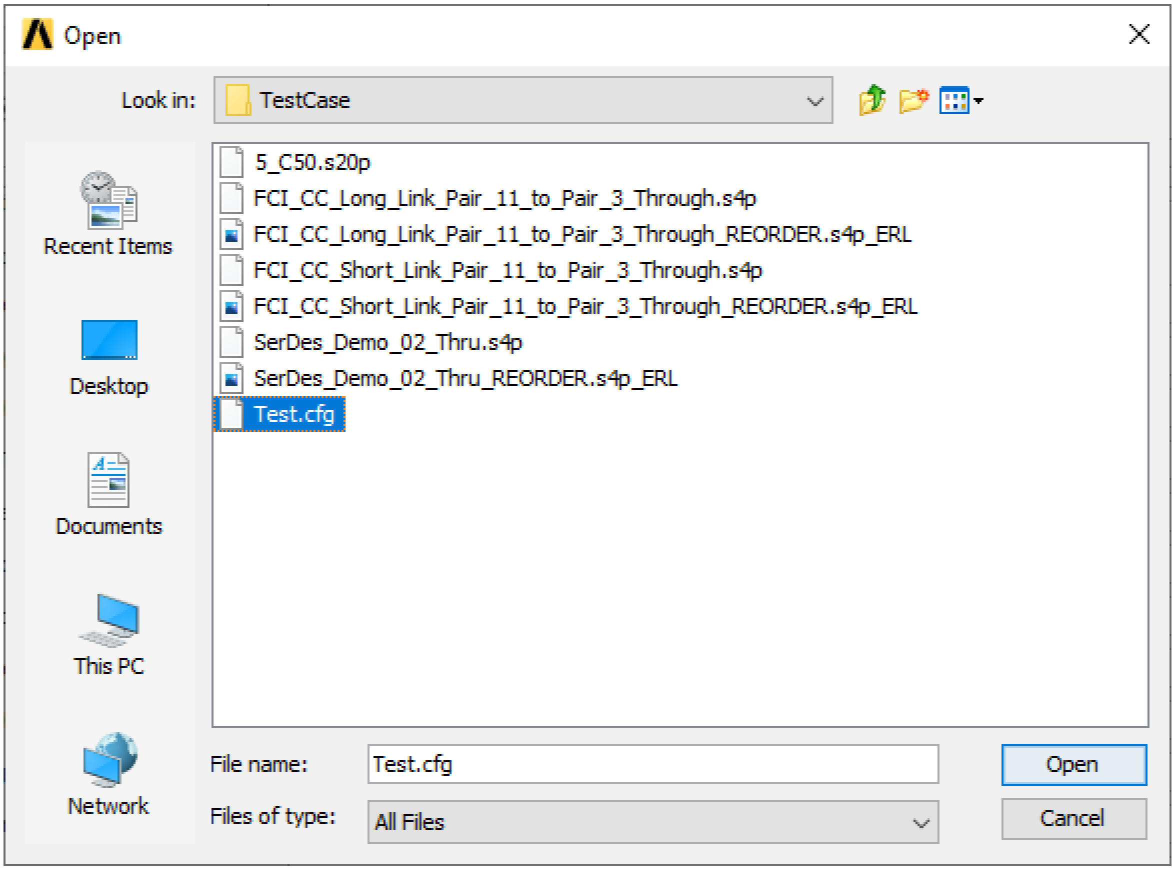
Importing an ERL Configuration File From the SPISIM User Interface

Complete the following steps to **Import** a *.cfg* file using the **SPISim** user interface, or refer to "Importing an ERL Configuration File Using a Batch Command Line to Invoke the User Interface" on page 15-121 or "Importing an ERL Configuration File Using Only a Non-GUI Batch Command Line" on page 15-127 for alternative methods.

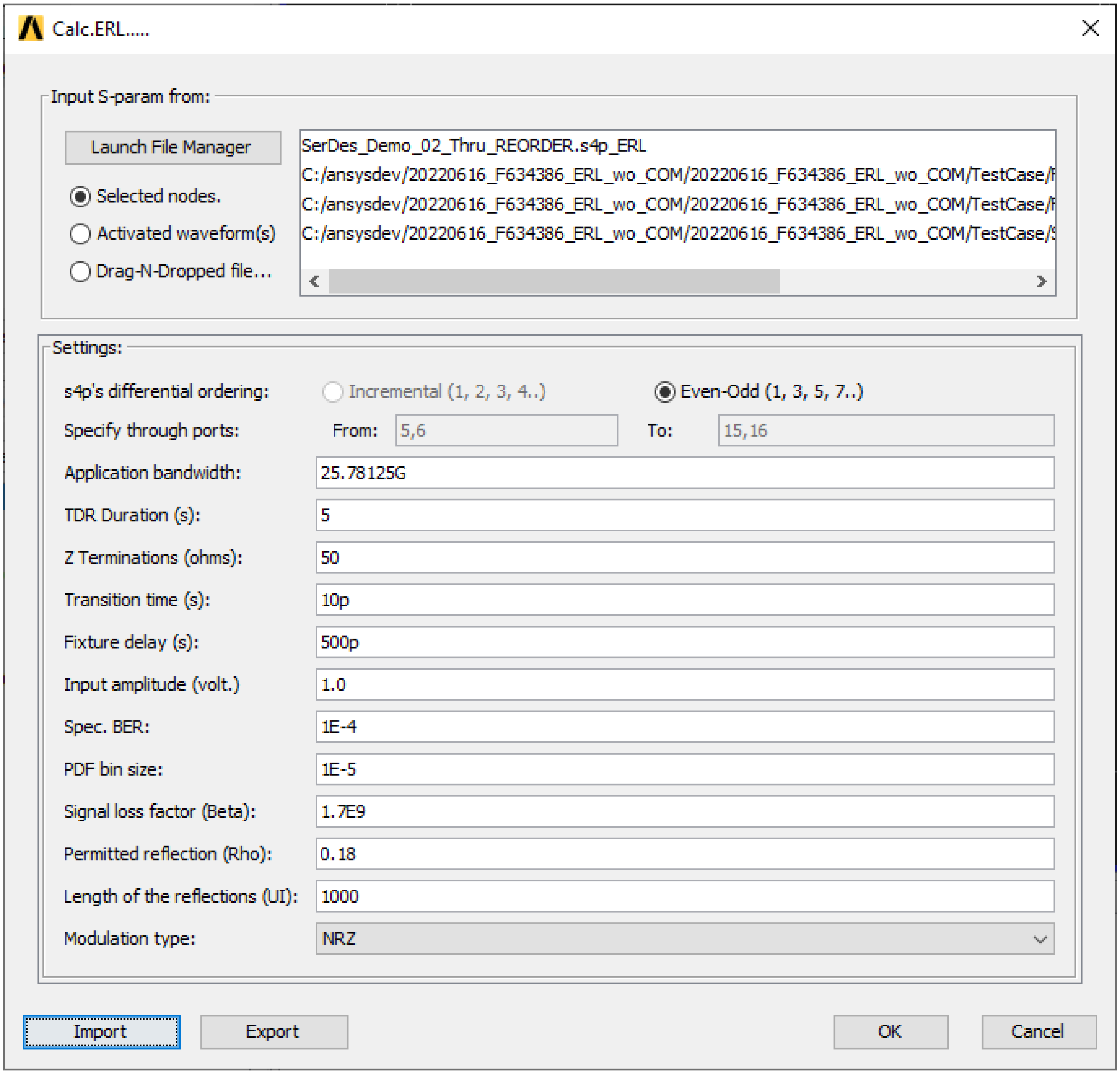
1. From the **Calc.ERL.....** window, click **Import** to open an explorer window.



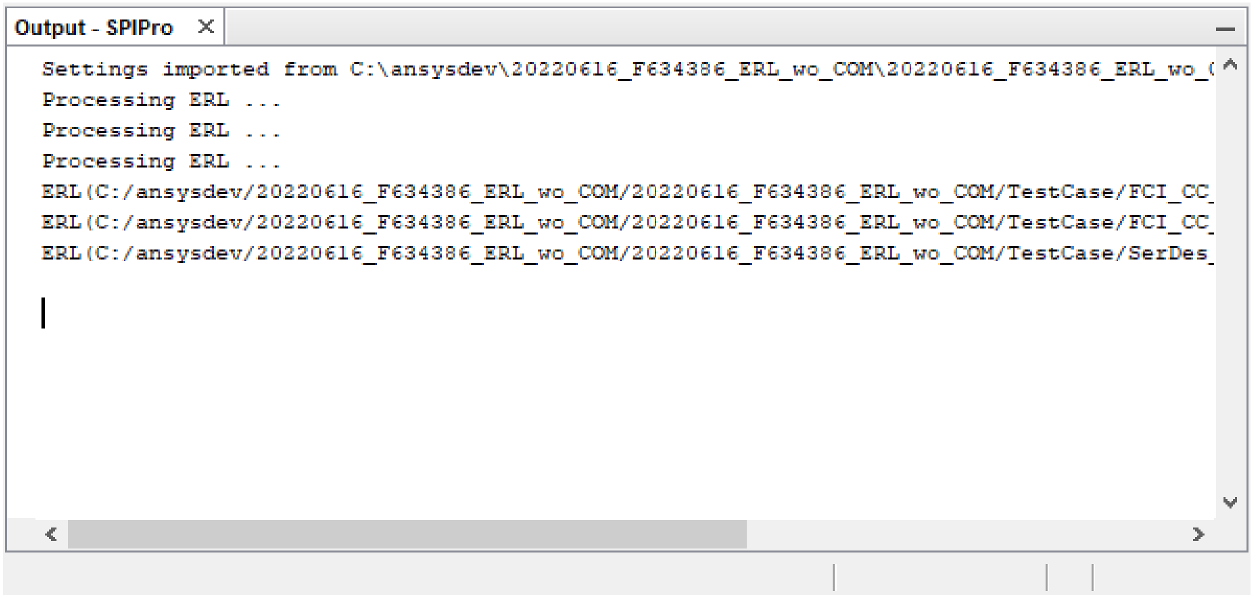
2. Navigate to the appropriate directory and select a *.cfg* file (i.e., **Test.*cfg***).
3. Click **Open** to close the explorer window.



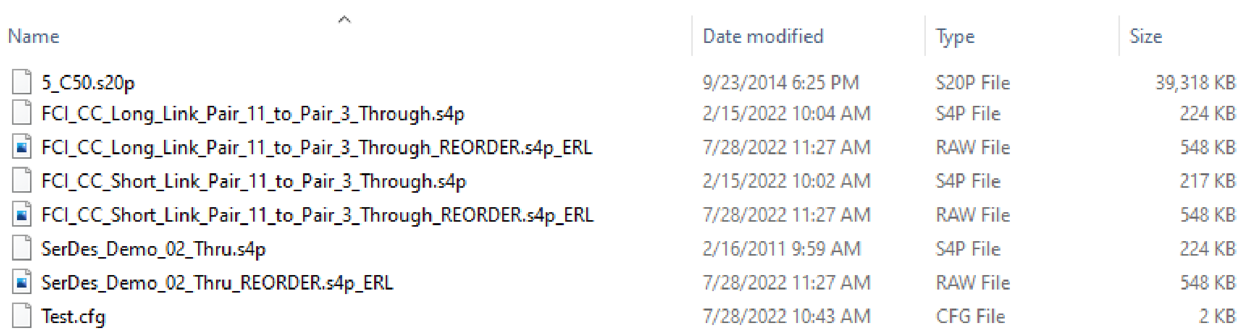
4. Make any necessary changes in the **Settings** group box. Refer to step 7 in Calculating ERL From the SPISim User Interface.



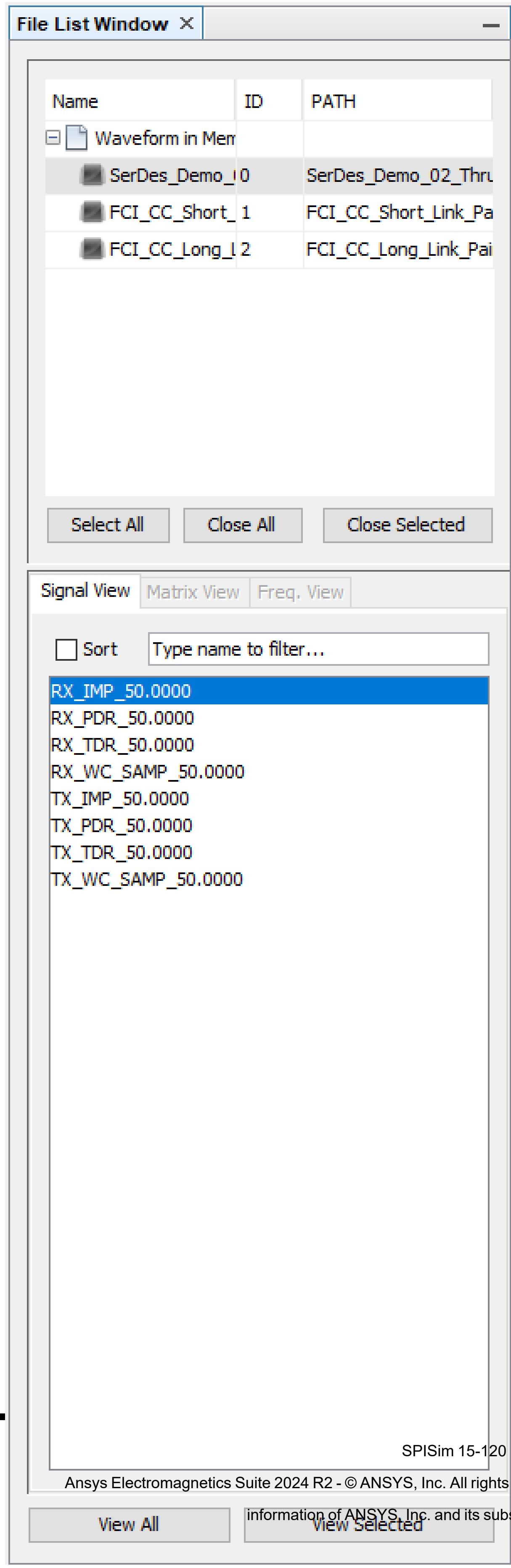
5. Click **OK** to close the **Calc.ERL.....** window and begin ERL calculation. View the process and result in the **Output - SPIPro** window.



When the process is finished, new *.RAW* waveform files appear in the directory from which the original *.s4p* files were dragged.



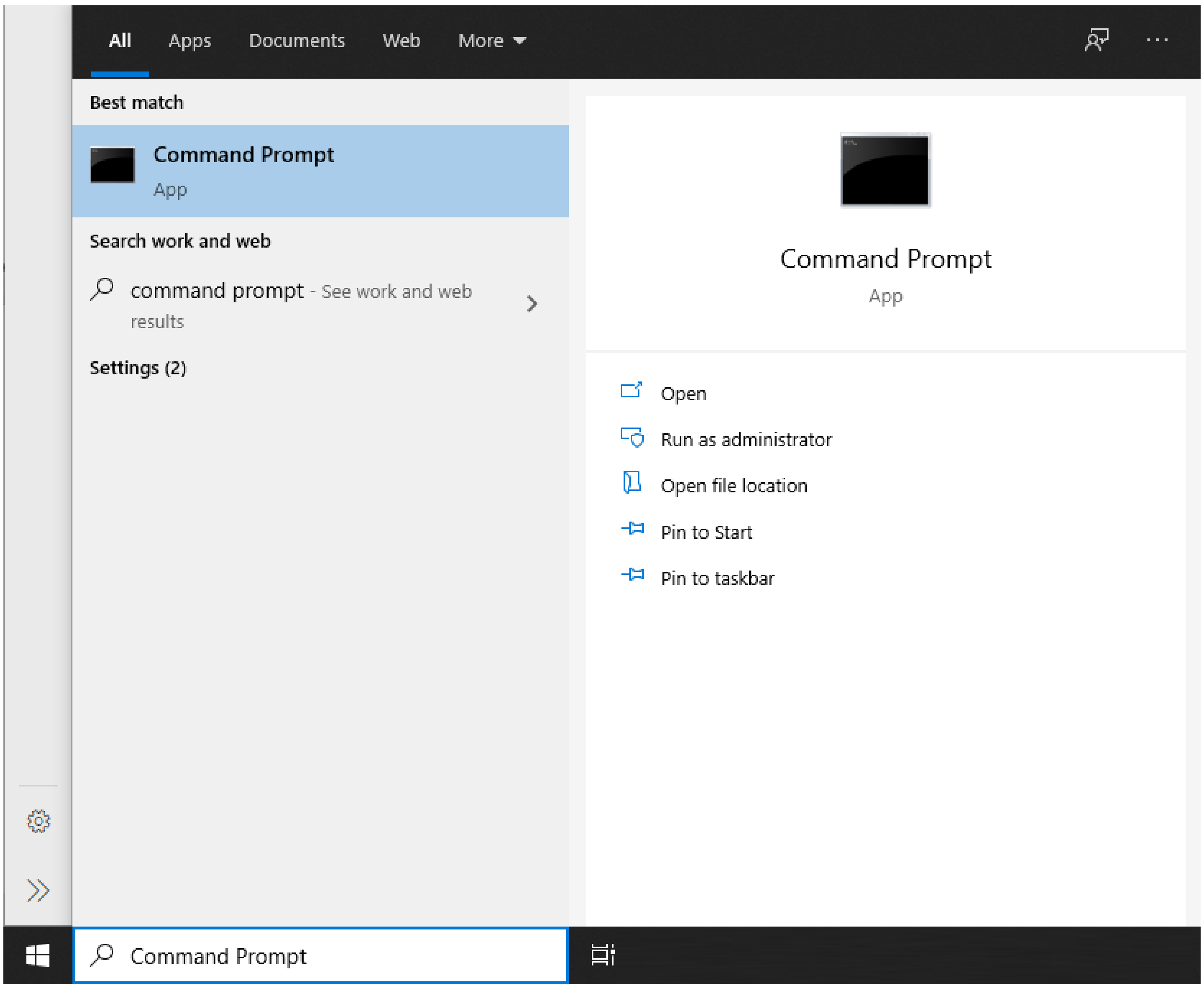
The **View** tabs populates with relevant information (.e.g., the **File List Window** is now populated with a waveform created for each *.s4p* file, which were identified by the configuration file).


File List Window
Name
ID
PATH
Waveform in Mem
SerDes_Demo_ 0 SerDes_Demo_02_Thru
FCI_CC_Short_ 1 FCI_CC_Short_Link_Pa
FCI_CC_Long_ 2 FCI_CC_Long_Link_Pai
Select All
Close All
Close Selected
Signal View
Matrix View
Freq. View
Sort
Type name to filter...
RX_IMP_50.0000
RX_PDR_50.0000
RX_TDR_50.0000
RX_WC_SAMP_50.0000
TX_IMP_50.0000
TX_PDR_50.0000
TX_TDR_50.0000
TX_WC_SAMP_50.0000
View All
View Selected

Importing an ERL Configuration File Using a Batch Command Line to Invoke the User Interface

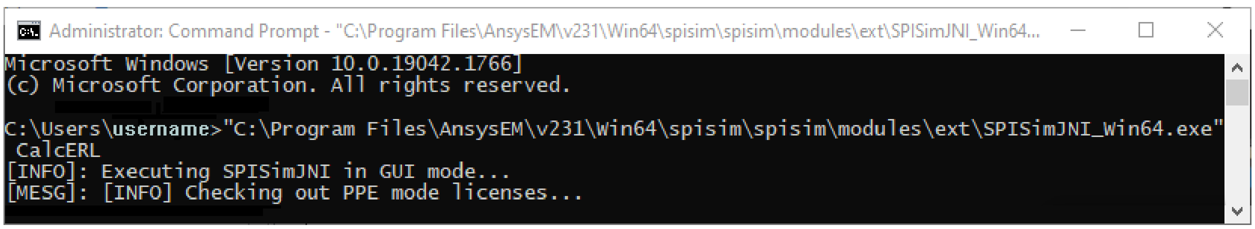
Complete the following steps to **Import** a *.cfg* file using a batch command line to open the **SPISim** user interface.

1. Run a command prompt (e.g., in a Windows environment, type *Command Prompt* into the **Search Bar** and press **Enter** to open a **Command Prompt** window).

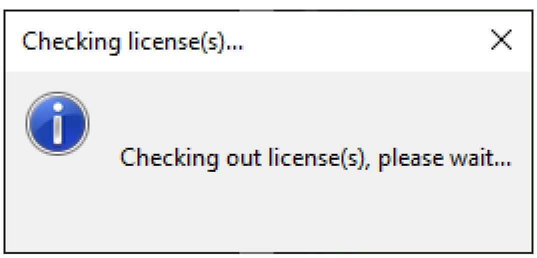


> **Note:** The following example assumes **Electronics Desktop** is installed in the default directory in a Windows operating system environment. Structure the command appropriately for a different directory, path, and/or operating system, as necessary. Do **not** move the ***SPISimJNI_WIN64.exe*** executable file. The executable file relies on the relative path to the **Electronics Desktop** installation folder.

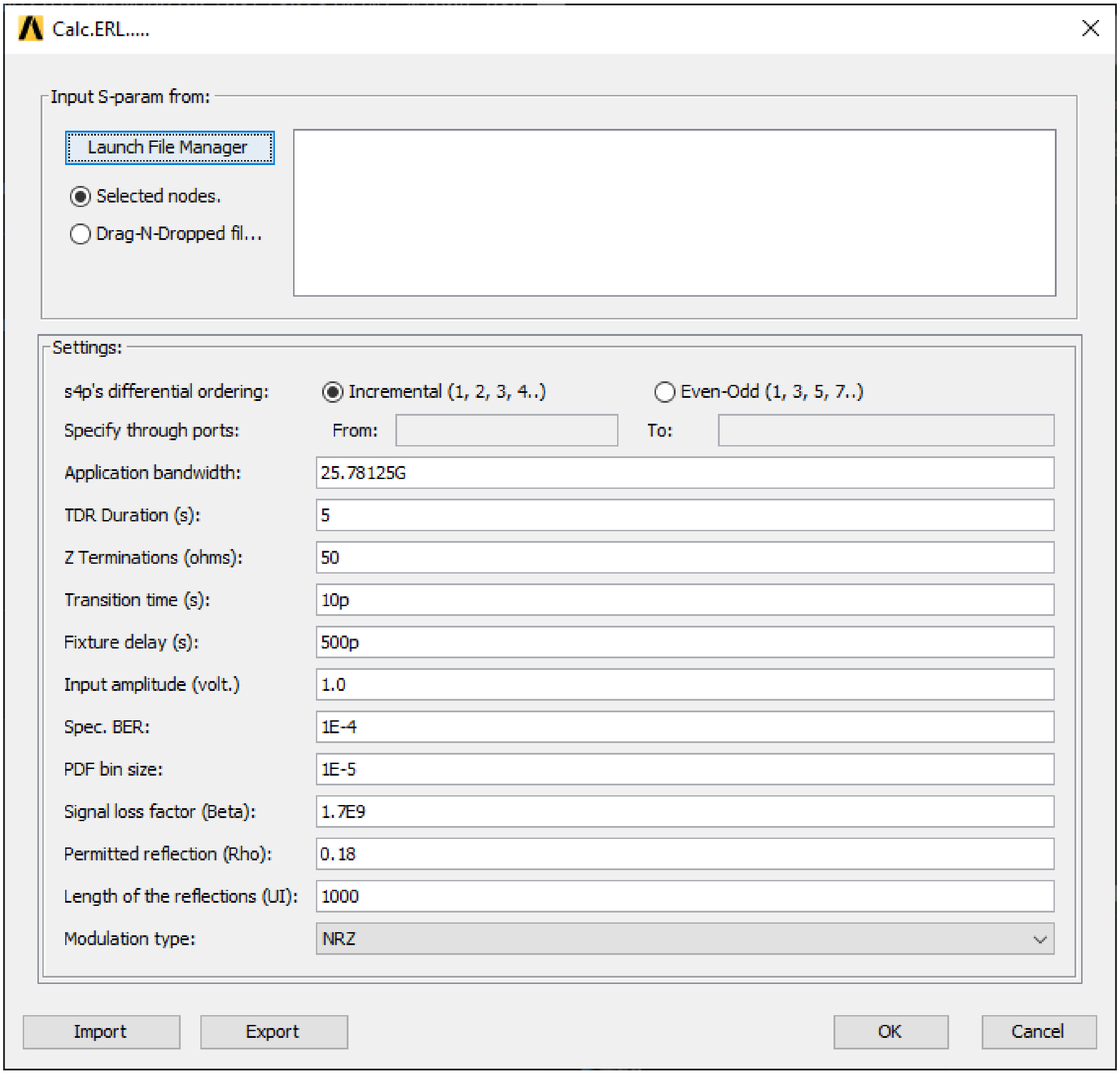
2. From the **Command Prompt** window, enter the following command, then press **Enter**:
   `"C:\Program Files\AnsysEm\v242\Win64\spisim\spisim\modules\ext\SPISimJNI_WIN64.exe" CalcERL`



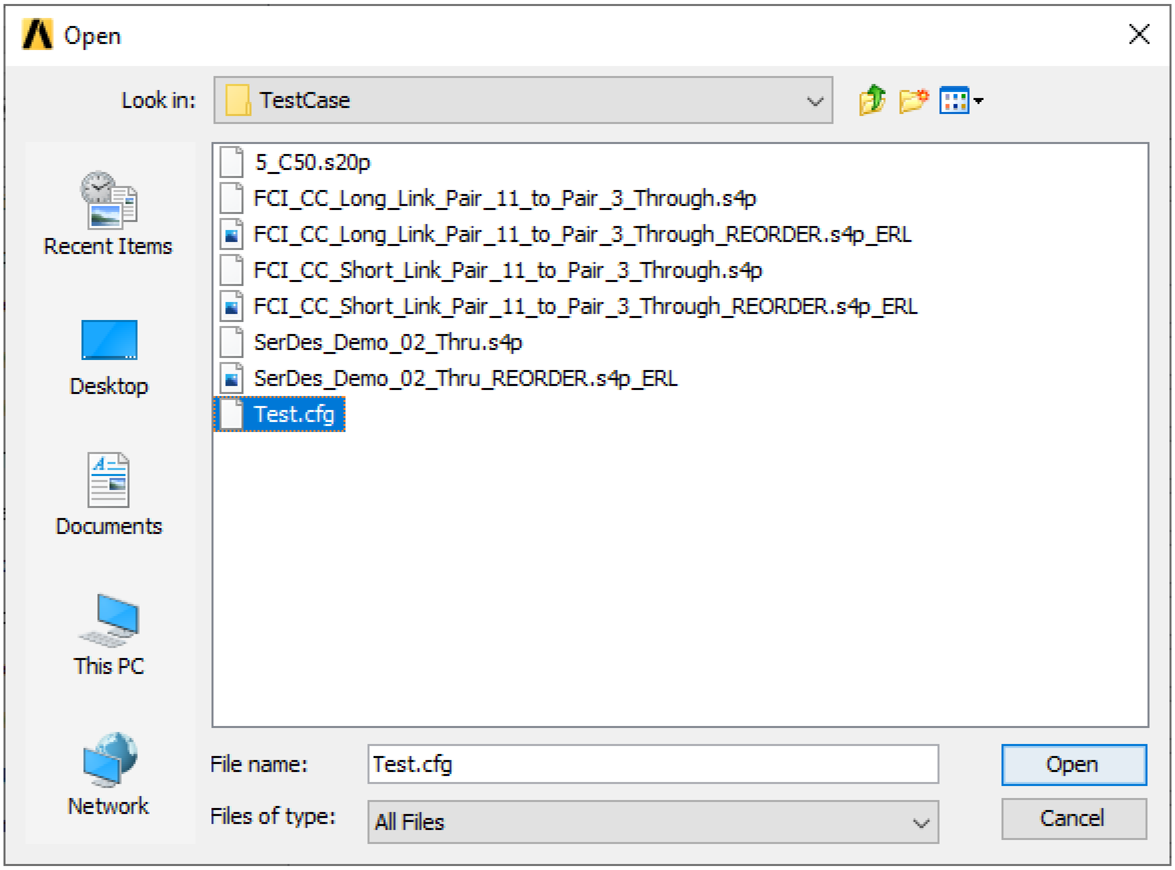
A **Checking License(s) ...** window opens.



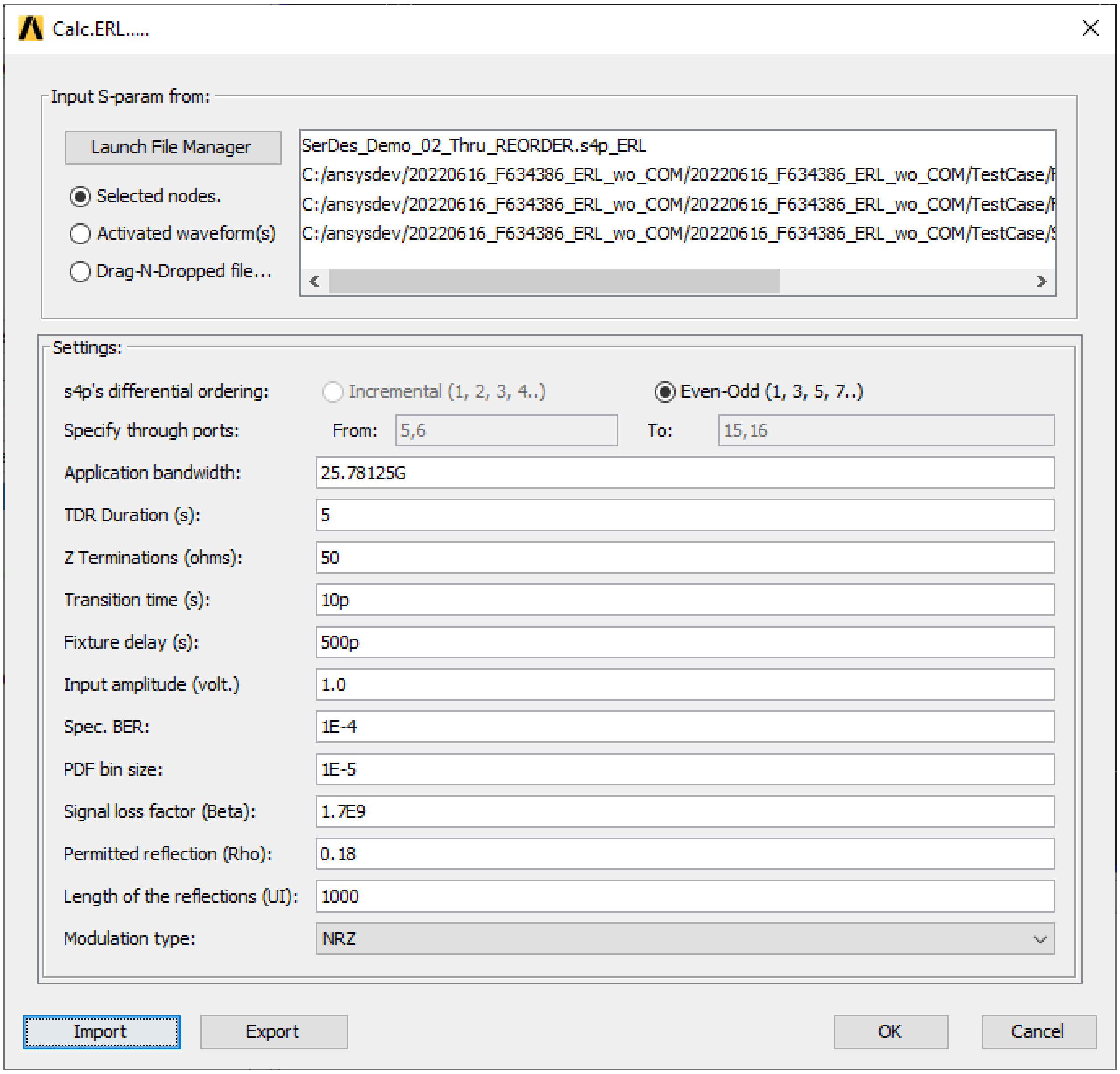
Once the software license is verified, the **Calc.ERL.....** window opens.



3. From the **Calc.ERL.....** window, click **Import** to open an explorer window.
4. Navigate to the appropriate directory and select a *.cfg* file (i.e., ***Test.cfg***).
5. Click **Open** to close the explorer window.



6. Make any necessary changes in the **Settings** group box. Refer to step 7 in Calculating ERL From the SPISim User Interface.



7. Click **OK** to close the **Calc.ERL.....** window and begin ERL calculation. View the process and result in the **Command Prompt** window.

```
Administrator: Command Prompt - "C:\Program Files\AnsysEM\v231\Win64\spisim\spisim\modules\ext\SPISimJNI_Win64...
Microsoft Windows [Version 10.0.19042.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\username>"C:\Program Files\AnsysEM\v231\Win64\spisim\spisim\modules\ext\SPISimJNI_Win64.exe"
 CalcERL
[INFO]: Executing SPISimJNI in GUI mode...
[MESG]: [INFO] Checking out PPE mode licenses...
[MESG]: [SPISimJNI] Info: Calling SPISim command.
[MESG]: UDP_COMM_CONCLUDE
[ParmDat]: TDR_DUR=5,SPECBER=1E-4,MODTYPE=NRZ,THRUS4P=null,BANDWID=25.78125G,REFIMPD=50,MIXMODE=EVENO
DD,TRSTIME=10p,BINSIZE=1E-5,REFLRHO=0.18,NCYCLES=1000,INPVOLT=1.0,FIXDELY=500p,ERL=0.6565315408892082
 -3.9896182972471195 0.6602794976412402,SIGBETA=1.7E9
[INFO]: Concluded.
```

In this example, three sets of results are output. The *.cfg* file, which is viewable in any text reader, calls three *.s4p* files.

```
Test - Notepad
File  Edit  Format  View  Help
################################################################################
# MODULE: CalcERL
# GENERATED ON 20220728104307
################################################################################

# INPARRY: INPARRY
INPARRY = C:/ansysdev/20220616_F634386_ERL_wo_COM/20220616_F634386_ERL_wo_COM/TestCase/FCI_CC_Long_Link_
C:/ansysdev/20220616_F634386_ERL_wo_COM/20220616_F634386_ERL_wo_COM/TestCase/FCI_CC_Short_Link_Pair_11_t
C:/ansysdev/20220616_F634386_ERL_wo_COM/20220616_F634386_ERL_wo_COM/TestCase/SerDes_Demo_02_Thru.s4p
# MIXMODE: MIXMODE
MIXMODE = EVENODD
# THRUS4P: THRUS4P
THRUS4P =
# BANDWID: BANDWID
BANDWID = 25.78125G
# TDR_DUR: TDR_DUR
TDR_DUR = 5
# REFIMPD: REFIMPD
REFIMPD = 50
# BINSIZE: BINSIZE
BINSIZE = 1E-5
# SPECBER: SPECBER
SPECBER = 1E-4
# MODTYPE: MODTYPE
MODTYPE = NRZ
# FIXDELY: FIXDELY
FIXDELY = 500p
# INPVOLT: INPVOLT
INPVOLT = 1.0
# TRSTIME: TRSTIME
TRSTIME = 10p
# SIGBETA: SIGBETA
SIGBETA = 1.7E9
# REFLRHO: REFLRHO
REFLRHO = 0.18
# NCYCLES: NCYCLES
NCYCLES = 1000
Ln 1, Col 1    100%    Unix (LF)    UTF-8
```

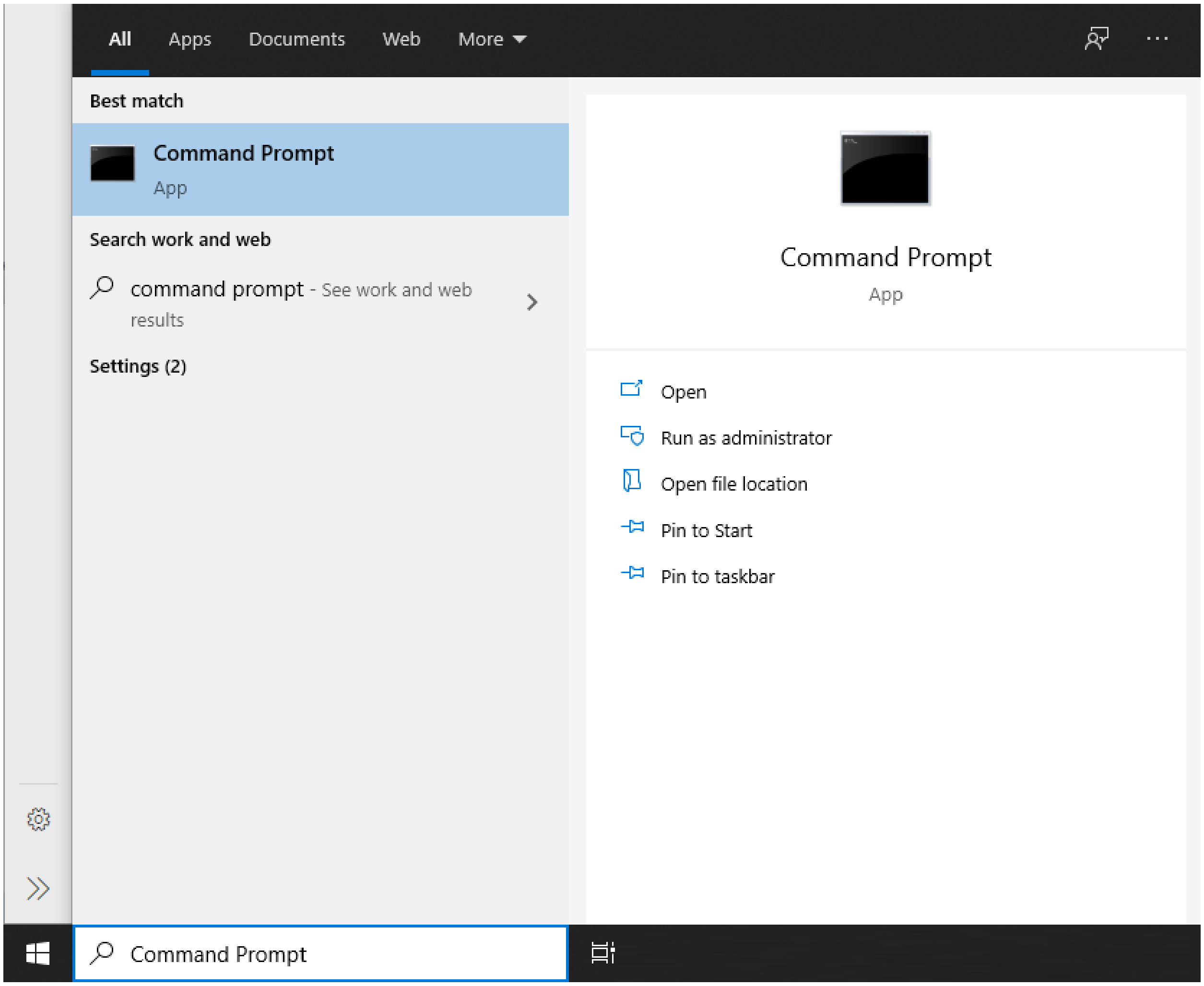
When the process is finished, new *.RAW* waveform files appear in the directory from which the *.s4p* files were called.

Name	Date modified	Type	Size
5_C50.s20p	9/23/2014 6:25 PM	S20P File	39,318 KB
FCI_CC_Long_Link_Pair_11_to_Pair_3_Through.s4p	2/15/2022 10:04 AM	S4P File	224 KB
FCI_CC_Long_Link_Pair_11_to_Pair_3_Through_REORDER.s4p_ERL	7/28/2022 11:27 AM	RAW File	548 KB
FCI_CC_Short_Link_Pair_11_to_Pair_3_Through.s4p	2/15/2022 10:02 AM	S4P File	217 KB
FCI_CC_Short_Link_Pair_11_to_Pair_3_Through_REORDER.s4p_ERL	7/28/2022 11:27 AM	RAW File	548 KB
SerDes_Demo_02_Thru.s4p	2/16/2011 9:59 AM	S4P File	224 KB
SerDes_Demo_02_Thru_REORDER.s4p_ERL	7/28/2022 11:27 AM	RAW File	548 KB
Test.cfg	7/28/2022 10:43 AM	CFG File	2 KB

Importing an ERL Configuration File Using Only a Non-GUI Batch Command Line

Complete the following steps to **Import** a *.cfg* file using only a non-GUI (NG) batch command line.

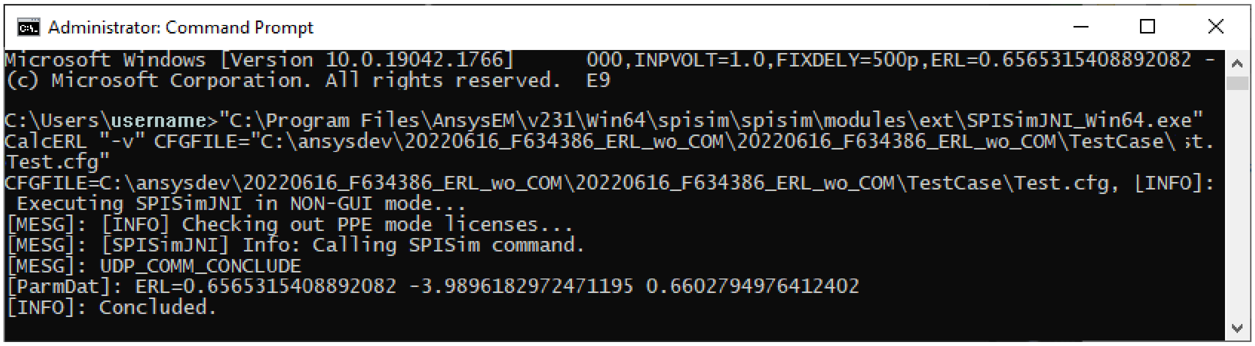
1. Run a command prompt (e.g., in a Windows environment, type *Command Prompt* into the **Search Bar** and press **Enter** to open a **Command Prompt** window.



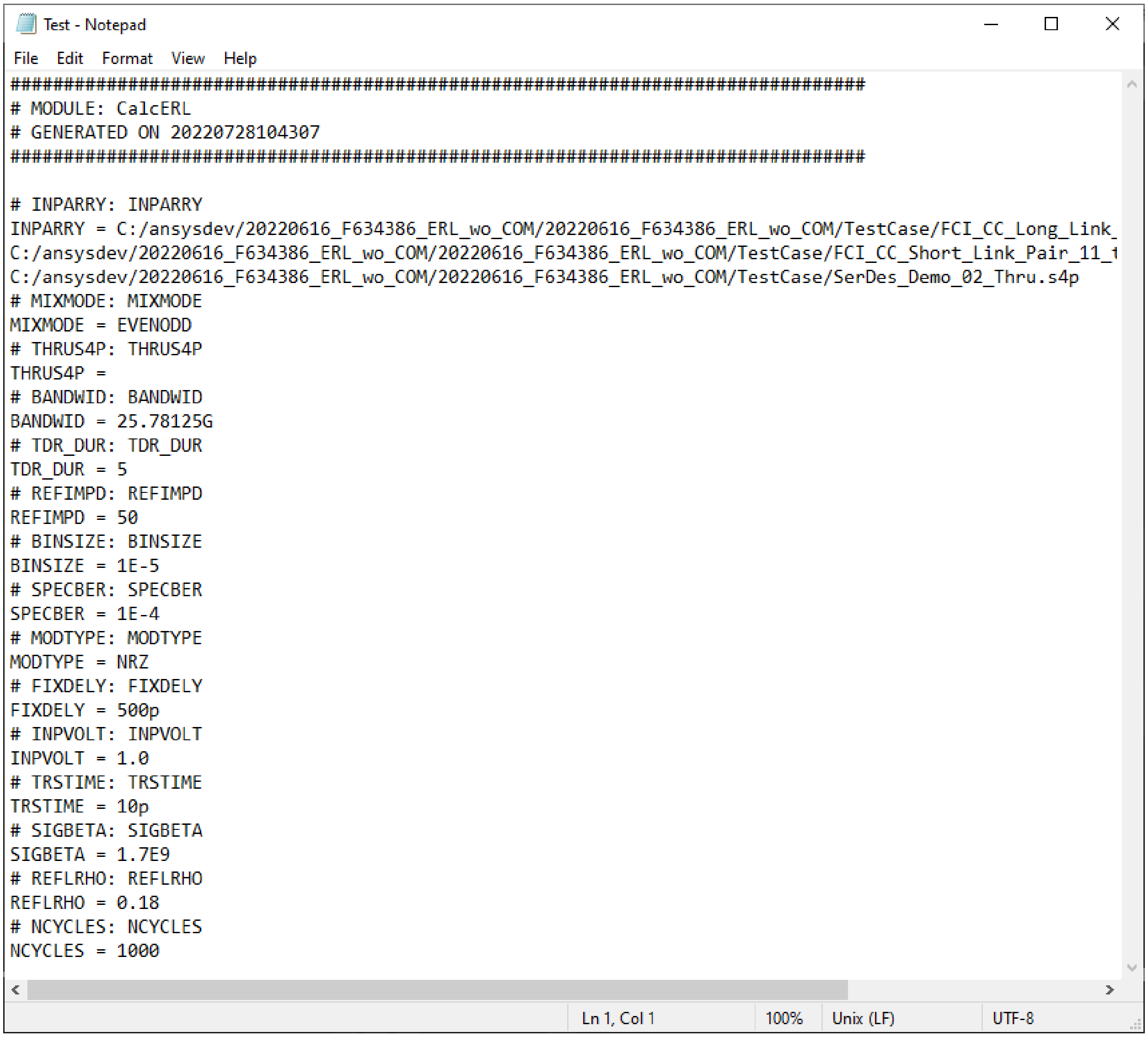
> **Note:** The following example assumes **Electronics Desktop** is installed in the default directory in a Windows operating system environment. Structure the command appropriately for a different directory, path, and/or operating system, as necessary. Do **not** move the ***SPISimJNI_WIN64.exe*** executable file. The executable file relies on the relative path to the **Electronics Desktop** installation folder.

2. From the **Command Prompt** window, enter the following command, then press **Enter**:
```
"C:\Program
Files\AnsysEm\v242\Win64\spisim\spisim\modules\ext\SPISimJNI_
WIN64.exe" CalcERL "-v" "CFGFILE="[The direct path to the .cfg
```

`file.]"`



SPISim checks software licenses, calculates ERL, and then outputs the result. In this example, three sets of results are output. The *.cfg* file, which is viewable in any text reader, calls three *.s4p* files.



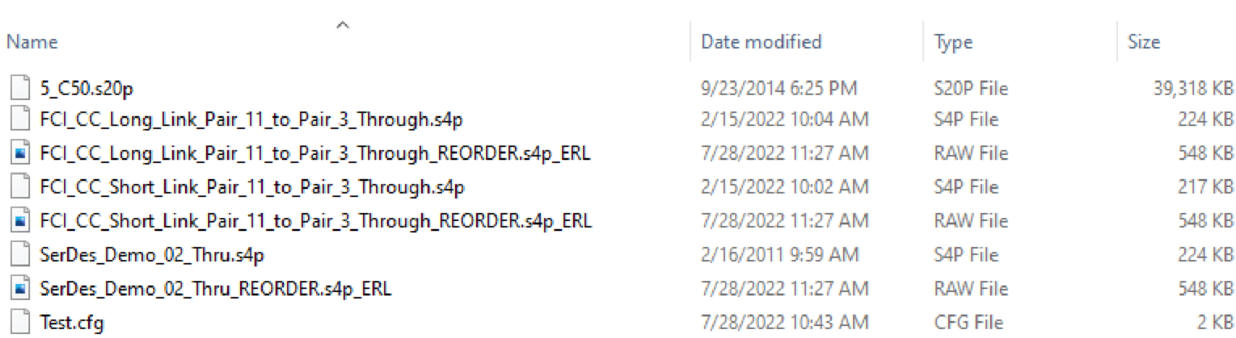
Test - Notepad

File Edit Format View Help

```
##############################################################################
# MODULE: CalcERL
# GENERATED ON 20220728104307
##############################################################################

# INPARRY: INPARRY
INPARRY = C:/ansysdev/20220616_F634386_ERL_wo_COM/20220616_F634386_ERL_wo_COM/TestCase/FCI_CC_Long_Link_
C:/ansysdev/20220616_F634386_ERL_wo_COM/20220616_F634386_ERL_wo_COM/TestCase/FCI_CC_Short_Link_Pair_11_t
C:/ansysdev/20220616_F634386_ERL_wo_COM/20220616_F634386_ERL_wo_COM/TestCase/SerDes_Demo_02_Thru.s4p
# MIXMODE: MIXMODE
MIXMODE = EVENODD
# THRUS4P: THRUS4P
THRUS4P =
# BANDWID: BANDWID
BANDWID = 25.78125G
# TDR_DUR: TDR_DUR
TDR_DUR = 5
# REFIMPD: REFIMPD
REFIMPD = 50
# BINSIZE: BINSIZE
BINSIZE = 1E-5
# SPECBER: SPECBER
SPECBER = 1E-4
# MODTYPE: MODTYPE
MODTYPE = NRZ
# FIXDELY: FIXDELY
FIXDELY = 500p
# INPVOLT: INPVOLT
INPVOLT = 1.0
# TRSTIME: TRSTIME
TRSTIME = 10p
# SIGBETA: SIGBETA
SIGBETA = 1.7E9
# REFLRHO: REFLRHO
REFLRHO = 0.18
# NCYCLES: NCYCLES
NCYCLES = 1000
```

Ln 1, Col 1 | 100% | Unix (LF) | UTF-8

When the process is finished, new *.RAW* waveform files appear in the directory from which the *.s4p* files were called.



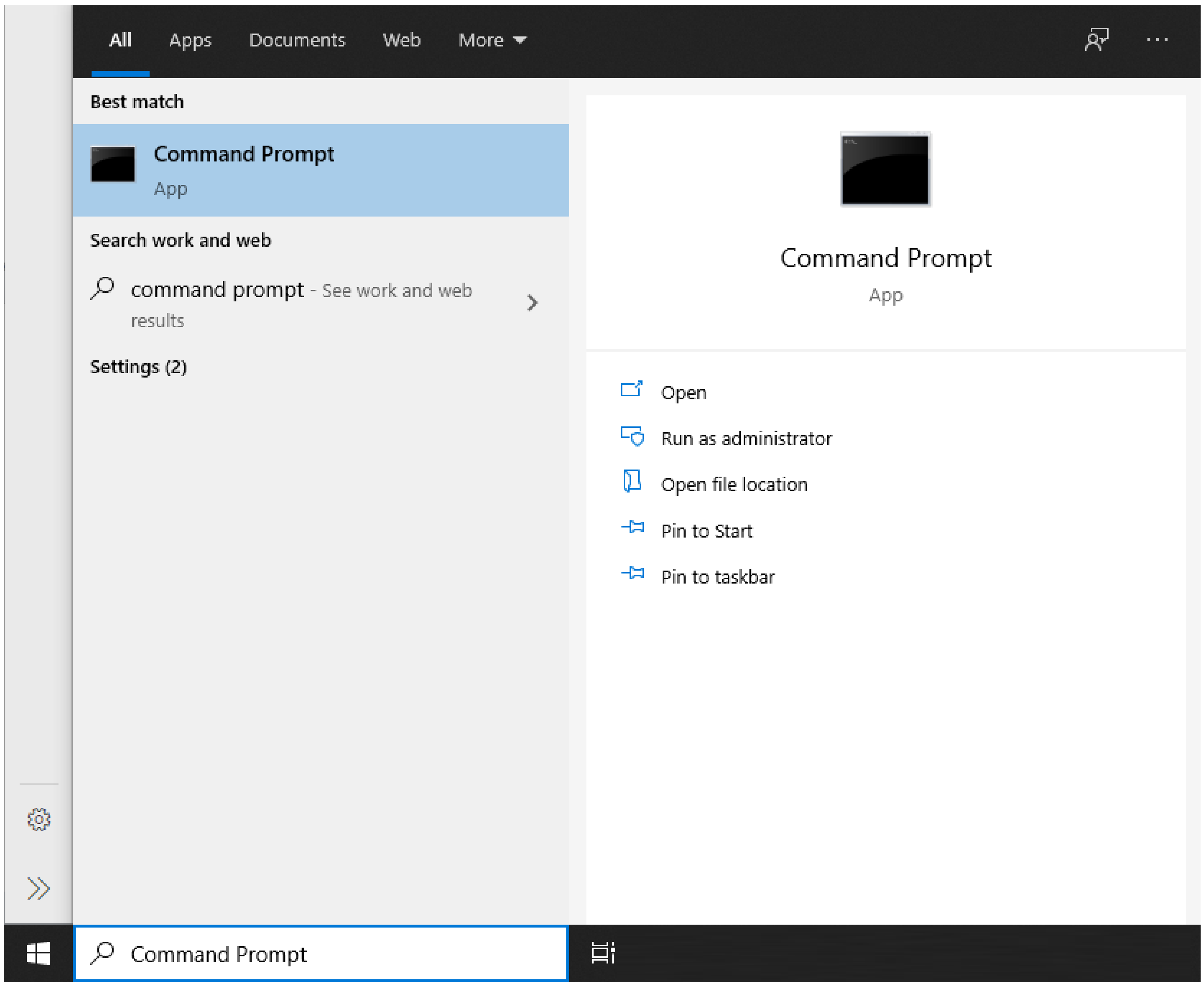
Name	Date modified	Type	Size
5_C50.s20p	9/23/2014 6:25 PM	S20P File	39,318 KB
FCI_CC_Long_Link_Pair_11_to_Pair_3_Through.s4p	2/15/2022 10:04 AM	S4P File	224 KB
FCI_CC_Long_Link_Pair_11_to_Pair_3_Through_REORDER.s4p_ERL	7/28/2022 11:27 AM	RAW File	548 KB
FCI_CC_Short_Link_Pair_11_to_Pair_3_Through.s4p	2/15/2022 10:02 AM	S4P File	217 KB
FCI_CC_Short_Link_Pair_11_to_Pair_3_Through_REORDER.s4p_ERL	7/28/2022 11:27 AM	RAW File	548 KB
SerDes_Demo_02_Thru.s4p	2/16/2011 9:59 AM	S4P File	224 KB
SerDes_Demo_02_Thru_REORDER.s4p_ERL	7/28/2022 11:27 AM	RAW File	548 KB
Test.cfg	7/28/2022 10:43 AM	CFG File	2 KB

\

Overriding ERL Configuration File Parameters

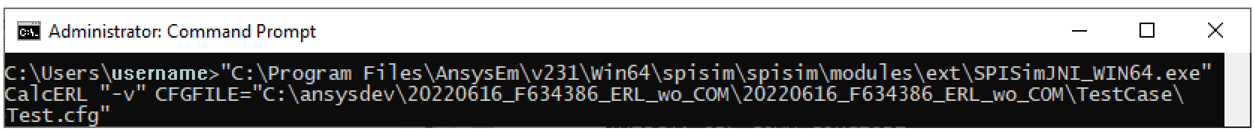
Complete the following steps to override parameter values, substitute input files, or remove input files from ERL calculation conducted via a configuration file from a command line.

1. Run a command prompt (e.g., in a Windows environment, type *Command Prompt* into the **Search Bar** and press **Enter** to open a **Command Prompt** window.



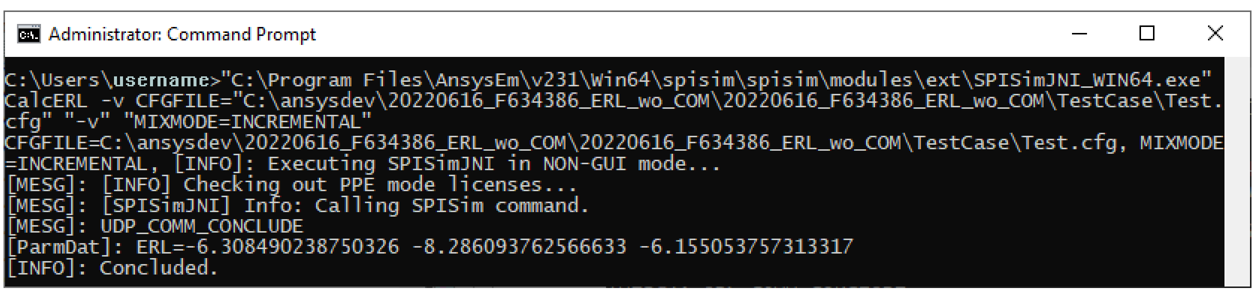
> **Note:** The following example assumes **Electronics Desktop** is installed in the default directory in a Windows operating system environment. Structure the command appropriately for a different directory, path, and/or operating system, as necessary. Do **not** move the ***SPISimJNI_WIN64.exe*** executable file. The executable file relies on the relative path to the **Electronics Desktop** installation folder.

1. From a **Command Prompt** window, enter the following command, but do **not** press **Enter**: `"C:\Program Files\AnsysEm\v242\Win64\spisim\spisim\modules\ext\SPISimJNI_WIN64.exe" CalcERL "-v" "CFGFILE="[The direct path to the .cfg file.]"`

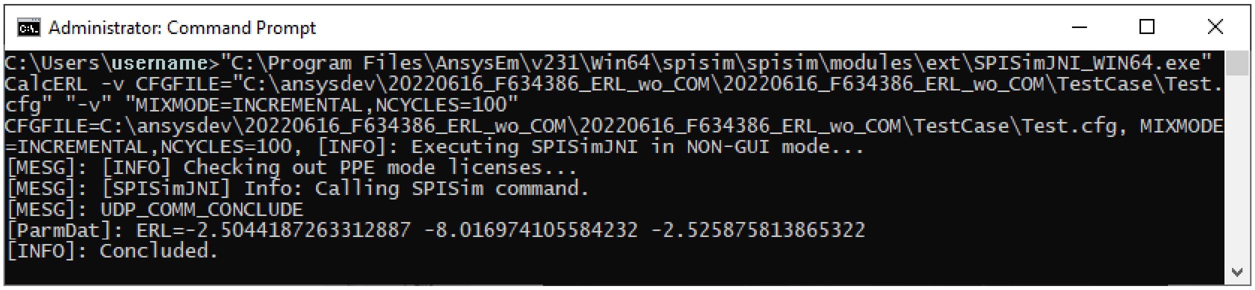


2. To override a configuration file, enter any combination of the following as a suffix (i.e., after the last quotation mark in step 1), then press Enter to run ERL calculation exactly as described in "Importing an ERL Configuration File Using Only a Non-GUI Batch Command Line" on page 15-127.

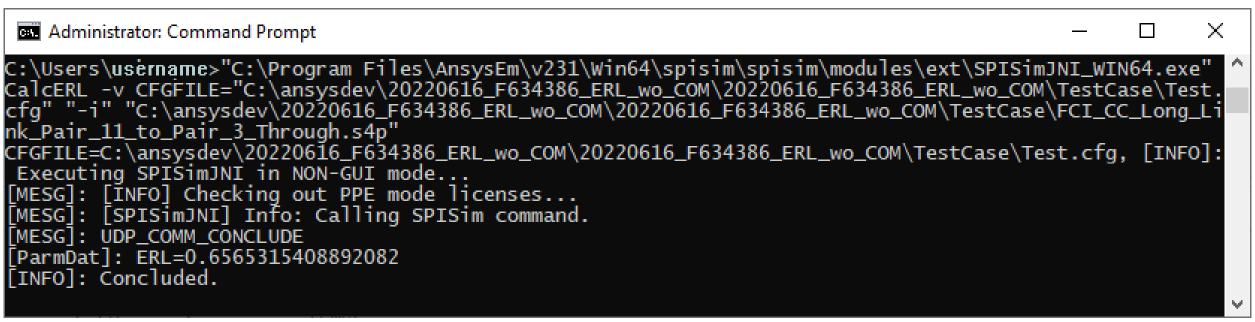
   - Enter `"-v"`, then a variable, and a new value (e.g., enter `"-v" "MIXMODE=INCREMENTAL"` to override EvenOdd mix).



   - Enter `"-v"`, then any number of variables, and new values, separated by commas (e.g., enter `"-v" "MIXMODE=INCREMENTAL,NCYCLES=100"` to override EvenOdd mix and N Cycles).



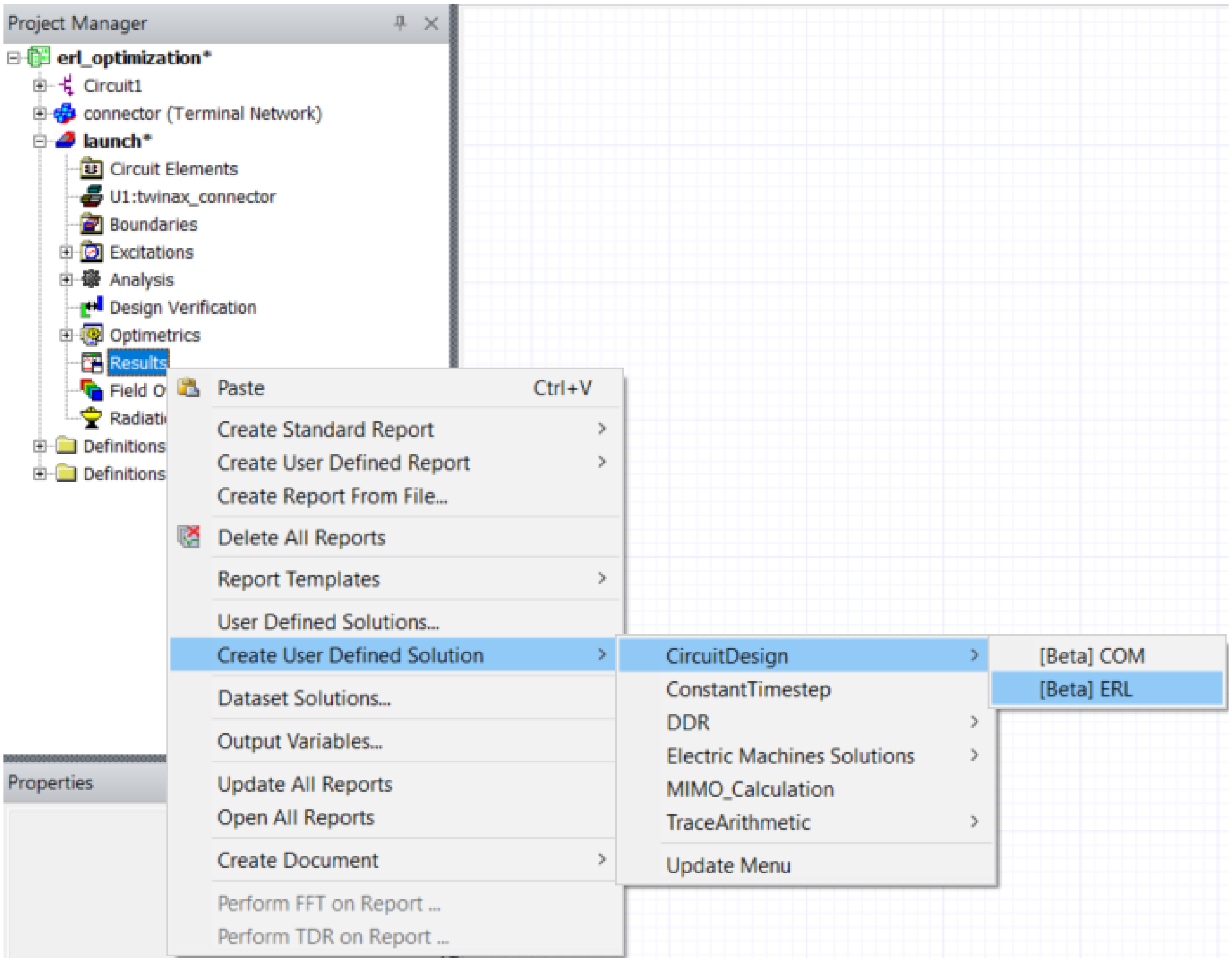
- Enter `-i` and the direct path to a *.sNp* or *.ts* file to override the input file(s) in the configuration file and calculate the override file instead (e.g., enter `"-i" "C:\ansysdev\20220616_F634386_ERL_wo_COM\20220616_F634386_ERL_wo_COM\TestCase\FCI_CC_Long_link_Pair_11_to_Pair_3_Through.s4p"` to override the input file(s) referred to the *.cfg* file).



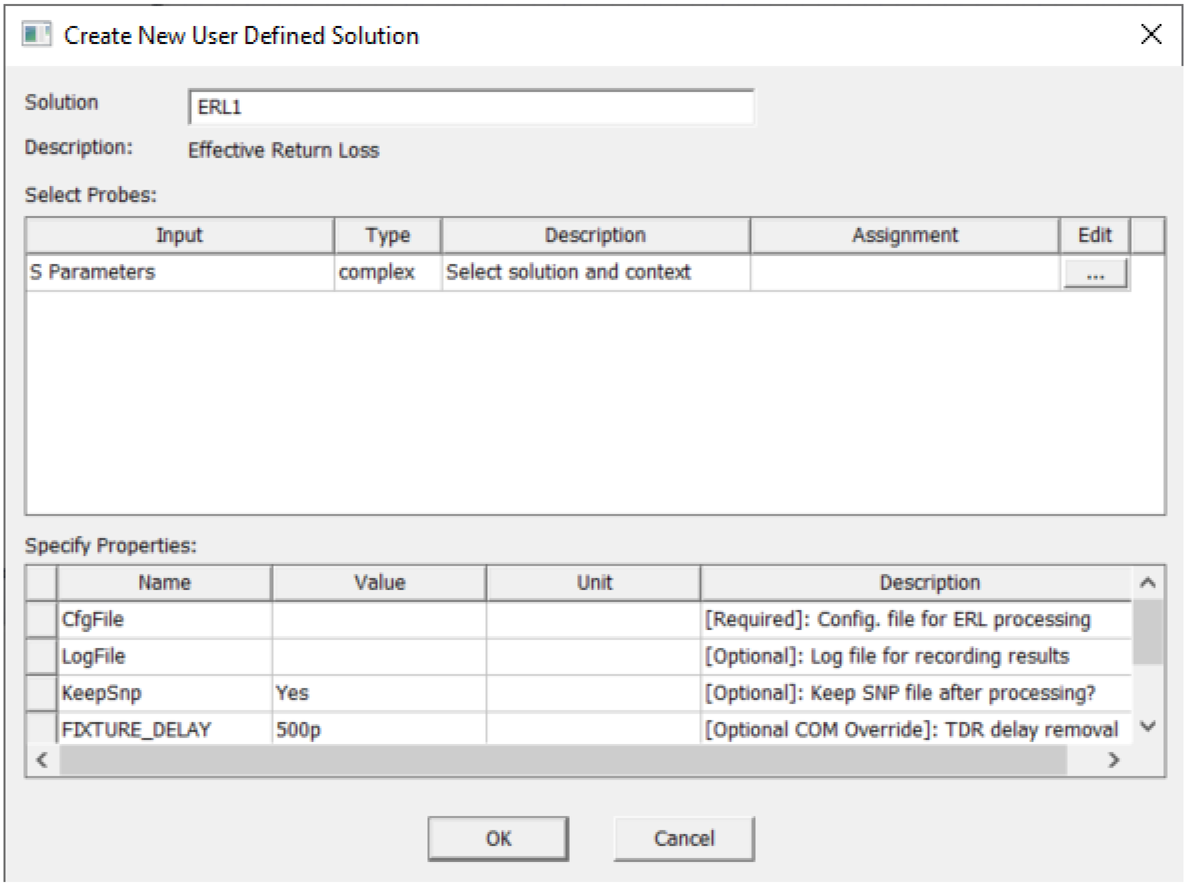
Generate an ERL Calculation Report with a User Defined Solution

Follow these steps to create a user defined solution (UDS) and calculate the effective return loss (ERL) from within **Electronics Desktop** > **HFSS 3D Layout** environment.

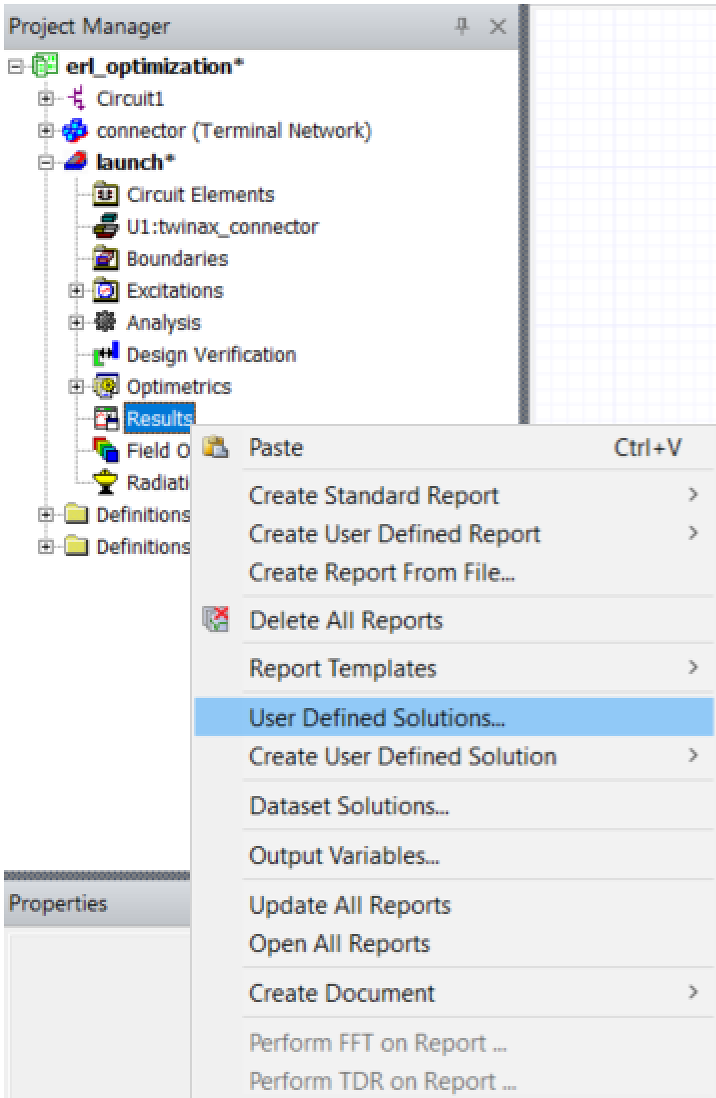
1. From the **Project Manager** window, expand the **Project Tree** > [*active design folder*]. Then right-click **Results** and select **Create User Defined Solution** > **CircuitDesign** > **[Beta] ERL** to open the **Create New User Defined Solution** window.



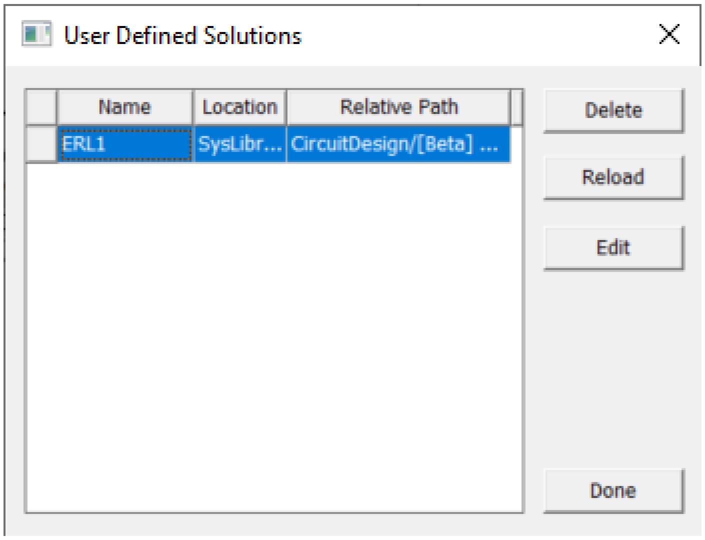
2. Click **OK** to create a new user-defined solution with default settings while simultaneously closing the **Create New User Defined Solution** window.



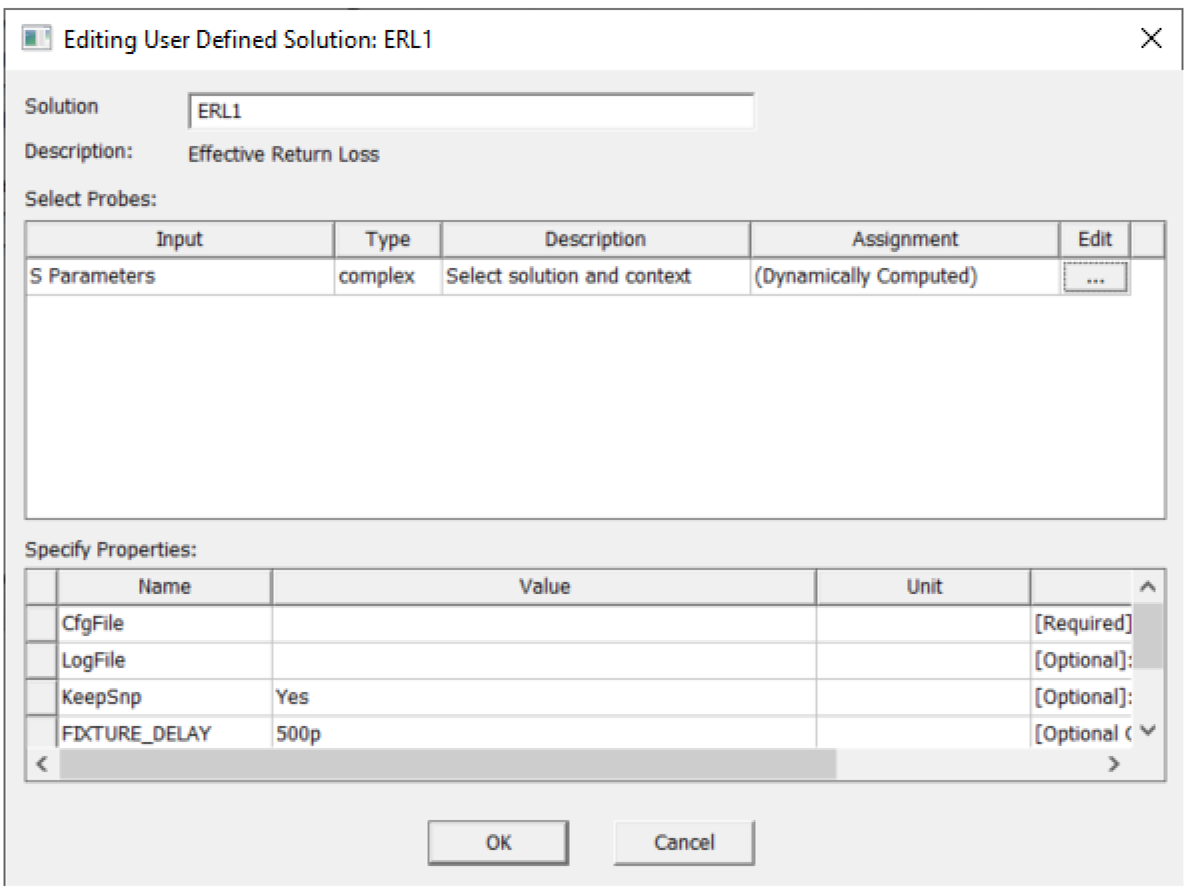
3. From the **Project Manager** window, right-click **Results** and select **User Defined Solutions...** to open the **User Defined Solutions** window.


Project Manager
erl_optimization*
Circuit1
connector (Terminal Network)
launch*
Circuit Elements
U1:twinax_connector
Boundaries
Excitations
Analysis
Design Verification
Optimetrics
Results
Field O
Radiati
Definitions
Definitions
Properties
Paste
Ctrl+V
Create Standard Report
Create User Defined Report
Create Report From File...
Delete All Reports
Report Templates
User Defined Solutions...
Create User Defined Solution
Dataset Solutions...
Output Variables...
Update All Reports
Open All Reports
Create Document
Perform FFT on Report ...
Perform TDR on Report ...

4. Select the new solution on the list (e.g., **ERL1**).



5. Click **Edit** to open the **Editing User Defined Solution** window.

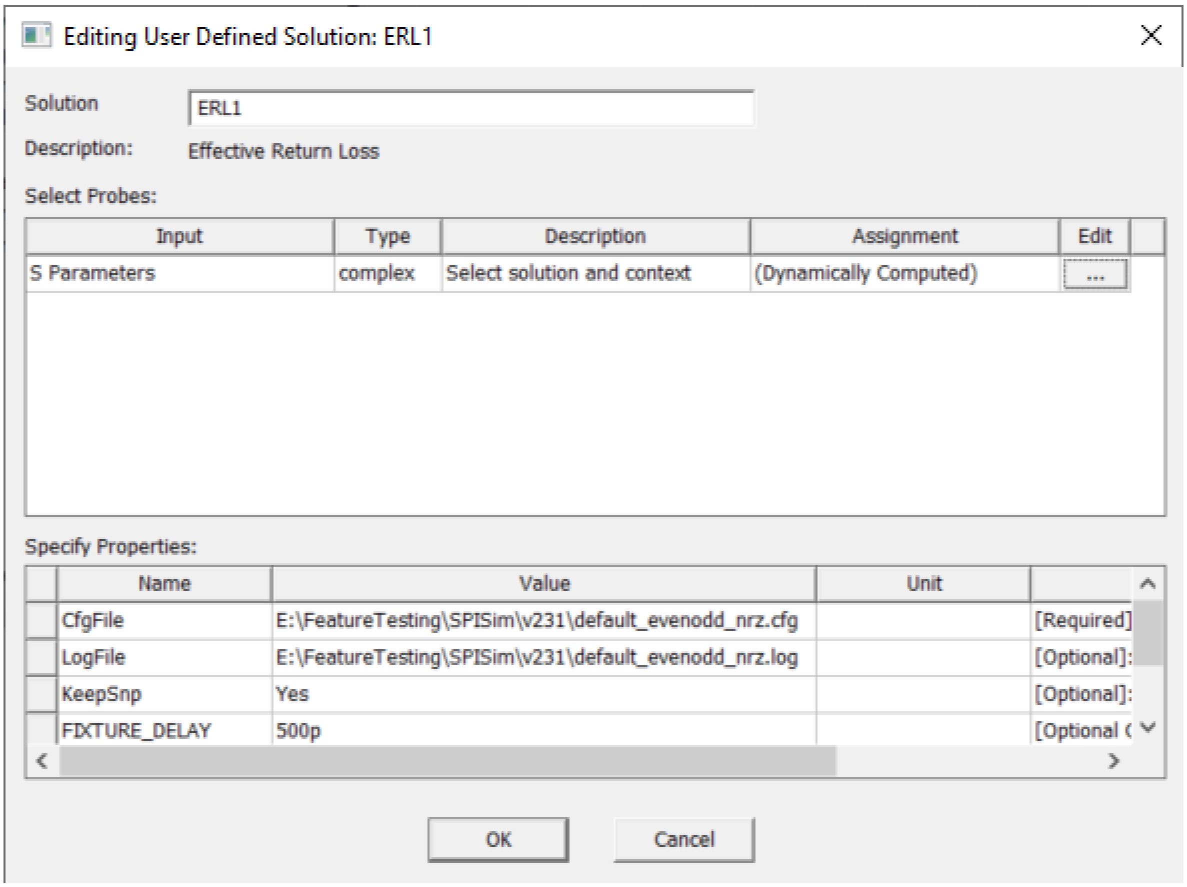


6. In the **Specify Properties** table, make the following changes:

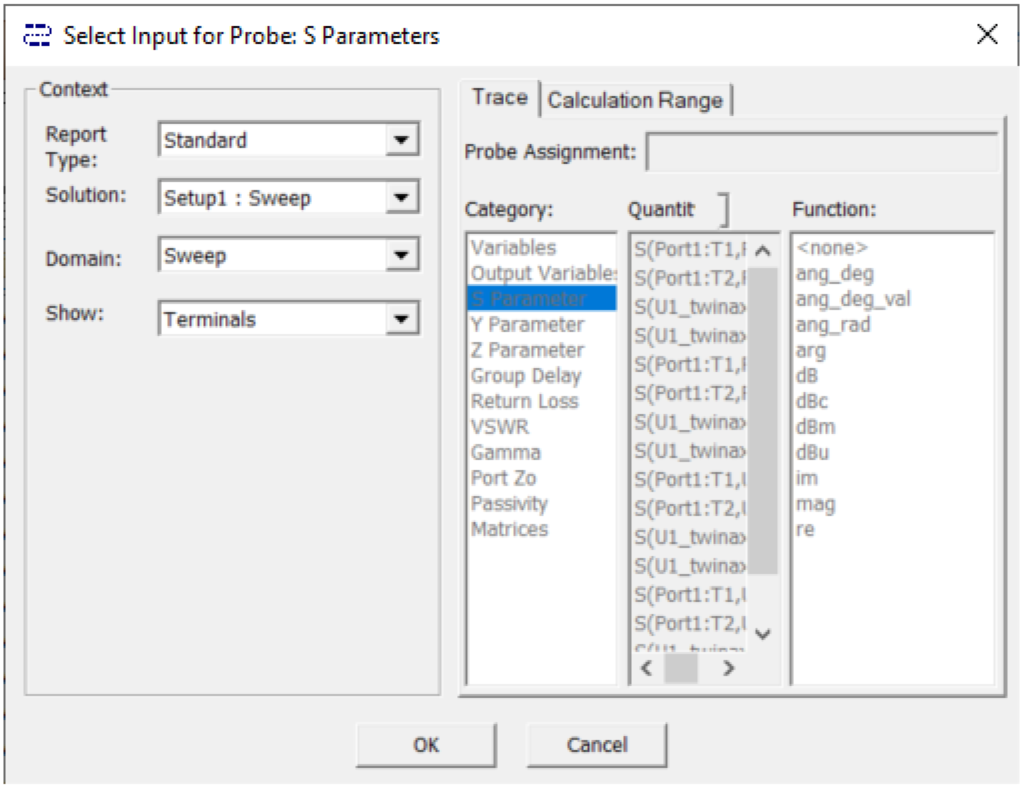
- Enter the path to the relevant *.cfg* file (e.g., **E:\FeatureTesting\SPISim\v242\default_ evenodd_nrz.cfg**) in the **CfgFile Value** field.

> **Note:** The path can include spaces but can **not** include commas or other punctuation.

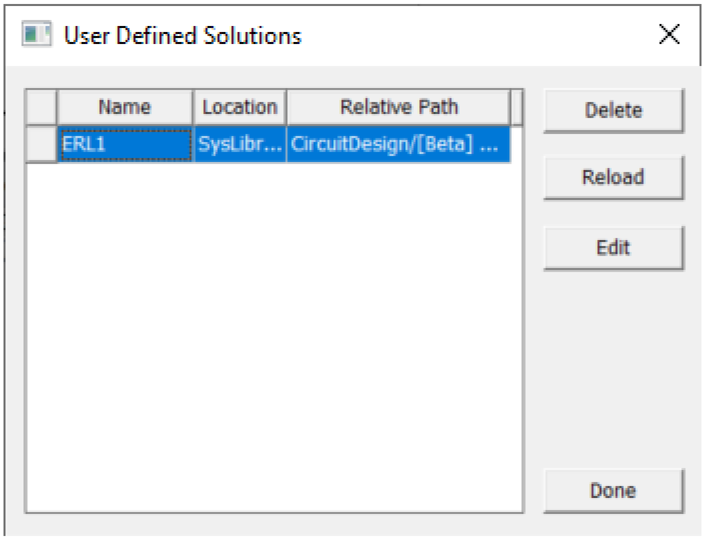
- If appropriate, enter the path to a *.log* file (e.g., **E:\FeatureTesting\SPISim\v242\default_ evenodd_nrz.log**) in the **LogFile Value** field.



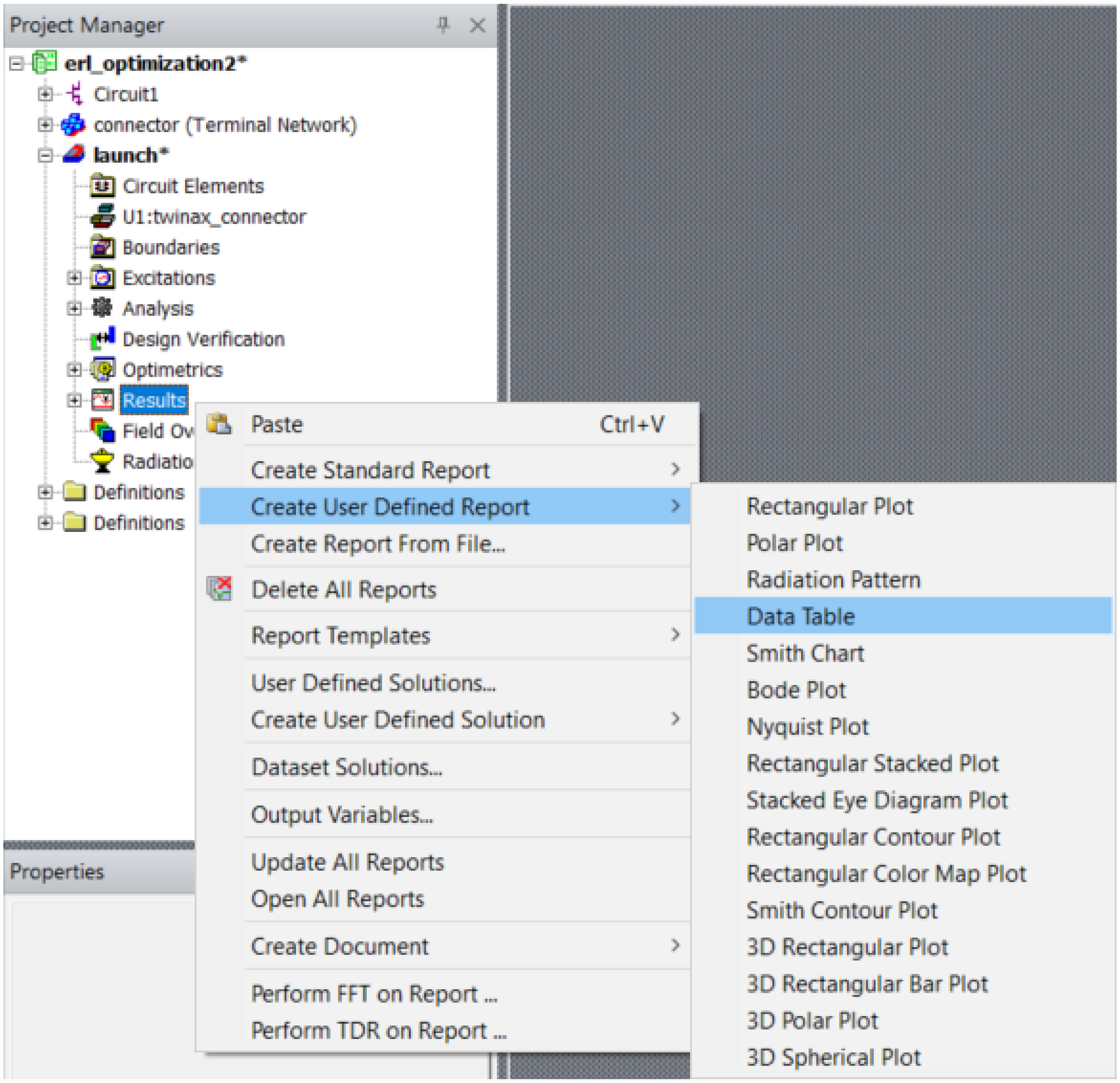
7. From the **Select Probes** table, click **...** in the **Edit** column to open the **Select Input for Probe: S Parameters** window.



8. Ensure that **Terminals** is selected on the **Context** group box > **Show** drop-down menu.
9. Make any additional changes, as appropriate. Then click **OK** to close the **Select Input for Probe: S Parameters** window.
10. Click **OK** to close the **Editing User Defined Solution** window.
11. Ensure the new solution (e.g., **ERL1**) is still highlighted in the **User Defined Solutions** window. Then click **Reload**.

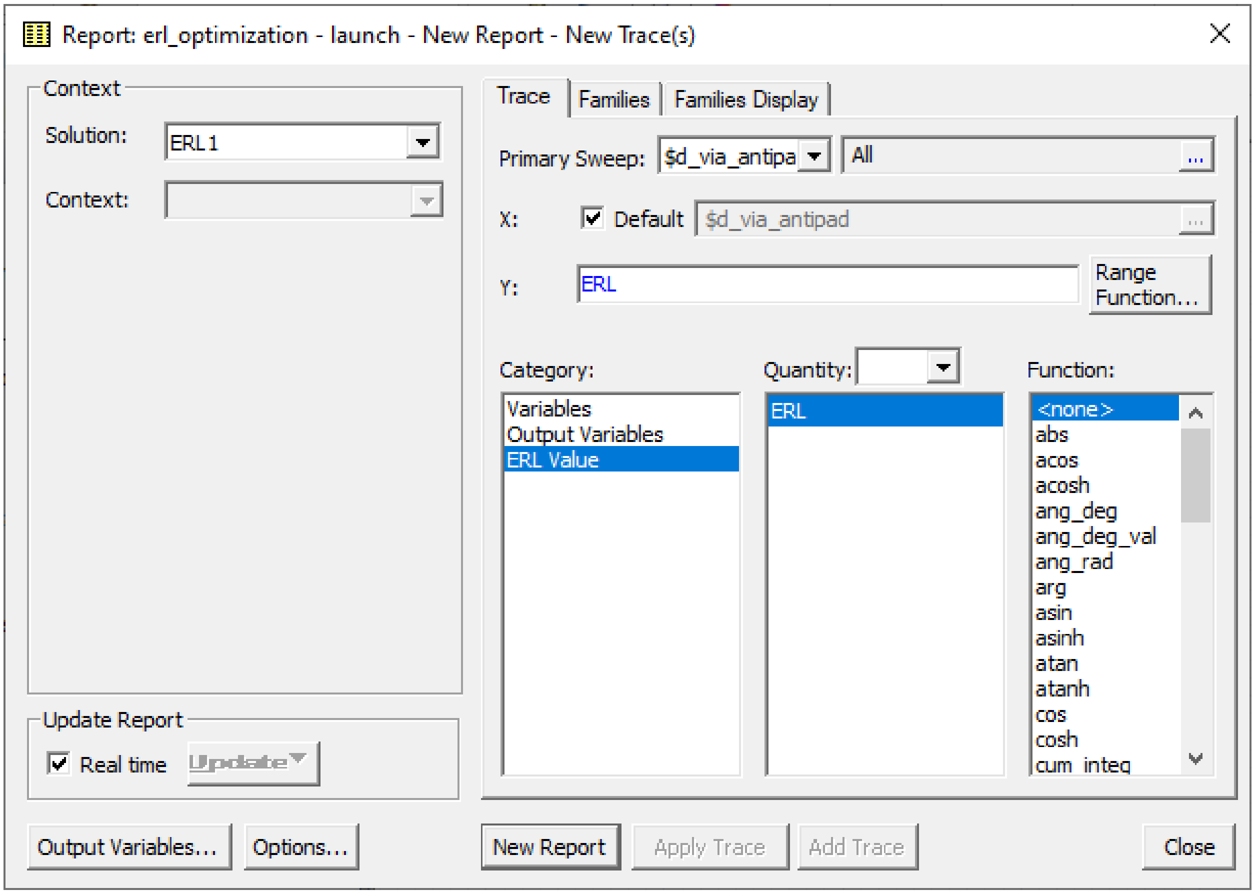


12. Click **Done** to close the **User Defined Solutions** window.
13. From the **Project Manager** window, right-click **Results** and select **Create User Defined Report** > **Data Table** to open a **Report** window to the **Trace** tab.

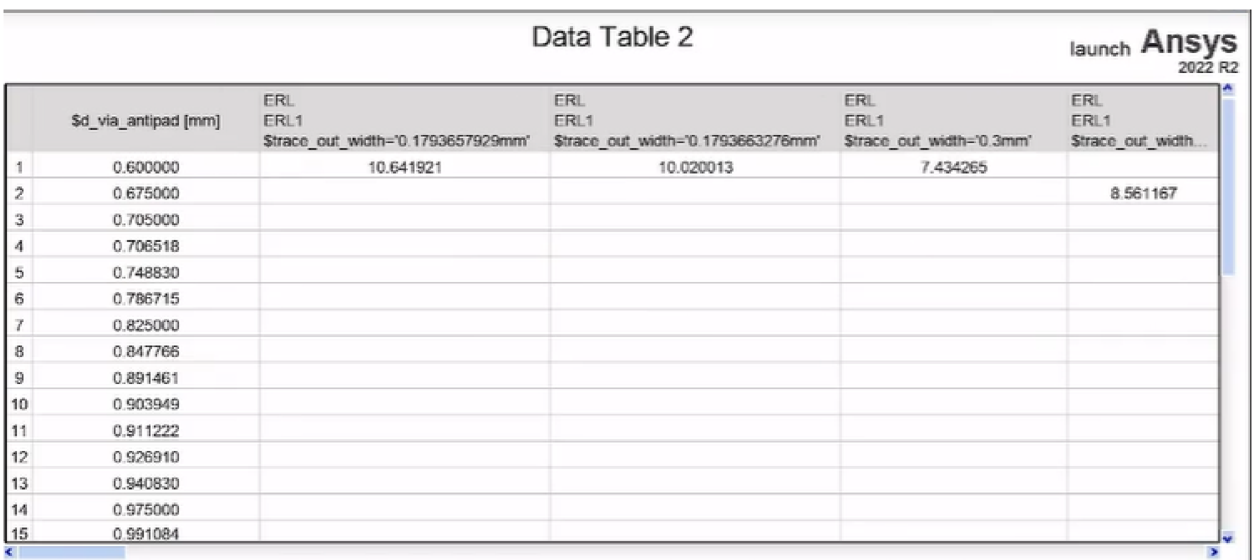


14. From the **Report** window, ensure the following settings are selected:

- From the **Context** group box, the appropriate **ERL** solution is selected on the **Solution** drop-down menu (i.e., **ERL1**).
- From the **Trace** tab, **ERL Value** is selected on the **Category** list.
- **ERL** is selected on the **Quantity** list.



15. Click **New Report** and the report appears.



	$d_via_antipad [mm]	ERL ERL1 $trace_out_width='0.1793657929mm'	ERL ERL1 $trace_out_width='0.1793663276mm'	ERL ERL1 $trace_out_width='0.3mm'	ERL ERL1 $trace_out_width...
1	0.600000	10.641921	10.020013	7.434265	
2	0.675000				8.561167
3	0.705000				
4	0.706518				
5	0.748830				
6	0.786715				
7	0.825000				
8	0.847766				
9	0.891461				
10	0.903949				
11	0.911222				
12	0.926910				
13	0.940830				
14	0.975000				
15	0.991084				

Channel Operating Margin Analysis

Follow the instructions on the following pages to generate Channel Operating Margin (COM) analysis reports through various methods.

" Generating a COM Analysis Report From the SPISim User Interface " below

" Generating a COM Analysis Report Using a Non-GUI Batch Command Line" on page 15-159

" Generating a COM Analysis Report Using IBIS-Based Transition Data" on page 15-163

"COM Analysis Report Summary" on page 15-175

" UDO/UDS-Based COM/ERL Calculations for HFSS-Based Solutions" on page 15-202

Generating a COM Analysis Report From the SPISim User Interface

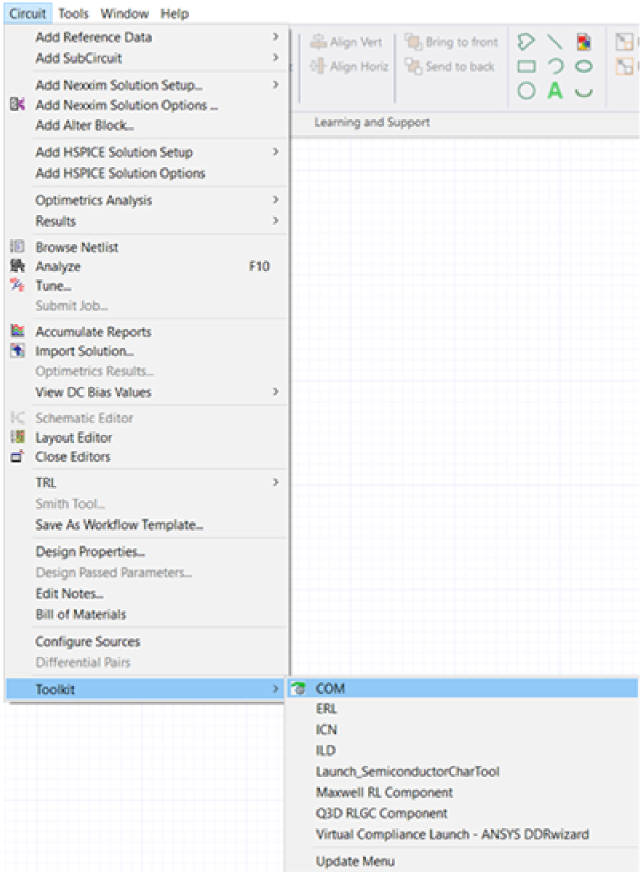
Channel operating margin (COM) analysis generates a *.csv* report file and binary waveform plots from analyzed data in a streamlined format. **Electronic Desktop 23.2** supports COM keywords up to V3.40 (with some experimental flags excluded, listed at the bottom of this page).

A COM analysis report can also be generated outside of the SPISim User Interface. Refer to " Generating a COM Analysis Report Using a Non-GUI Batch Command Line" on page 15-159 and " UDO/UDS-Based COM/ERL Calculations for HFSS-Based Solutions" on page 15-202 for details.

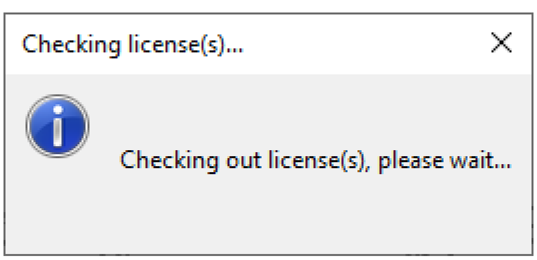
To begin the process of generating a COM analysis report via the SPISim User Interface, complete the following steps to open the **Config. and generate COM report ...** window. Alternatively, refer to the animated demonstration following these instructions.

From Electronics Desktop

From the **Circuit** menu, select **Toolkit** > **COM**.

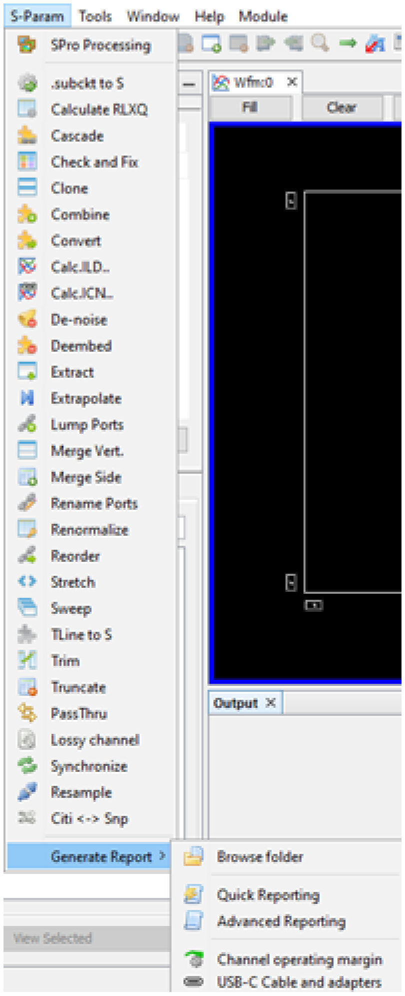


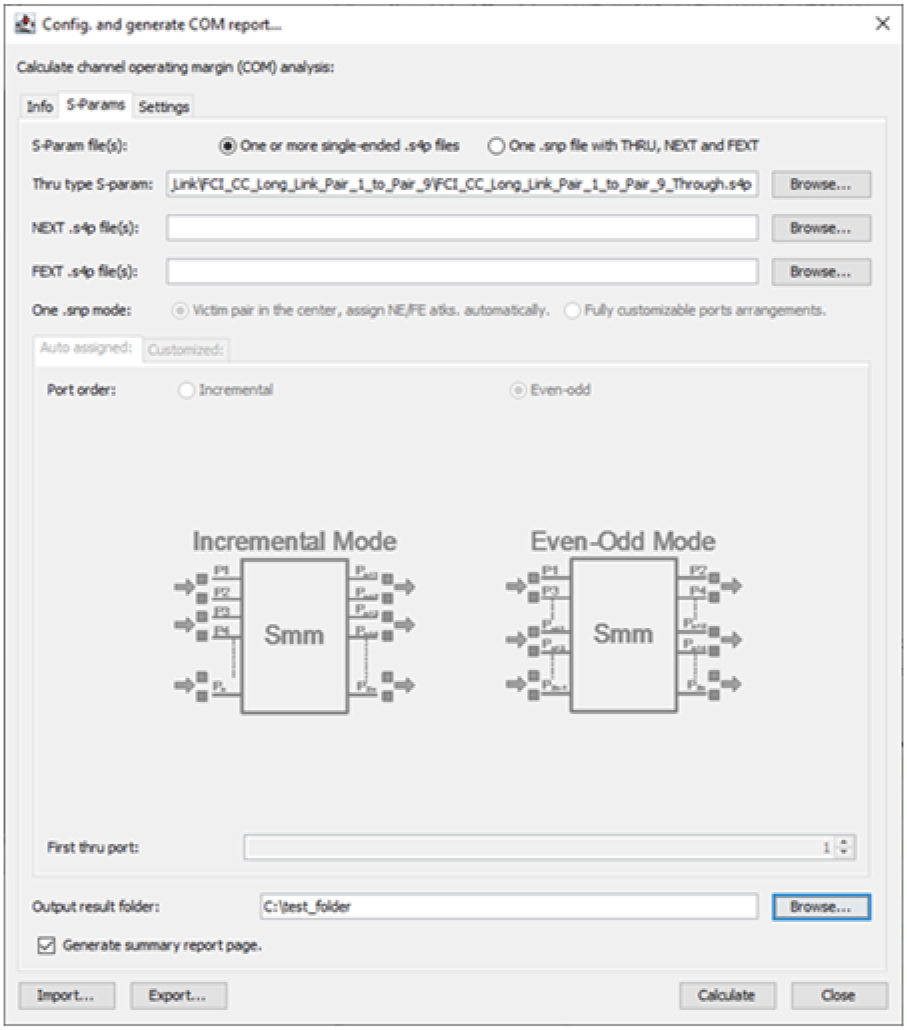
Before the **Config. and generate COM report ...** window opens, a **Checking License(s) ...** window opens.



From SPISim

From the **S-Param** menu, select **Generate Report** > **Channel operating margin** to open the **Config. and generate COM report ...** window.


S-Param
Tools
Window
Help
Module
SPro Processing
.subckt to S
Calculate RLXQ
Cascade
Check and Fix
Clone
Combine
Convert
Calc.ILD..
Calc.ICN..
De-noise
Deembed
Extract
Extrapolate
Lump Ports
Merge Vert.
Merge Side
Rename Ports
Renormalize
Reorder
Stretch
Sweep
TLine to S
Trim
Truncate
PassThru
Lossy channel
Synchronize
Resample
Citi <-> Snp
Generate Report
Browse folder
Quick Reporting
Advanced Reporting
Channel operating margin
USB-C Cable and adapters
View Selected
Wfm:0
Fill
Clear
Output

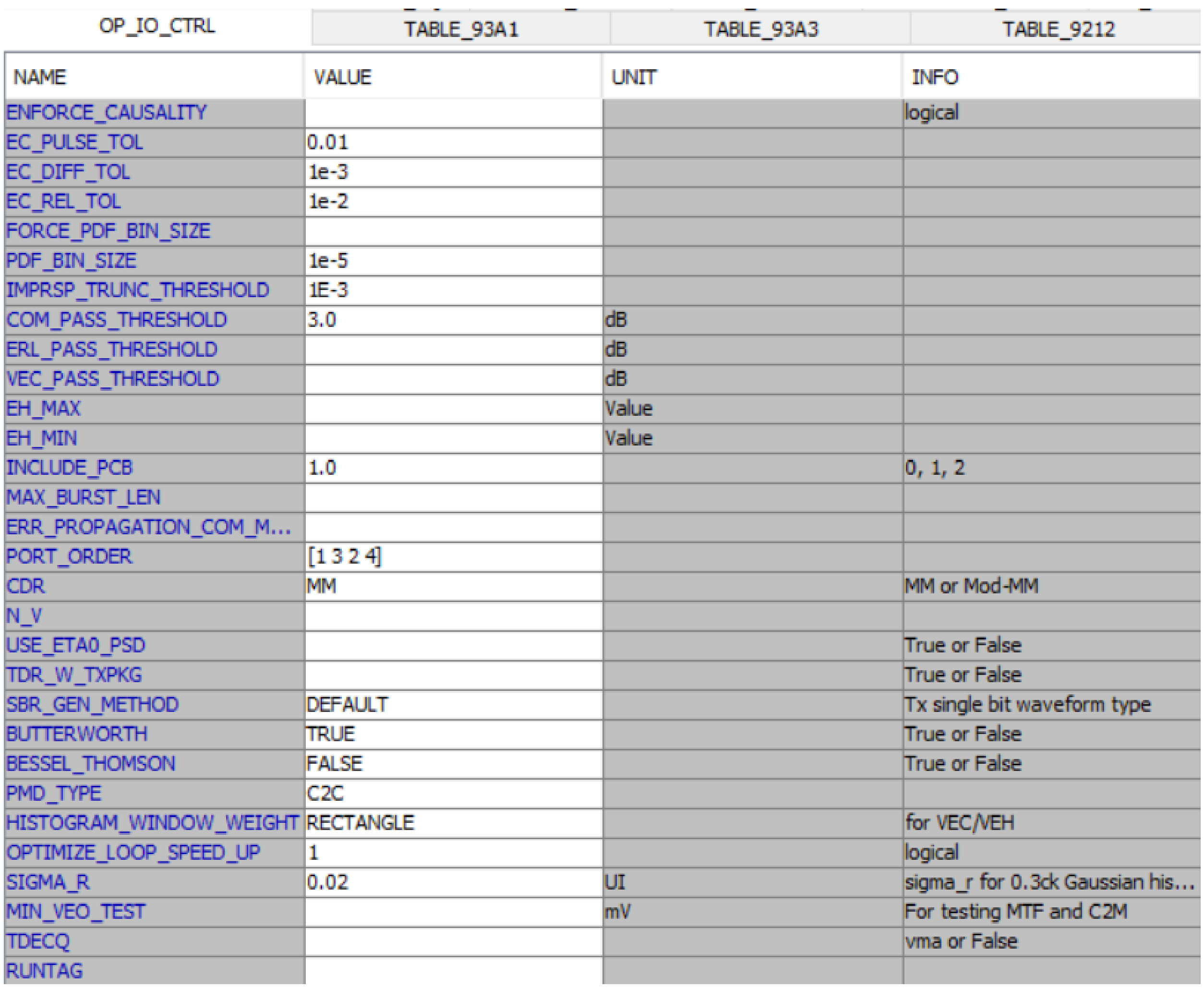


Follow the steps below to configure and then generate a COM analysis report.

1. Click the **Browse ...** button in line with the **Thru type S-param** field to open an explorer window.
2. Navigate to and choose an *.s*N*p* file to include it in the COM analysis report. Then click **Open**.
3. Click the **Browse ...** button in line with the **Output result folder** field to open an explorer window. Choose a location to save the COM analysis report output files, which includes *.raw*, *.csv*, and *.png* files that remain accessible to the user outside of the context of the COM analysis report.
4. Click **Open**.

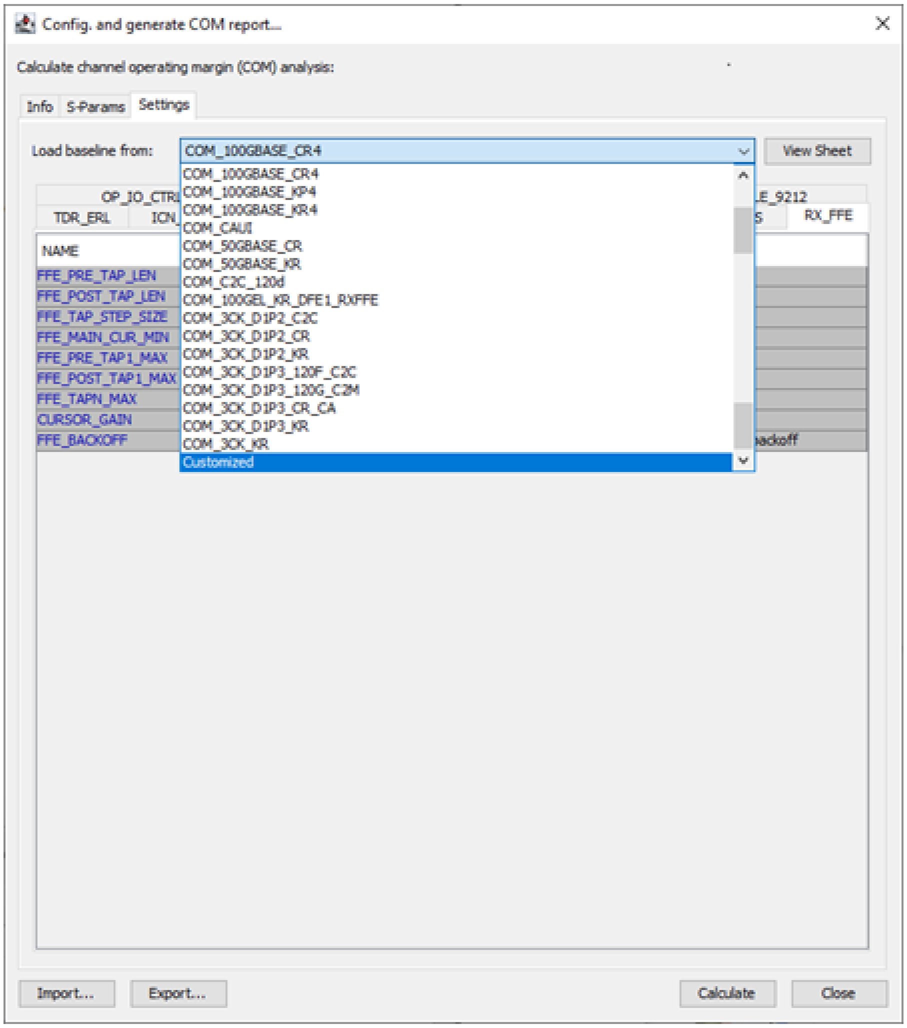
5. Before continuing, ensure that the **Generate summary report page.** check box, beneath the **Output result folder** field, is checked (checked by default).

6. Click the **Settings** tab to view the current configuration of the analyzed data.

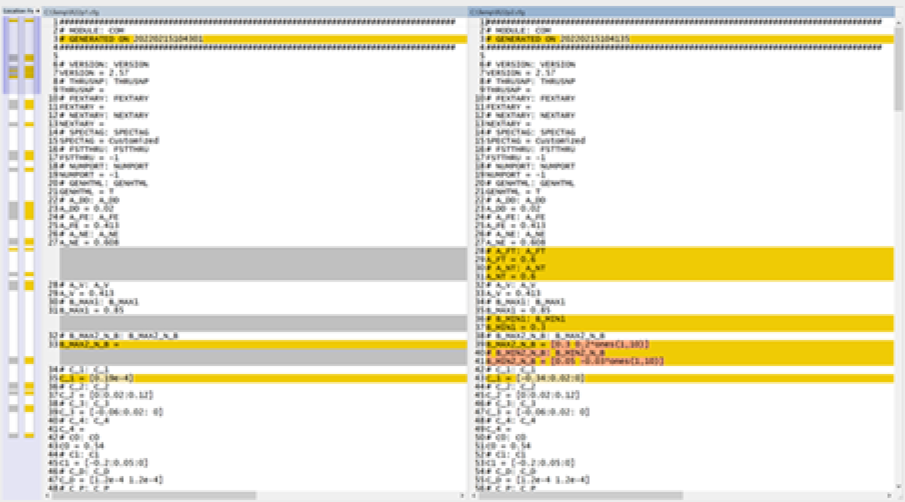


NAME	VALUE	UNIT	INFO
ENFORCE_CAUSALITY			logical
EC_PULSE_TOL	0.01		
EC_DIFF_TOL	1e-3		
EC_REL_TOL	1e-2		
FORCE_PDF_BIN_SIZE			
PDF_BIN_SIZE	1e-5		
IMPRSP_TRUNC_THRESHOLD	1E-3		
COM_PASS_THRESHOLD	3.0	dB	
ERL_PASS_THRESHOLD		dB	
VEC_PASS_THRESHOLD		dB	
EH_MAX		Value	
EH_MIN		Value	
INCLUDE_PCB	1.0		0, 1, 2
MAX_BURST_LEN			
ERR_PROPAGATION_COM_M...			
PORT_ORDER	[1 3 2 4]		
CDR	MM		MM or Mod-MM
N_V			
USE_ETA0_PSD			True or False
TDR_W_TXPKG			True or False
SBR_GEN_METHOD	DEFAULT		Tx single bit waveform type
BUTTERWORTH	TRUE		True or False
BESSEL_THOMSON	FALSE		True or False
PMD_TYPE	C2C		
HISTOGRAM_WINDOW_WEIGHT	RECTANGLE		for VEC/VEH
OPTIMIZE_LOOP_SPEED_UP	1		logical
SIGMA_R	0.02	UI	sigma_r for 0.3ck Gaussian his...
MIN_VEO_TEST		mV	For testing MTF and C2M
TDECQ			vma or False
RUNTAG			

If necessary, choose another baseline configuration on the **Spec** drop-down menu, or select **Customized** to make individual changes to each field.

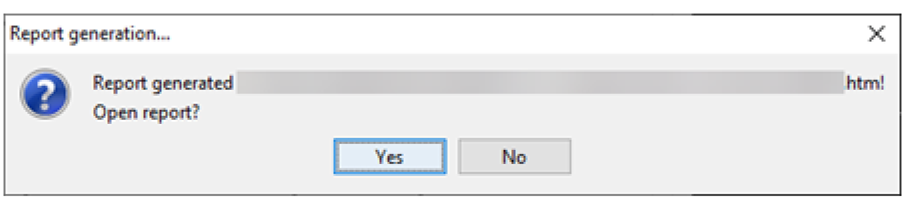

Config. and generate COM report...
Calculate channel operating margin (COM) analysis:
Info
S-Params
Settings
Load baseline from:
COM_100GBASE_CR4
View Sheet
COM_100GBASE_CR4
COM_100GBASE_KP4
COM_100GBASE_KR4
COM_CAUI
COM_50GBASE_CR
COM_50GBASE_KR
COM_C2C_120d
COM_100GEL_KR_DFE1_RXFFE
COM_3CK_D1P2_C2C
COM_3CK_D1P2_CR
COM_3CK_D1P2_KR
COM_3CK_D1P3_120F_C2C
COM_3CK_D1P3_120G_C2M
COM_3CK_D1P3_CR_CA
COM_3CK_D1P3_KR
COM_3CK_KR
Customized
TDR_ERL
RX_FFE
NAME
FFE_PRE_TAP_LEN
FFE_POST_TAP_LEN
FFE_TAP_STEP_SIZE
FFE_MAIN_CUR_MIN
FFE_PRE_TAP1_MAX
FFE_POST_TAP1_MAX
FFE_TAPN_MAX
CURSOR_GAIN
FFE_BACKOFF
Import...
Export...
Calculate
Close

Note: The configurable settings are extensive, see Generate A COM Analysis Report - Settings Tab. If the user is upgrading from a previous version of *Electronic Desktop*, and currently has both versions installed, they can **Export** their configuration settings (e.g., *.cfg* file) from both versions and compare them. New parameters will be clearly visible in this side by side perspective. While **Electronic Desktop** supports COM keywords up to V3.40, not all keywords will be defined. Undefined keywords will function at their default values.



For more information, refer to the IEEE 802 LAN/MAN Standards Committee website and the IEEE P802.3ck 100 Gb/s, 200 Gb/s, and 400 Gb/s Electrical Interfaces Task Force Home Page.

7. Click **Calculate** and the output folder selected in **Step 3** populates with .raw (i.e., Berkley spice), *.csv*, and *.png* files.
8. When the report is ready, a **Report generation ...** window appears. Click **Yes** to open the COM analysis report in the system's default browser.



Continue to COM Analysis Report Summary

> **Note:** New features have been added to SPISim that are not mentioned in the following video tutorial. Relevant features, such as the number of configurable parameters, their locations within the Settings tab, and the expanded subtabs available, are documented in the prior instructions.

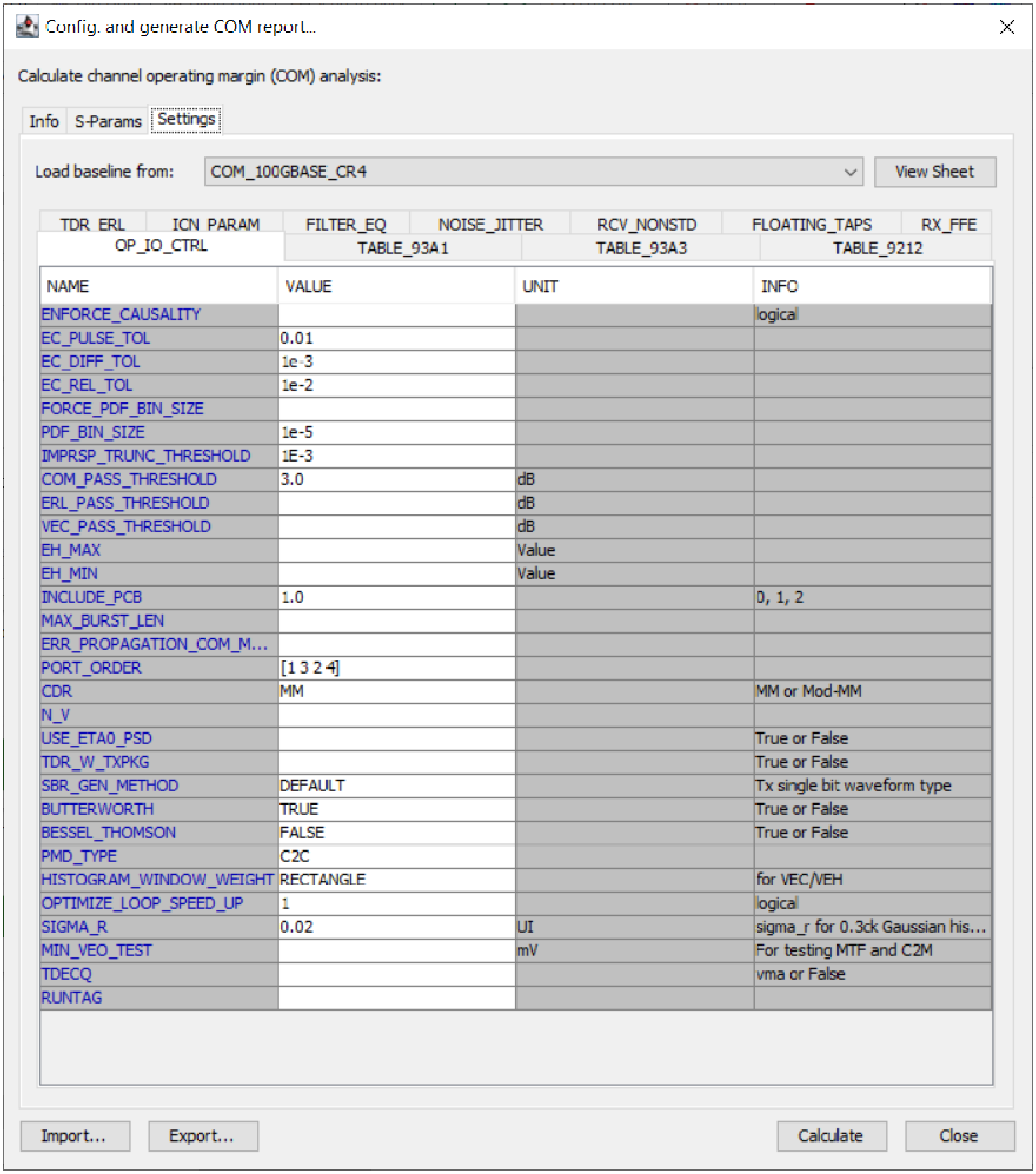
Excluded Experimental Flags

> **Note:** Experimental flags excluded from the V3.40 COM keywords are: RILN related flags (COMPUTE_RILN, COMPUTE_TDLIN, RILN) and TDMODE, Save_TD.

Generating a COM Analysis Report From the SPISim User Interface - Settings Tab

Channel operating margin (COM) analysis generates a *.csv* report file and binary waveform plots from analyzed data in a streamlined format. **Electronic Desktop 2023 R2 and following releases** support COM keywords up to V3.40 (with some experimental flags excluded, listed at the bottom of this page).

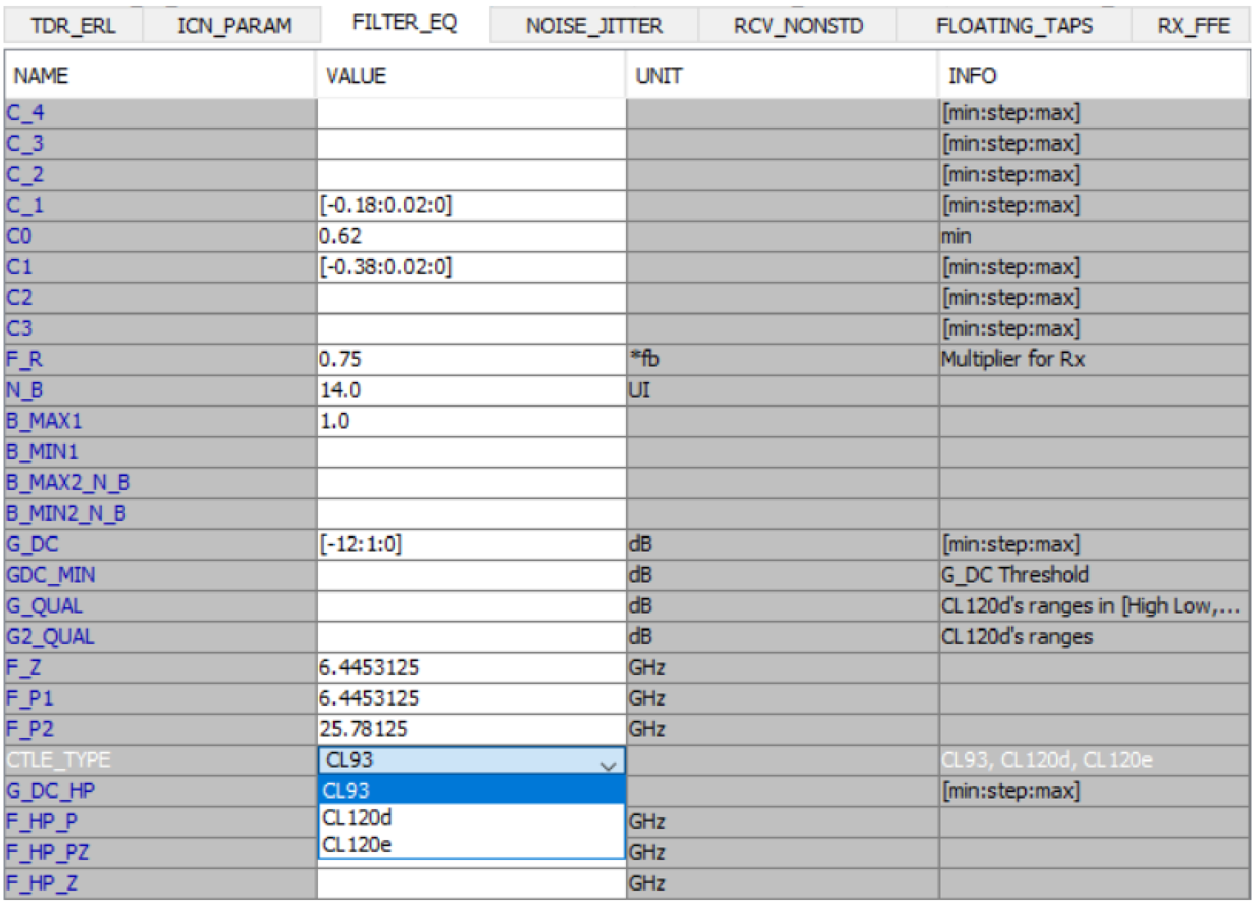
Click the **Settings** tab to view the current configuration of the analyzed data.



For an exhaustive list of the parameters, descriptions, and usage notes, refer to the COM Analysis Keywords reference page.

The settings are divided in eleven subtabs (i.e., *TDR_ERL*, *ICN_PARAM*, *FILTER_EQ*, *NOISE_JITTER*, *RCV_NONSTD*, *FLOATING_TAPS*, *RX_FFE*, *OP_IO_CTRL*, *TABLE_93A1*, *TABLE_93A3*, and *TABLE_9212*). Beside each parameter **NAME** is a configurable **Value** field, **UNIT** field (if applicable) and useful **INFO** field. These parameters include:

- **C_4** and **C_3** TXFFE taps and **CTLE_TYPE** (on the *FILTER_EQ* subtab). **CTLE_TYPE** includes three choices on the **VALUE** field drop-down menu (i.e., **CL93**, **CL120d**, and **CL120e**)



TDR_ERL | ICN_PARAM | FILTER_EQ | NOISE_JITTER | RCV_NONSTD | FLOATING_TAPS | RX_FFE

NAME	VALUE	UNIT	INFO
C_4			[min:step:max]
C_3			[min:step:max]
C_2			[min:step:max]
C_1	[-0.18:0.02:0]		[min:step:max]
C0	0.62		min
C1	[-0.38:0.02:0]		[min:step:max]
C2			[min:step:max]
C3			[min:step:max]
F_R	0.75	*fb	Multiplier for Rx
N_B	14.0	UI	
B_MAX1	1.0		
B_MIN1			
B_MAX2_N_B			
B_MIN2_N_B			
G_DC	[-12:1:0]	dB	[min:step:max]
GDC_MIN		dB	G_DC Threshold
G_QUAL		dB	CL120d's ranges in [High Low,...
G2_QUAL		dB	CL120d's ranges
F_Z	6.4453125	GHz	
F_P1	6.4453125	GHz	
F_P2	25.78125	GHz	
CTLE_TYPE	CL93		CL93, CL120d, CL120e
G_DC_HP	CL93		[min:step:max]
F_HP_P	CL120d	GHz	
F_HP_PZ	CL120e	GHz	
F_HP_Z		GHz	

- **FFE_BackOff** (on the *RX_FFE* subtab) compels RxFEE post taps to decrement up to the increment assigned in the **FFE_BackOff Value** field, and to scan for FFE settings based on DFE probes. The best resolution will be used as the tap selection for COM calculation (default is 4, i.e., deactivated). The *RX_FFE* subtab also includes parameters to optimize the pulse before DFE (i.e., **FFE_PRE_TAP_LEN** and **FFE_POST_TAP_LEN**, configurable pre and post taps for RxFFe). The subsequent parameters provide incremental and thresholds to qualify solutions (e.g., **FFE_TAP_STEP_SIZE**, **FFE_MAIN_CUR_MIN**, etcetera).

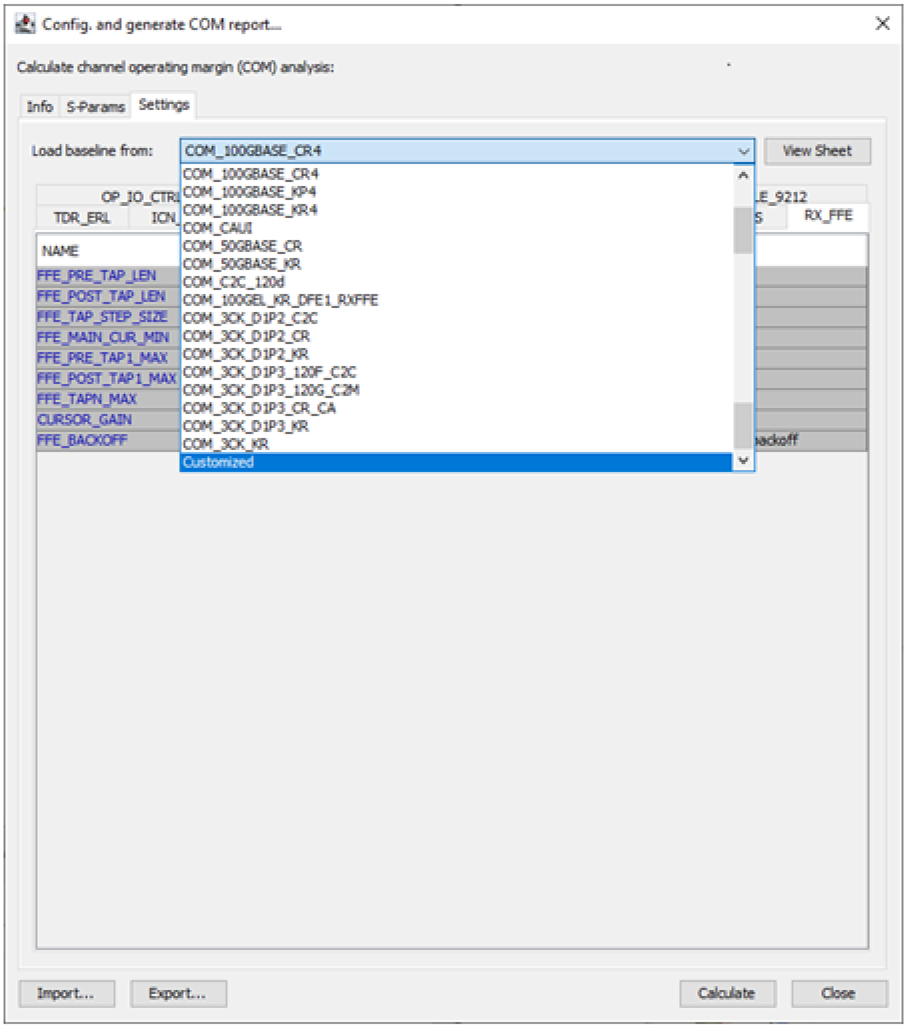
> **Note:** For more information, refer to the IEEE 802 LAN/MAN Standards Committee COM 2.51 with rxFFE Updates presentation files.

OP_IO_CTRL | TABLE_93A1 | TABLE_93A3 | TABLE_9212

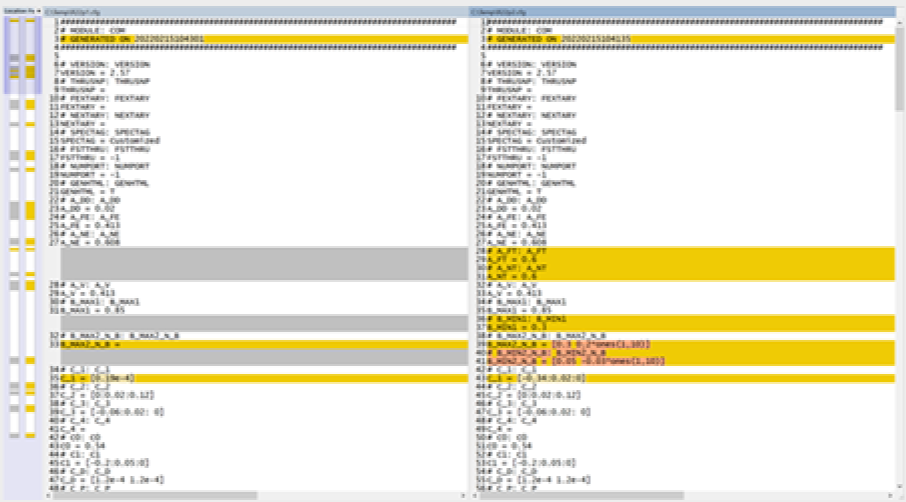
TDR_ERL | ICN_PARAM | FILTER_EQ | NOISE_JITTER | RCV_NONSTD | FLOATING_TAPS | RX_FFE

NAME	VALUE	UNIT	INFO
FFE_PRE_TAP_LEN		UI	
FFE_POST_TAP_LEN		UI	
FFE_TAP_STEP_SIZE	0		
FFE_MAIN_CUR_MIN	0.7		
FFE_PRE_TAP1_MAX	0.7		
FFE_POST_TAP1_MAX	0.7		
FFE_TAPN_MAX	0.7		
CURSOR_GAIN		dB	
FFE_BACKOFF	4		No. of taps to backoff

> **Note:** There are various options for baseline configuration.


Config. and generate COM report...
Calculate channel operating margin (COM) analysis:
Info
S-Params
Settings
Load baseline from:
COM_100GBASE_CR4
View Sheet
COM_100GBASE_CR4
COM_100GBASE_KP4
COM_100GBASE_KR4
COM_CAUI
COM_50GBASE_CR
COM_50GBASE_KR
COM_C2C_120d
COM_100GEL_KR_DFE1_RXFFE
COM_3CK_D1P2_C2C
COM_3CK_D1P2_CR
COM_3CK_D1P2_KR
COM_3CK_D1P3_120F_C2C
COM_3CK_D1P3_120G_C2M
COM_3CK_D1P3_CR_CA
COM_3CK_D1P3_KR
COM_3CK_KR
Customized
TDR_ERL
RX_FFE
NAME
FFE_PRE_TAP_LEN
FFE_POST_TAP_LEN
FFE_TAP_STEP_SIZE
FFE_MAIN_CUR_MIN
FFE_PRE_TAP1_MAX
FFE_POST_TAP1_MAX
FFE_TAPN_MAX
CURSOR_GAIN
FFE_BACKOFF
Import...
Export...
Calculate
Close

Note: The configurable settings are extensive. If the user is upgrading from a previous version of *Electronic Desktop*, and currently has both versions installed, they can **Export** their configuration settings (e.g., *.cfg* file) from both versions and compare them. New parameters will be clearly visible in this side by side perspective. While **Electronic Desktop** supports COM keywords up to V3.40, not all keywords will be defined. Undefined keywords will function at their default values.



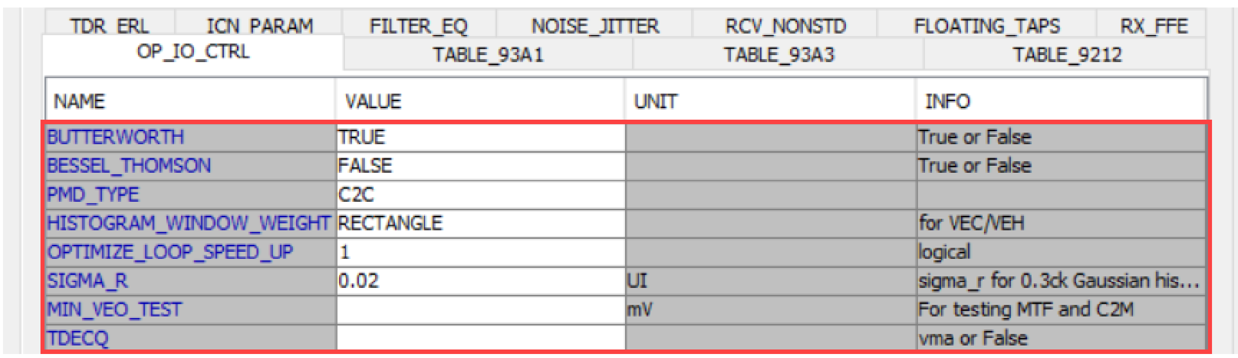
For more information, refer to the IEEE 802 LAN/MAN Standards Committee website and the IEEE P802.3ck 100 Gb/s, 200 Gb/s, and 400 Gb/s Electrical Interfaces Task Force Home Page.

AEDT/SPISim's COM analysis flow has been upgraded from V2.95 to match IEEE's reference implementation V3.40 (with some of experimental flags excluded).

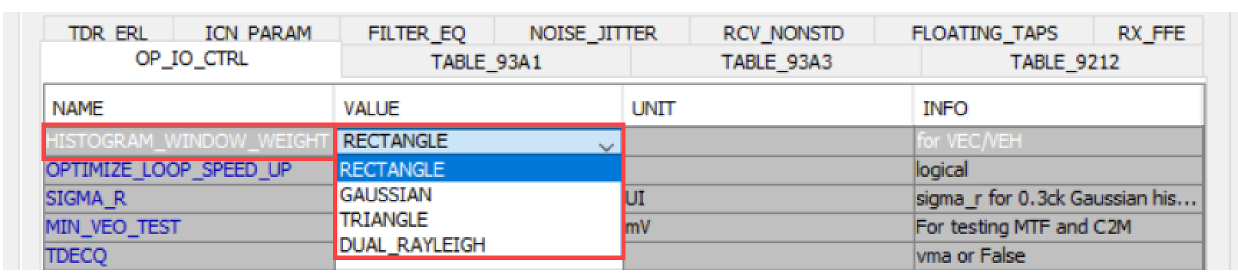
Flags	Impact
• **BUTTERWORTH** • **BESSEL_THOMSON** • **BTorder**	Affect filtering applied to impulse response and COM/ERL results in general.
• **HISTOGRAM_WINDOW_WEIGHT** • **SIGMA_R** • **MIN_VEO_TEST** • **SAMPLES_FOR_C2M** • **T_O: VE_MV/VEC_DB**	Affect VEO_MV/VEC_DB values, these are used in PMD type = C2M only.
• **TDECQ**	Affect TDECQ output
• **AC_CM_RMS** • **ACCM_MAX_FREQ**	Affect sigma_ACCM_at_tp0_mv output (AC RMS at TP0)
• **AUTO_TFX** • **FIXTURE_DELAY_TIME**	Affect ERL output, these are amount to trim before TDR data is used for ERL calculations.
• **OPTIMIZE_LOOP_SPEED_UP**	Speed-up parameter sweeps.
• **C2** • **C3** • **GDC_MIN**	Additional parameters for FFE (C2, C3) and CTLE (GDC_MIN) sweeping to find best FOM (figure of merit).

Locations by Tab

OP_IO_CTRL
BUTTERWORTH
BESSEL_THOMSON
PMD_TYP
HISTOGRAM_WINDOW_WEIGHT
OPTIMIZE_LOOP_SPEED_UP
SIGMA_R
MIN_VEO_TEST
TDECQ



HISTOGRAM_WINDOW_WEIGHT includes RECTANGLE, GAUSSIAN, TRIANGLE, and DUAL_RAYLEIGH options.



TDECQ (Transmitter and Dispersion Eye Closure Quality) for PAM4 supports VMA mode or not (0, False):

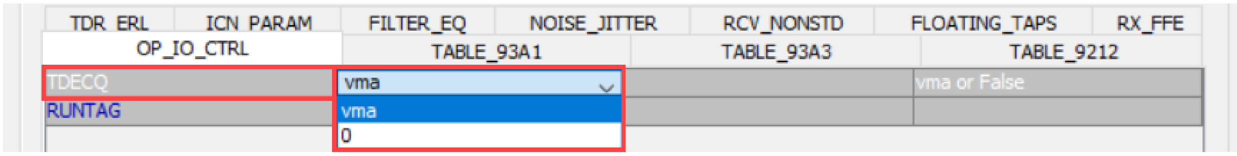
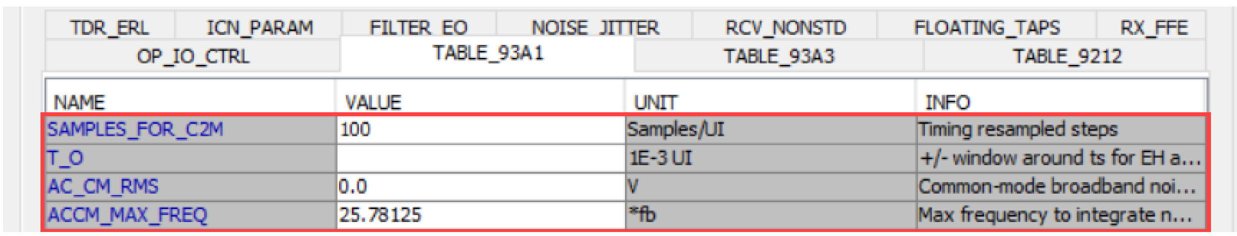
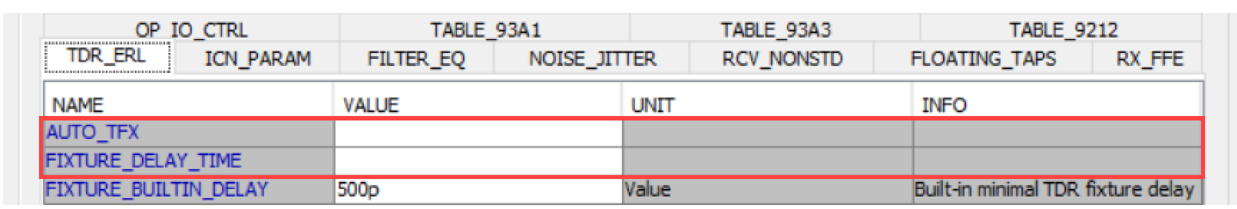


TABLE_93A1
SAMPLES_FOR_C2M
T_O
AC_CM_RMS
ACCM_MAX_FREQ

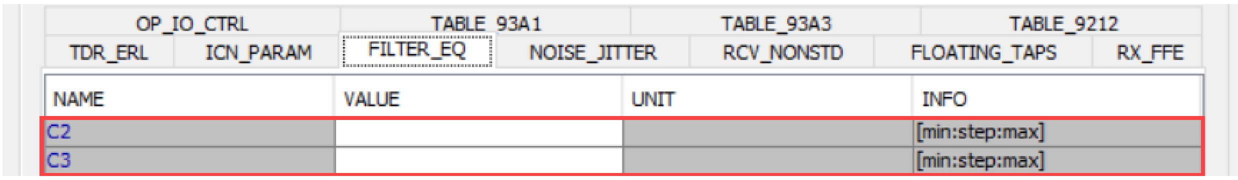


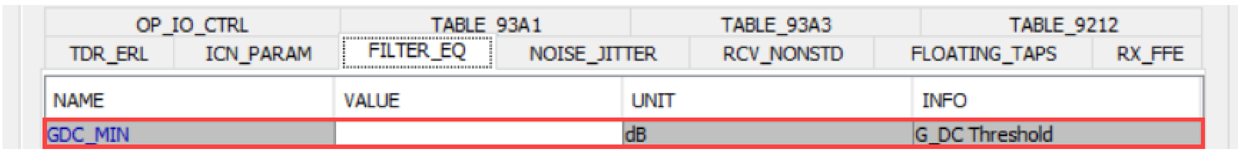
TDR_ERL
AUTO_FIX
FIXTURE_DELAY_TIME



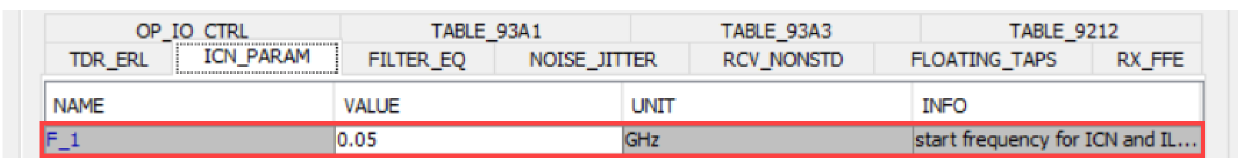
FILTER_EQ
C2

FILTER_EQ
C3
GCD_MIN





ICN_PARAM
F_1



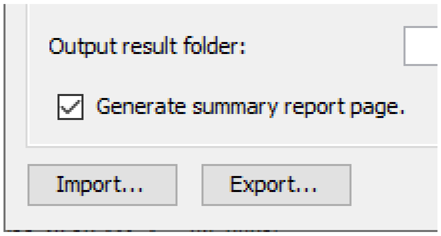
> **Important:** Experimental flags excluded from the V3.40 COM keywords are: RILN related flags (COMPUTE_RILN, COMPUTE_TDLIN, RILN) and TDMODE, Save_TD.

Generating a COM Analysis Report Using a Non-GUI Batch Command Line

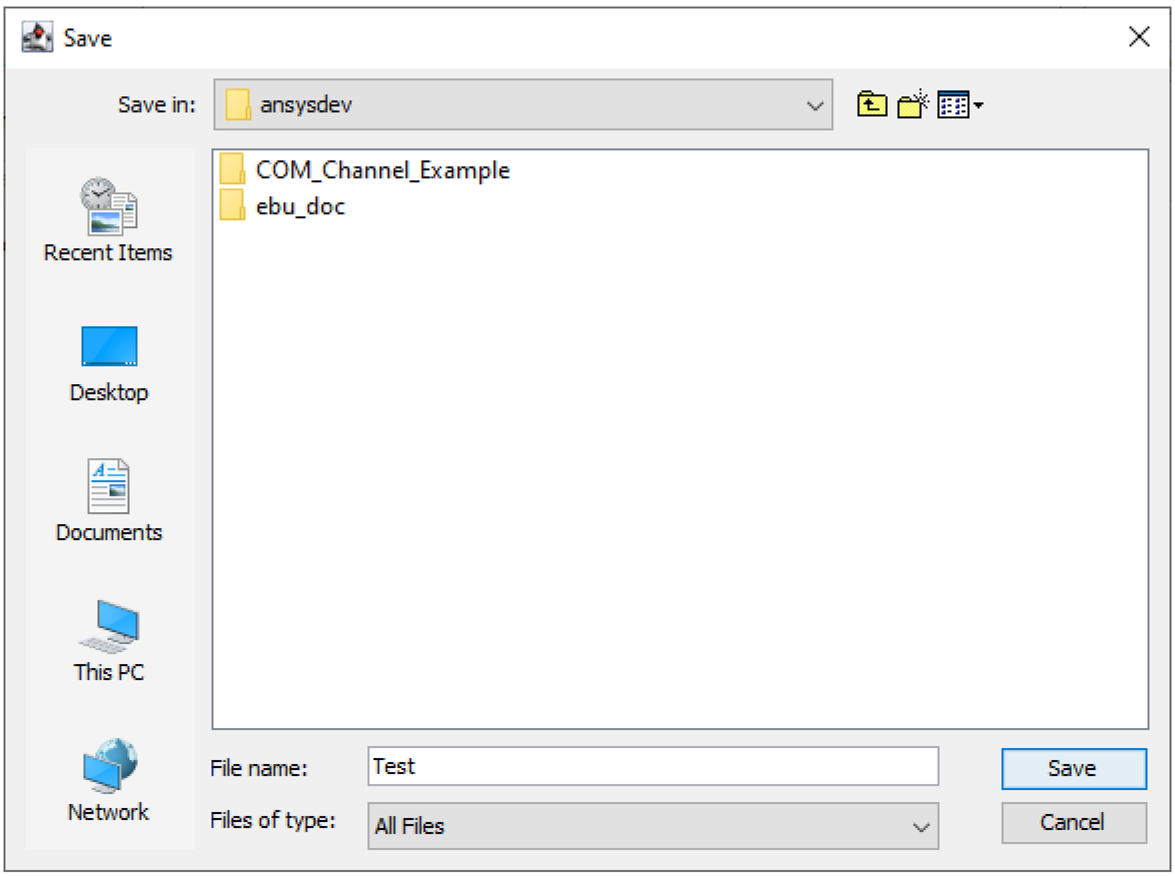
Generate a COM analysis report in a non-GUI (NG) batch command line by completing the following steps.

1. Launch the COM analysis report GUI interface. Refer to the first section of " Generating a COM Analysis Report From the SPISim User Interface " on page 15-143 for details.

2. Complete steps 1-6 in the " Generating a COM Analysis Report From the SPISim User Interface " on page 15-143 section to configure a COM analysis report.
3. From the **Config. and generate COM report ...** window, select **Export...** to open an explorer window.



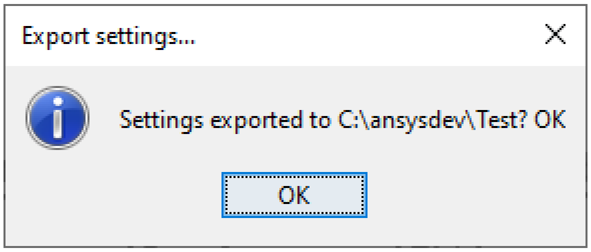
4. Choose and enter a name in the **File name** field. Then use the **Save in** drop-down menu or the navigable explorer field to choose a location to save the *.cfg* file.



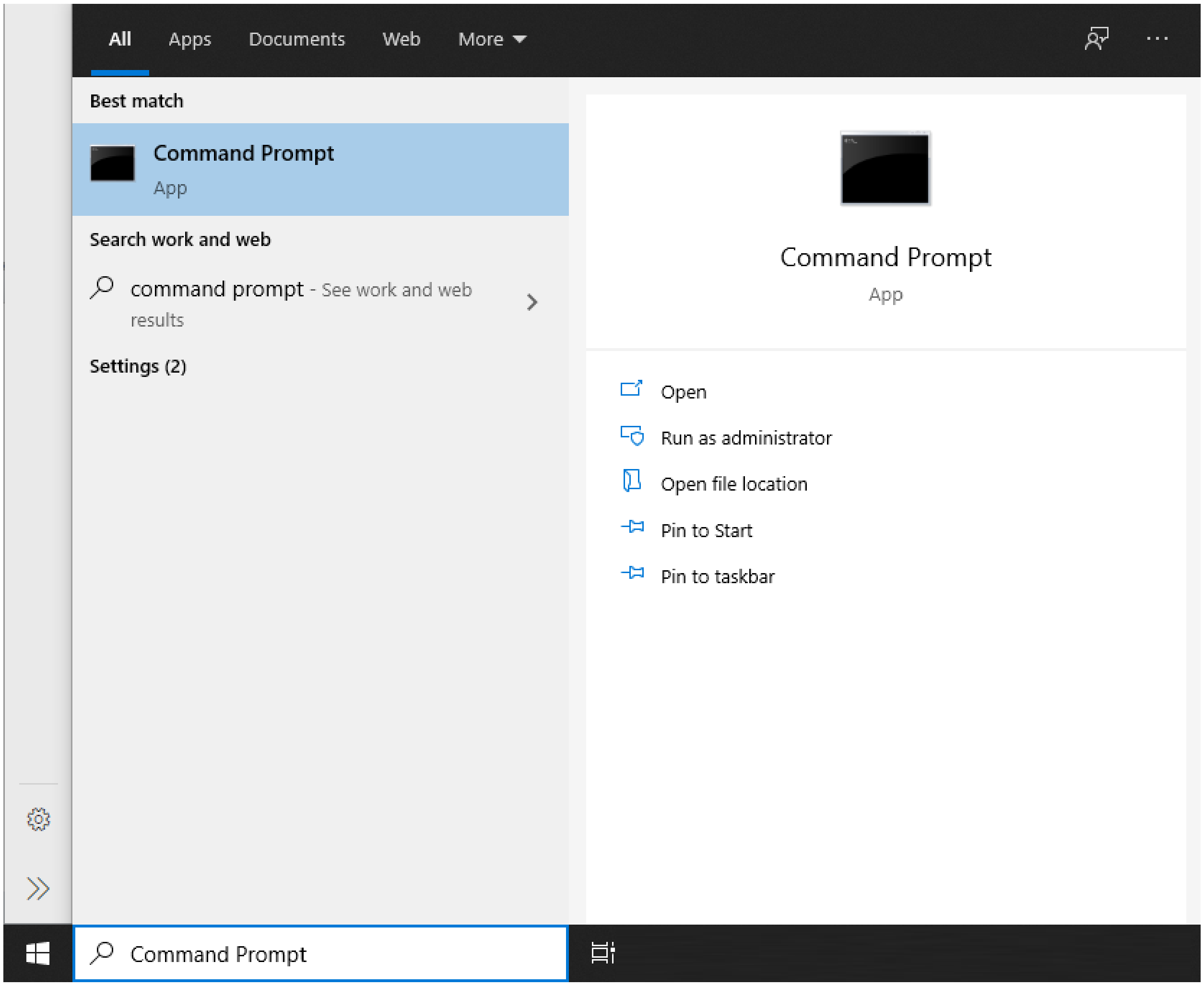
5. Click **Save** to close the explorer window and open the **Export settings...** confirmation window.

> **Note:** The *.cfg* file contains configuration settings in plain text format. These settings can be edited, if necessary.

6. Click **OK**

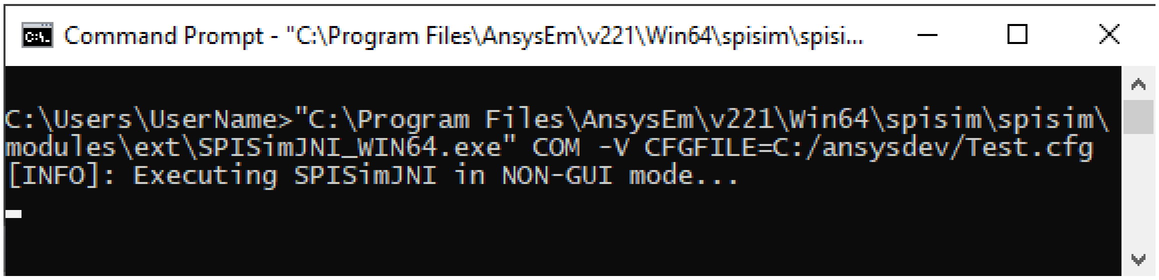


7. Run a command prompt (e.g., in a Windows environment, type *Command Prompt* into the **Search Bar** and press **Enter** to open a **Command Prompt** window.



> **Note:** The following example assumes **Electronics Desktop** is installed in the default directory in a Windows operating system environment. Structure the command appropriately for a different directory, path, and/or operating system, as necessary. Do **not** move the ***SPISimJNI_WIN64.exe*** executable file. The executable file relies on the relative path to the **Electronics Desktop** installation folder.

8. From the window, enter the following command, then press **Enter** to execute the COM analysis report in NG mode: `"C:\Program Files\AnsysEm\v242\Win64\spisim\spisim\modules\ext\SPISimJNI_WIN64.exe" COM -v CFGFILE=[`path to .cfg file`]`



> **Note:** The *.cfg* file contains configuration settings in plain text format. These settings can be edited, if necessary.

The command prompt generates a COM analysis report in standard output (stdout), and generates the same *.htm*, *.raw*, *.csv*, and *.png* files that are created in a standard GUI report. The files are saved in the **Output result folder:** (selected on the **Config. and generate COM report ...** window in "Complete steps 1-6 in the " Generating a COM Analysis Report From the SPISim User Interface " on page 15-143 section to configure a COM analysis report." on page 15-160).

Generating a COM Analysis Report Using IBIS-Based Transition Data

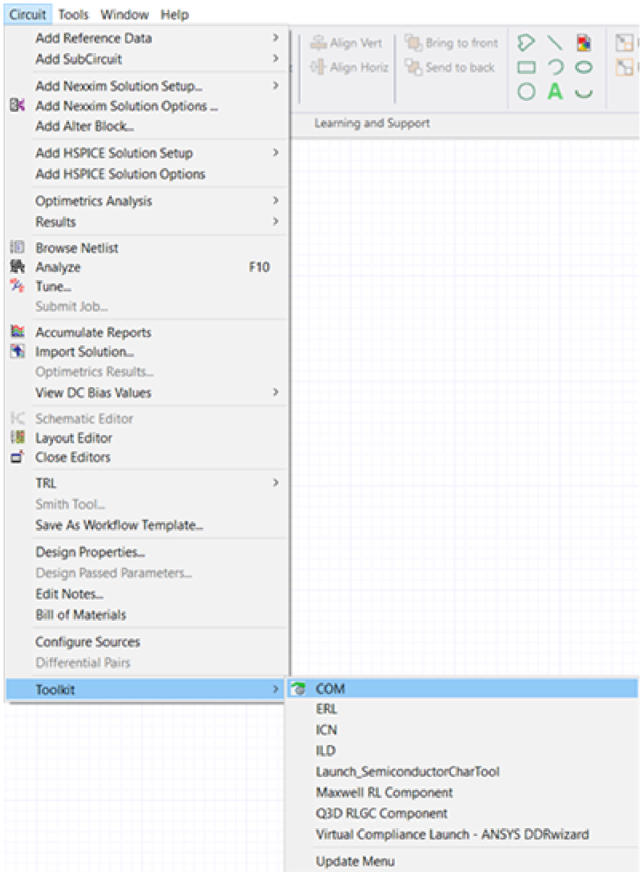
Channel operating margin (COM) analysis generates a *.csv* report file and binary waveform plots from analyzed data in a streamlined format. While AMI configuration data (e.g., Tx and Rx package data) can be entered manually, follow the steps in this section to generate a COM analysis report using data parsed from an IBIS file and then easily selected from drop-down menus.

> **Note:** A COM analysis report can also be generated in other ways. If the user is familiar with the process, the pertinent differences begin at **step 8**. Refer to " Generating a COM Analysis Report From the SPISim User Interface " on page 15-143.

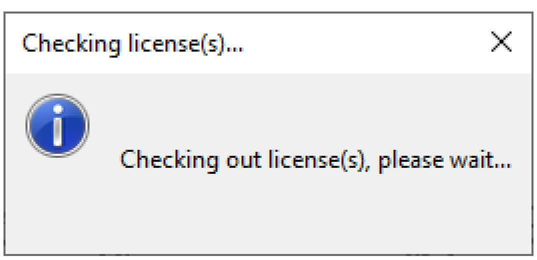
Complete the following steps to open the **Config. and generate COM report ...** window.

From Electronics Desktop

From the **Circuit** menu, select **Toolkit** > **COM**.

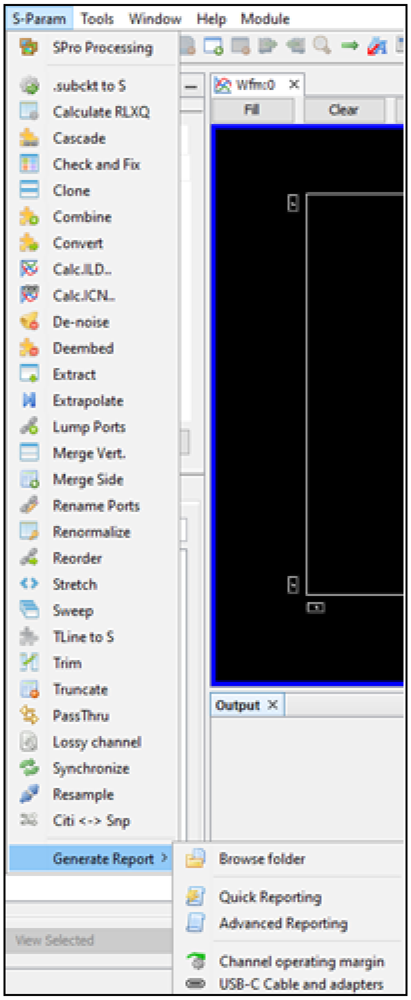


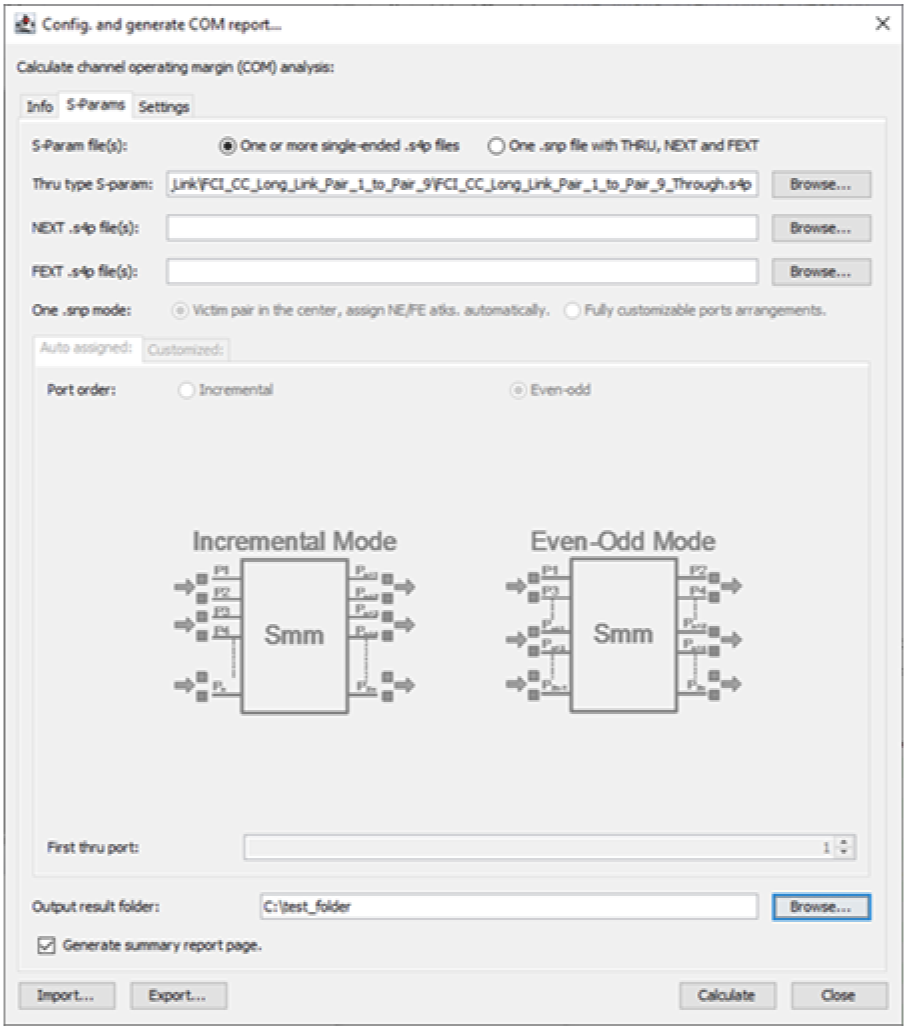
Before the **Config. and generate COM report ...** window opens, a **Checking License(s) ...** window opens.



From SPISim

From the **S-Param** menu, select **Generate Report** > **Channel operating margin** to open the **Config. and generate COM report ...** window.

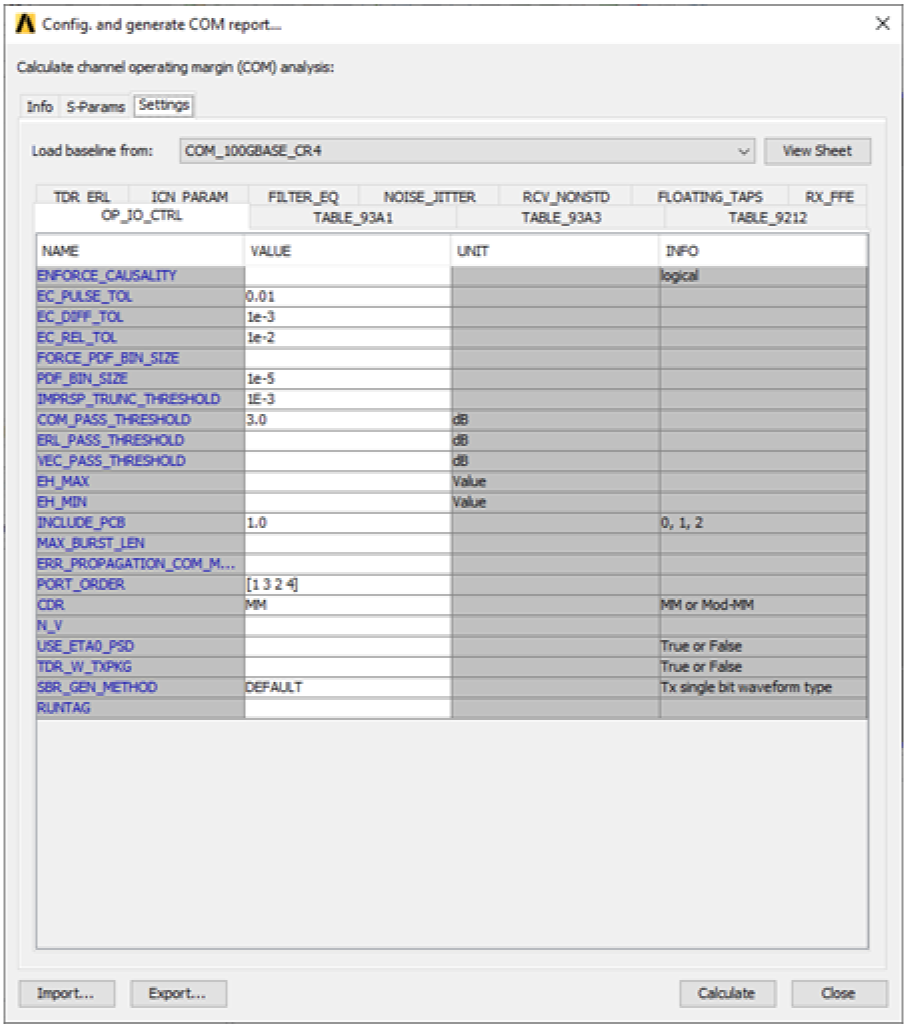

S-Param
Tools
Window
Help
Module
SPro Processing
.subckt to S
Calculate RLXQ
Cascade
Check and Fix
Clone
Combine
Convert
Calc.ILD..
Calc.ICN..
De-noise
Deembed
Extract
Extrapolate
Lump Ports
Merge Vert.
Merge Side
Rename Ports
Renormalize
Reorder
Stretch
Sweep
TLine to S
Trim
Truncate
PassThru
Lossy channel
Synchronize
Resample
Citi <-> Snp
Generate Report
Browse folder
Quick Reporting
Advanced Reporting
Channel operating margin
USB-C Cable and adapters
View Selected
Wfm:0
Fill
Clear
Output



Follow the steps below to configure and then generate a COM analysis report.

1. Click the **Browse ...** button in line with the **Thru type S-param** field to open an explorer window.
2. Navigate to and choose an *.s*N*p* file to include it in the COM analysis report. Then click **Open**.
3. Click the **Browse ...** button in line with the **Output result folder** field to open an explorer window. Choose a location to save the COM analysis report output files, which includes *.raw*, *.csv*, and *.png* files that remain accessible to the user outside of the context of the COM analysis report.
4. Click **Open**.

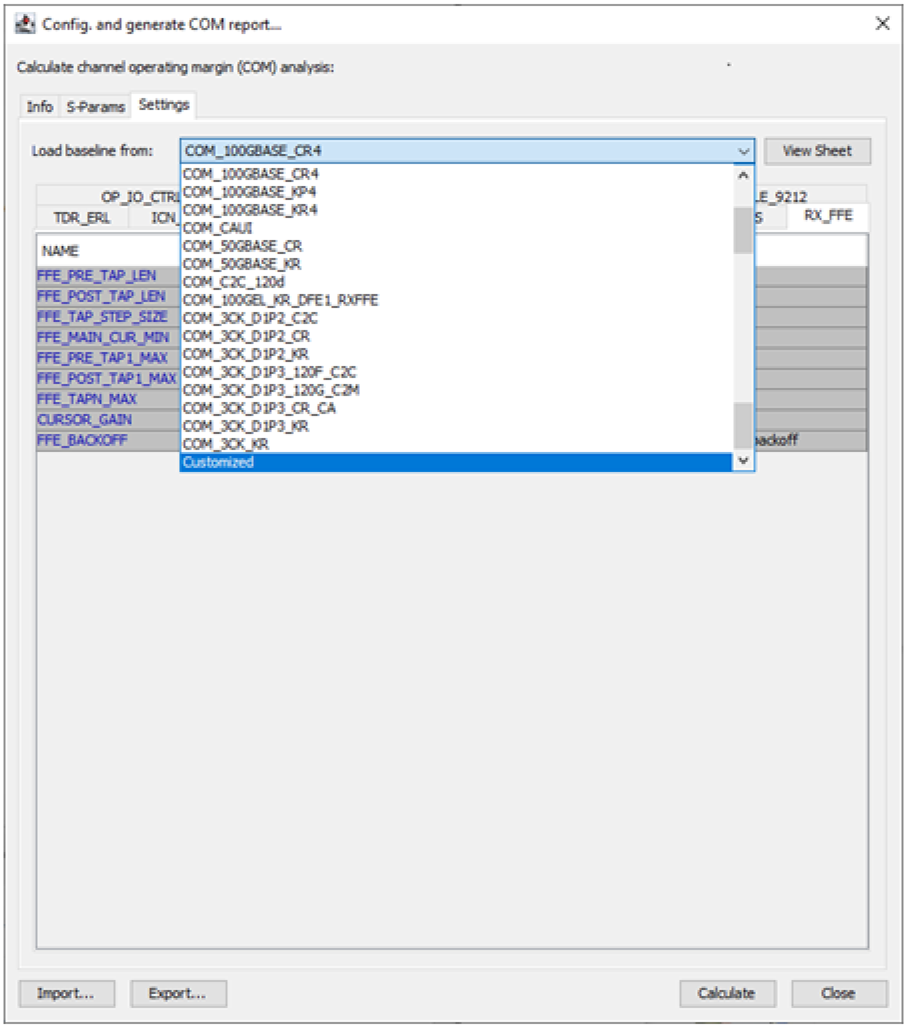
5. Before continuing, ensure that the **Generate summary report page.** check box, beneath the **Output result folder** field, is checked (checked by default).
6. Click the **Settings** tab to view the analyzed data's current configuration.



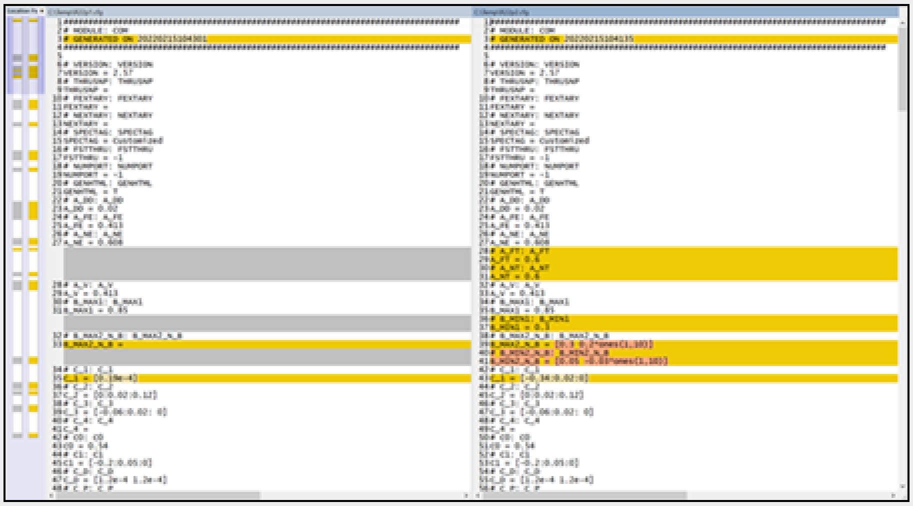
The settings are divided in eleven subtabs (i.e., **TDR_ERL**, **ICN_PARAM**, **FILTER_EQ**, **NOISE_JITTER**, **RCV_NONSTD**, **FLOATING_TAPS**, **RX_FFE**, **OP_IO_CTRL**, **TABLE_93A1**, **TABLE_93A3**, and **TABLE_9212**). Beside each parameter **NAME** is a configurable field, **UNIT** field (if applicable) and useful **INFO** field.

> **Note:** For more information, refer to the IEEE 802 LAN/MAN Standards Committee COM 2.51 with rxFFE Updates presentation files.

7. If necessary, choose another baseline configuration on the **Spec** drop-down menu, or select **Customized** to make individual changes to each field.



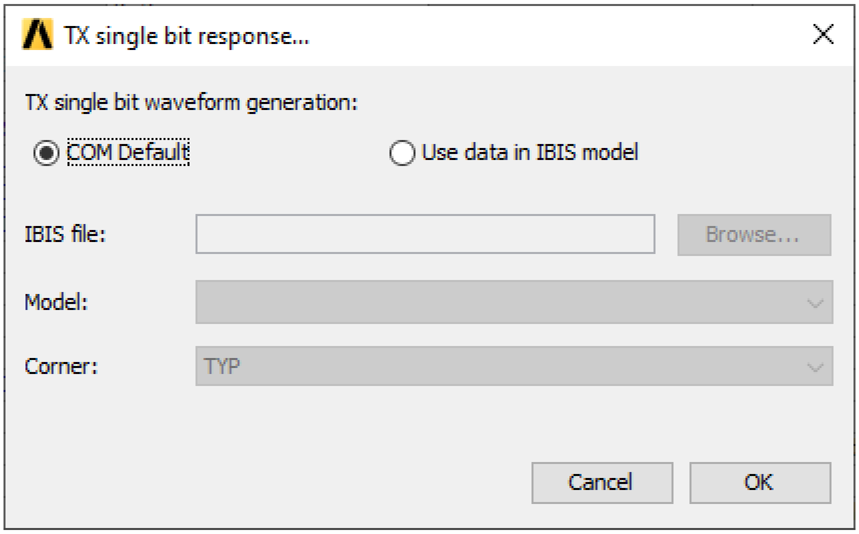
Note: The configurable settings are extensive. If the user is upgrading from a previous version of *Electronic Desktop*, and currently has both versions installed, they can **Export** their configuration settings (e.g., *.cfg* file) from both versions and compare them. New parameters will be obviously represented in this side by side perspective. While **Electronic Desktop** supports COM keywords up to V2.95, not all keywords will be defined. Undefined keywords will function at their default values.



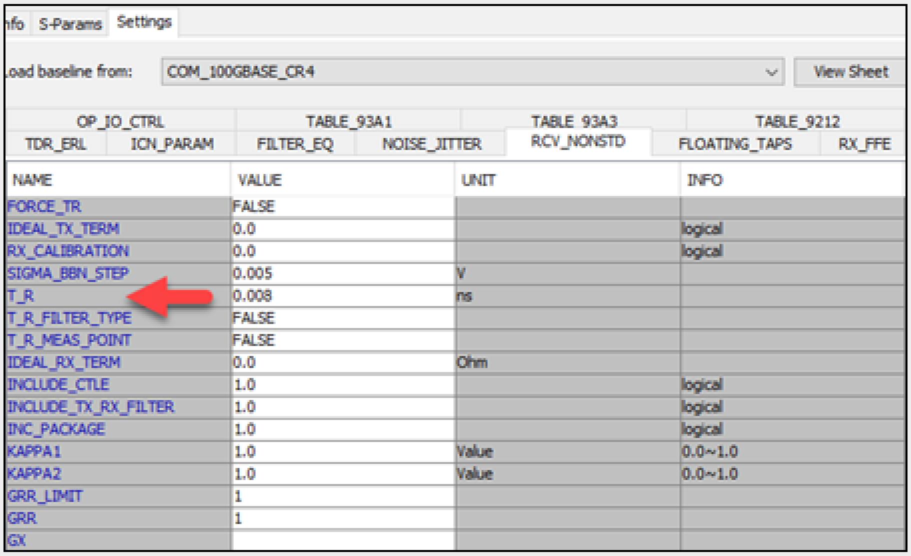
For more information, refer to the IEEE 802 LAN/MAN Standards Committee website and the IEEE P802.3ck 100 Gb/s, 200 Gb/s, and 400 Gb/s Electrical Interfaces Task Force Home Page.

8. Navigate to the **OP_IO_CTRL** subtab. Then click the field adjacent to **SBR_GEN_ METHOD** to reveal the **...** button.

9. Click the **...** button to open the **TX single bit response...** window.

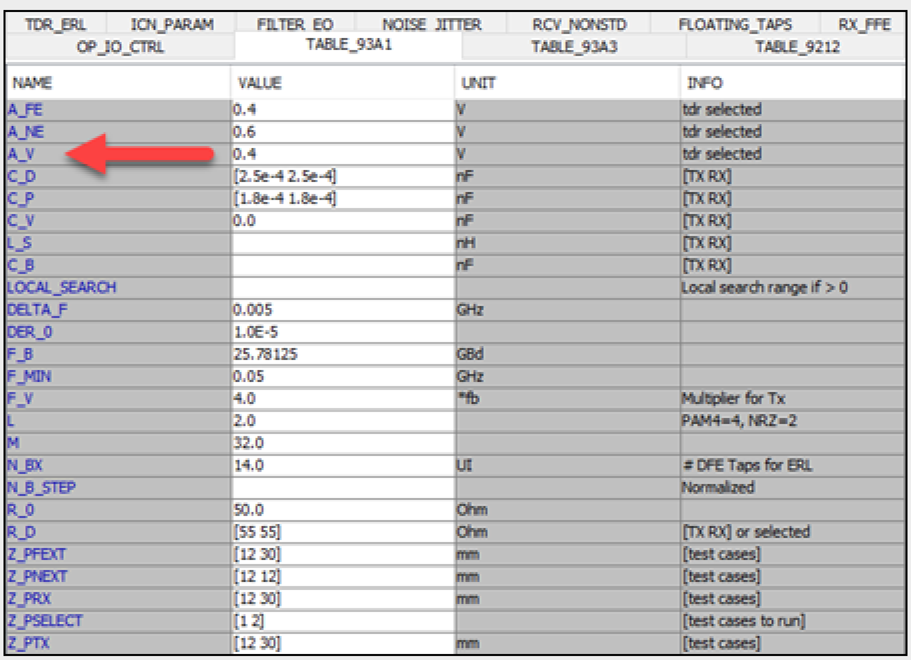


Note: If **COM Default** is selected, the user must enter Tx/Rx IBIS data (e.g., Tx and Rx package data, and amplitude) manually.
First, navigate to the **RCV_NONSTD** subtab to enter Tx/Rx values in the **T_R** field.



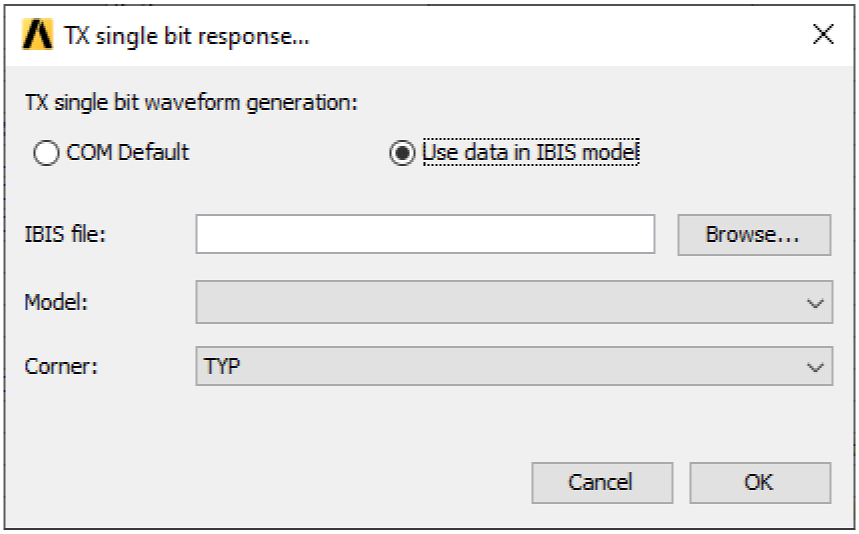
NAME	VALUE	UNIT	INFO
FORCE_TR	FALSE		
IDEAL_TX_TERM	0.0		logical
RX_CALIBRATION	0.0		logical
SIGMA_BBN_STEP	0.005	V	
T_R	0.008	ns	
T_R_FILTER_TYPE	FALSE		
T_R_MEAS_POINT	FALSE		
IDEAL_RX_TERM	0.0	Ohm	
INCLUDE_CTLE	1.0		logical
INCLUDE_TX_RX_FILTER	1.0		logical
INC_PACKAGE	1.0		logical
KAPPA1	1.0	Value	0.0~1.0
KAPPA2	1.0	Value	0.0~1.0
GRR_LIMIT	1		
GRR	1		
GX			

Then navigate to the **TABLE_93A1** subtab to enter amplitude values in the **A_V** field.

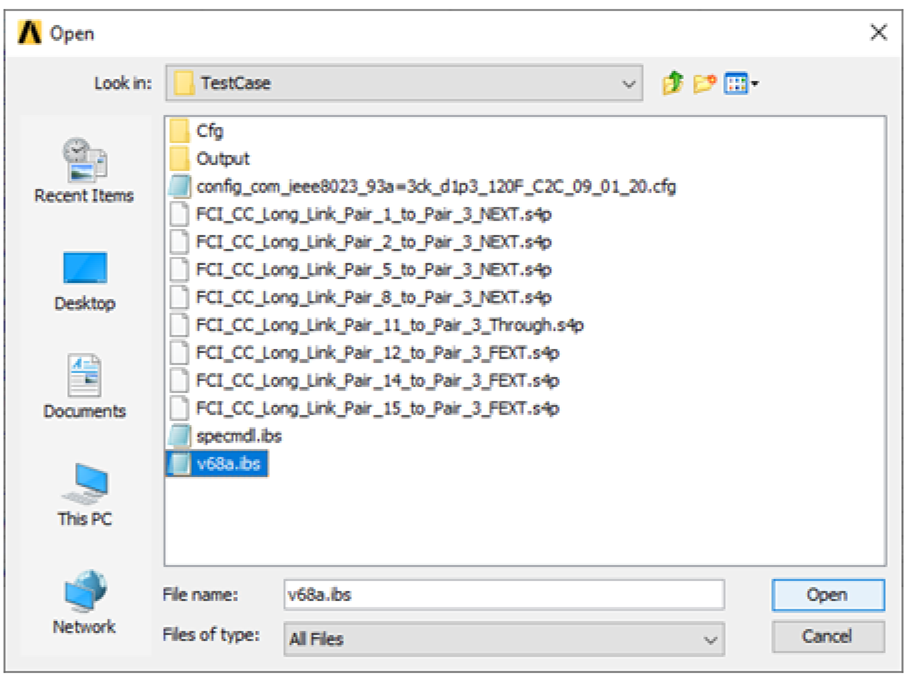


NAME	VALUE	UNIT	INFO
A_FE	0.4	V	tdr selected
A_NE	0.6	V	tdr selected
A_V	0.4	V	tdr selected
C_D	[2.5e-4 2.5e-4]	nF	[TX RX]
C_P	[1.8e-4 1.8e-4]	nF	[TX RX]
C_V	0.0	nF	[TX RX]
L_S		nH	[TX RX]
C_B		nF	[TX RX]
LOCAL_SEARCH			Local search range if > 0
DELTA_F	0.005	GHz	
DER_0	1.0E-5		
F_B	25.78125	GBd	
F_MIN	0.05	GHz	
F_V	4.0	*fb	Multiplier for Tx
L	2.0		PAM4=4, NRZ=2
M	32.0		
N_BX	14.0	UI	# DFE Taps for ERL
N_B_STEP			Normalized
R_0	50.0	Ohm	
R_D	[55 55]	Ohm	[TX RX] or selected
Z_PFEXT	[12 30]	mm	[test cases]
Z_PNEXT	[12 12]	mm	[test cases]
Z_PRX	[12 30]	mm	[test cases]
Z_PSELECT	[1 2]		[test cases to run]
Z_PTX	[12 30]	mm	[test cases]

10. To use an IBIS file (i.e., *.*ibs* file) to create a COM analysis report, select the appropriate **TX single bit waveform generation** option (i.e., **Use data in IBIS model**).

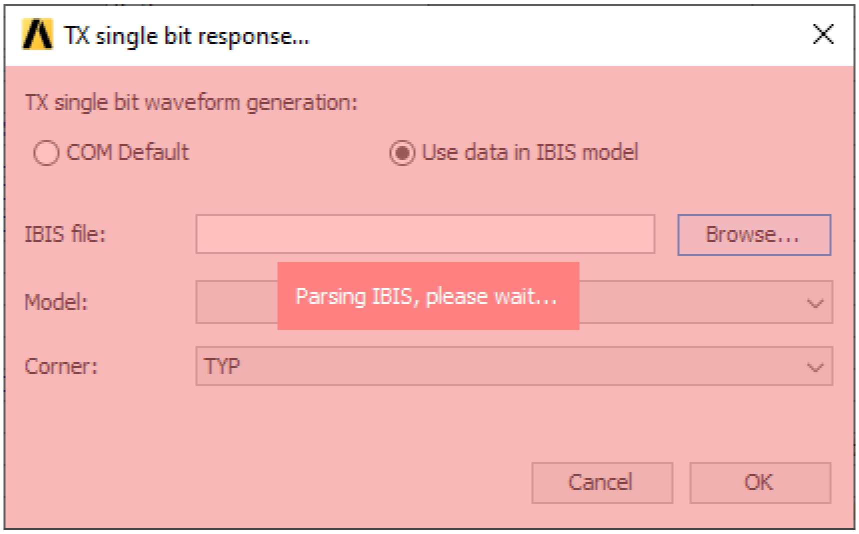


11. Click **Browse...** to open an explorer window. Then navigate to and select the appropriate IBIS file (e.g., **v68a.ibs**).

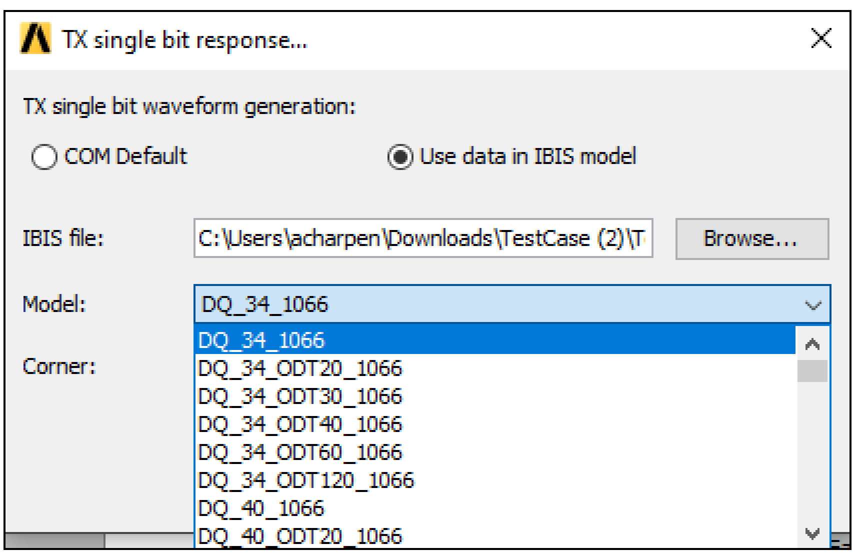


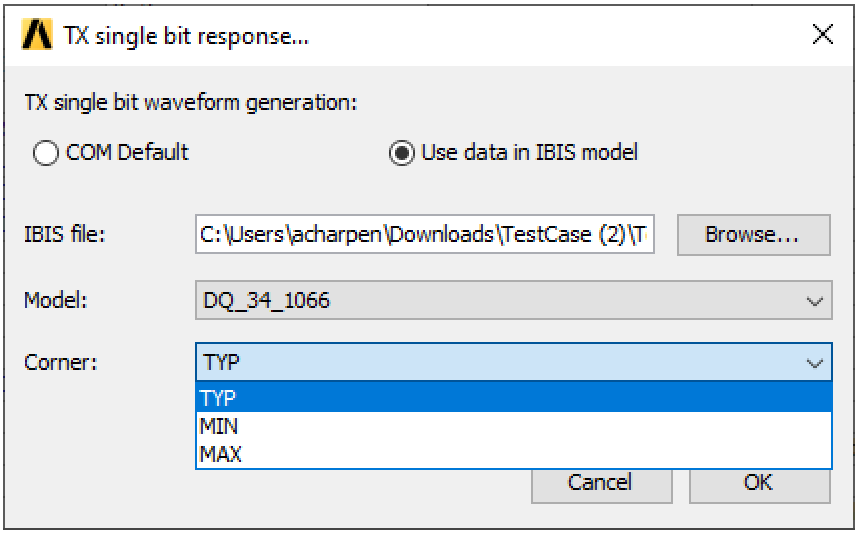
12. Click **Open** to close the explorer window and return to the **TX single bit waveform generation** window. **SPISim** immediately parses the selected IBIS file, scales based on the IBIS model's voltage range and the amplitude specified in the **A_V** field. The updated

value is used in the final report calculation and present in the **Output** window as **IBIS Tr Time**. Depending on the size of the IBIS file, the **TX single bit waveform generation** window may be momentarily inaccessible. In that case, the following message may appear.



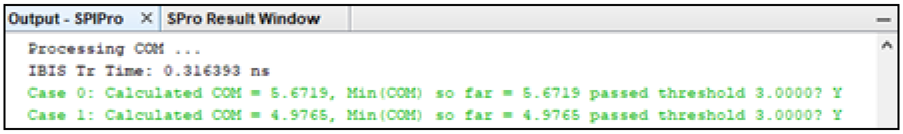
13. Once the IBIS file is parsed, select the appropriate **Model** and **Corner** from the adjacent drop-down menus.





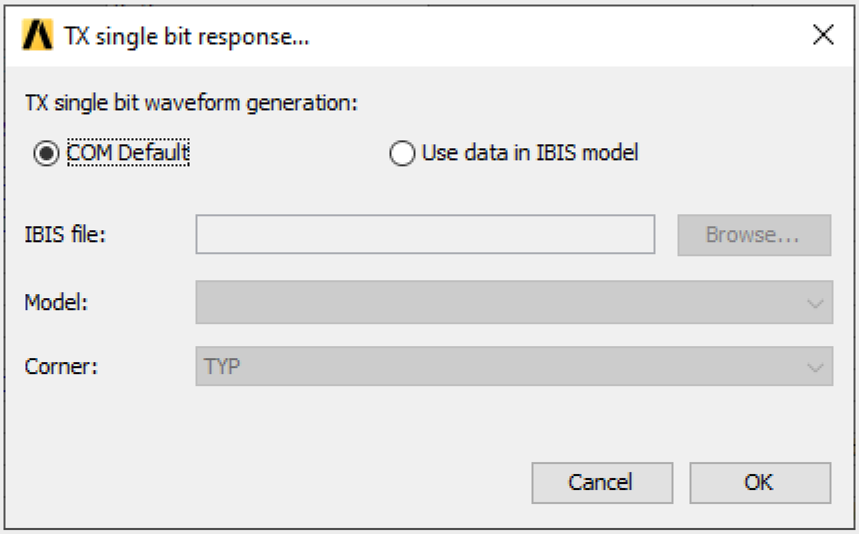
14. Click **OK** to return to the **Config. and generate COM report ...** window.

15. Click **Calculate** and the output folder selected in **Step 3** populates with .raw (i.e., Berkley spice), .csv, and .png files. The **Output - SPIPro** window will display relevant data, including the **IBIS Tr Time** (e.g., **0.316393 ns**).



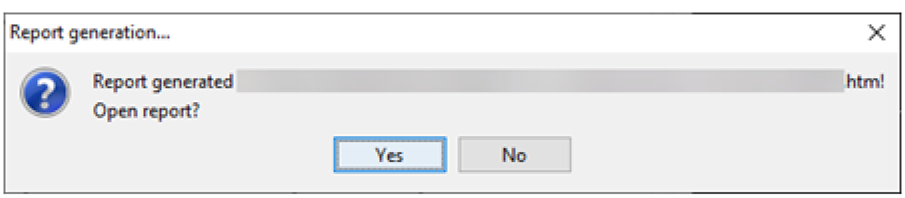
Note: To test the data reported in the method described in this section, enter the **IBIS Tr Time** value in the **Config. and generate COM report ...** > **Settings** tab > **RCV_NONSTD** subtab > **T_R** field.

Then set the **OP_IO_CTRL** > **SBR_GEN_METHOD** > **TX single bit response...** window > **TX single bit waveform generation** option to **Default**.



Click **OK** to close the **TX single bit response...** window, and then click **Calculate**. The resulting data should match.

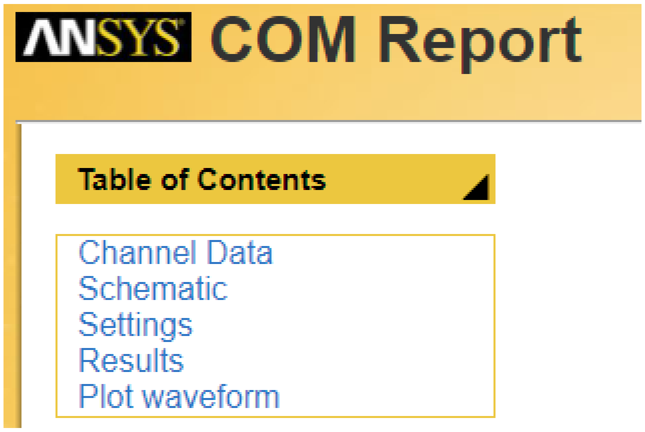
16. When the report is ready, a **Report generation ...** window appears. Click **Yes** to open the COM analysis report in the system's default browser.



Continue to COM Analysis Report Summary

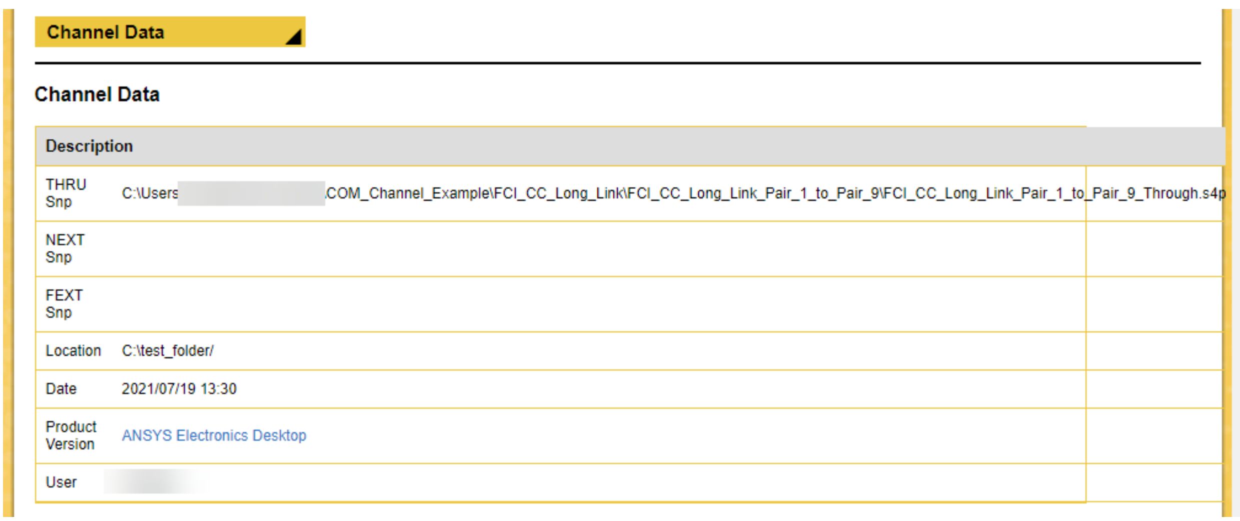
COM Analysis Report Summary

The COM Analysis Report is organized into a single window, navigable by a Table of Contents (TOC).



The TOC contains the following subsections.

Channel Data – Contains statistics about the report, including the location of the raw file(s) used to generate the report, the output location, date of generation, version of Electronic Desktop/SPISim software used to generate the report, and the user responsible for generating the report.

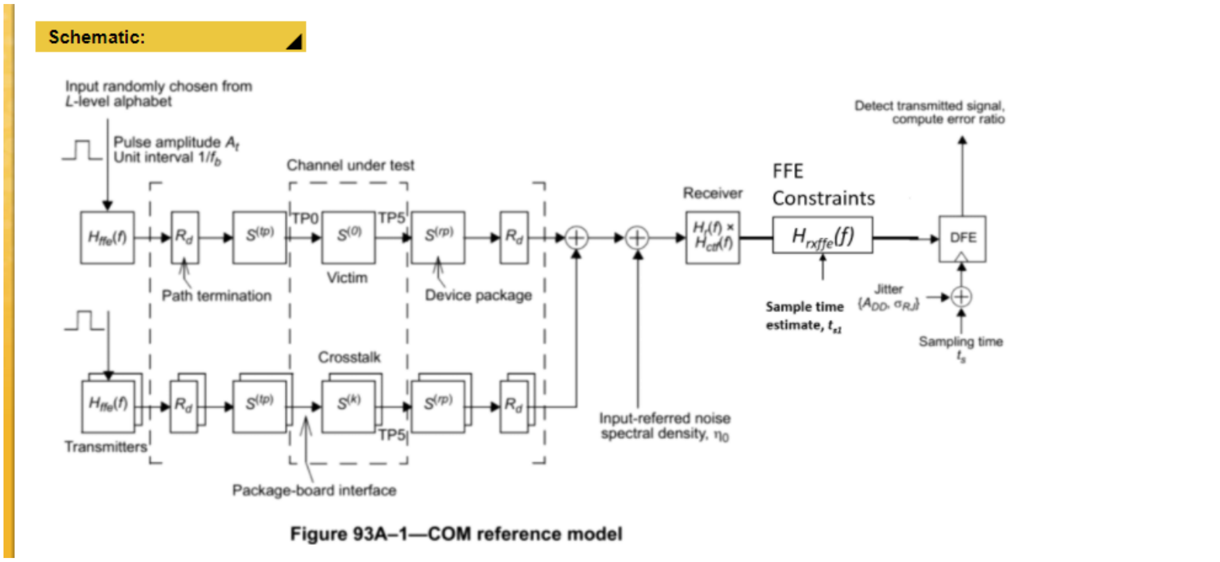

Channel Data

Channel Data

Description	
THRU Snp	C:\Users COM_Channel_Example\FCI_CC_Long_Link\FCI_CC_Long_Link_Pair_1_to_Pair_9\FCI_CC_Long_Link_Pair_1_to_Pair_9_Through.s4p
NEXT Snp	
FEXT Snp	
Location	C:\test_folder/
Date	2021/07/19 13:30
Product Version	ANSYS Electronics Desktop
User	

Schematic: – A *.png* of the schematic used to produce the original analysis, upon which the COM analysis report is based. Click the *.png* file to open the image at full resolution.

> **Note:** The schematic *.png*, as well as other *.pngs*, *.raw*, and *.csv* files created during generation of the COM Analysis Report remain accessible in the Output folder.



Settings – Shows extensive tables of the parameters and settings used to generate the COM analysis report.

> **Note:** The tables mirror the settings accepted or chosen by the user on the **Config. and generate COM report ...** window **Settings** tab and the number of tabs differs from report to report.

Settings

Click on the buttons inside the tabbed menu :

OP_IO_CTRL   TABLE_93A1   TABLE_93A2   TABLE_92A2   RCV_NONSTD   RX_FFE

Results – Contains three subsections:

- A hyperlink to a .csv file.
- Multiple tables with the measured analysis results (the exact number of tabs is determined at the **Config. and generate COM report ...** > Settings tab). The results mirror the contents of the *.csv* file.

Results

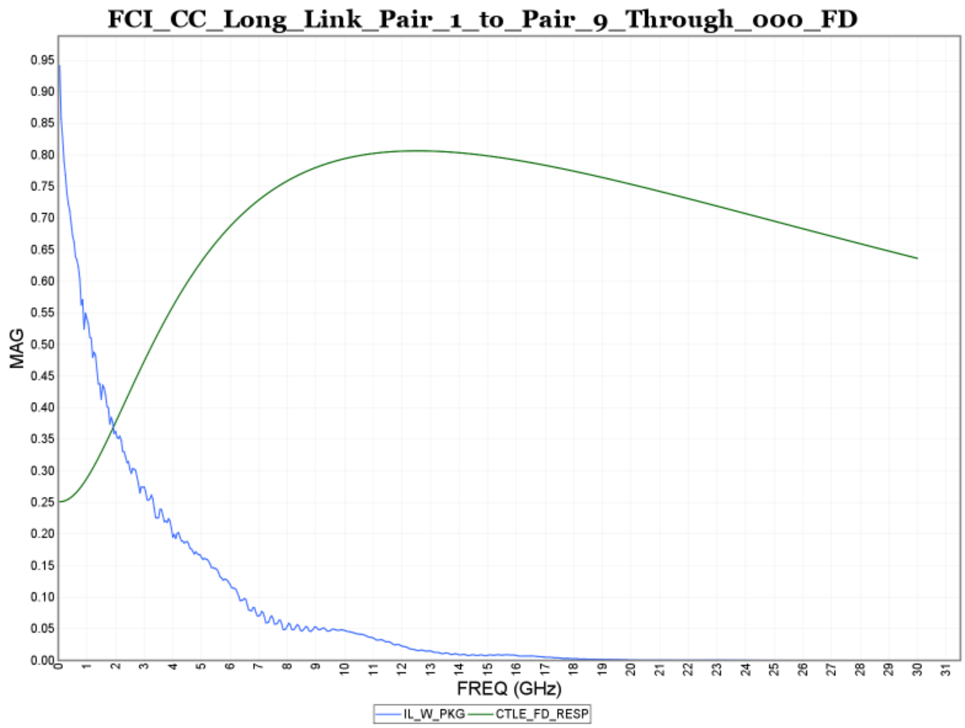
FCI_CC_Long_Link_Pair_1_to_Pair_9_Through.csv:

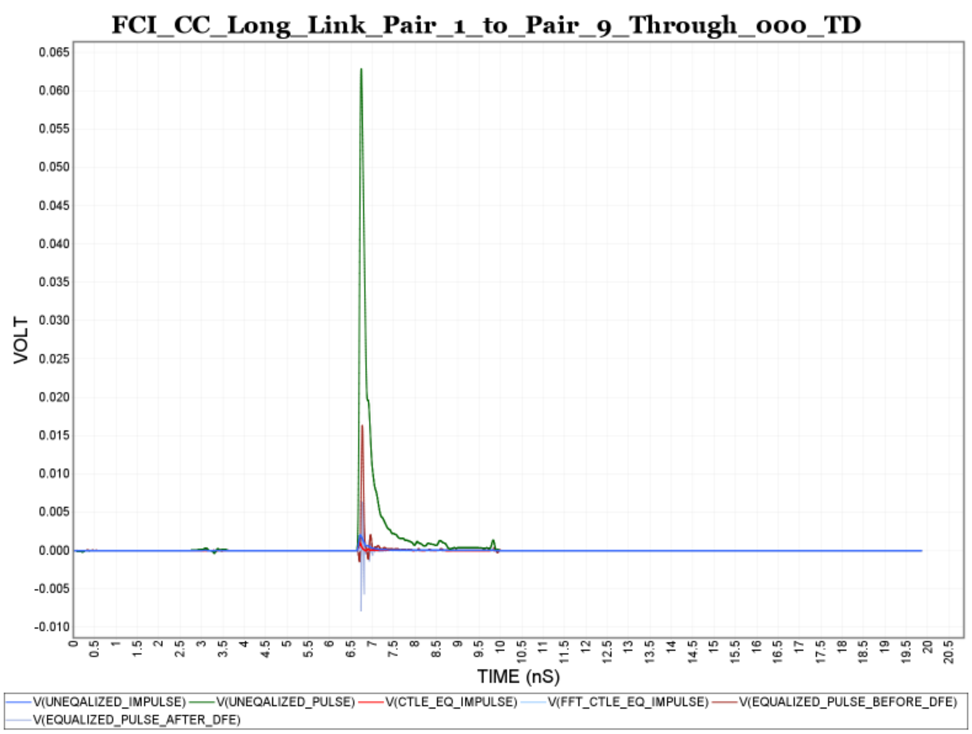
Click on the buttons inside the tabbed menu:

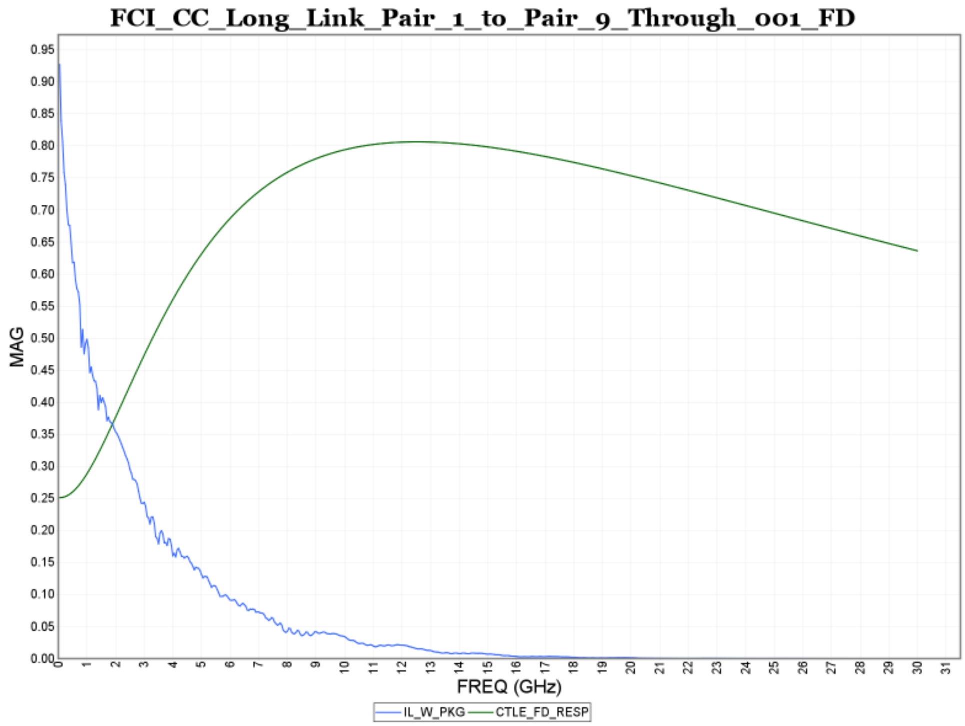
Case00 Case01

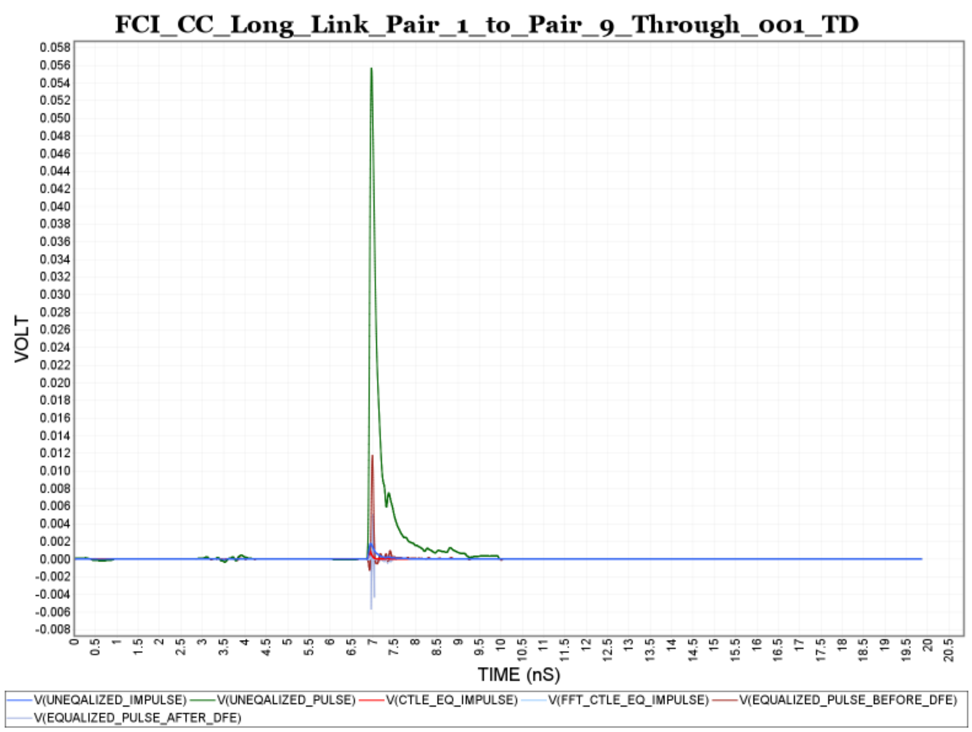
Name	Value
VER	2.95
WFM	FCI_CC_Long_Link_Pair_1_to_Pair_9_Through.s4p
IDX	Case 0
COM	6.2994
ICN	0.0000
ILD	0.3672
LEVELS	2
PKG_LEN_TX	[D@4a125ffa
PKG_LEN_NEXT	-1.0
PKG_LEN_FEXT	-1.0
PKG_LEN_RX	[D@f10c8dc
BAUD_RATE_GHZ	25.7812
F_NYQUIST_GHZ	12.8906
CHANNEL_OPERATING_MARGIN_DB	6.2994
PEAK_INTERFERENCE_MV	7.1700
PEAK_CHANNEL_INTERFERENCE_MV	4.1800
PEAK_ISI_MV	4.1800
PEAK_MDXTK_INTERFERENCE_MV	0.0000
PEAK_MDNEXT_INTERFERENCE_MV	0.0000
PEAK_MDFEXT_INTERFERENCE_MV	0.0000
AVAILABLE_SIGNAL_AFTER_EQ_MV	14.8077

- An extensive list of waveform plots generated during analysis. Click any of the *.png* files to open the images at full resolution.

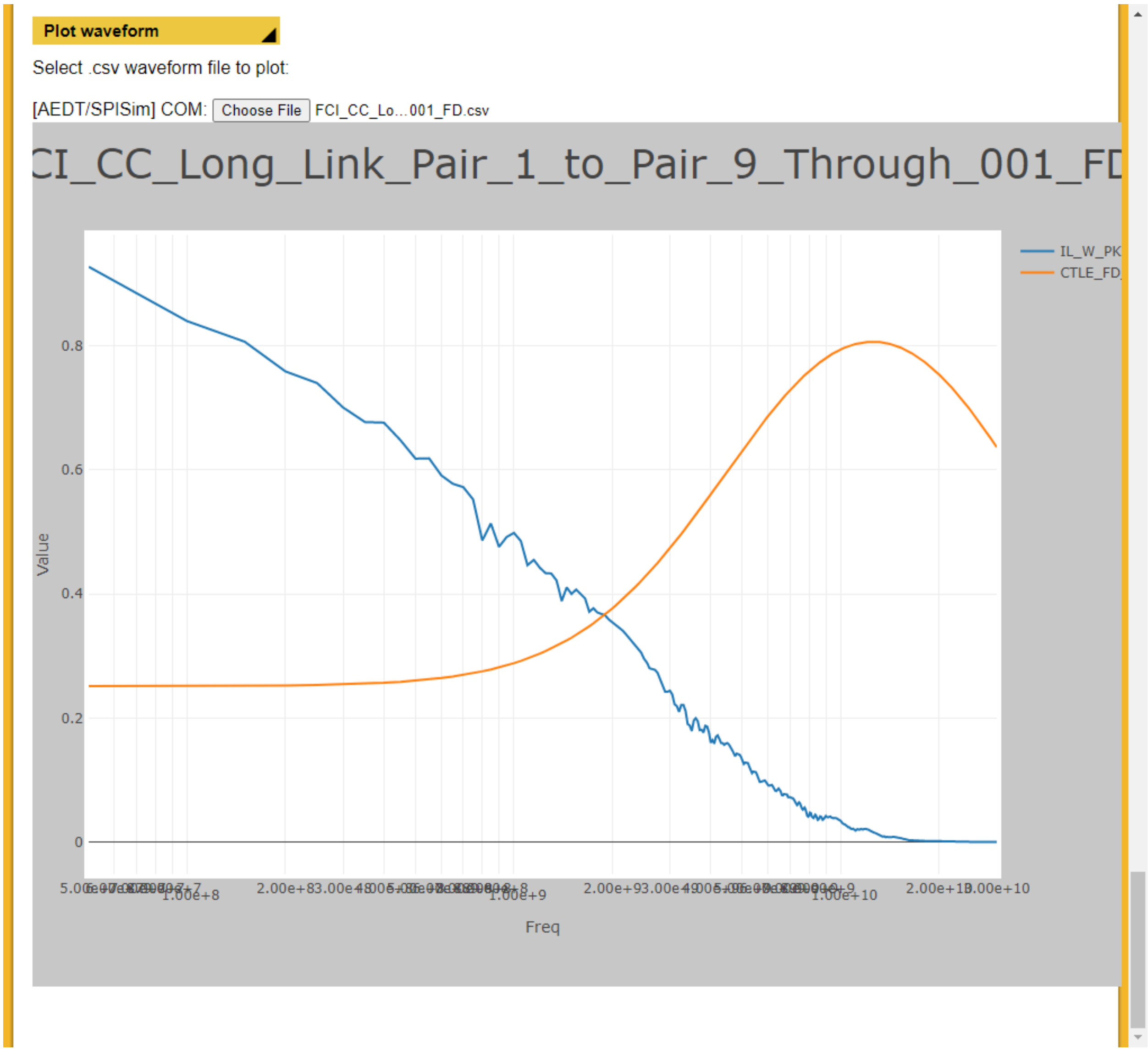

FCI_CC_Long_Link_Pair_1_to_Pair_9_Through_000_FD
MAG
FREQ (GHz)
IL_W_PKG
CTLE_FD_RESP


FCI_CC_Long_Link_Pair_1_to_Pair_9_Through_000_TD
VOLT
TIME (nS)
V(UNEQALIZED_IMPULSE)
V(UNEQALIZED_PULSE)
V(CTLE_EQ_IMPULSE)
V(FFT_CTLE_EQ_IMPULSE)
V(EQUALIZED_PULSE_BEFORE_DFE)
V(EQUALIZED_PULSE_AFTER_DFE)


FCI_CC_Long_Link_Pair_1_to_Pair_9_Through_001_FD
MAG
FREQ (GHz)
IL_W_PKG
CTLE_FD_RESP

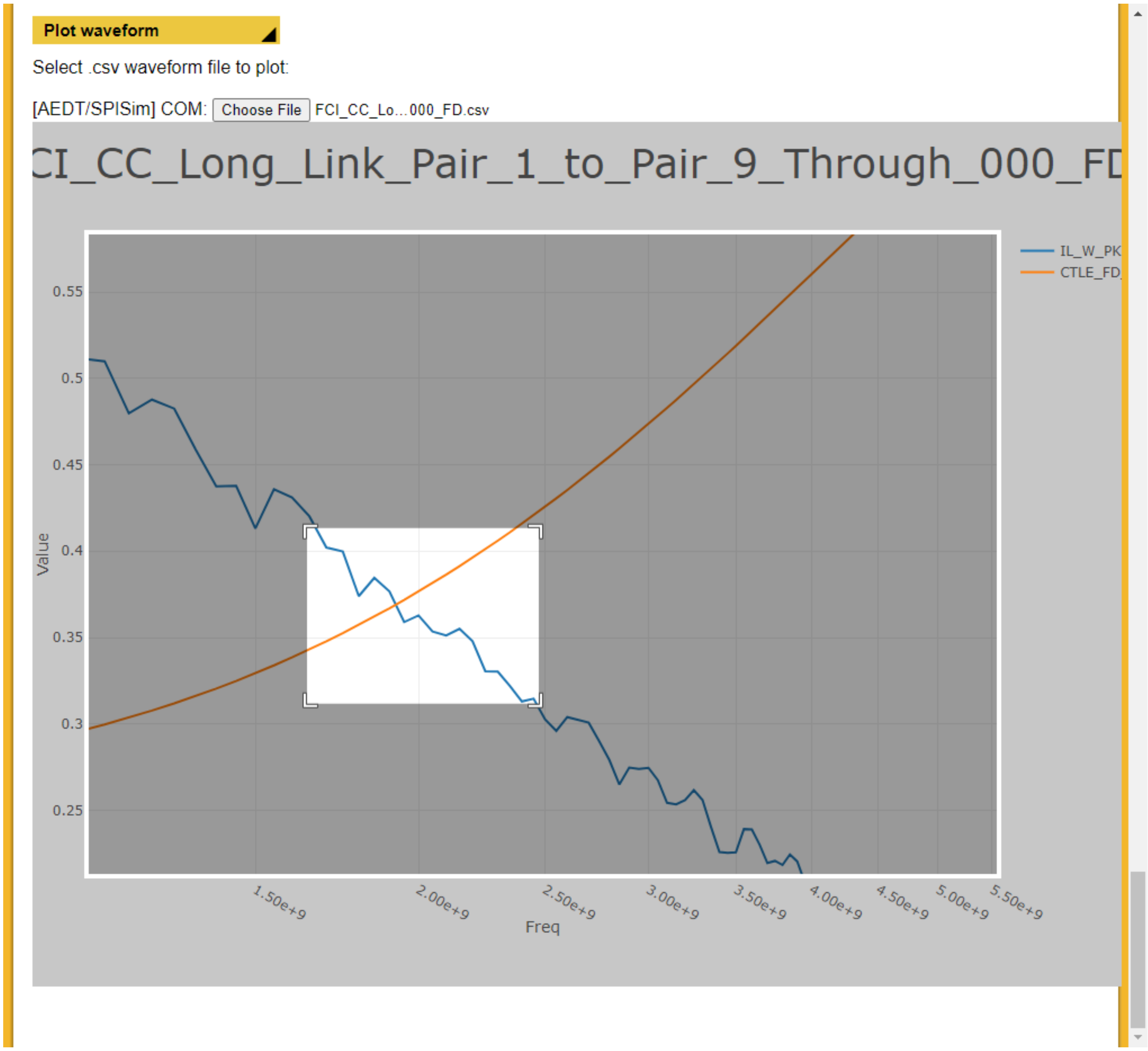

FCI_CC_Long_Link_Pair_1_to_Pair_9_Through_001_TD
VOLT
TIME (nS)
V(UNEQALIZED_IMPULSE)
V(UNEQALIZED_PULSE)
V(CTLE_EQ_IMPULSE)
V(FFT_CTLE_EQ_IMPULSE)
V(EQUALIZED_PULSE_BEFORE_DFE)
V(EQUALIZED_PULSE_AFTER_DFE)

Plot waveform – To view an additional waveform plot, click **Choose File** to open an explorer window. Then navigate to and select a *.csv* file. Select **OK** and the waveform appears below.



> **Note:** Hover over the waveform to see various controls that allow you to **Pan**, **Zoom in**, **Zoom out**, **Autoscale**, or isolate specific trace lines (via the legend).

The following is an example of using the mouse (i.e., **clicking+dragging** to zoom in on a particular quadrant of the waveform).



COM Analysis Keywords Reference Tables

OP_IO_CTRL

OP_IO_CTRL

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
MAX_BURST_LEN	nburst	Used to calculate burst error rate (not normally

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		used).
OPTIMIZE_LOOP_SPEED_UP	Optimize_loop_speed_up	If set to 0 (or default) normal looping, If set to 1 loop speedup by slightly reducing PD Fbin and FIR_ threshold for optimize looping only
HISTOGRAM_WINDOW_WEIGHT	Histogram_Window_Weight	%Weighting for VEC and VEO for histogram processing. Type are Gaussian,Dual Rayleigh,Triangle, and Rectangle (default)
SBR_GEN_METHOD		Pulse generation method: use IBIS's slew rate or filtered rectangular pause.
COM_PASS_THRESHOLD	COM Pass threshold	The pass/fail threshold for COM in dB.
TDR_W_TXPKG		Adds tx package for TDR, PTDR, and ERL. Default is 0.
EH_MAX		Used when PMD_ type is C2M and is not really computed per spec.
EC_DIFF_TOL	Enforce Causality DIFF_TOL	Difference Tolerance parameter for causality. Hard

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		enforcement, 1e-4,Soft enforcement, 1e-3.
PDF_BIN_SIZE	Force PDF bin size	Do not use.
CDR		CDR method, default is 'MM' (Mueller-Muller).
RUNTAG		This string is appended to the beginning of results files.
ERR_PROPAGATION_COM_MARGIN	Error propagation COM margin	Use to calculate error propagation (not normally used).
MIN_VEO_TEST	Min_VEO_Test	Used when PMD_type is C2M. This allows EH to go below EH_min. If set to zero, it is ignored.
EC_PULSE_TOL	Enforce Causality pulse start tolerance	Tolerance parameter for causality. Hard enforcement, 0.05, Soft enforcement, .01.
IMPRSP_TRUNC_THRESHOLD	Impulse response truncatio threshold	Zero padding threshold in fraction of IR peak for the impulse response. Effectively controls the length of time for the PR. Larger values decrease run time and accuracy.

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		Default is 1e-3.
FORCE_PDF_BIN_SIZE	Force PDF bin size	Do not use.
EC_REL_TOL	Enforce Causality REL_TOL	Difference Tolerance parameter for causality, Hard enforcement, 1e-4,Soft enforcement, 1e-3
SIGMA_R	sigma_r	sigma_r for 0.3ck Gaussian histogram window. Unit are UI. Preferred usage.
ERL_PASS_THRESHOLD	ERL Pass threshold	The pass fail threshold for ERL in dB.
N_V	N_v	Number of UI used to compute Vf.
BESSEL_THOMSON		Enable Bessel Thomsen filter for COM.
PORT_ORDER	Port Order	S parameter port order [tx+ tx- rx+ rx-].
ENFORCE_CAUSALITY		Default is 0. Not recommended to use.
USE_ETA0_PSD		Used eta_0 PSD equaiton for sigma_n. Default is 0. Do not use.
BUTTERWORTH	Butterworth	Enable Butterworth filter for TDR, PTDR, and ERL.
INCLUDE_PCB	Include PCB	Used to add a PCB, one each

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		side of the passed s-parameters.
EH_MIN		Used when PMD_type is C2M.
VEC_PASS_THRESHOLD	VEC Pass threshold	The pass fail threshold for VEC in dB; only used when PMD_type is C2M.

TABLE_93A1

TABLE_93A1

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
R_0		Reference impedance.
N_BX	N_bx	Used for ERL to compensate for a number of UI associated with the DFE.
Z_PSELECT	z_p select	List of package length indexes used to run COM.
C_B	C_b	C_b in nF (single sided).
Z_PNEXT	z_p (NEXT)	List of NEXT transmitter package trace lengths in mm, one per case.
F_B	f_b	Baud (Signaling) rate in Gbaud.

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
C_D	C_d	C_d in nF (single sided).
L		Number of symbols levels (PAM-4 is 4, NRZ is 2).
M		Samples per UI.
Z_PTX	z_p (TX)	List of victim transmitter package trace lengths in mm, one per case.
Z_PRX	z_p (RX)	List of FEXT receiver package trace lengths in mm, one per case.
C_P	C_p	C_p in nF (single sided).
R_D	R_d	Die source termination resistance (single sided).
A_V	A_v	Victim differential peak source output voltage (half of peak to peak).
C_V		C_v in nF (via cap)(single sided).
F_V		For FOM_ILD: Transiton rate cut off frequency for ICN/ILD calc in terms of fb.

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
AC_CM_RMS		AC_CM_RMS is the CM BBN AWGN RMS at COM source point. Default is 0. Adds common mode noise source to the COM signal path for the through channel.
L_S	L_s	L_s in nH (single sided).
A_FE	A_fe	FEXT aggressor differential peak source output voltage (half of peak to peak).
LOCAL_SEARCH	Local Search	Decreases COM compute time. Setting to 2 seems ok, if 0 search is full grid.
DELTA_F	Delta_f	Frequency step.
T_O	T_h	Superceded with T_O but is the internal values that is used. Do not use.
ACCM_MAX_FREQ	ACCM_MAX_Freq	F max for integrating ACCM voltage in Hz. Default is

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		fb.
A_NE	A_ne	NEXT aggressor differential peak source output voltage (half of peak to peak).
SAMPLES_FOR_C2M	samples_for_C2M	Finer sampling in terms of samples per UI for c2m histogram analysis.
Z_PFEXT	z_p (FEXT)	List of FEXT transmitter package trace lengths in mm, one per case.
N_B_STEP		Discretization of DFE. 0 disables and is not normally used.
DER_0		Target detector error ratio.
F_MIN	f_min	Minimum required frequency start for s parameters.

TABLE_93A3

TABLE_93A3

SPISim Keyword	Matlab Keywor (Case sensitive)	Description and Notes
PACKAGE_Z_C	package_Z_c	Package model transmission line characteristic impedance [Tx , Rx].
PACKAGE_TL_GAMMA0_A1_A2	package_tl_gamma0_a1_a2	Fitting parameters for package model per unit length. First element is in 1/mm and affects DC loss of package model . Second element is in ns1/2/mm and affects loss proportional to sqrt(f). Third element is in ns/mm and affects loss proportional to f.
PACKAGE_TL_TAU	package_tl_tau	Package model transmission line delay ns/mm.

TABLE_9212

TABLE_9212

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
CB0	C_0	If include PCB is set to 1, near device single ended capacitance C0 in nF is added.
Z_BPNEXT	z_bp (NEXT)	Next Assessor transmitter board trace lengths in mm.
CB1	C_1	If include PCB is set to 1, connector side single ended capacitance C1 in nF is added.
Z_BPRX	z_bp (RX)	Victim receiver board trace lengths in mm.
BOARD_TL_TAU	board_tl_tau	Board model transmission line delay ns/mm.
Z_BPTX	z_bp (TX)	Victim transmitter board trace lengths in mm.
BOARD_Z_C	board_Z_c	Board model transmission line characteristic impedance [Tx , Rx].
BOARD_TL_GAMMA0_A1_A2	board_tl_gamma0_a1_a2	Fitting parameters for package model per unit length.

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		First element is in 1/mm and affects DC loss of package model . Second element is in ns1/2/mm and affects loss proportional to sqrt(f). Third element is in ns/mm and affects loss proportional to f.
Z_BPFEXT	z_bp (FEXT)	Fext Assessor transmitter board trace lengths in mm.

TDR_ERL

TDR_ERL

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
TR_TDR		Gaussian shaped transition time for TDR source in ns.
FIXTURE_DELAY_TIME	fixture delay time	Fixture delay time (for ERL).
ERL_ONLY		Compute ERL only.
AUTO_TFX		Mostly used for device ERL. If

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		sent to 1 the fixture tfx will be estimated.
Z_T		Single sided source termination reference resistance for TDR and ERL.
N		Duration time in UI which is used for ERL (PTDR).
TDR_DURATION		Only used if N*UI is longer than the TDR duration time. Default is 5 times the raw s-parameter transit time.
BT_ORDER	BTorder	Bessel function order.
TUKEY_WINDOW	Tukey_Window	Required for ERL. Set to 1. Default is 0.
RHO_X	rho_x	(For ERL) use default 0.618.
BETA_X	beta_x	(For ERL) use default 0.
FIXTURE_BUILTIN_DELAY		Built-in fixture delay.
ERL		Enables ERL. Needs TDR to be set as well.

ICN_PARAM

ICN_PARAM

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
A_FT	A_ft	FEXT aggressor amplitude for ICN. Defaults to A_fe if not specified.
F_N		For ICN: Next transiton rate cut off frequency for ICN calc in terms of fb.
F_2	f_2	Frequency in GHz for intergration compuation of ICN or FOM_Ild in GHz.
F_1	f_1	Start frequency for ICN and ILD calculations in GHz.
A_NT	A_nt	NEXT aggressor amplitude for ICN. Defaults to A_ne if not specified.
F_F		For ICN: Fext transition rate cut off frequency for ICN calc in terms of fb.

FILTER_EQ

FILTER_EQ

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
G_DC_HP	g_DC_HP	CTF AC-DC gain list (GDC2).
F_HP_PZ	f_HP_PZ	CFT pole pole zero pair in GHz for low frequency CTF.
B_MAX2_N_B	b_max(2..N_b)	DFE magnitude limit for second coefficient and on (ignored if Nb<2). Can be a regular expression.
C0	c(0)	TX equalizer cursor minimum value (actual value is calculated as 1-sum(abs(tap)), Grid seat ignored for when C(0) is below this value.
C1	c(1)	TX equalizer post cursor tap 1.
G2_QUAL	G2_Qual	G2_Qual limit values of g_ DC_HP (g DC2) which corresponds to ranges of g_DC g DC specified with G_QUAL.
C2	c(2)	TX equalizer post cursor tap

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		2.
G_QUAL	G_Qual	G_Qual are the dB ranges of g_DC (g DC)which correspond tog_DC_HP (g DC2).
C3	c(3)	TX equalizer post cursor tap 3.
N_B	N_b	Decision feedback fixed equalizer (DFE) length.
B_MIN2_N_B	b_min(2..N_b)	DFE negative magnitude limit; if not specified it defaults to -b_max(2..N_b).
G_DC	g_DC	AC-DC gain list.
F_R	f_r	Reference receiver filter in COM and in ICN/FOM_ILD calcs in terms of fb.
F_P2	f_p2	CTLE pole 2 in GHz.
F_P1	f_p1	CTLE pole 1 in GHz.
F_Z	f_z	CTLE zero in GHz.
B_MIN1	b_min(1)	DFE negative magnitude limit. If not specified it defaults to -bmax.

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
CTLE_TYPE		Sets the CTLE type default as poles and zeros (i.e. not a list of poles as in 120e).
F_HP_Z		CFT pole pole zero pair in GHz for low frequency CTF.
C_1	c(-1)	TX equalizer pre cursor tap -1.
B_MAX1	b_max(1)	DFE negative magnitude limit. If not specified it defaults to -bmax.
C_3	c(-3)	TX equalizer pre cursor tap -3.
C_2	c(-2)	TX equalizer pre cursor tap -2.
C_4	c(-4)	TX equalizer pre cursor tap -4.
GDC_MIN		Max ACDC gain, if 0 ignore.
F_HP_P		CFT pole fp2 is in GHz. Normally a list for 120e. Not normally used otherwise.

NOISE_JITTER

NOISE_JITTER

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
A_DD		Normalized peak dual-Dirac noise, this is half of the total bound uncorrelated jitter (BUJ) in UI.
ETA_0	eta_0	One-sided noise spectral density (V^2/GHz). Input referred noise at TP5.
SNR_TX		Transmitter SNDR noise in dB.
R_LM		Ratio of level separation mismatch. Relevant when not PAM-2 (NRZ).
SIGMA_RJ	sigma_RJ	rms of random jitter.

RCV_NONSTD

RCV_NONSTD

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
GRR_LIMIT		Either do not use or set to 1 (for ERL).
INCLUDE_CTLE		Do not use.

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
INC_PACKAGE		Warning: INC_ PACKAGE=0 not fully supported, instead, set Zp,Cd, and Cp parameters to zero and Zp to 1.
T_R	T_r	20% to 80% transition time used for the Gaussian shaped source.
GRR	Grr	Either do no use or set to 1 (for ERL).
FORCE_TR		Included for earlier version support but should be set to 1 in most later config sheets.
GX	Gx	ERL parameter param.Grr; This is used in the COM code.
IDEAL_TX_TERM		Not supported, instead, set Zp ,Cd, and Cp parameters to zero and Zp to 1
KAPPA2	kappa2	If set 0, reflection at tp5 are omitted from COM.
KAPPA1	kappa1	If set 0, reflection at tp0

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		are omitted from COM.
SIGMA_BBN_STEP	Sigma BBN step	BBN step for Rx Calibration in volts. Defaults is 0.5e-3.
T_R_FILTER_TYPE		Included for earlier version support. Not recommended to use.
IDEAL_RX_TERM		IDEAL_RX_TERM not supported, instead, set Zp, Cd, and Cp parameters to zero and Zp to 1
INCLUDE_TX_RX_FILTER		Do not use.
RX_CALIBRATION		Turn on RX_Calibration loop.
T_R_MEAS_POINT		Included for earlier version support. Not recommended to use.

FLOATING_TAPS

FLOATING_TAPS

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
N_BF	N_bf	Number of taps

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		in group.
N_BG	N_bg	Number of group of floating tap. Used as a switch, 0 means no float.
BMAXG	bmaxg	Max DFE value for floating taps.
N_F	N_f	UI span for floating taps. Replaced by N_bmax.
N_TAIL_START	N_tail_start	Start range for max RSS limit for DFE taps.
B_FLOAT_RSS_MAX	B_float_RSS_MAX	Floating DFE tap start for RSS floating tap limit.

RX_FFE

RX_FFE

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
FFE_TAP_STEP_SIZE		Rx FFE tap step size.
FFE_PRE_TAP1_MAX		Rx FFE pre cursor tap1 limit.
FFE_POST_TAP1_MAX		Rx FFE post cursor tap1 limit.
FFE_TAPN_MAX		Rx FFE

SPISim Keyword	Matlab Keyword (Case sensitive)	Description and Notes
		precursor tap N limit.
FFE_MAIN_CUR_MIN		Rx FFE main cursor miminum.
FFE_BACKOFF		See if better zero forced solution is better by backing off the number specified FFE taps one at a time.
FFE_PRE_TAP_LEN	ffe_pre_tap_len	Rx FFE pre cursor tap length.
CURSOR_GAIN		Only FFE and not supported.
FFE_POST_TAP_LEN	ffe_post_tap_len	Rx FFE post cursor tap length.

The COM analysis keywords can also be referenced at https://www.ieee802.org/3/ck/public/tools/index.html.

UDO/UDS-Based COM/ERL Calculations for HFSS-Based Solutions

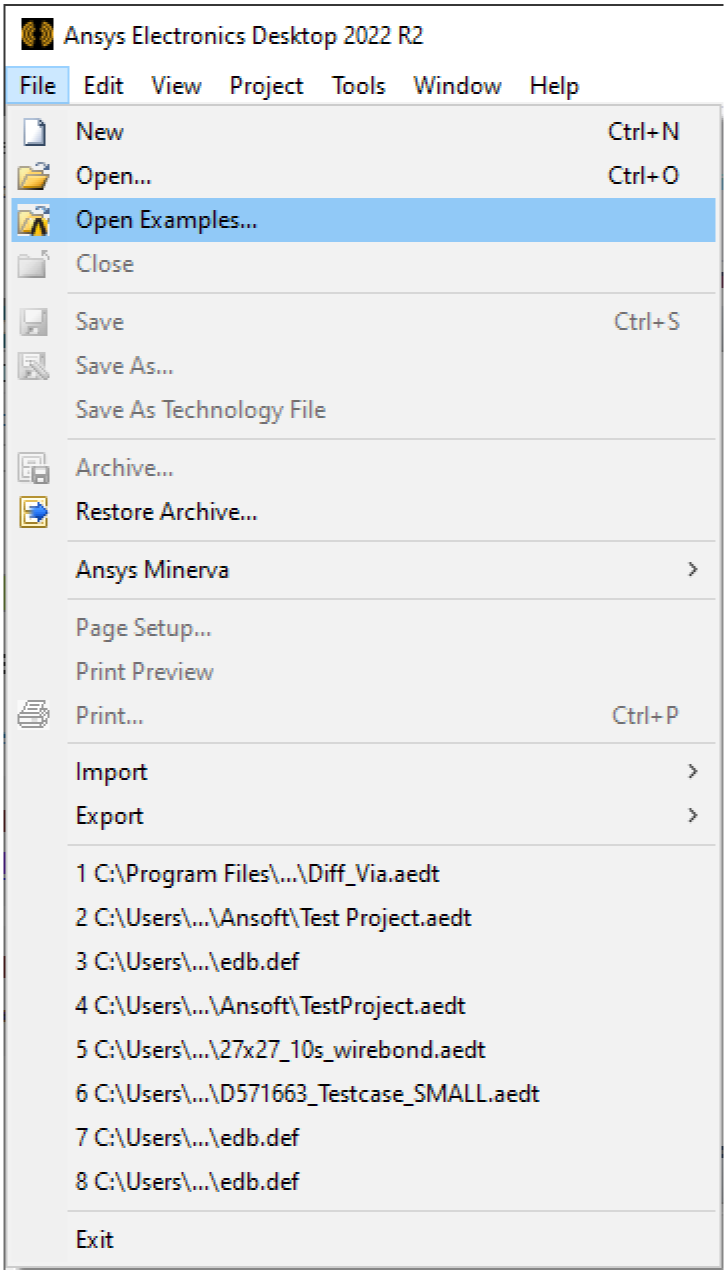
Generate an User-Defined Output (UDO)-based COM analysis report by completing the steps in the following subsections.

> **Note:** A COM analysis report can also be generated in other ways. Refer to " Generating a COM Analysis Report From the SPISim User Interface " on page 15-143 and " Generating a COM Analysis Report Using a Non-GUI Batch Command Line" on page 15-159 for details.

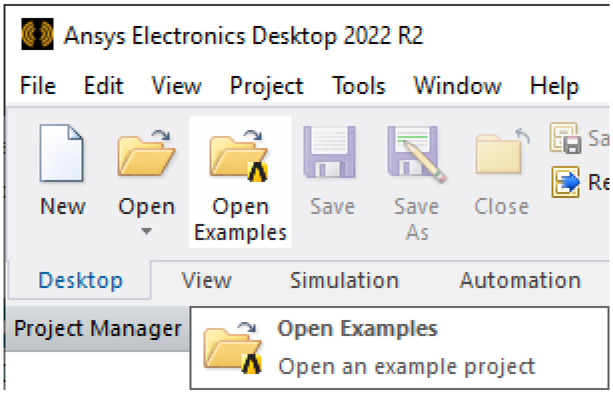
Open the Example Project

1. Open the example project **Diff_Via.aedt** by doing one of the following:

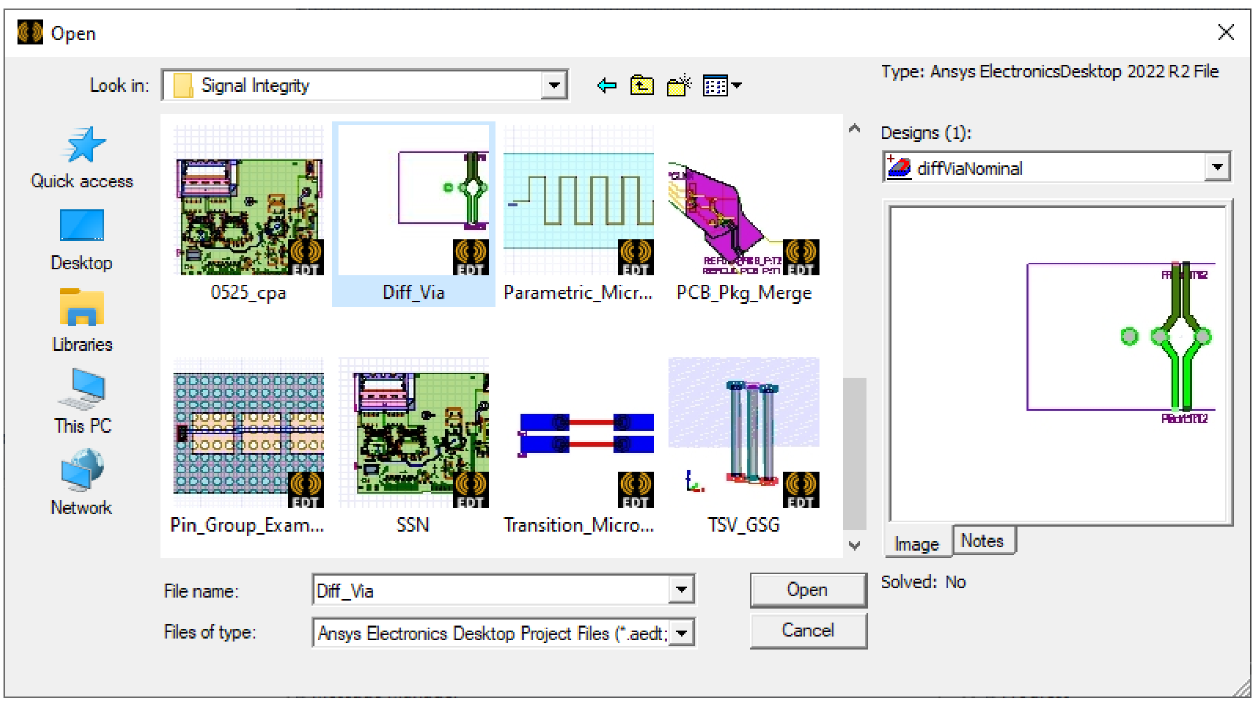
- From **File**, select **Open Examples**.



- From the **Desktop** ribbon, click **Open Examples**.



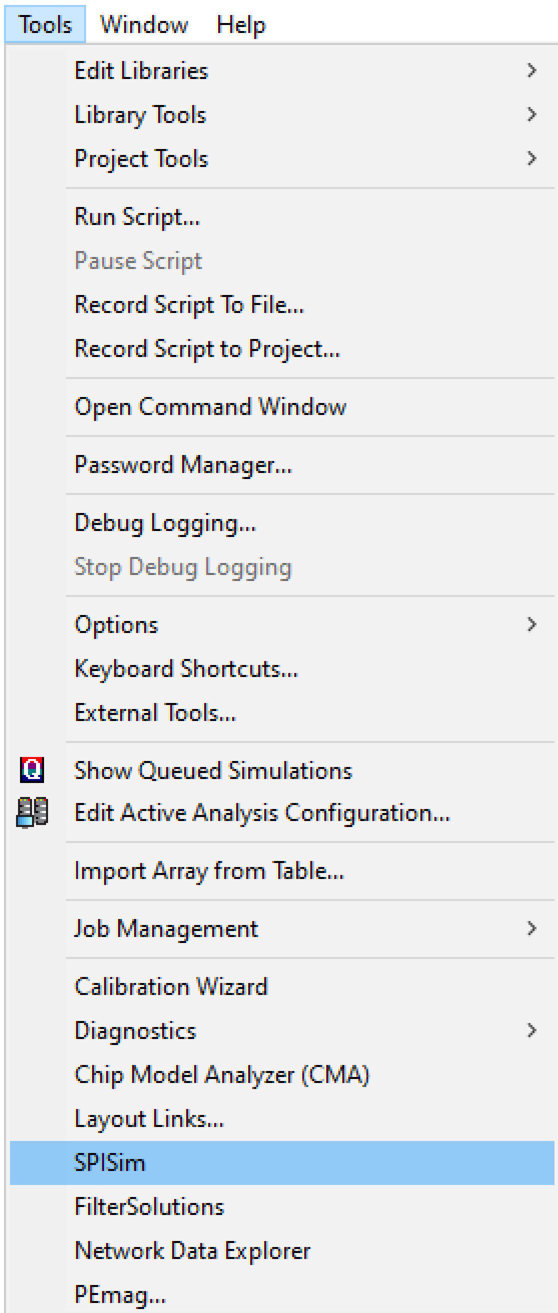
2. From the explorer window that opens, navigate to **HFSS 3D Layout** > **Signal Integrity** and either double-click **Diff_Via.aedt** or click **Diff_Via.aedt** and then click **Open**.



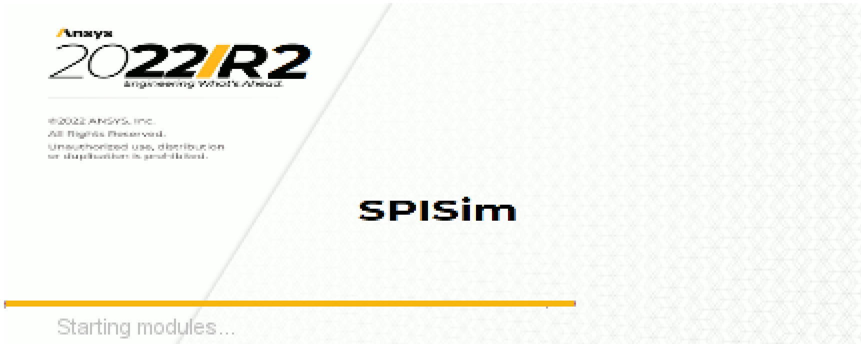
3. If necessary, navigate to an appropriate directory and save the project.

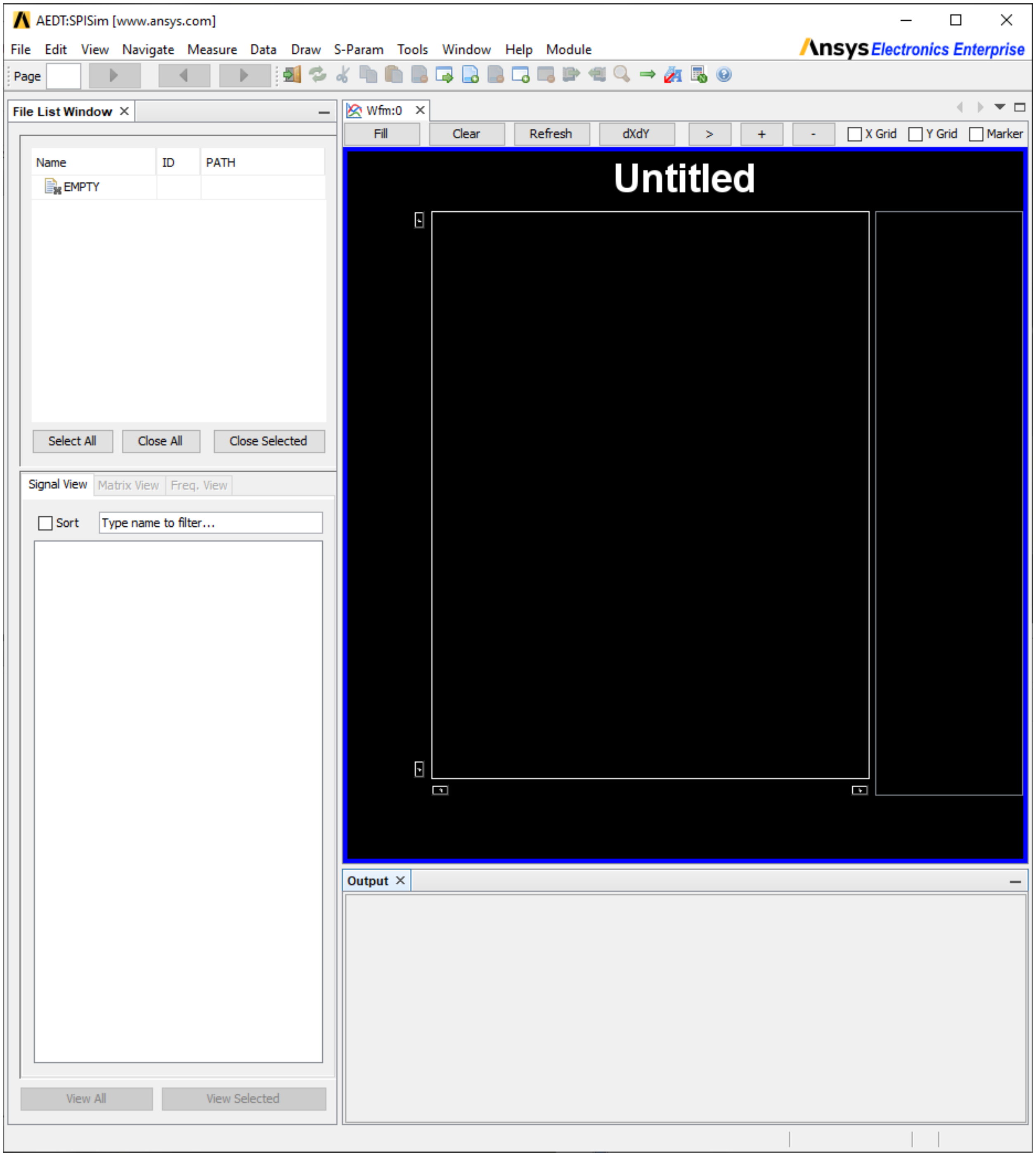
Create a Configuration File in SPISim

1. From **Tools**, select **SPISim**.

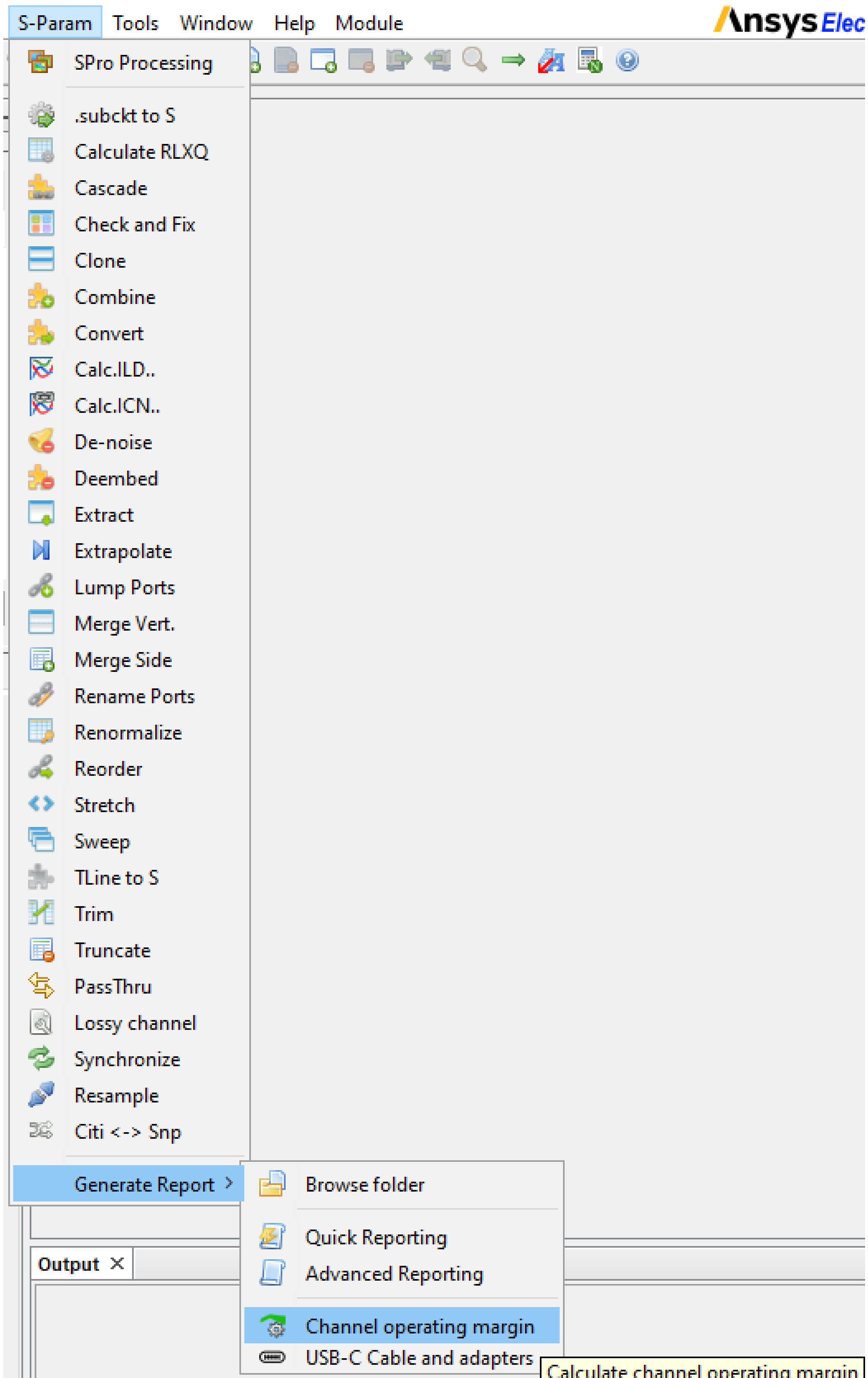


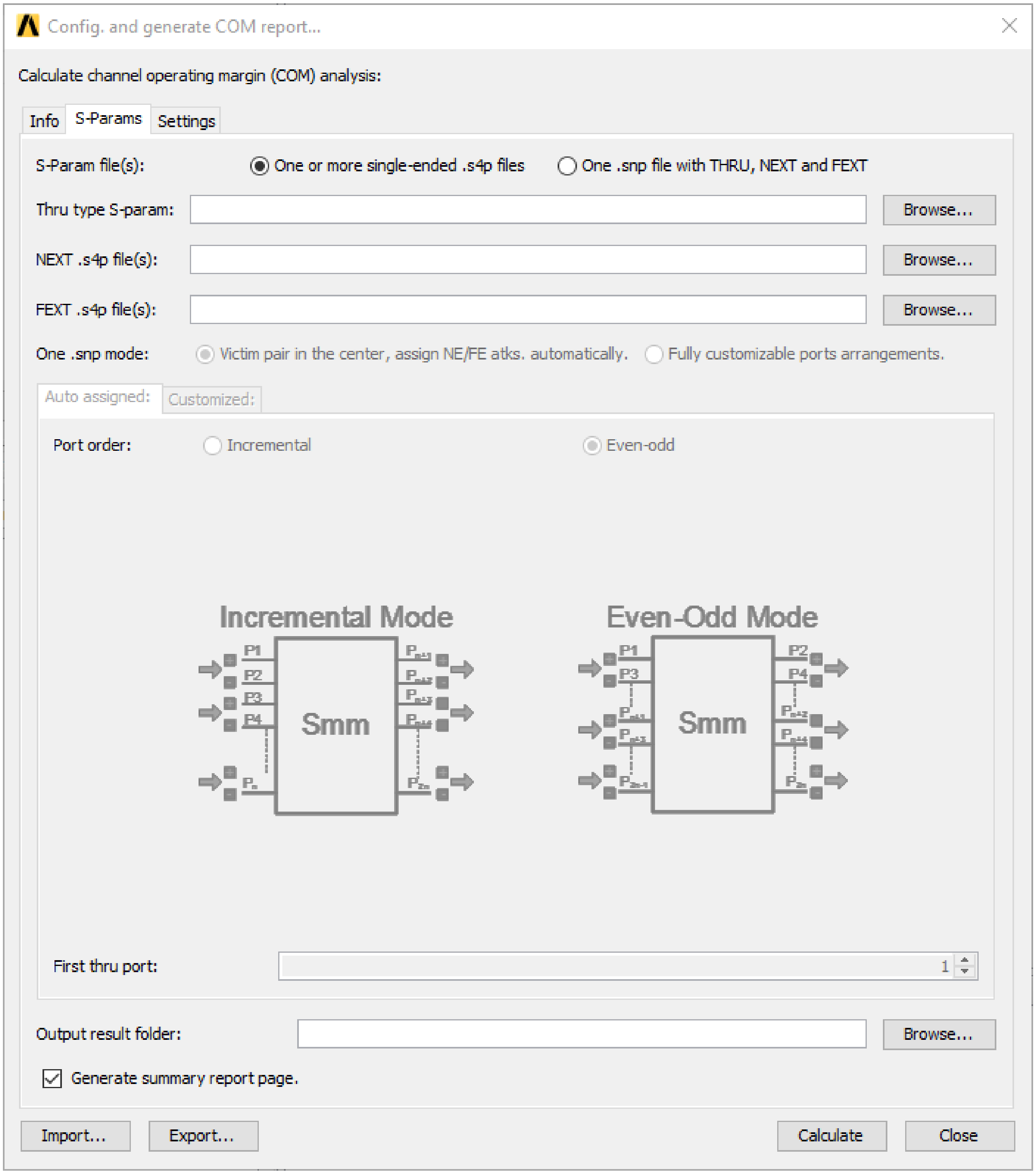
2. Wait for **SPISim** to load.





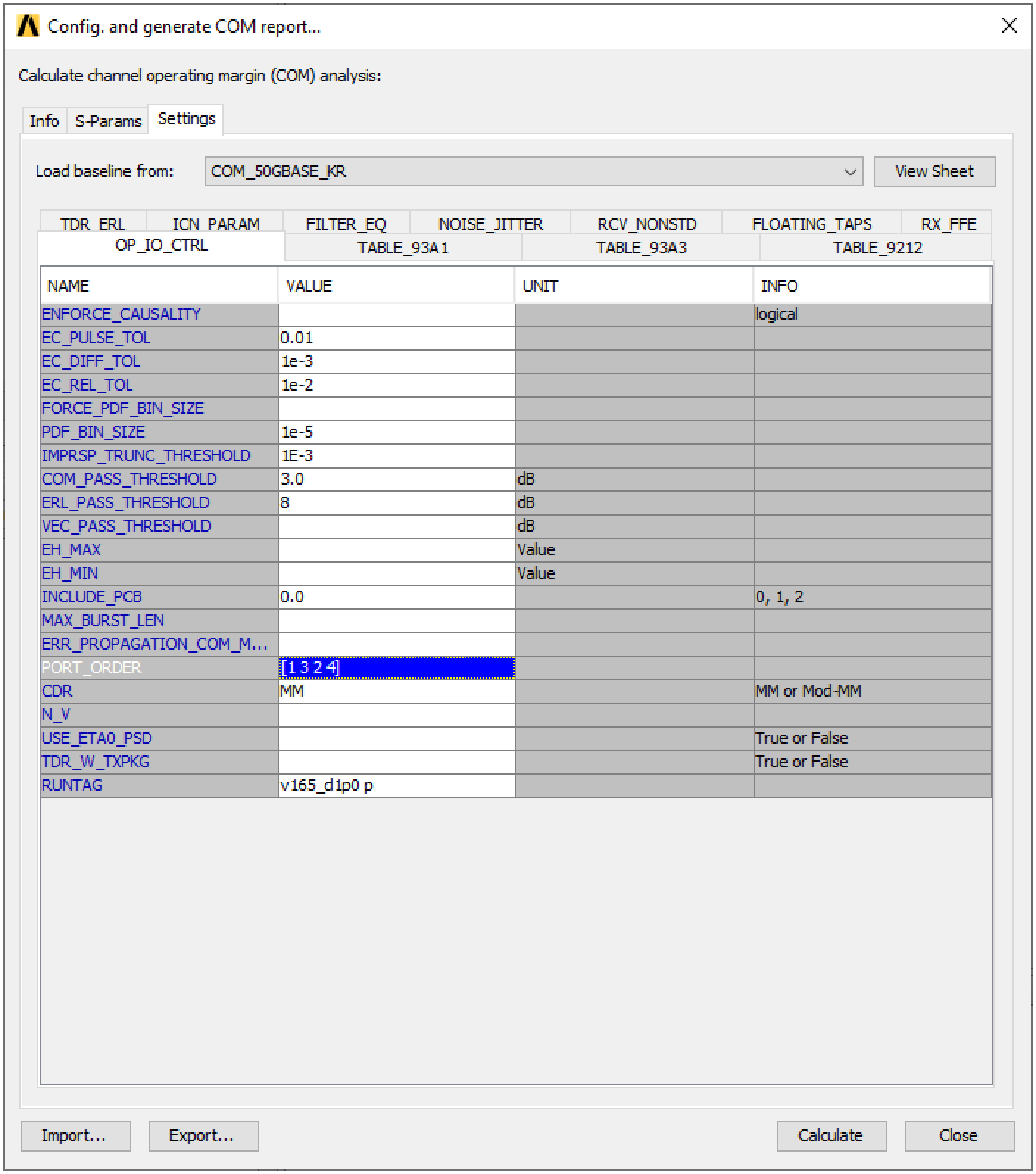
3. From **SPISim**, select **S-Param** > **Generate Report** > **Channel operating margin** to open the **Config. and generate COM report...** window.


S-Param
Tools
Window
Help
Module
Ansys Elec
SPro Processing
.subckt to S
Calculate RLXQ
Cascade
Check and Fix
Clone
Combine
Convert
Calc.ILD..
Calc.ICN..
De-noise
Deembed
Extract
Extrapolate
Lump Ports
Merge Vert.
Merge Side
Rename Ports
Renormalize
Reorder
Stretch
Sweep
TLine to S
Trim
Truncate
PassThru
Lossy channel
Synchronize
Resample
Citi <-> Snp
Generate Report
Browse folder
Quick Reporting
Advanced Reporting
Channel operating margin
USB-C Cable and adapters
Calculate channel operating margin
Output



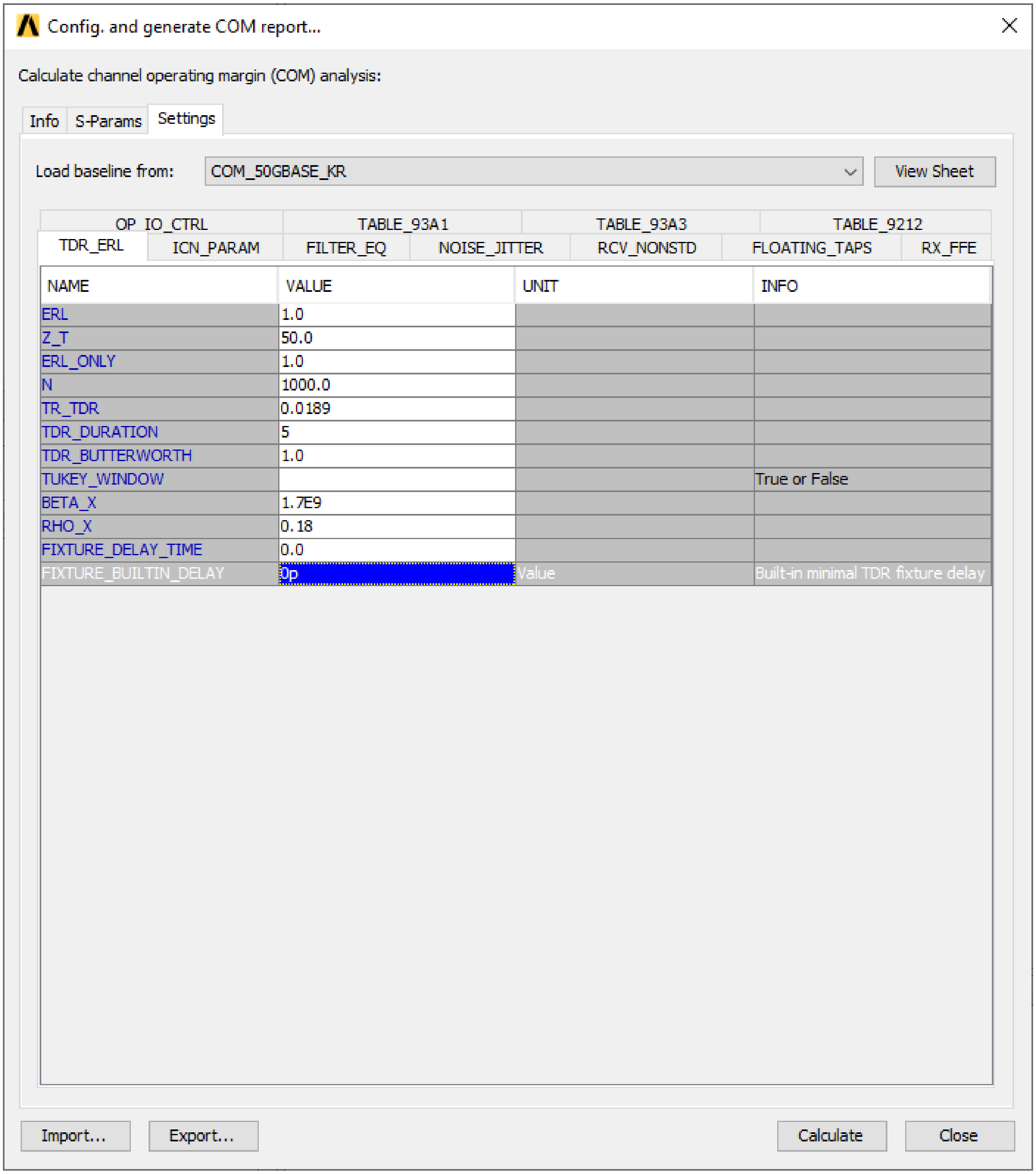
4. Navigate to the **Settings** tab. Then do the following:

- From the **Load baseline from** drop-down menu, select **COM_50GBASE_KR**.
- From the **OP_IO_CTRL** subtab, type **[1 2 3 4]** in the **PORT_ORDER VALUE** field.



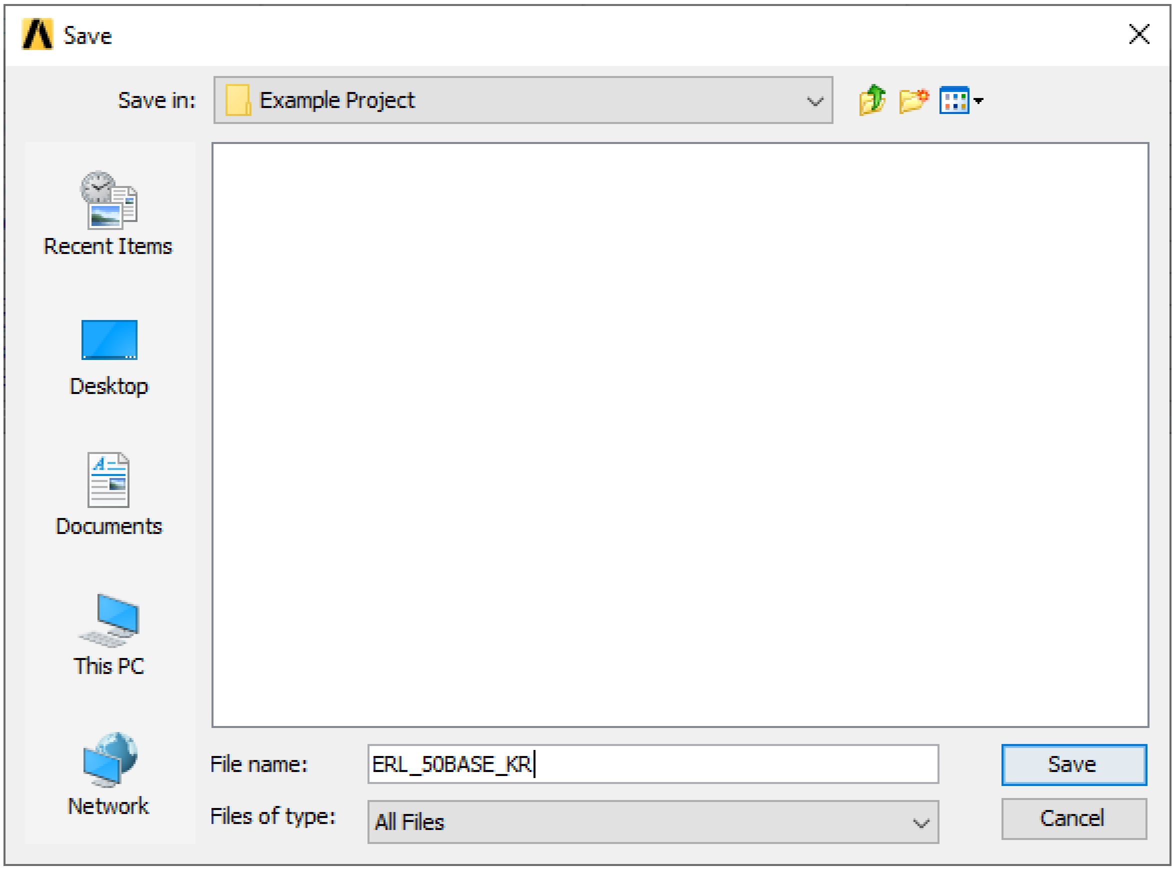
5. Navigate to the **TDR_ERL** subtab and make the following changes:

- Enter **1.0** in the **ERL_ONLY Value** field.
- Enter **0p** in the **FIXTURE_BUILTIN_DELAY Value** field.

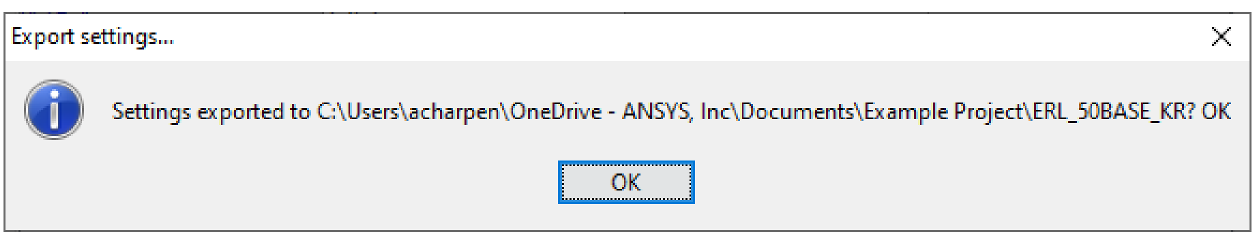


6. Click **Export...** to open an explorer window. Then do the following:

- Navigate to an appropriate directory to save the configuration file (*.*cfg*).
- Enter an appropriate **File Name** (e.g., **ERL_50BASE_KR.cfg**).

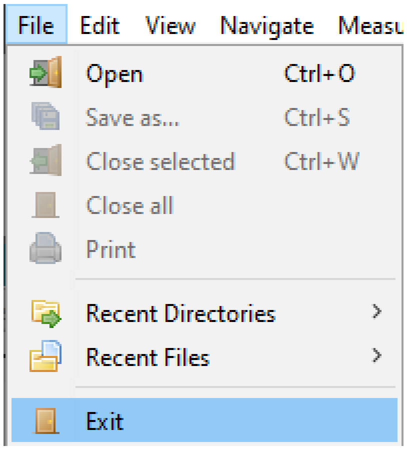


7. Click **Save** to close the explorer window and simultaneously open an **Export settings...** window.



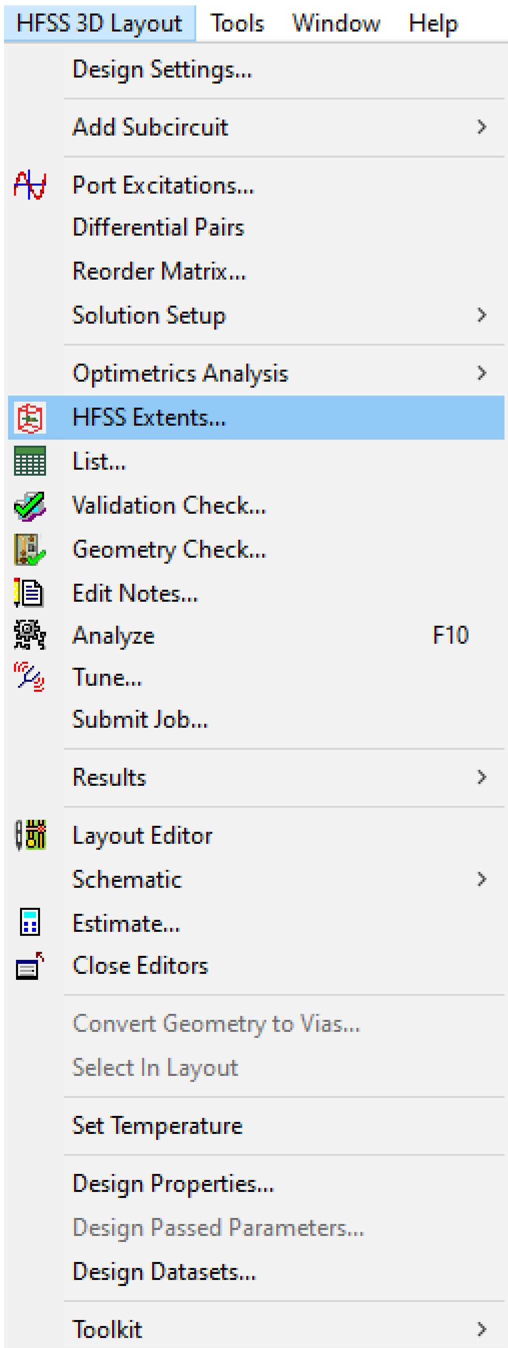
8. Click **OK** to close the Export Settings window.
9. Click **Close** to close the **Config. and generate COM report...** window.

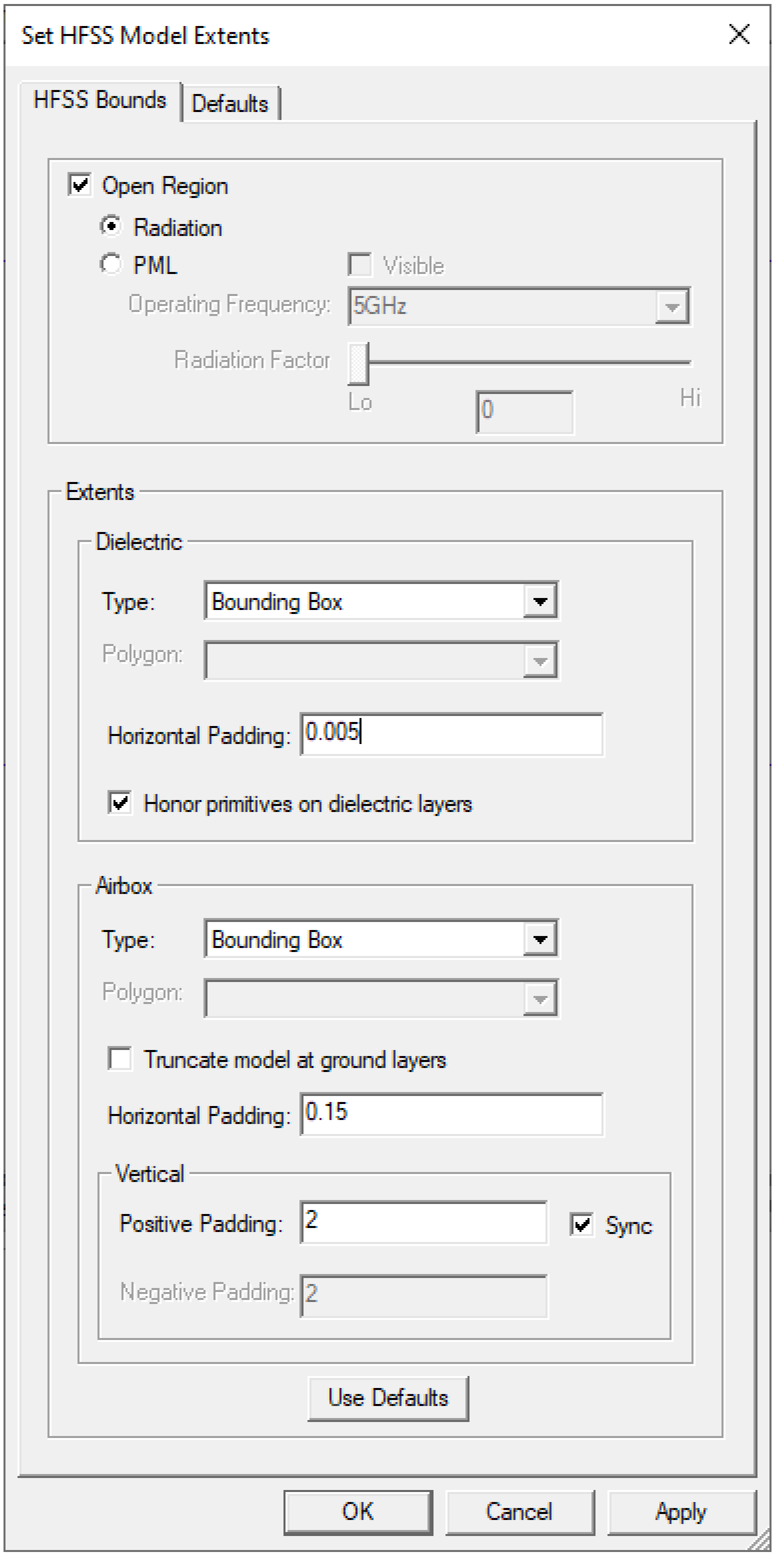
10. From **File**, select **Exit** to close **SPISim**.



Set Up an HFSS Solution

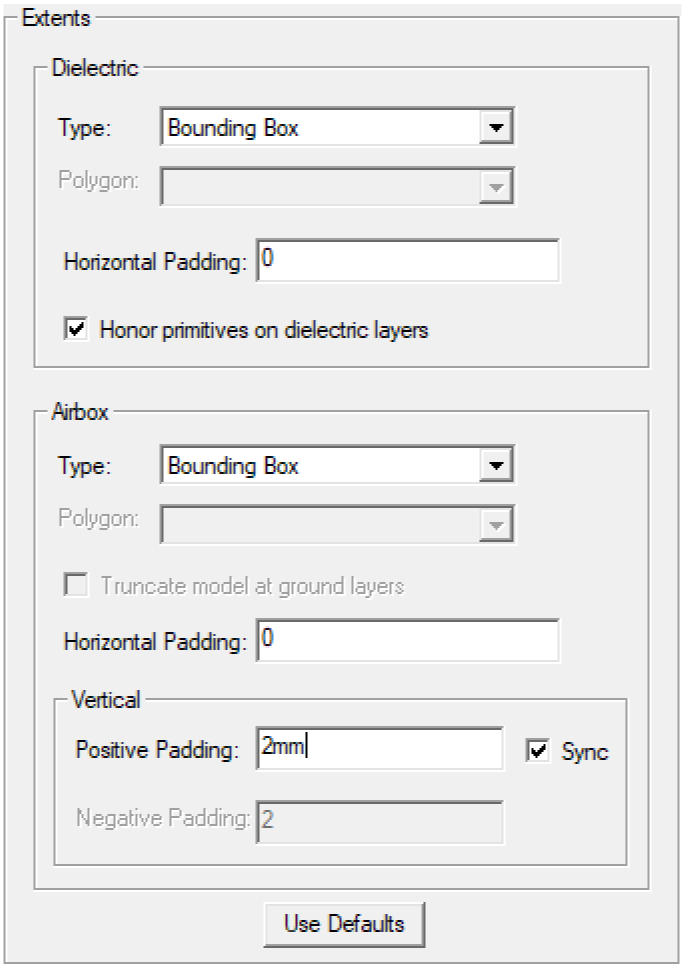
1. From **Electronics Desktop**, select **HFSS 3D Layout** > **HFSS Extents...** to open the **Set HFSS Model Extents** window.


HFSS 3D Layout
Tools
Window
Help
Design Settings...
Add Subcircuit
Port Excitations...
Differential Pairs
Reorder Matrix...
Solution Setup
Optimetrics Analysis
HFSS Extents...
List...
Validation Check...
Geometry Check...
Edit Notes...
Analyze
F10
Tune...
Submit Job...
Results
Layout Editor
Schematic
Estimate...
Close Editors
Convert Geometry to Vias...
Select In Layout
Set Temperature
Design Properties...
Design Passed Parameters...
Design Datasets...
Toolkit

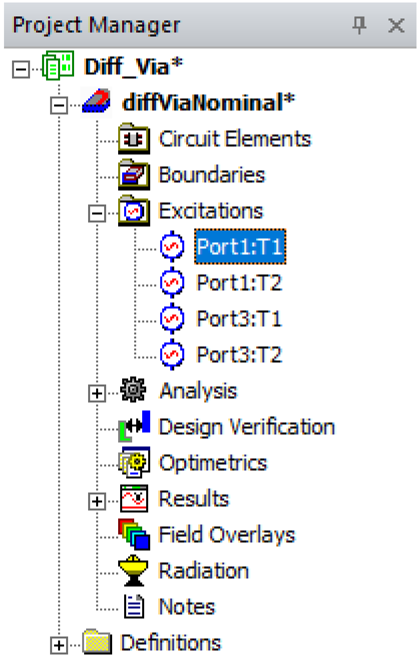


2. From the **Extents** group box, do the following:

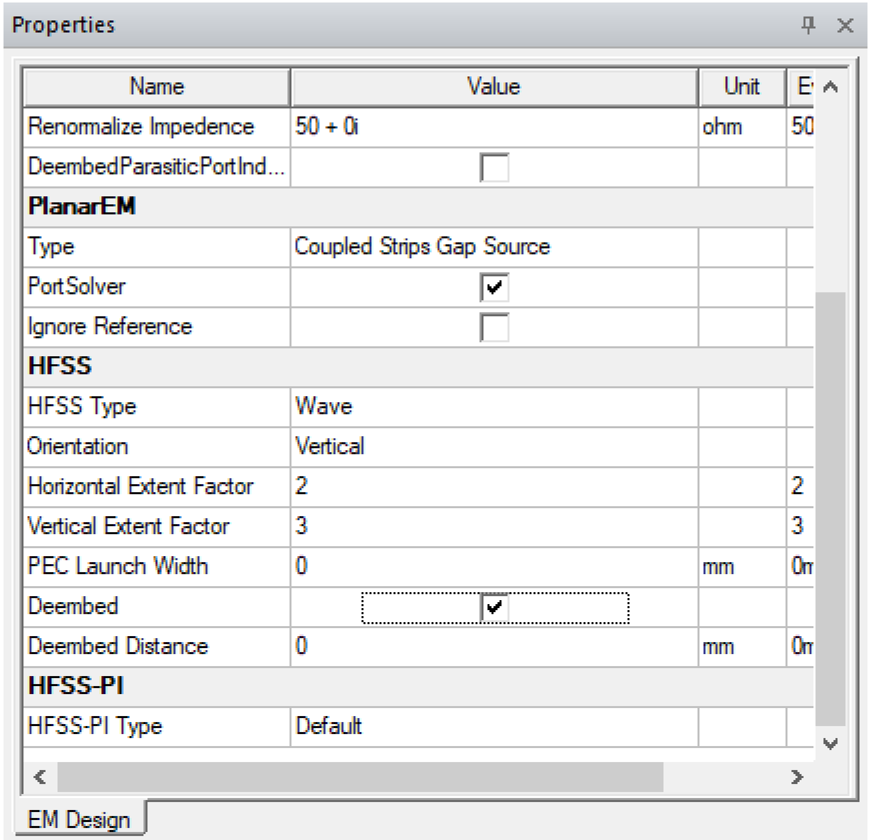
- In the *Dielectric* subgroup, enter **0** in the **Horizontal Padding** field.
- In the *Airbox* subgroup, enter **0** in the **Horizontal Padding** field.
- In the *Airbox* subgroup, enter **2mm** in the **Positive Padding** field.



3. Click **Apply** to save the configuration. Then click **OK** to close the **Set HFSS Model Extents** window.
4. From the **Project Manager** window, expand the **Project Tree** > [*active design folder*] > **Excitations**. Then click **Port1:T1**.

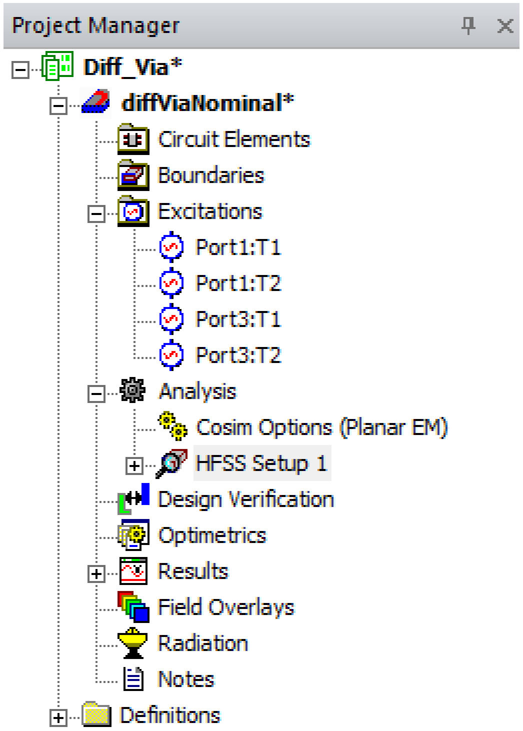


5. From the **Properties** window, type **0** in the **PEC Launch Width Value** field. Then press **Enter**.

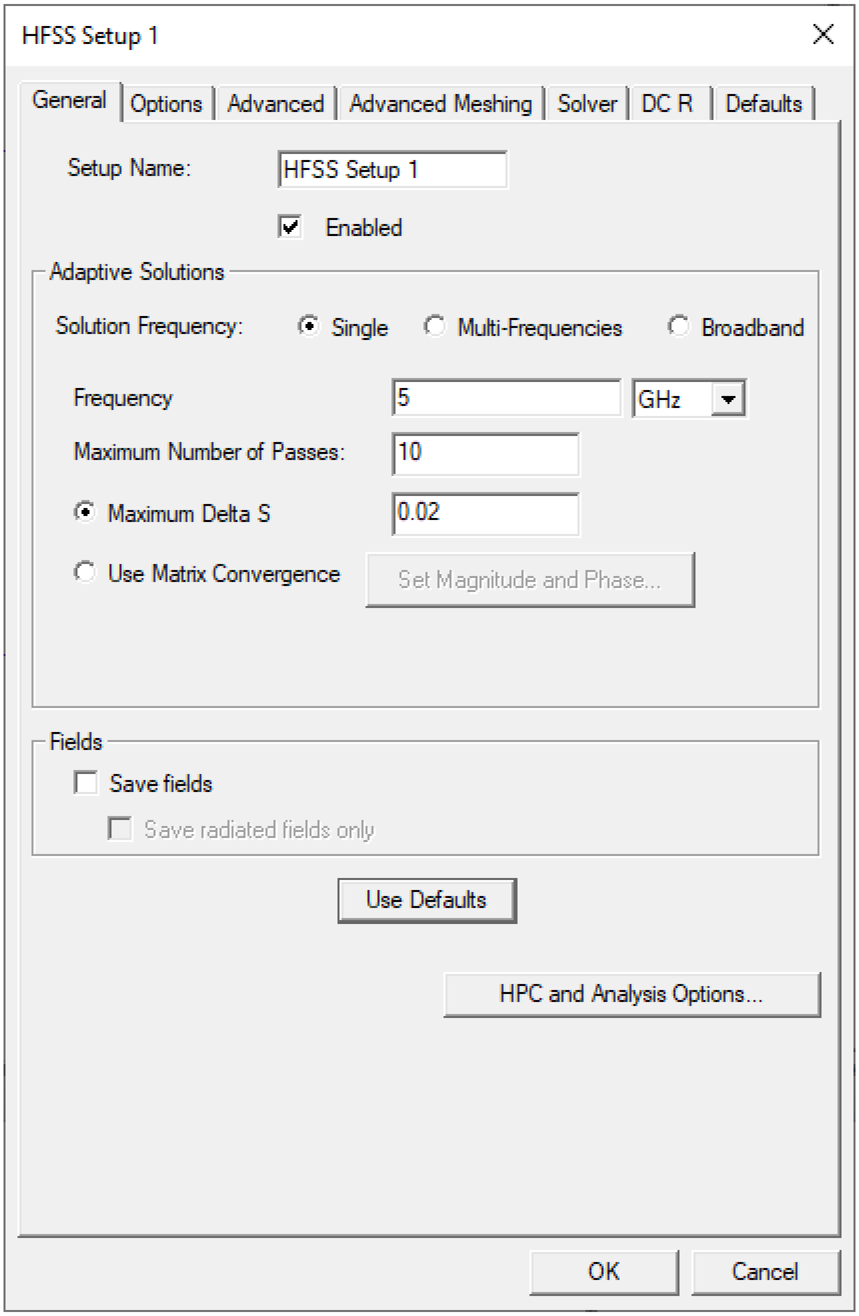


6. Repeat steps **4-5** for **Port3:T1**.

7. From the **Project Manager** window, expand **Analysis**.

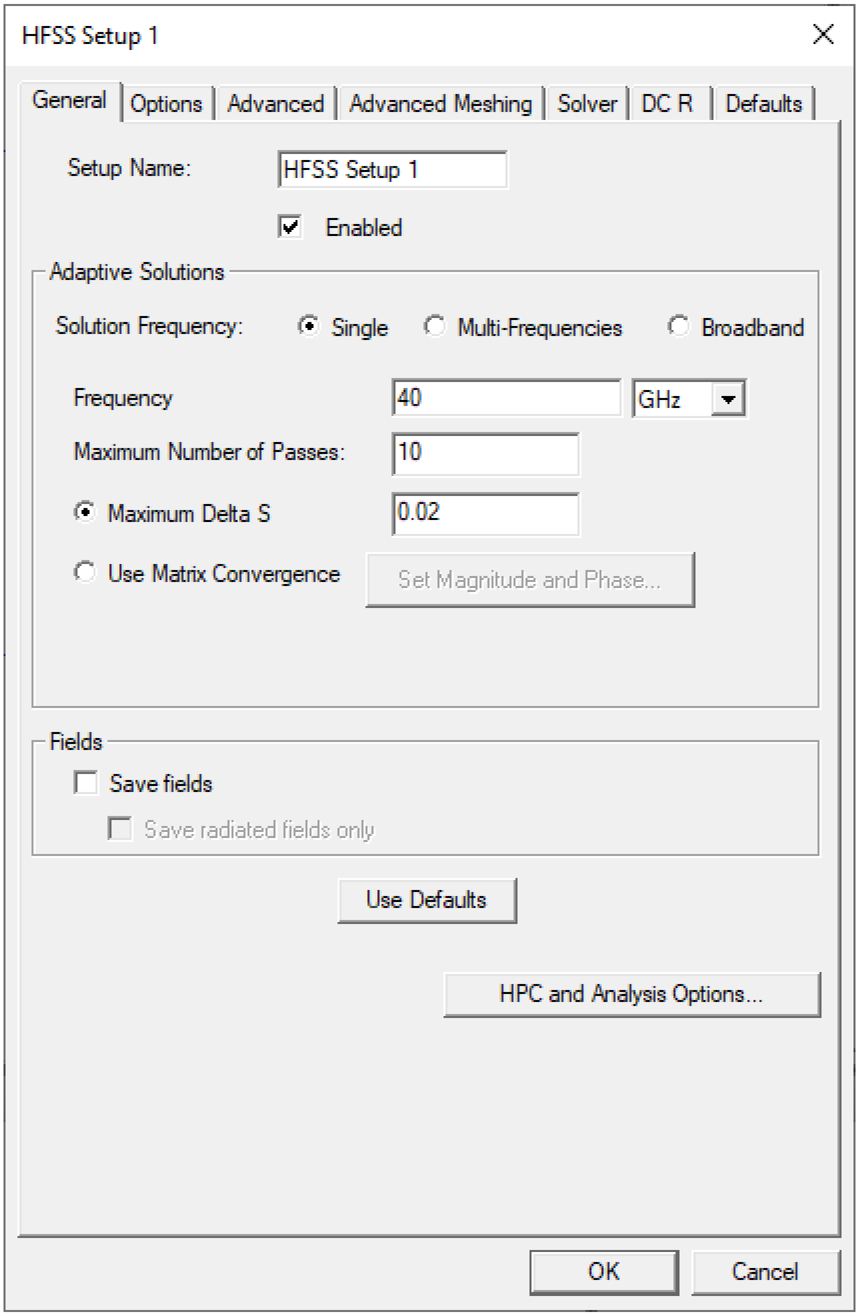


8. Double-click **HFSS Setup 1** to open the **HFSS Setup 1** window.



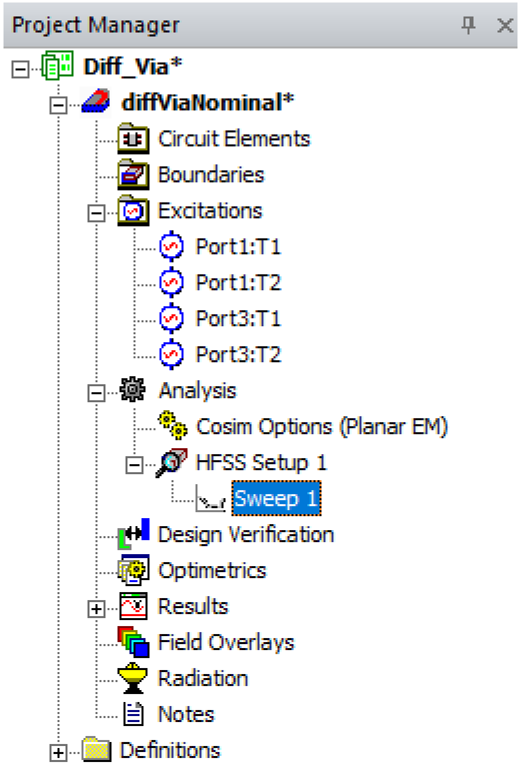
9. From the **HFSS Setup 1** window, make the following changes:

- From the **Adaptive Solutions** group box, enter **40** in the **Frequency** field.
- If necessary, enter **10** in the **Maximum Number of Passes** field.

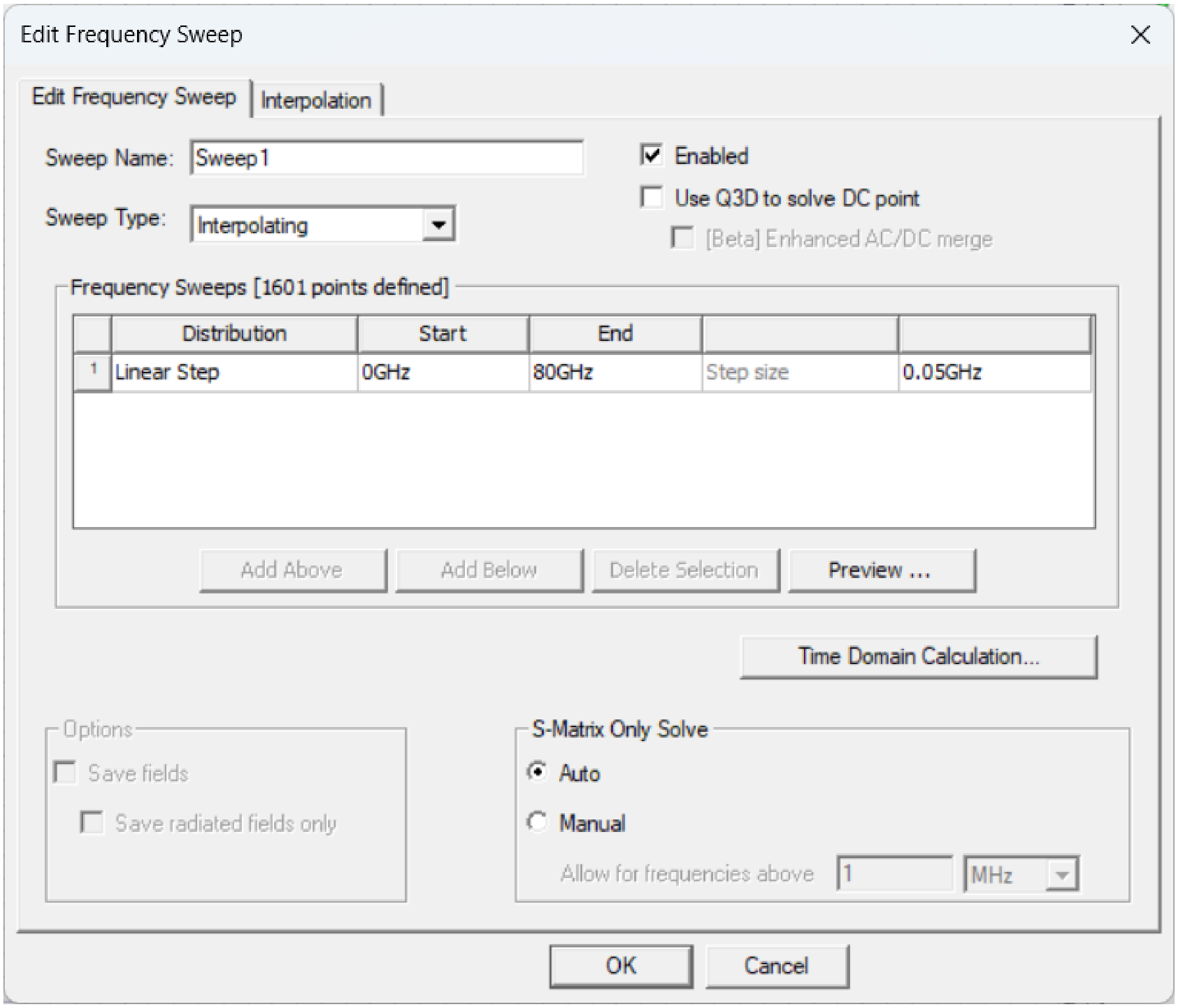

HFSS Setup 1
General
Options
Advanced
Advanced Meshing
Solver
DC R
Defaults
Setup Name:
HFSS Setup 1
Enabled
Adaptive Solutions
Solution Frequency:
Single
Multi-Frequencies
Broadband
Frequency
40
GHz
Maximum Number of Passes:
10
Maximum Delta S
0.02
Use Matrix Convergence
Set Magnitude and Phase...
Fields
Save fields
Save radiated fields only
Use Defaults
HPC and Analysis Options...
OK
Cancel

10. Click **OK** to close the **HFSS Setup 1** window.

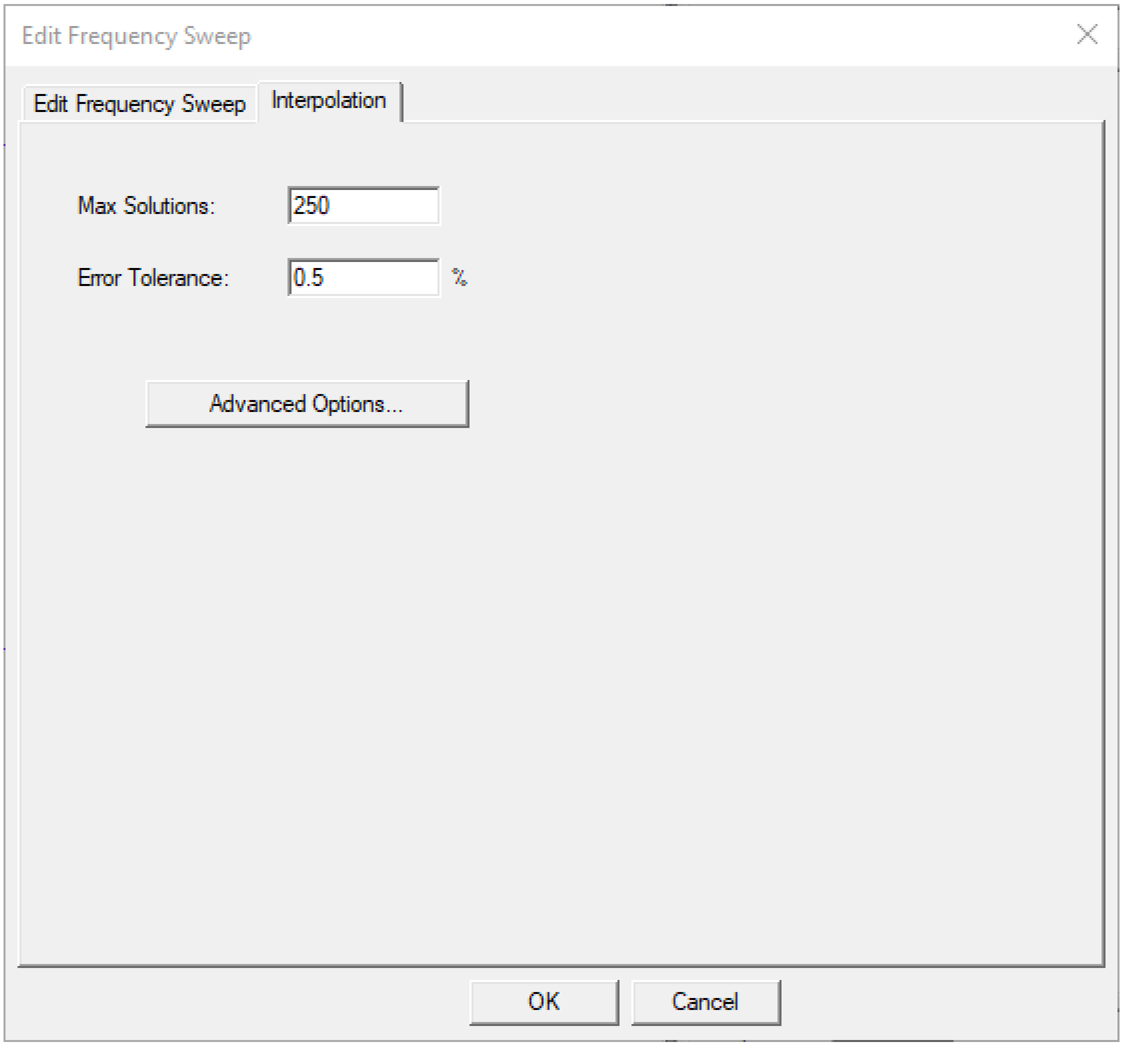
11. From the **Project Manager** window, expand **HFSS Setup 1**.



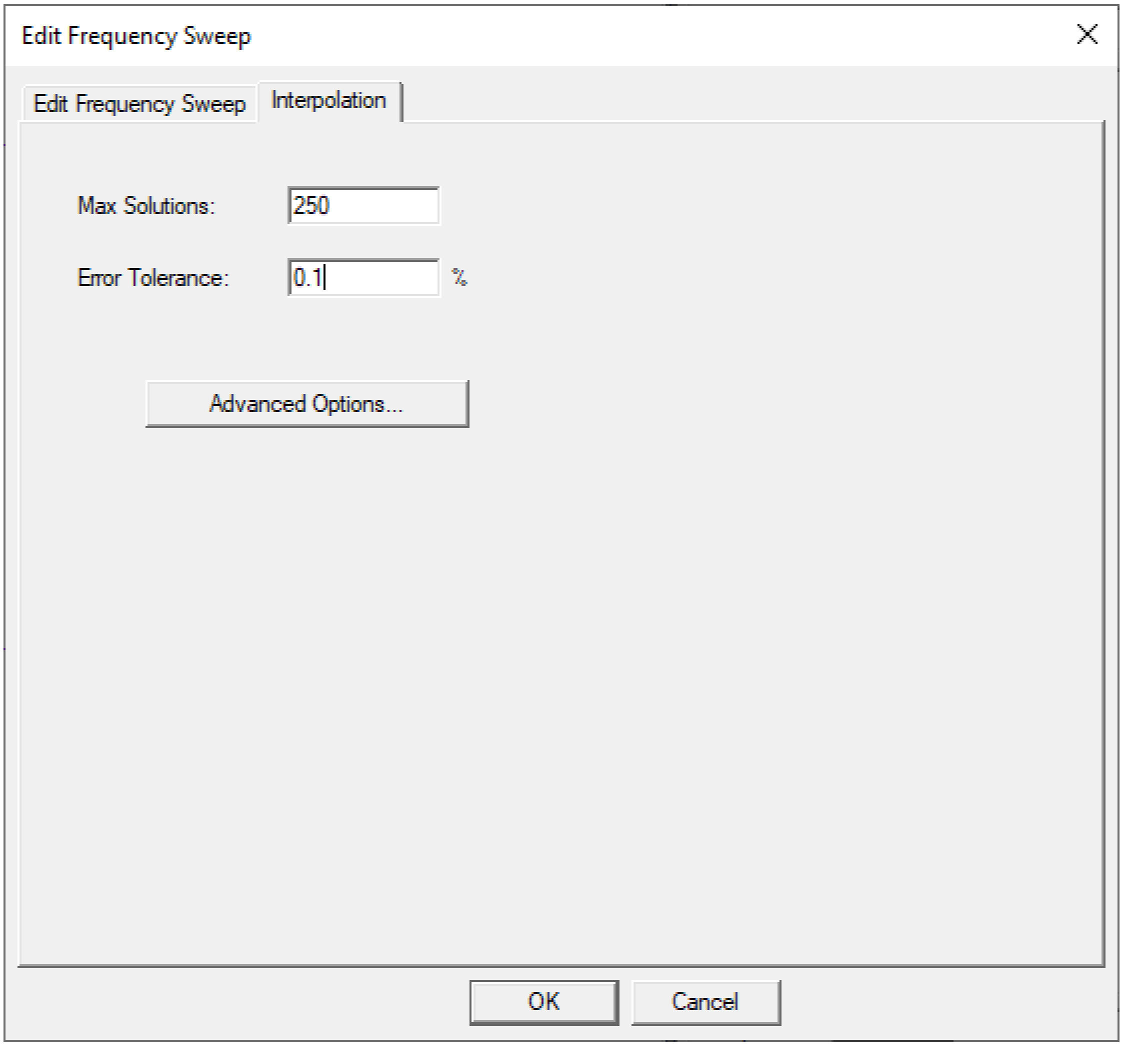
12. Double-click **Sweep 1** to open the **Edit Frequency Sweep** window. Then type **80GHz** in the **Frequency Sweeps** table > **Linear Step** > **End** column and press **Enter**.


Edit Frequency Sweep
Edit Frequency Sweep
Interpolation
Sweep Name:
Sweep1
Enabled
Use Q3D to solve DC point
Sweep Type:
Interpolating
[Beta] Enhanced AC/DC merge
Frequency Sweeps [1601 points defined]
Distribution
Start
End
1
Linear Step
0GHz
80GHz
Step size
0.05GHz
Add Above
Add Below
Delete Selection
Preview ...
Time Domain Calculation...
Options
Save fields
Save radiated fields only
S-Matrix Only Solve
Auto
Manual
Allow for frequencies above
1
MHz
OK
Cancel

13. Navigate to the **Interpolation** tab.



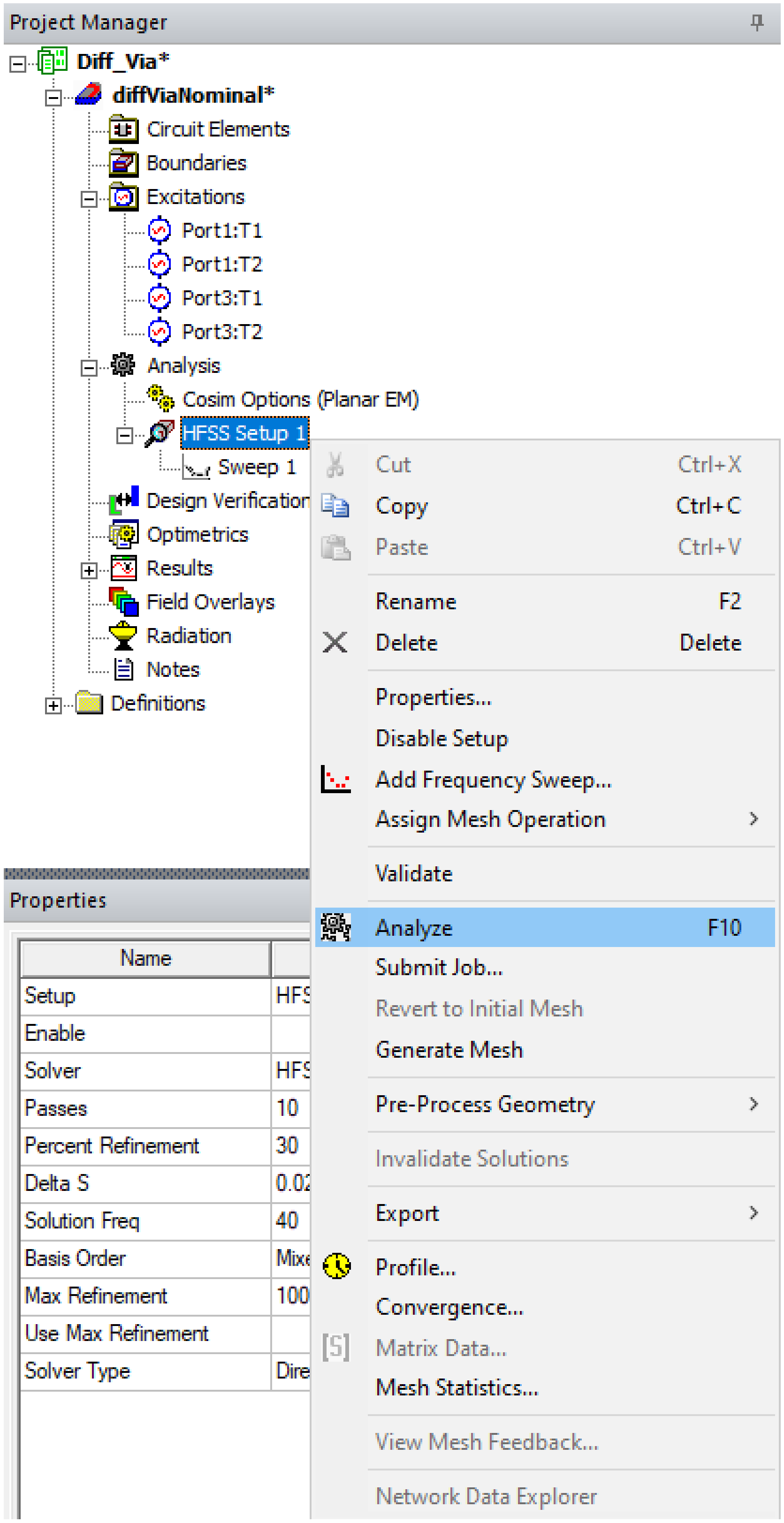
14. Enter **0.1** in the **Error Tolerance** field.



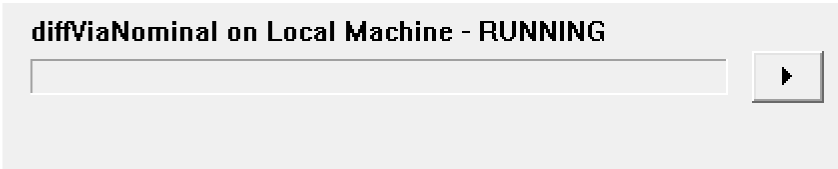
15. Click **OK** to close the **Edit Frequency Sweep** window.

Analyze the Solution, Then Create a Plot and Report

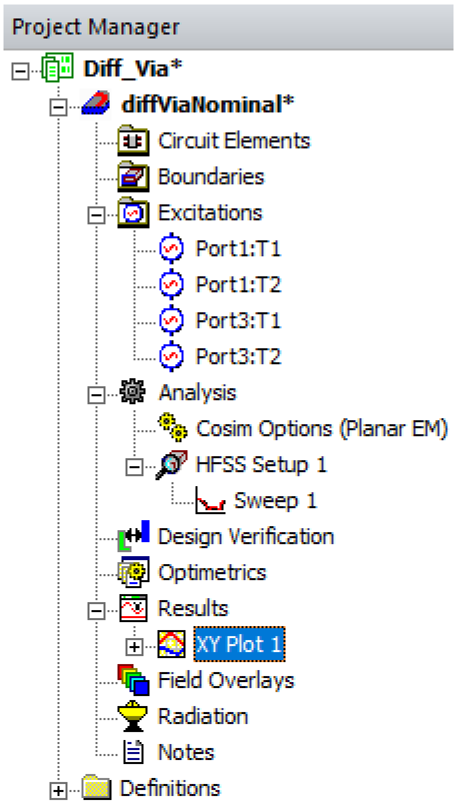
1. From the **Project Manager** window, right-click **HFSS Setup 1** and select **Analyze**.


Project Manager
Diff_Via*
diffViaNominal*
Circuit Elements
Boundaries
Excitations
Port1:T1
Port1:T2
Port3:T1
Port3:T2
Analysis
Cosim Options (Planar EM)
HFSS Setup 1
Sweep 1
Design Verification
Optimetrics
Results
Field Overlays
Radiation
Notes
Definitions
Properties
Name
Setup
Enable
Solver
Passes
Percent Refinement
Delta S
Solution Freq
Basis Order
Max Refinement
Use Max Refinement
Solver Type
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Rename F2
Delete Delete
Properties...
Disable Setup
Add Frequency Sweep...
Assign Mesh Operation
Validate
Analyze F10
Submit Job...
Revert to Initial Mesh
Generate Mesh
Pre-Process Geometry
Invalidate Solutions
Export
Profile...
Convergence...
Matrix Data...
Mesh Statistics...
View Mesh Feedback...
Network Data Explorer

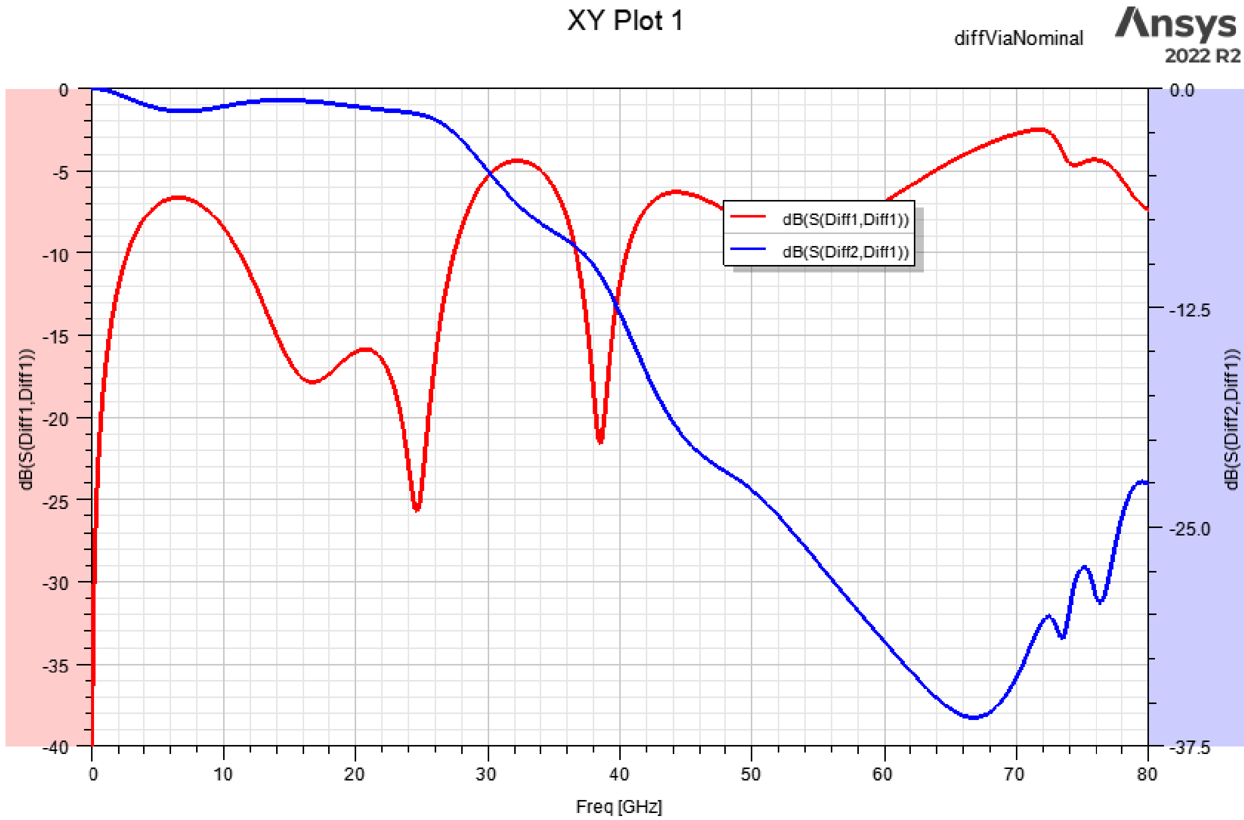
2. Wait for the solution to complete.



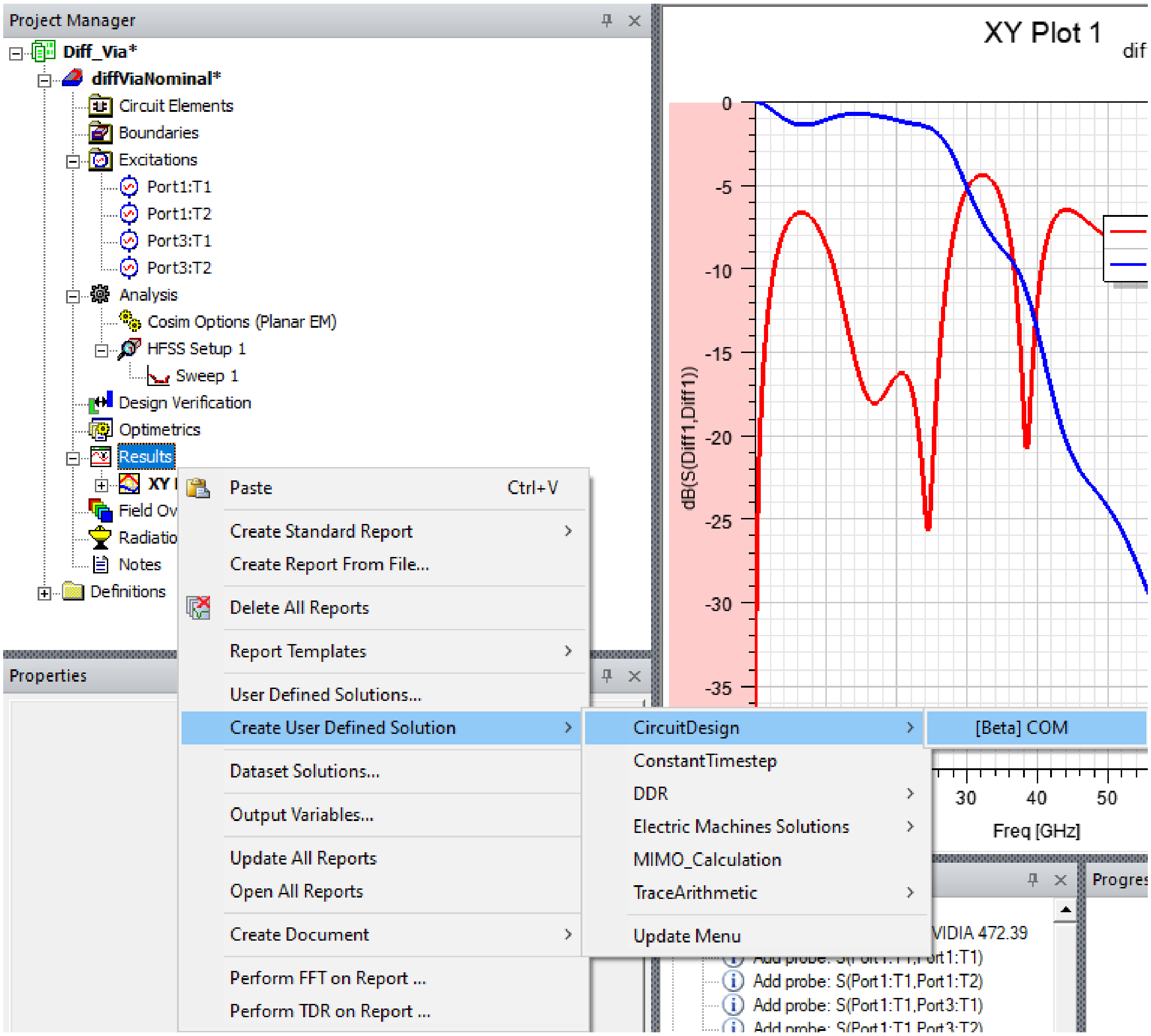
3. From the **Project Manager** window, expand **Results**.



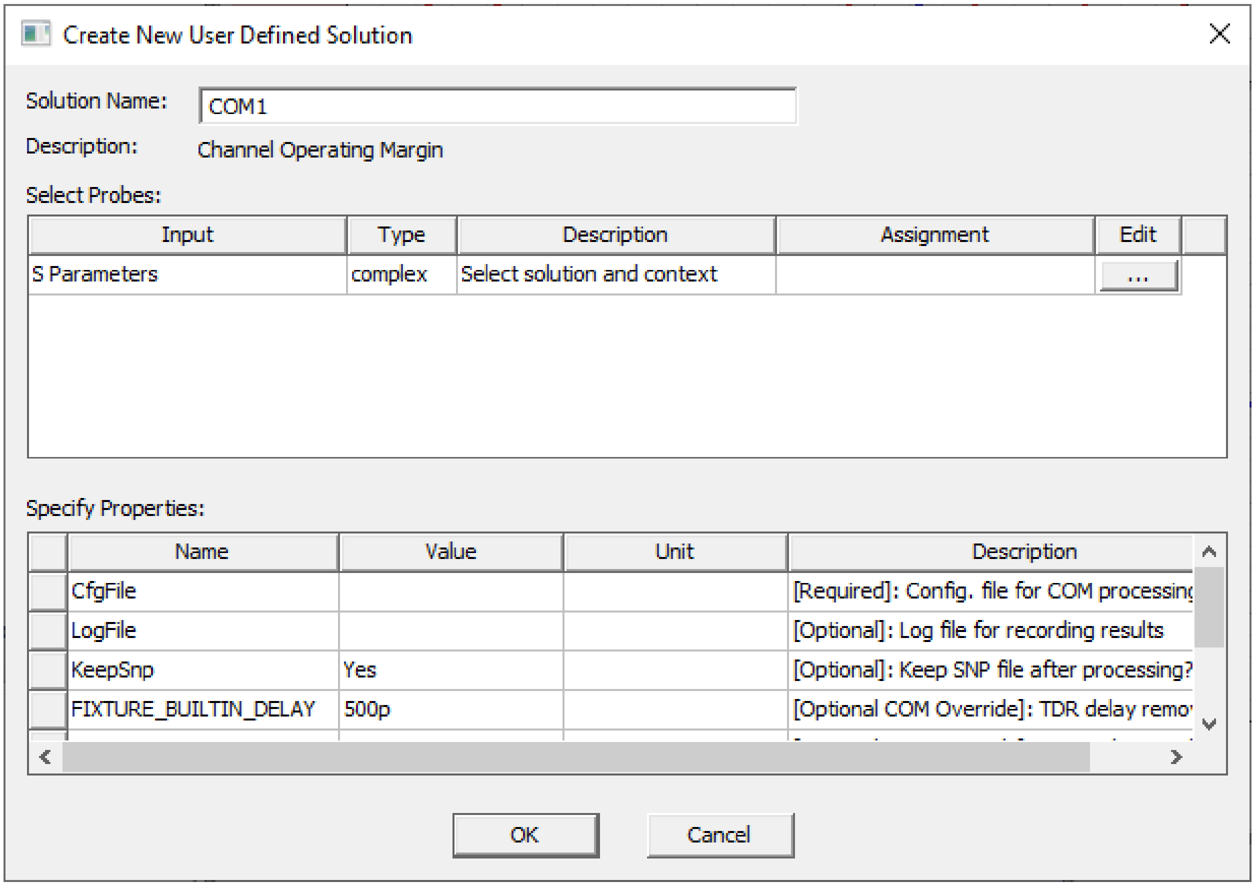
4. Double-click **XY Plot 1** to view the differential insertion and return loss plot.



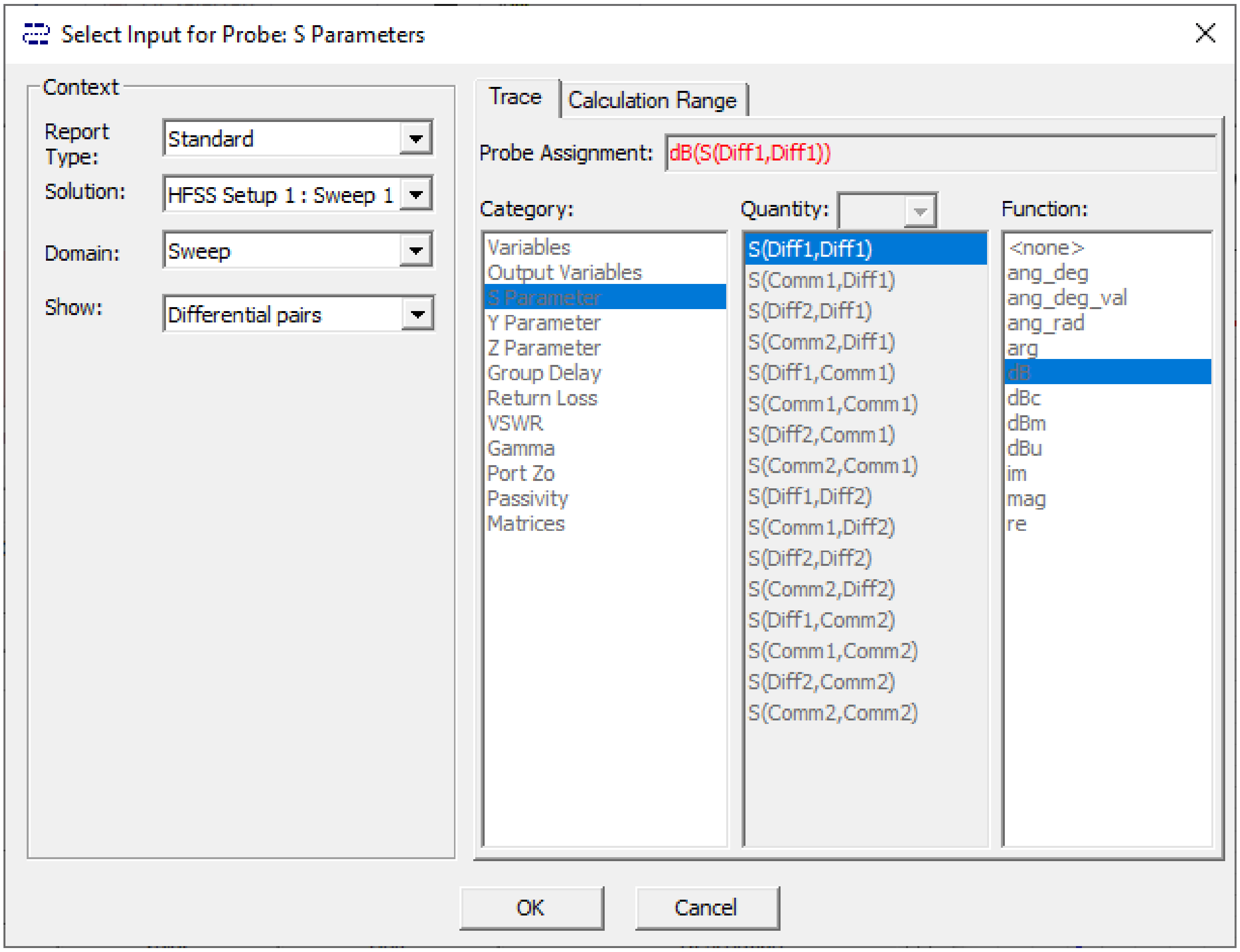
5. From the **Project Manager** window, right click **Results** and select **Create User-Defined Solution** > **CircuitDesign** > **[Beta] COM** to open the **Create New User-Defined Solution** window.



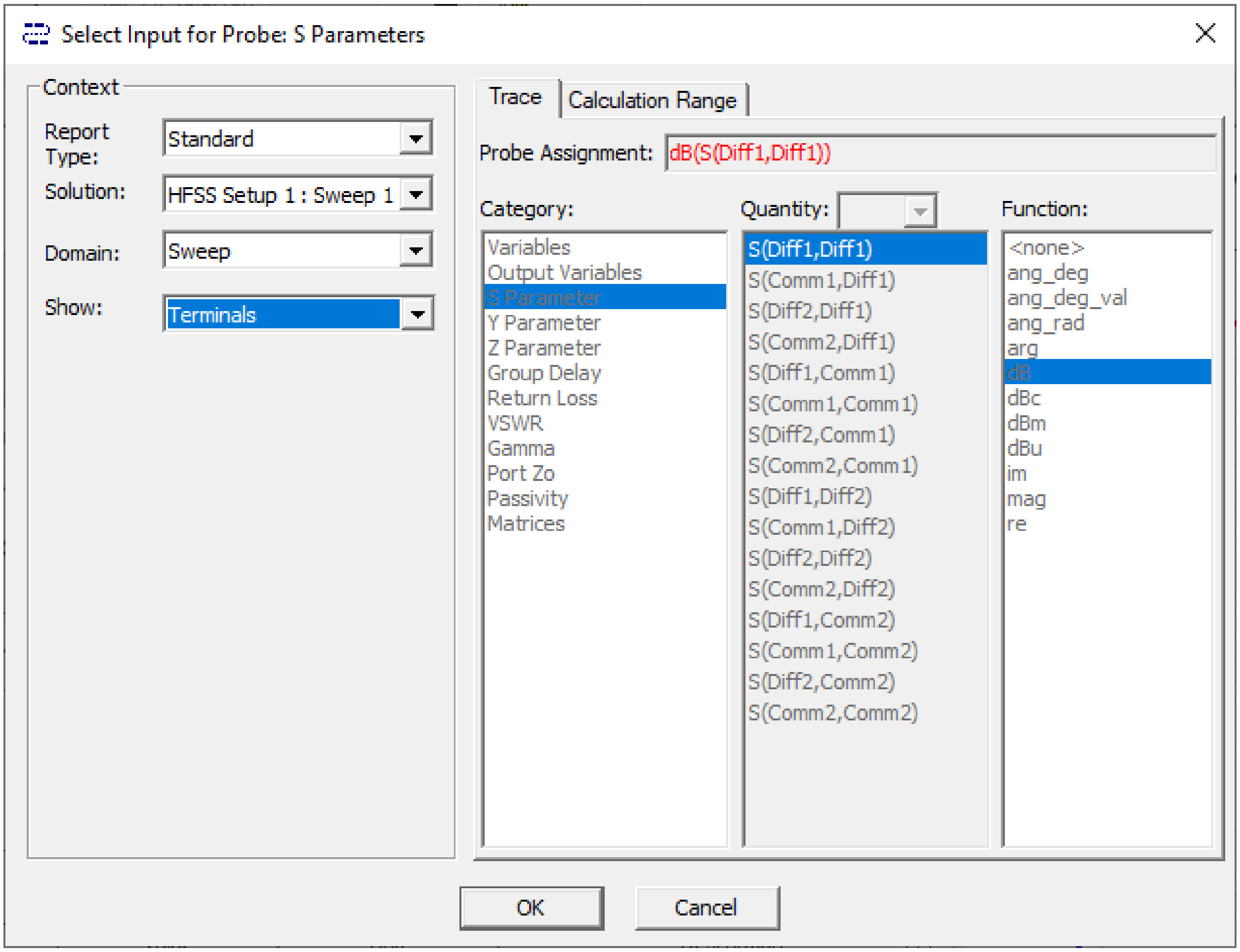
Project Manager
Diff_Via*
diffViaNominal*
Circuit Elements
Boundaries
Excitations
Port1:T1
Port1:T2
Port3:T1
Port3:T2
Analysis
Cosim Options (Planar EM)
HFSS Setup 1
Sweep 1
Design Verification
Optimetrics
Results
Notes
Definitions
Properties
Paste Ctrl+V
Create Standard Report
Create Report From File...
Delete All Reports
Report Templates
User Defined Solutions...
Create User Defined Solution
Dataset Solutions...
Output Variables...
Update All Reports
Open All Reports
Create Document
Perform FFT on Report ...
Perform TDR on Report ...
CircuitDesign
ConstantTimestep
DDR
Electric Machines Solutions
MIMO_Calculation
TraceArithmetic
Update Menu
[Beta] COM
XY Plot 1
dB(S(Diff1,Diff1))
Freq [GHz]
Add probe: S(Port1:T1,Port1:T2)
Add probe: S(Port1:T1,Port3:T1)



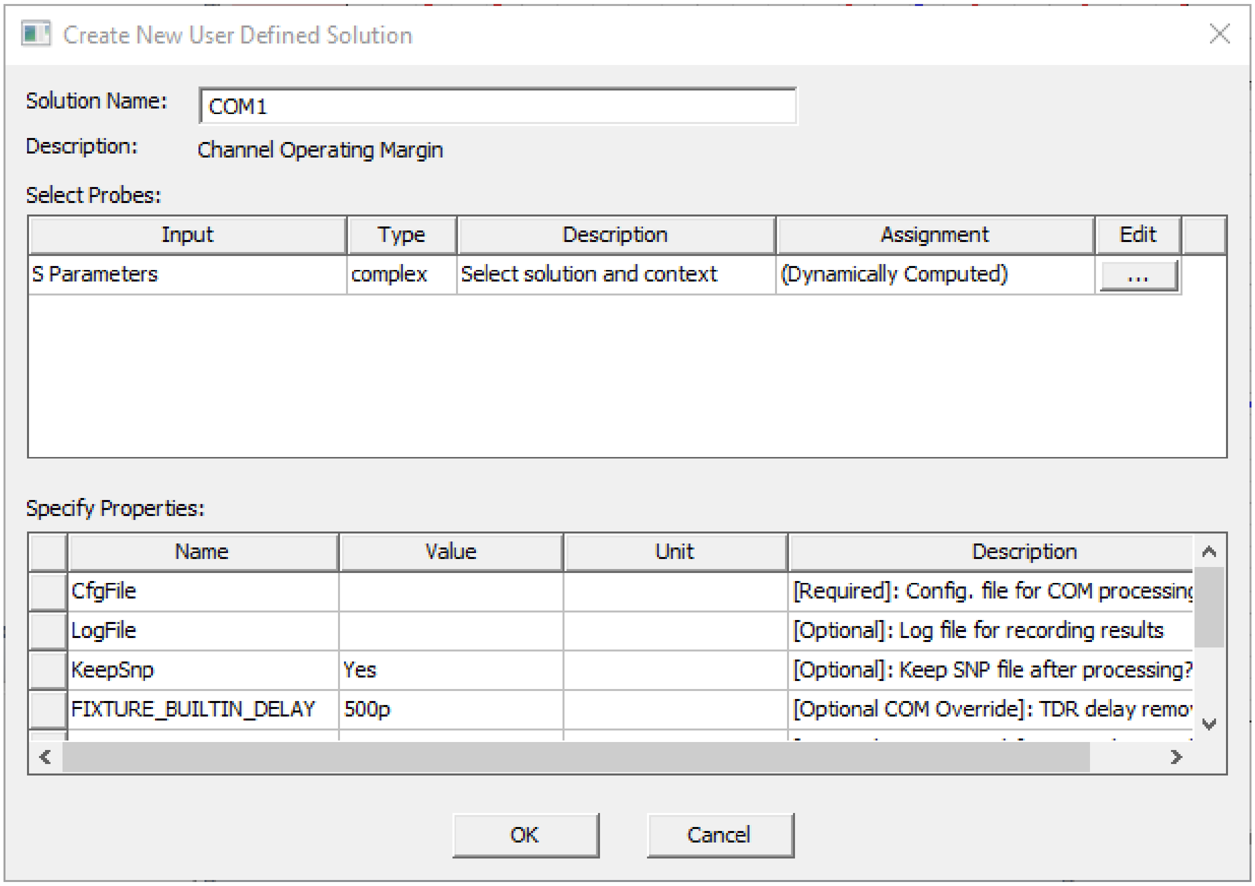
6. From the **Select Probes** table > **S-Parameters** row, click **...** to open the **Select Input for Probe** window.


Select Input for Probe: S Parameters
Context
Report Type:
Standard
Solution:
HFSS Setup 1 : Sweep 1
Domain:
Sweep
Show:
Differential pairs
Trace
Calculation Range
Probe Assignment:
dB(S(Diff1,Diff1))
Category:
Variables
Output Variables
S Parameter
Y Parameter
Z Parameter
Group Delay
Return Loss
VSWR
Gamma
Port Zo
Passivity
Matrices
Quantity:
S(Diff1,Diff1)
S(Comm1,Diff1)
S(Diff2,Diff1)
S(Comm2,Diff1)
S(Diff1,Comm1)
S(Comm1,Comm1)
S(Diff2,Comm1)
S(Comm2,Comm1)
S(Diff1,Diff2)
S(Comm1,Diff2)
S(Diff2,Diff2)
S(Comm2,Diff2)
S(Diff1,Comm2)
S(Comm1,Comm2)
S(Diff2,Comm2)
S(Comm2,Comm2)
Function:
<none>
ang_deg
ang_deg_val
ang_rad
arg
dB
dBc
dBm
dBu
im
mag
re
OK
Cancel

7. In the **Context** group box, select **Terminals** on the **Show** drop-down menu.



8. Click **OK** to close the **Select Input for Probe** window and return to the **Create New User-Defined Solution** window.

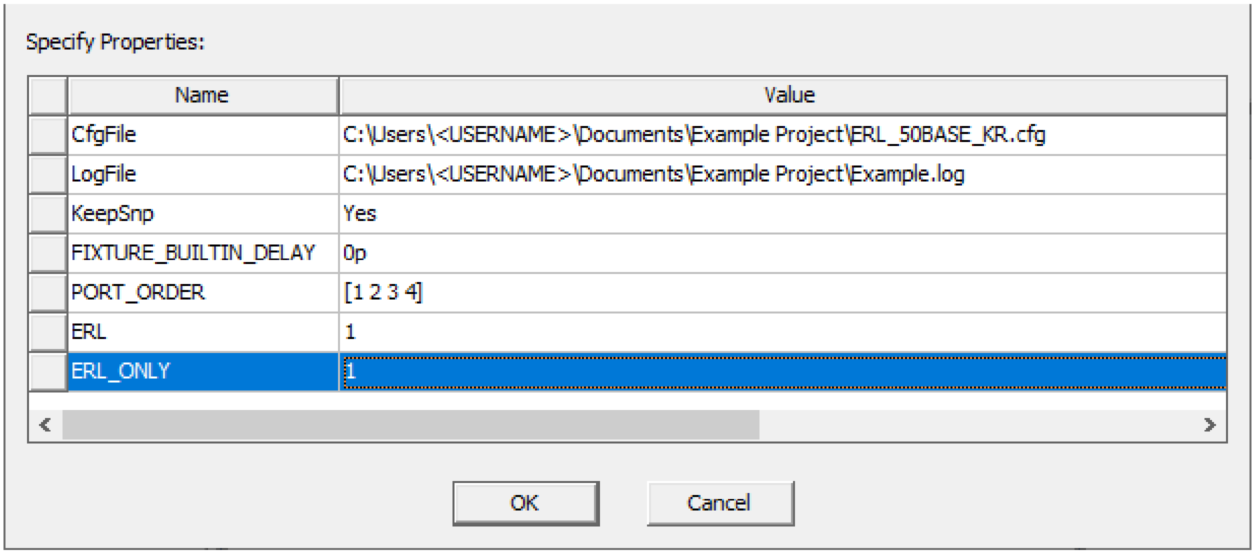


9. In the **Specify Properties** table, make the following changes.

- Enter the path to the configuration file exported in **"Create a Configuration File in SPISim" on page 15-205** (e.g., **C:\Users\<USERNAME>\Documents\Example Project\ERL_50BASE_KR.cfg**) in the **CfgFile Value** field.

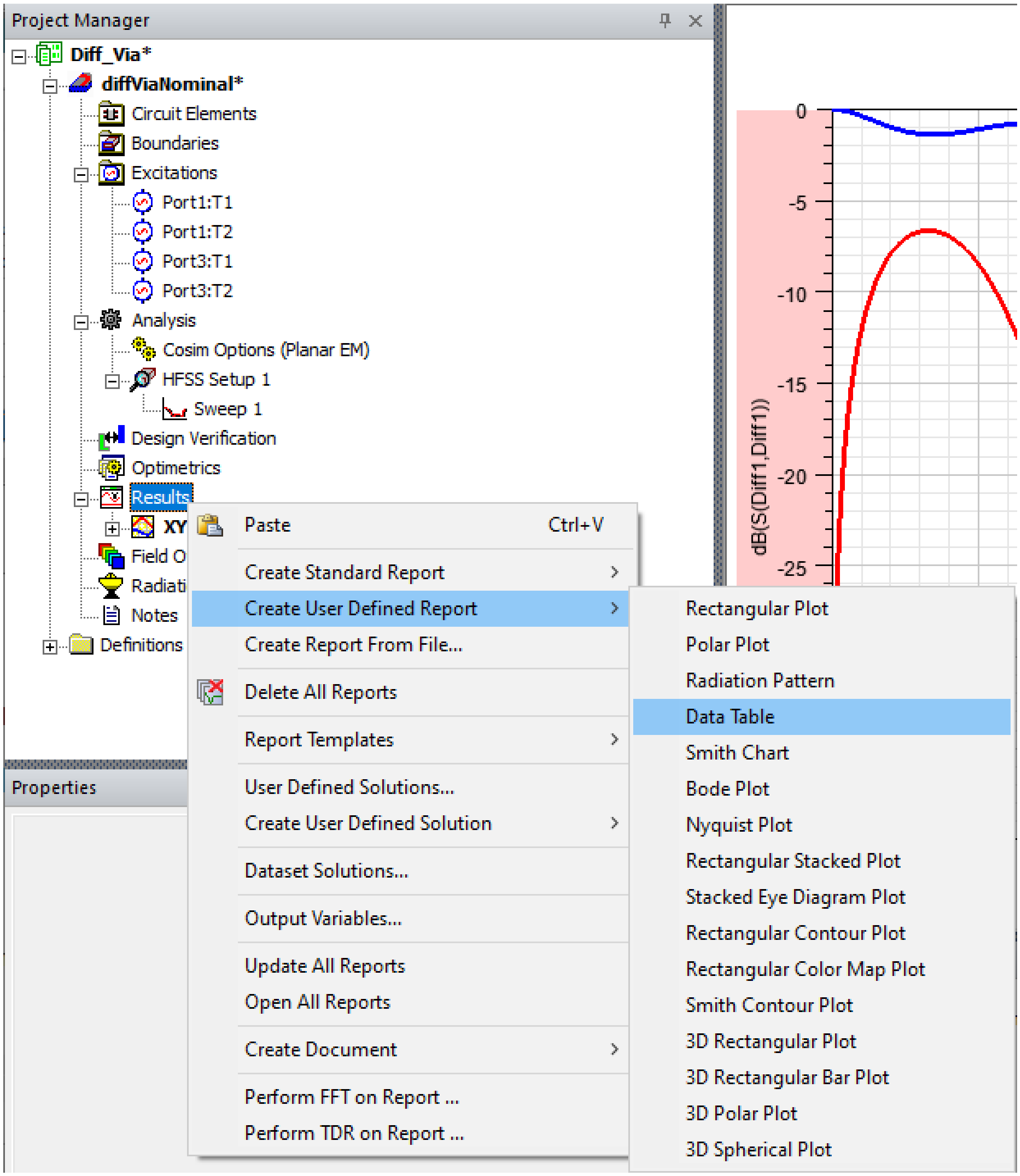
> **Note:** The path can include spaces but can **not** include commas or other punctuation.

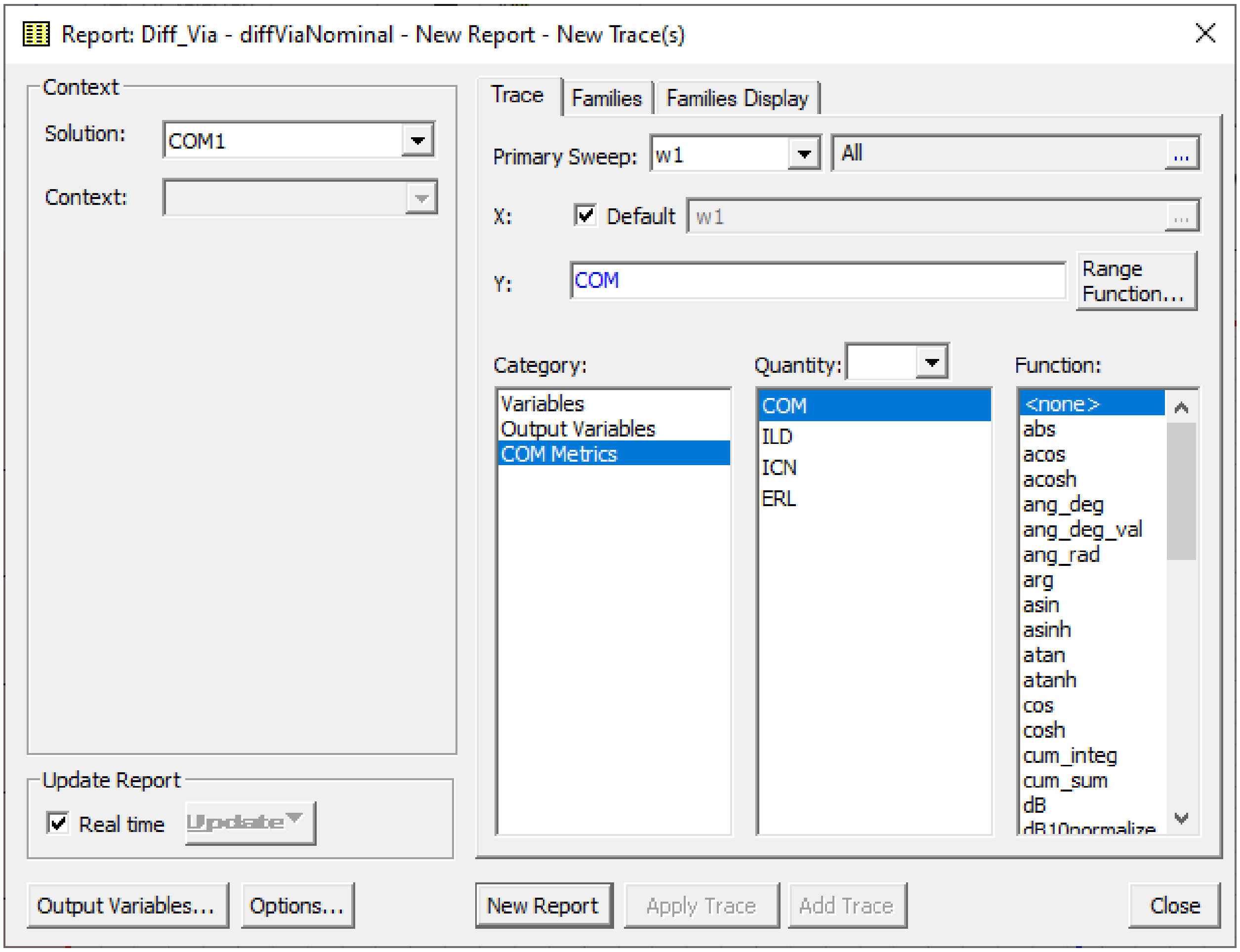
- If appropriate, enter the path to a log file in the **LogFile Value** field (e.g., **C:\Users\<USERNAME>\Documents\Example Project\Example.log**).
- Enter **0p** in the **FIXTURE_BUILTIN_DELAY Value** field.
- Select **[1 2 3 4]** on the drop-down menu in the **PORT_ORDER Value** field.
- Select **1** on the drop-down menu in the **ERL Value** field.
- Select **1** on the drop-down menu in the **ERL_ONLY Value** field.



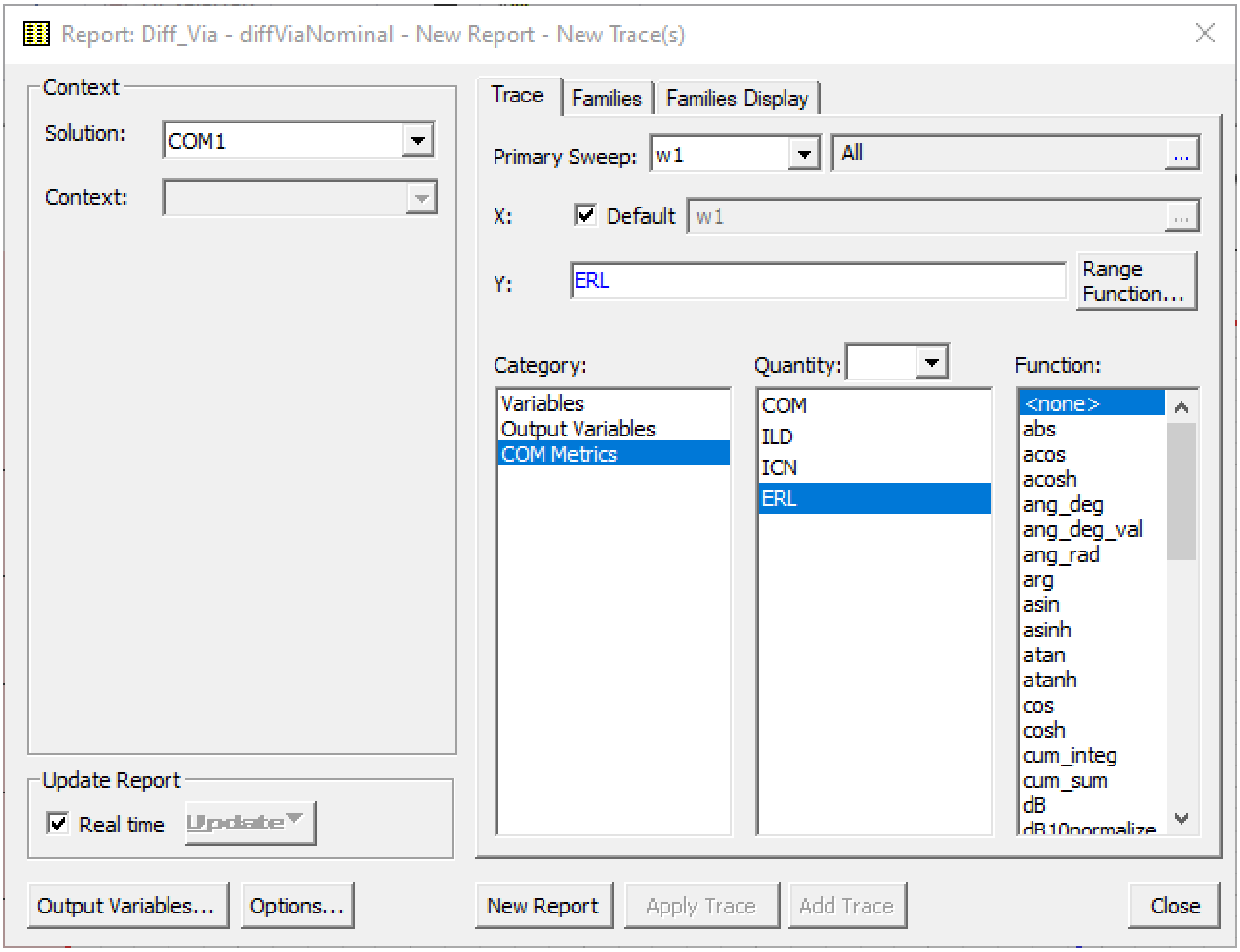
10. Click **OK** to close the **Create New User-Defined Solution** window.

11. From the **Project Manager** window, right-click **Results** and select **Create User-Defined Report** > **Data Table** to open the **Report** window.


Project Manager
Diff_Via*
diffViaNominal*
Circuit Elements
Boundaries
Excitations
Port1:T1
Port1:T2
Port3:T1
Port3:T2
Analysis
Cosim Options (Planar EM)
HFSS Setup 1
Sweep 1
Design Verification
Optimetrics
Results
XY
Field O
Radiati
Notes
Definitions
Properties
Paste
Ctrl+V
Create Standard Report
Create User Defined Report
Create Report From File...
Delete All Reports
Report Templates
User Defined Solutions...
Create User Defined Solution
Dataset Solutions...
Output Variables...
Update All Reports
Open All Reports
Create Document
Perform FFT on Report ...
Perform TDR on Report ...
Rectangular Plot
Polar Plot
Radiation Pattern
Data Table
Smith Chart
Bode Plot
Nyquist Plot
Rectangular Stacked Plot
Stacked Eye Diagram Plot
Rectangular Contour Plot
Rectangular Color Map Plot
Smith Contour Plot
3D Rectangular Plot
3D Rectangular Bar Plot
3D Polar Plot
3D Spherical Plot
dB(S(Diff1,Diff1))
0
-5
-10
-15
-20
-25

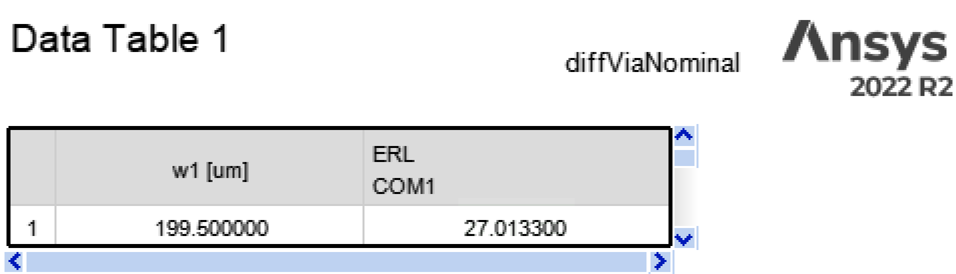

Report: Diff_Via - diffViaNominal - New Report - New Trace(s)
Context
Solution:
COM1
Context:
Update Report
Real time
Update
Trace
Families
Families Display
Primary Sweep:
w1
All
X:
Default
w1
Y:
COM
Range Function...
Category:
Variables
Output Variables
COM Metrics
Quantity:
COM
ILD
ICN
ERL
Function:
<none>
abs
acos
acosh
ang_deg
ang_deg_val
ang_rad
arg
asin
asinh
atan
atanh
cos
cosh
cum_integ
cum_sum
dB
Output Variables...
Options...
New Report
Apply Trace
Add Trace
Close

12. From the **Report** window, select **ERL** on the **Quantity** list.



13. Click **New Report** to begin the ERL calculation. The **Close** button will be temporarily unavailable until ERL calculation is complete.

14. Click **Close** to close the **Report** window and view the **Data Table 1** report. The report is accessible from **Electronics Desktop View** tab and from expanding the **Project Manager** window > **Data Table 1**.



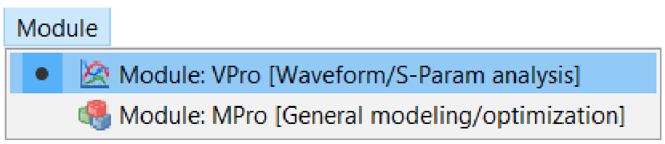
The procedure is complete.

Calculating Integrated Crosstalk Noise

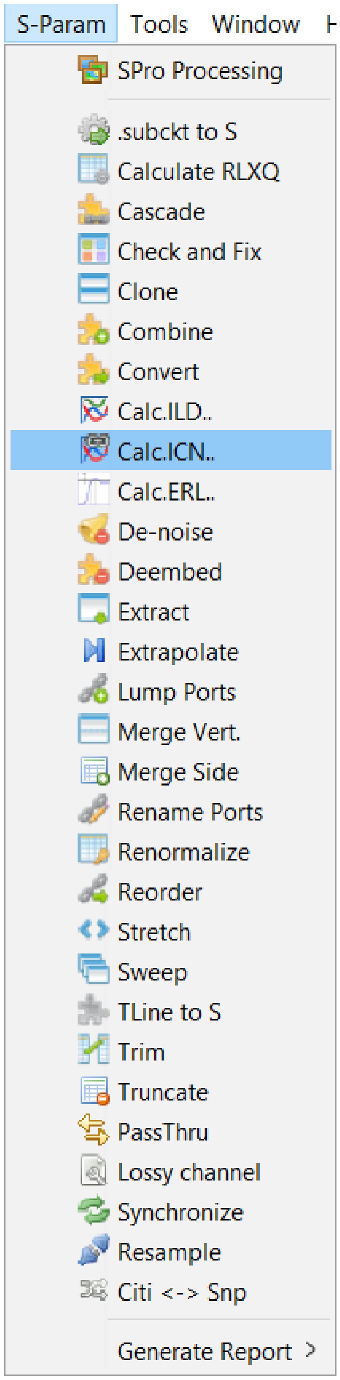
> **Note: NEW in 23.2 Release:** calculation method **PCIe5 ccICN** added. This option enables calculating ccICN for given s-param files according to PCIe5 spec, (see step 5).

To launch the Calc. ICN window:

1. Select **Module: VPro [Waveform/ S-Param analysis]** from the **Module** menu.



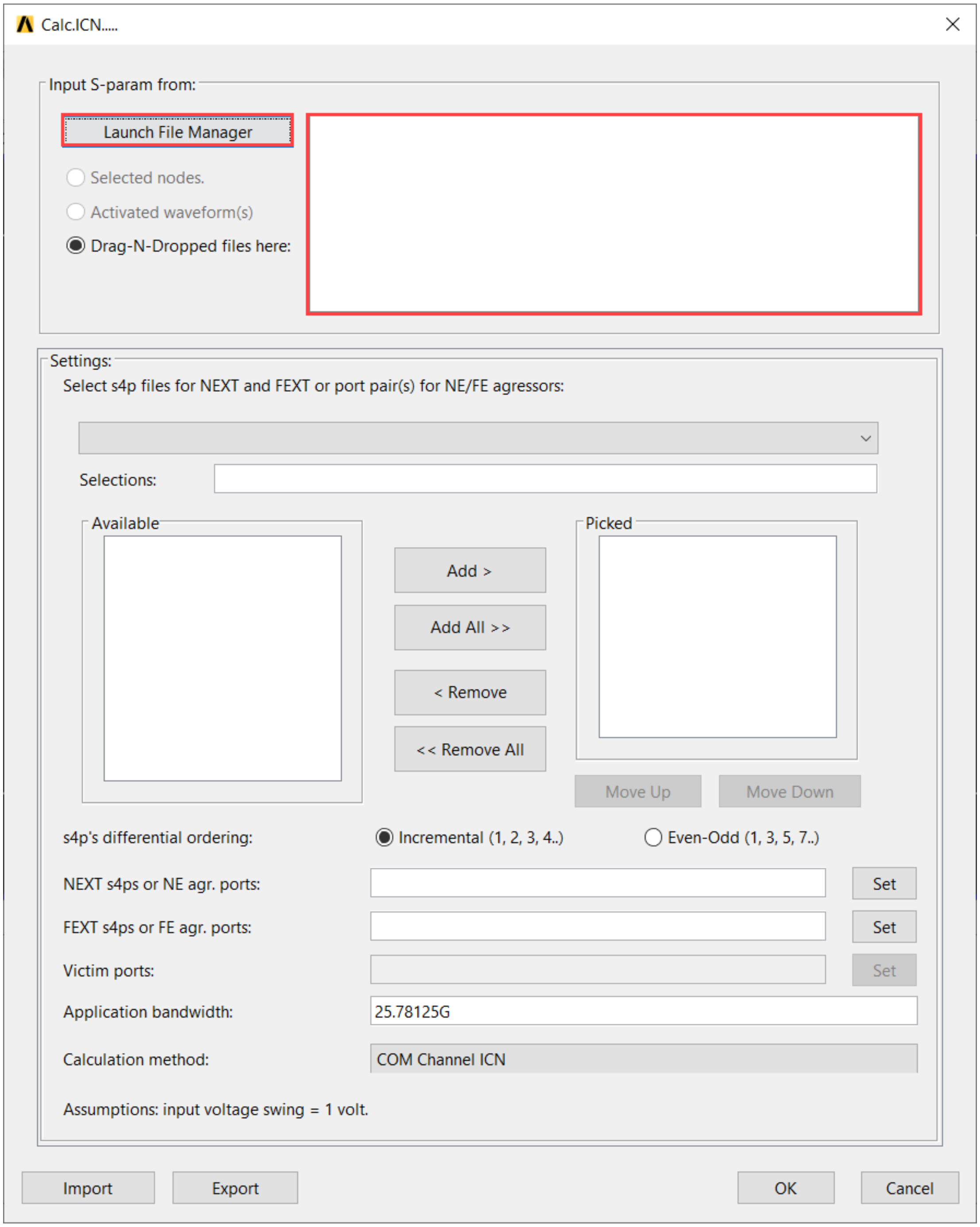
2. From the S. Param menu, select Calc. ICN.



To use the Calc. ICN feature:

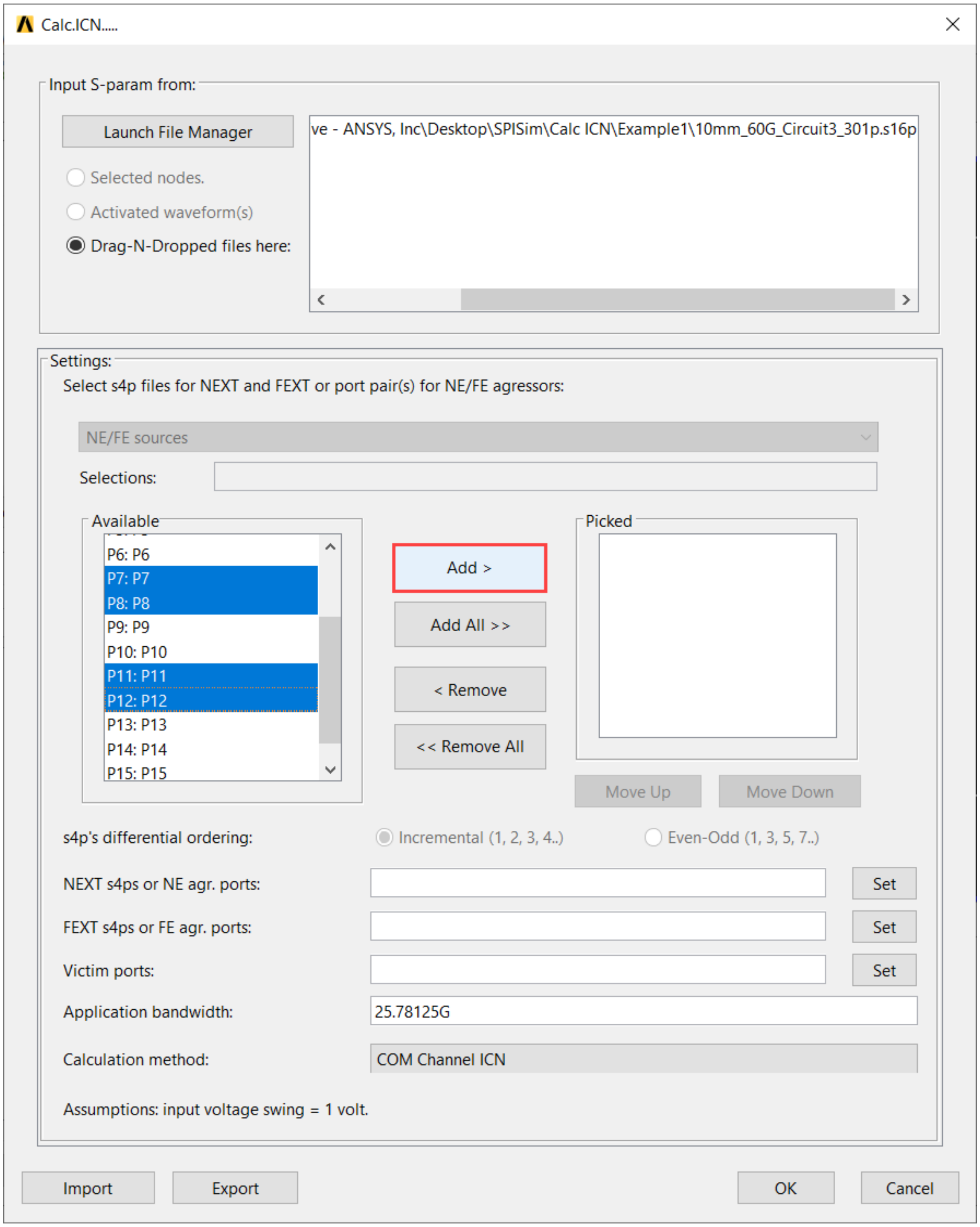
1. Choose the S. parameter files by selecting **Launch File Manager** or by dragging and dropping the files into the box.

   The **Available** group box will be populated with the selected files.

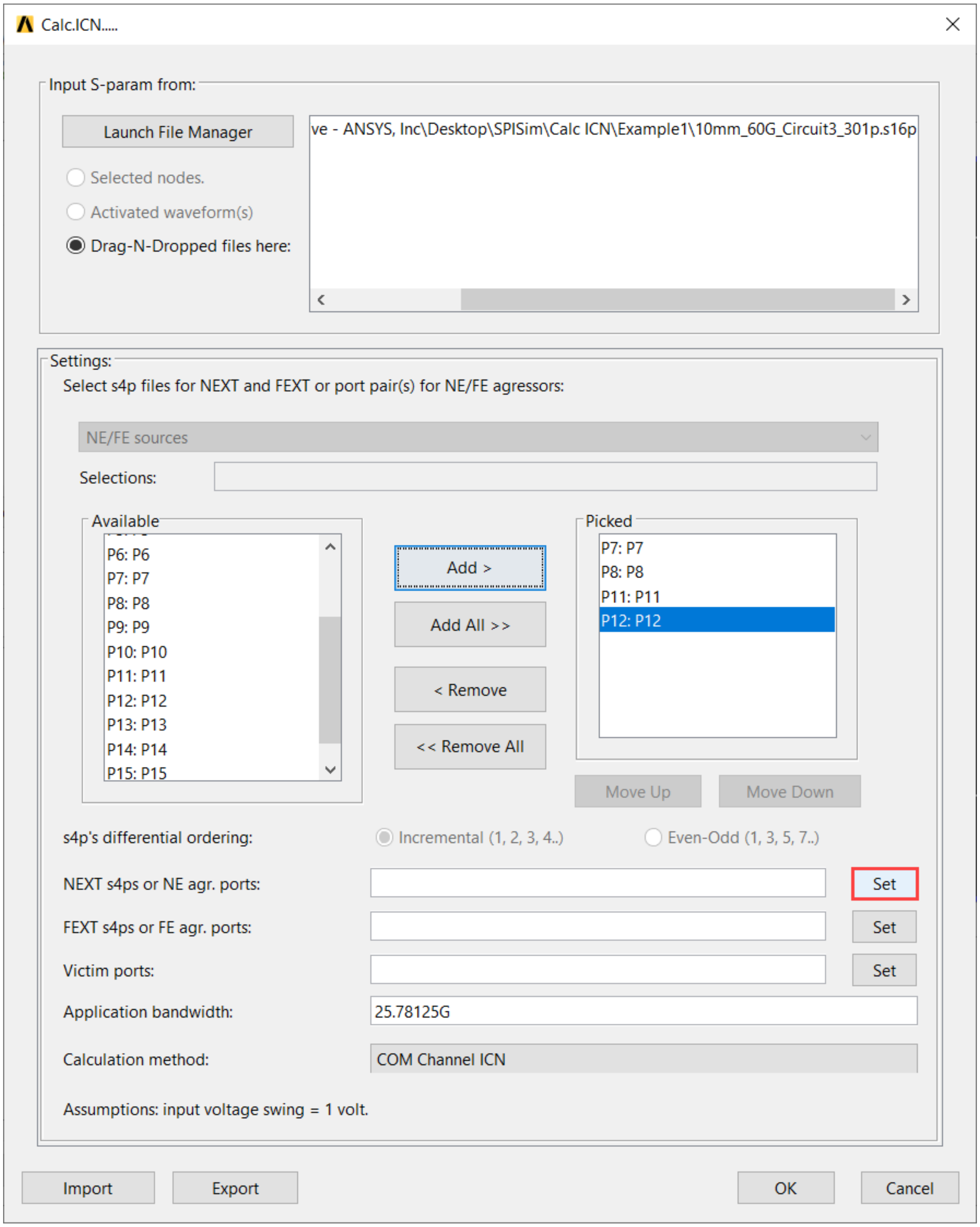


2. Select **NEXT s4ps or NE agr. ports**:

i. Select from the **Available** group box and click **Add >**

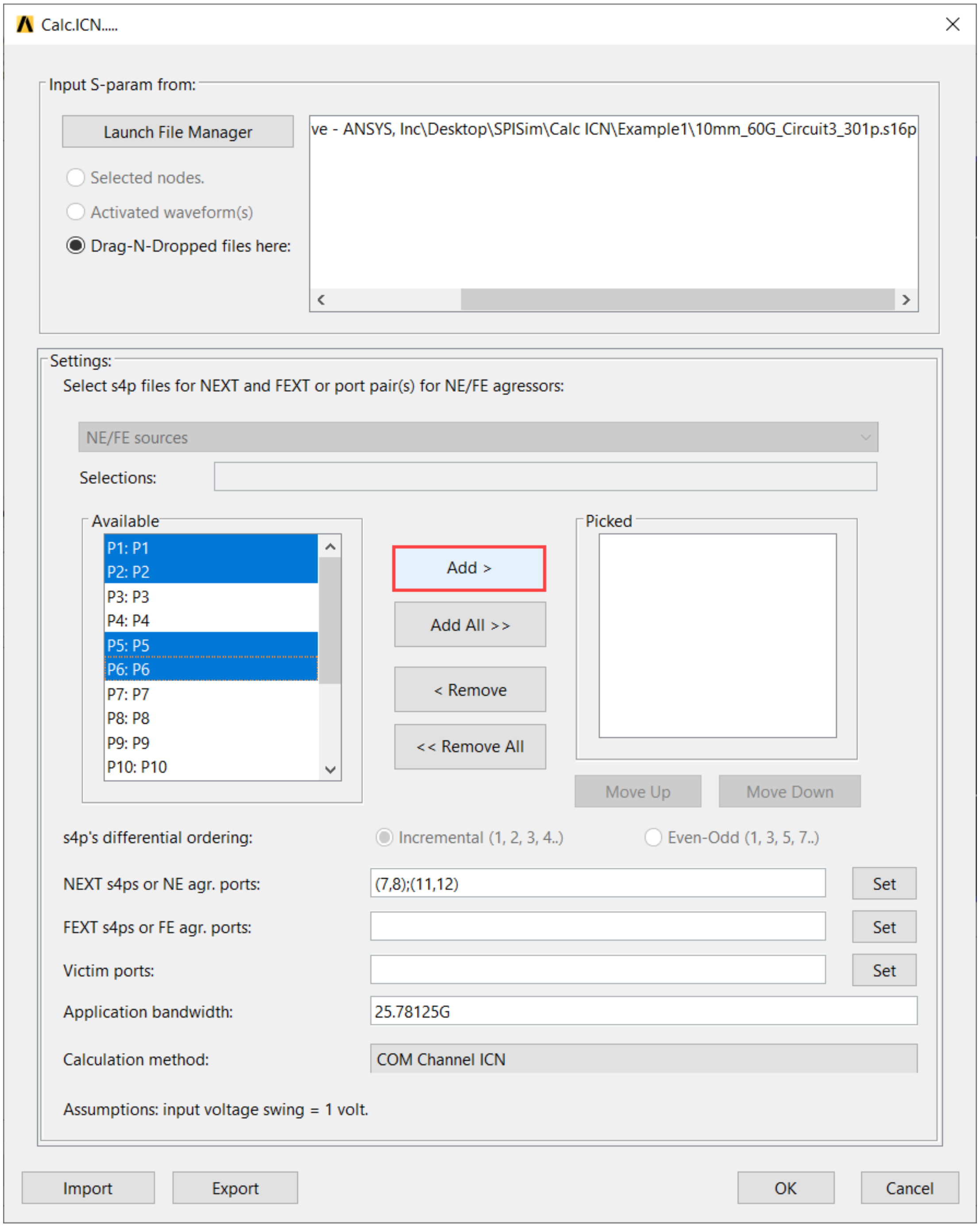


ii. The selections will be populated in **Picked**. Click **Set.**

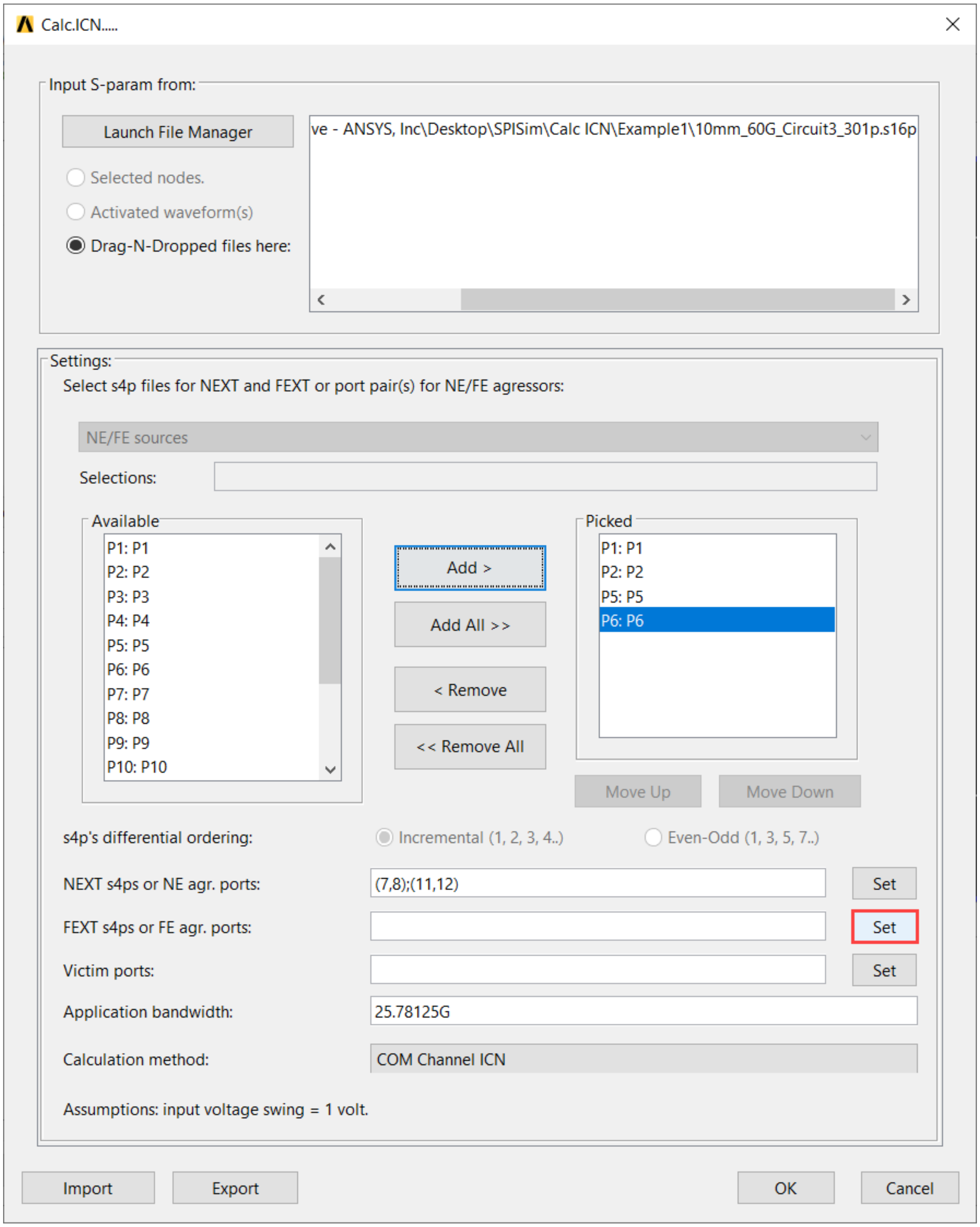


3. Select **FEXT s4ps or FE agr. ports**:

i. Select from the **Available** group box and click **Add >**

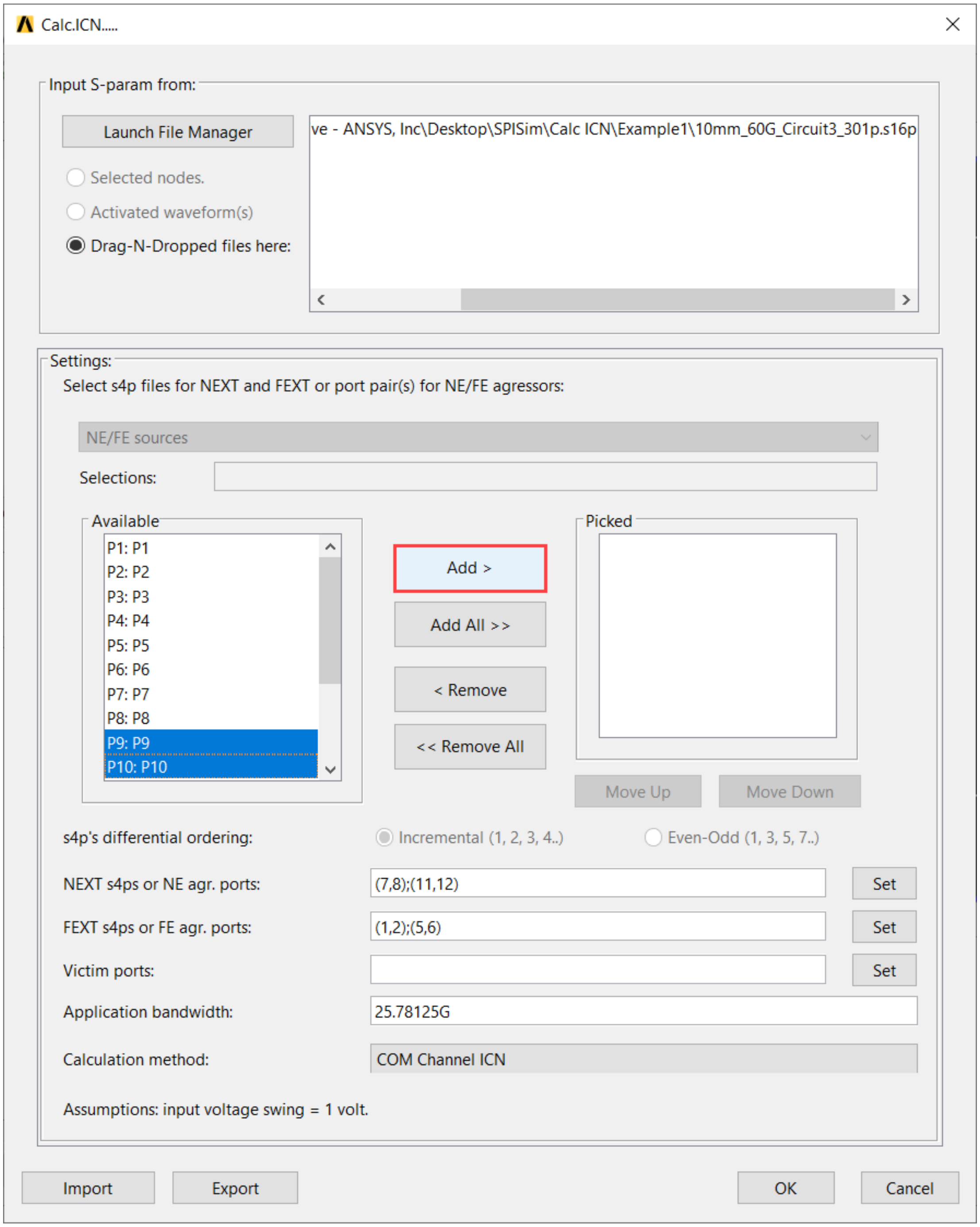


ii. The selections will be populated in **Picked**. Click **Set**

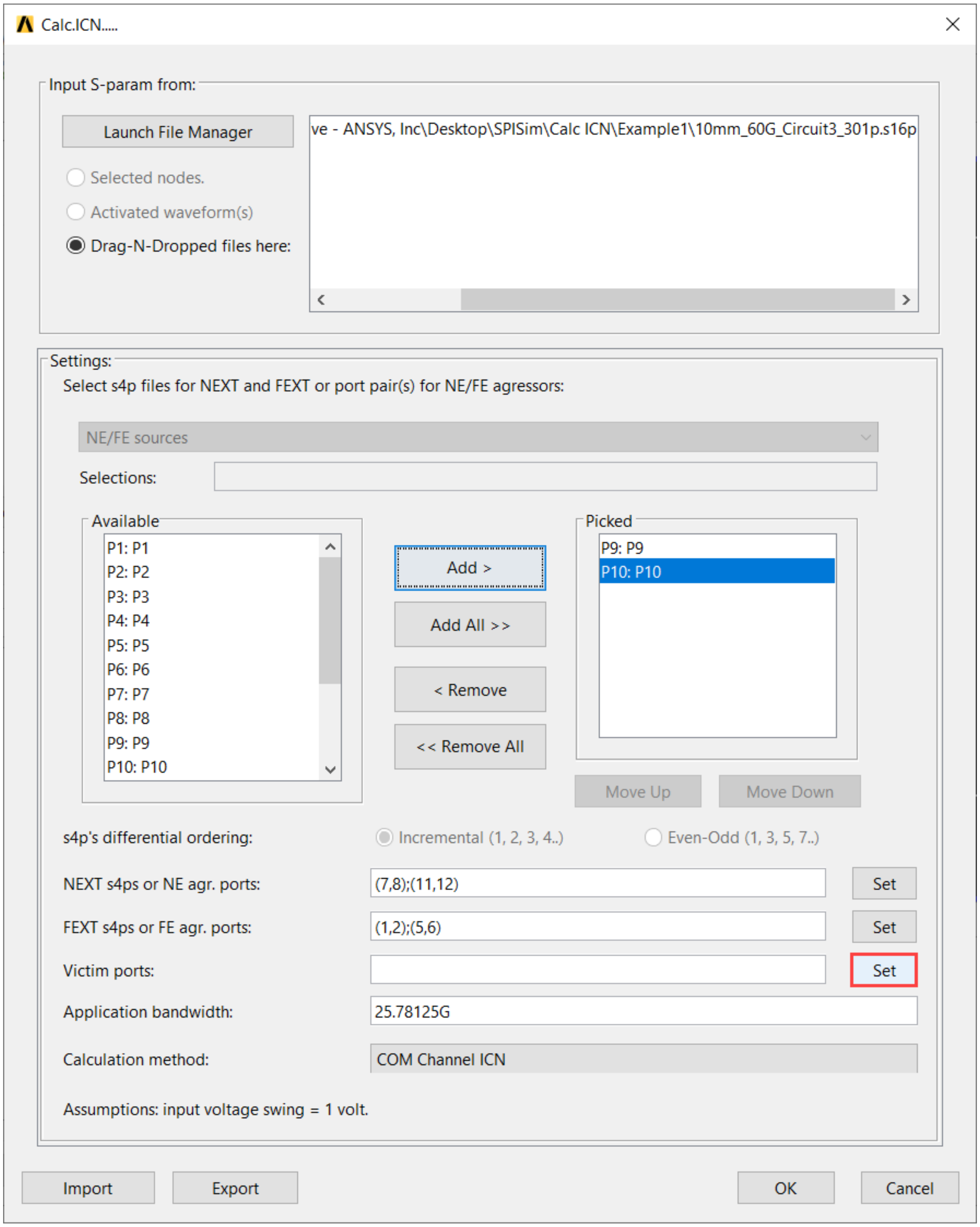


4. Select **Victim ports**:

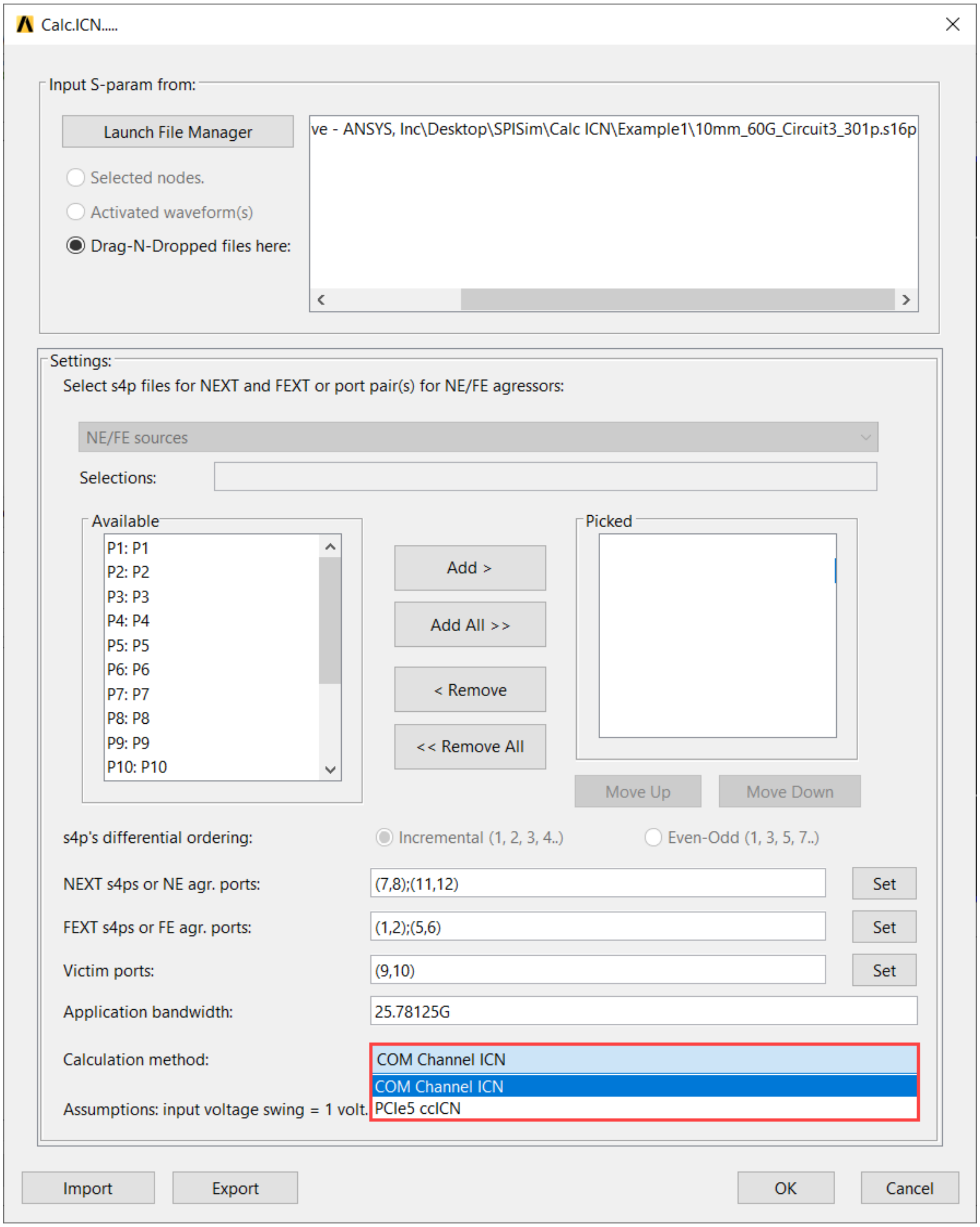
i. Select the ports from the **Available** group box and click **Add >**



ii. The selections will be populated in **Picked**. Click **Set**

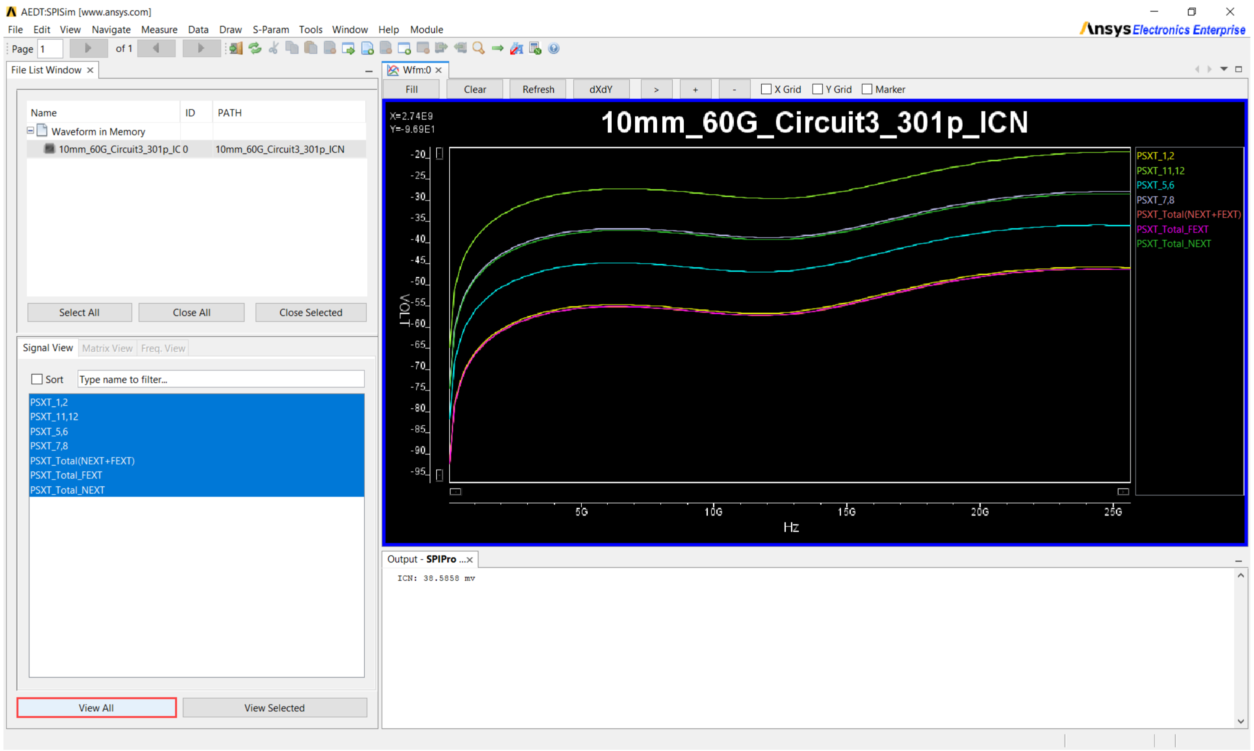


5. Choose a calculation method from **COM Channel ICN** or **PCIe5 ccICN**



6. Select **OK**to close the window.

Select **View All** in the SPISim **VPro Signal View** group box.



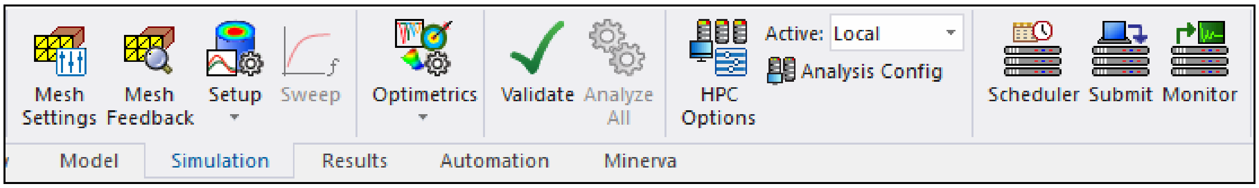
> **Note:** To specify PCIe5 CCICN in batch command, add the following setting as part of the config settings: ICNCALC=PCIE_CCICN
>
> The default is COM_CHNICN

16 - FilterSolutions

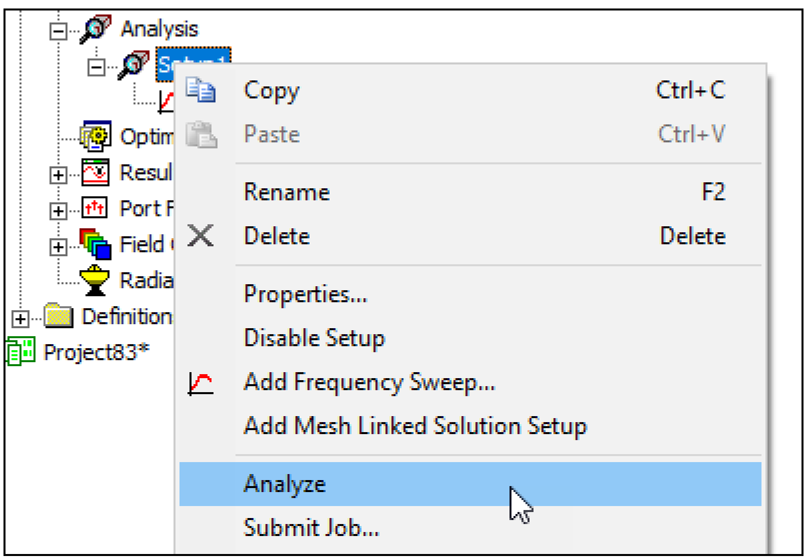
Ansys FilterSolutions allows you to design frequency filters using specifications. This stand-alone application simplifies filter design. With it, custom-design a filter and import the resulting components into Circuit. FilterSolutions is available through **Electronics Desktop**. It is launched from the **Electronics Desktop**. In the menu bar, click **Tools** > **FilterSolutions**. Refer to the Welcome to FilterSolutions Help for more details.

17 - Running Simulations

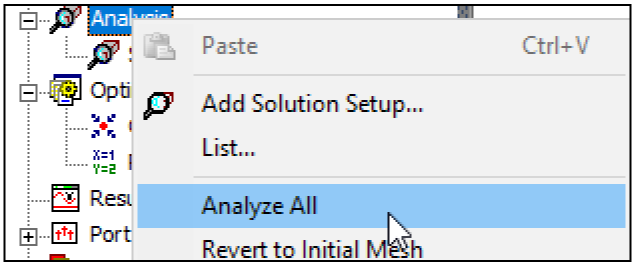
After you specify how Ansys Electronics Desktop is to compute the solution, you need to begin the solution process. You can access the **Analyze** commands for a specific setup by right-clicking on a solution or sweep in the **Project** tree and selecting from the shortcut menu, you by selecting the **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 **Circuit**> **Analyze All** or right-click the **Analysis** icon in the *Project Manager* and select **Analyze All**.



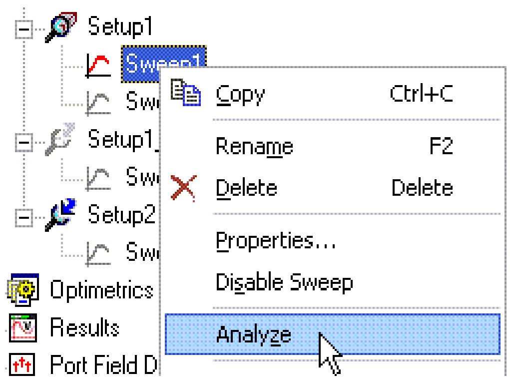
What do you want to do?

- 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

Solving a Single Setup or Sweep

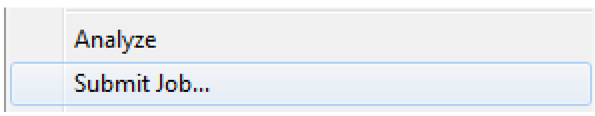
To run a single setup or sweep:

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. For a select sweep, it is computed for the sweep variables.

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.

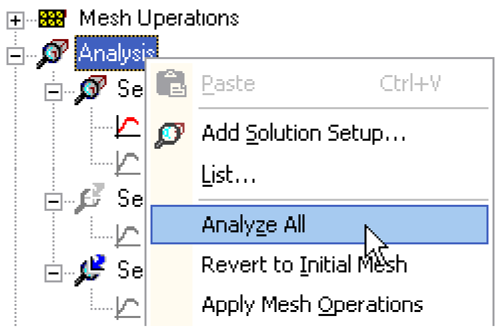
Running More than One Simulation

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 **Circuit**> **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.

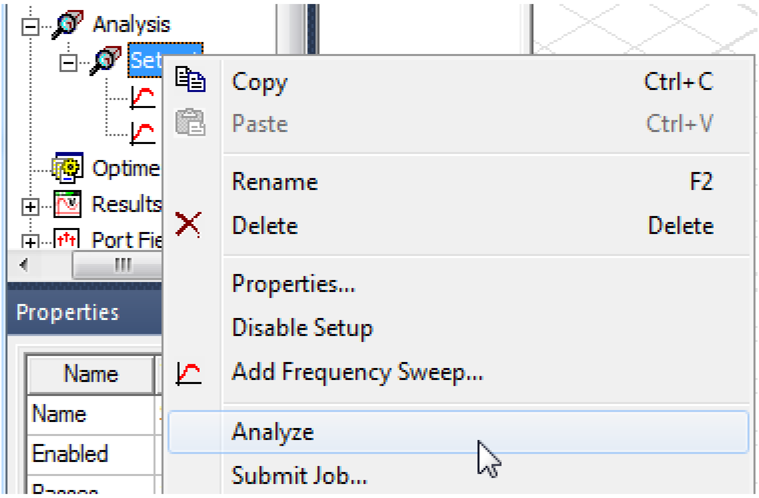


> **Note:**
>
> The **General** tab for the Setup includes an **Enabled** check box. By default, this is checked. Clearing the **Enabled** check box excludes a setup from running

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.

To view the *Solutions* window:

1. Right-click the solution **Setup** in the Project Manager.

   A shortcut menu appears.

2. Select **Convergence**, **Matrix Data**, **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:

- Right-click **Analysis** in the Project Manager and choose **Browse Solutions** from the shortcut menu.

  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."

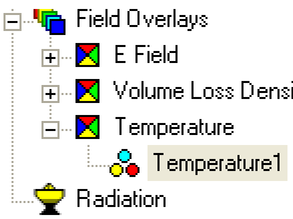
To generate a new solution after modifying a design, follow the procedure for running a simulation.

Also see Re-Solving with Ansys Workbench Thermal Feedback.

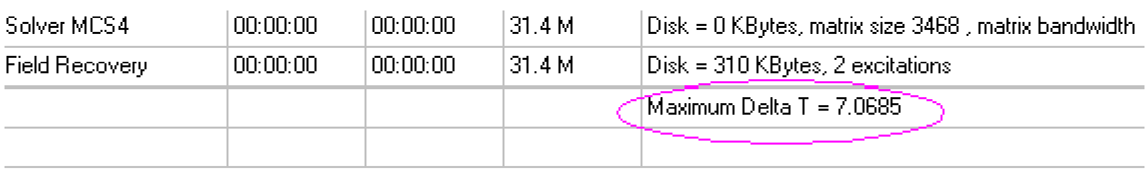
Re-solving after Ansys Workbench Thermal Feedback

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.

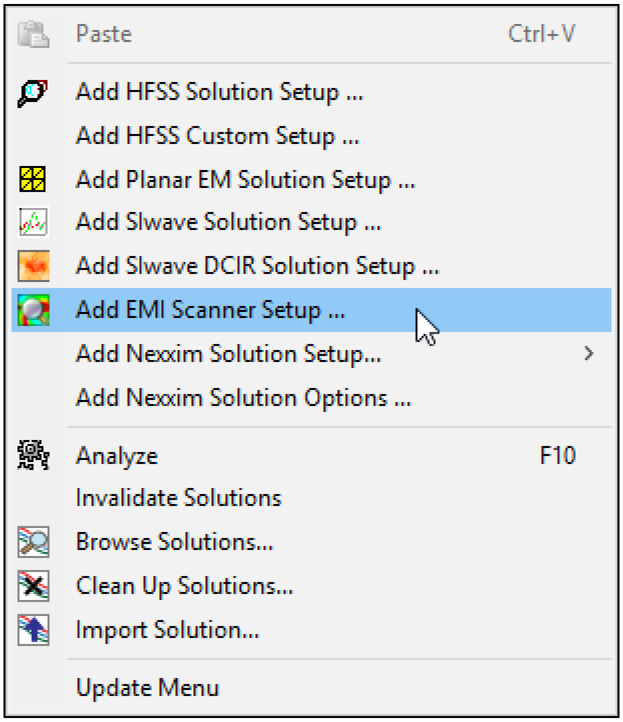
Running EMI Scanner in HFSS 3D Layout

Ansys EMI Scanner identifies design issues that might result in electromagnetic interference problems during operation. Such problems can stem from a compromised power delivery network or from improperly shielded signal nets. This feature, which is based on geometric rule checks, scans complete layouts quickly and can be used to identify subsections of the design requiring more rigorous analysis using the SIwave or HFSS finite element-based solvers.

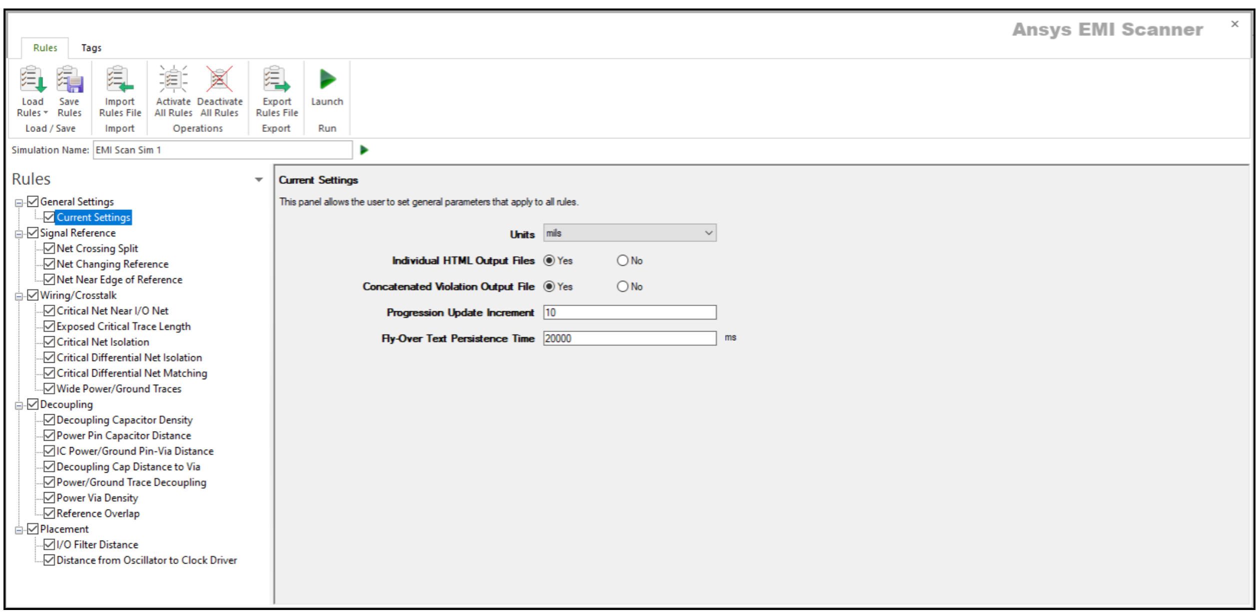
To set up EMI scanner:

1. From the **Simulation** tab, click **EMI**, or

   From the **Project Manager** window, right-click **Analysis** and select **Add EMI Scanner Setup**.



2. From the **Project Manager** window, double-click **EMI Setup**. to open the **Ansys EMI Scanner** window, on the **Rules** tab.

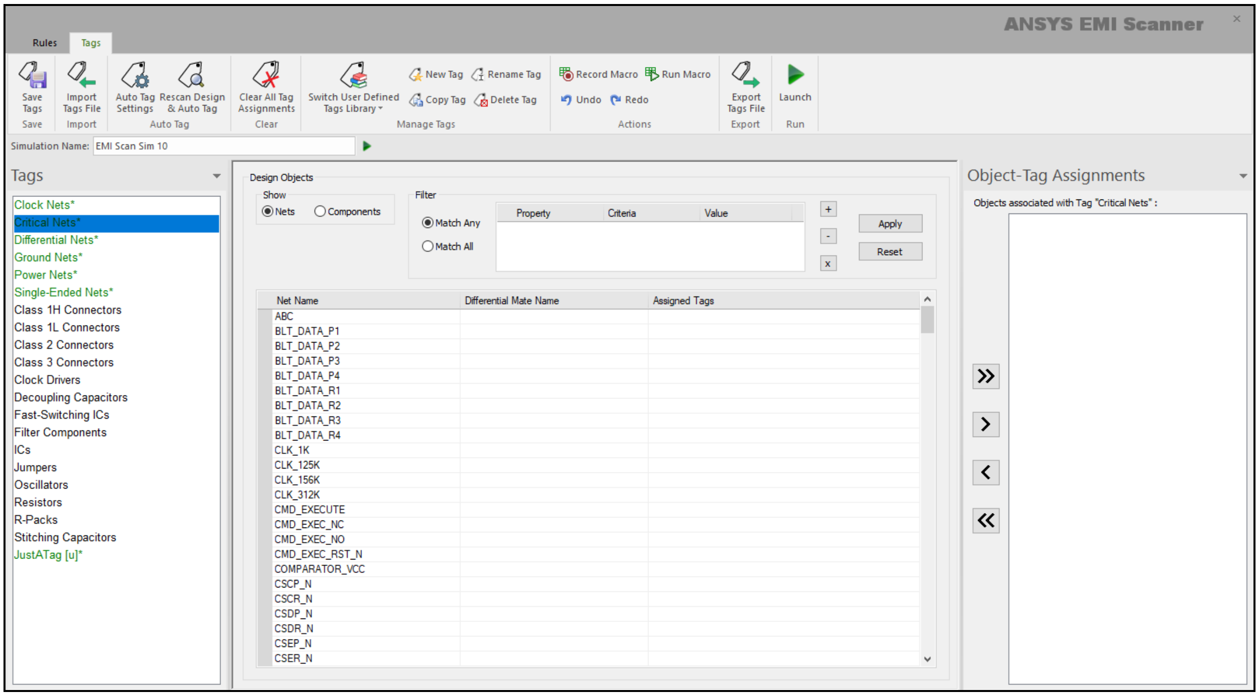


If chosen, click the **Simulation Name** field to rename the scan.

3. Set rules, as chosen.

4. Select the **Tags** tab.

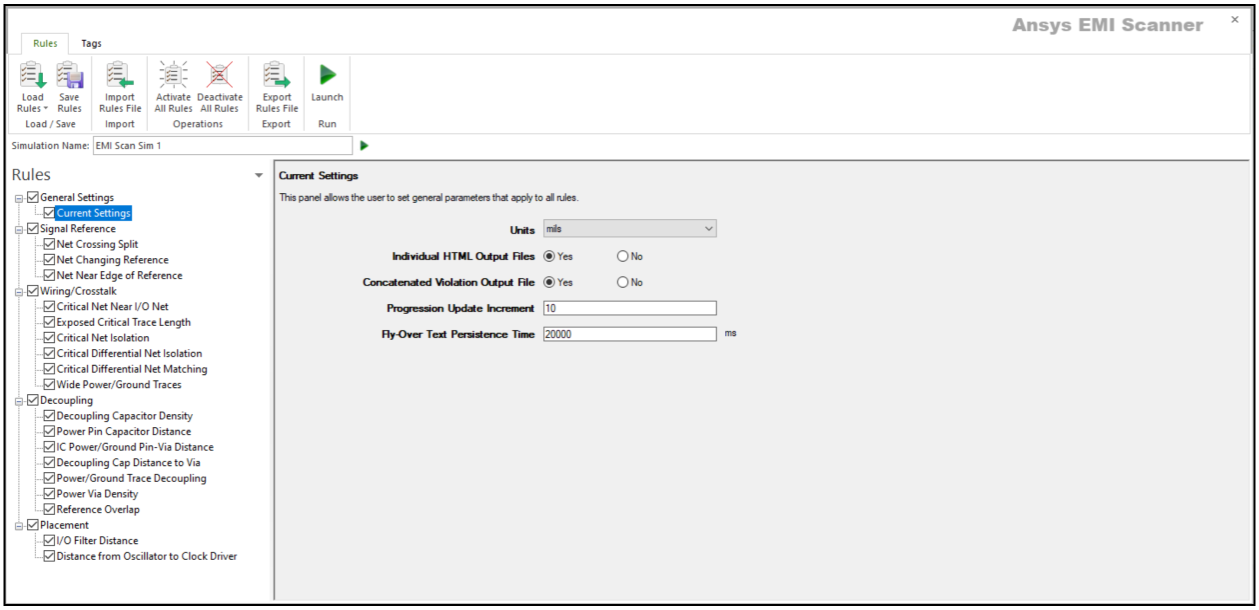
   The **Tags** tab displays.



5. Assign tags, as chosen.
6. Click **Launch**.

   The **Progress** window updates to show EMI Scanner progress. Simulation Status messages are logged in the **Message Manager**.

Setting Rules in EMI Scanner



The EMI Scanner **Rules** tab contains the following buttons:

- **Load Rules** – allows you to select from a series of pre-defined rule sets to load into the Rules pane:
  - **Default EM Rules Profile** – contains the basic rule set for analyzing electromagnetic interference.
  - **Default SI Rules Profile** – contains the basic rule set for analyzing signal integrity.
  - **Default EM+SI Rules Profile** – contains all rules from both the Default EM Rules Profile and the Default SI Rules Profile.
  - **Default Rules Profile (0 - 100MHz)** – contains the recommended rule set for frequencies ranging from 0 - 100MHz.
  - **Default Rules Profile (100MHz - 500MHz)** – contains the recommended rule set for frequencies ranging from 100 - 500MHz.
  - **Default Rules Profile (500MHz - 1000MHz)** – contains the recommended rule set for frequencies ranging from 500 - 1000MHz.
  - **Default Rules Profile (1GHz+)** – contains the recommended rule set for frequencies 1GHz and above.
- **Save Rules** – applies changes to the rule profile.
- **Import Rules File** – allows you to import rules via an XML file.
- **Activate All Rules** – selects all check boxes in the Rules pane.
- **Deactivate All Rules** – deselects all check boxes in the Rules pane.

- **Export Rules File** – allows you to export the current rule settings in XML format.
- **Launch** – begins the EMI scan.

To set rules for an EMI Scan:

1. Click the **Load Rules** drop-down menu to select a pre-defined rules profile, or click **Import Rules File**.

   The **Rules** pane updates to show all applicable rules for the selected profile.

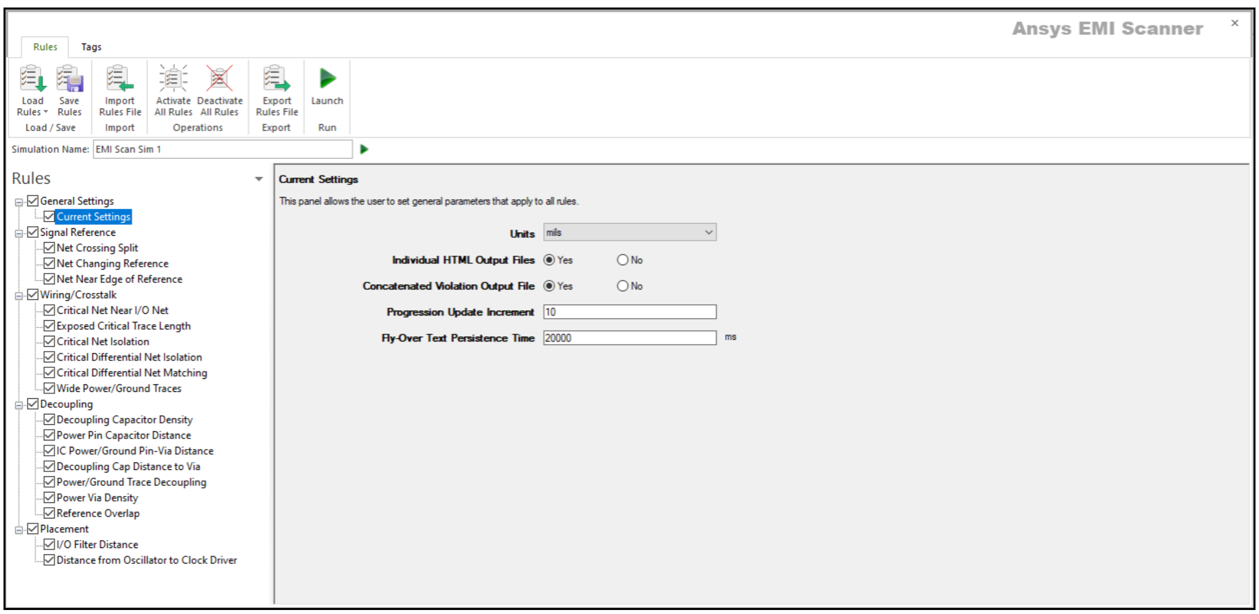
2. From the **Rules** pane, use the check boxes to activate and deactivate rules, as appropriate. Also **Activate All Rules** or **Deactivate All Rules**.
3. Change the General Settings and settings for Signal Reference, Wiring/Crosstalk, Decoupling, Placement, Net Integrity, Via Integrity, and Power Integrity rules, as appropriate.

Applying EMI Scanner General Settings

To change EMI Scanner General Settings:

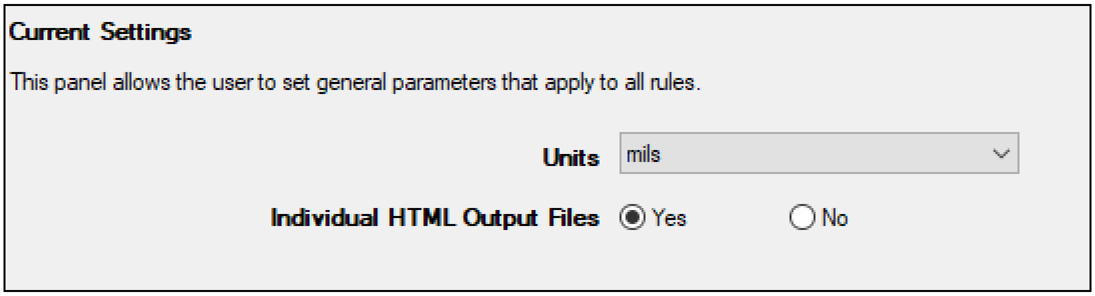
1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



2. The **Current Settings** display by default. If they do not, in the Rules pane, select **General Settings** > **Current Settings**.

   The **Current Settings** display.



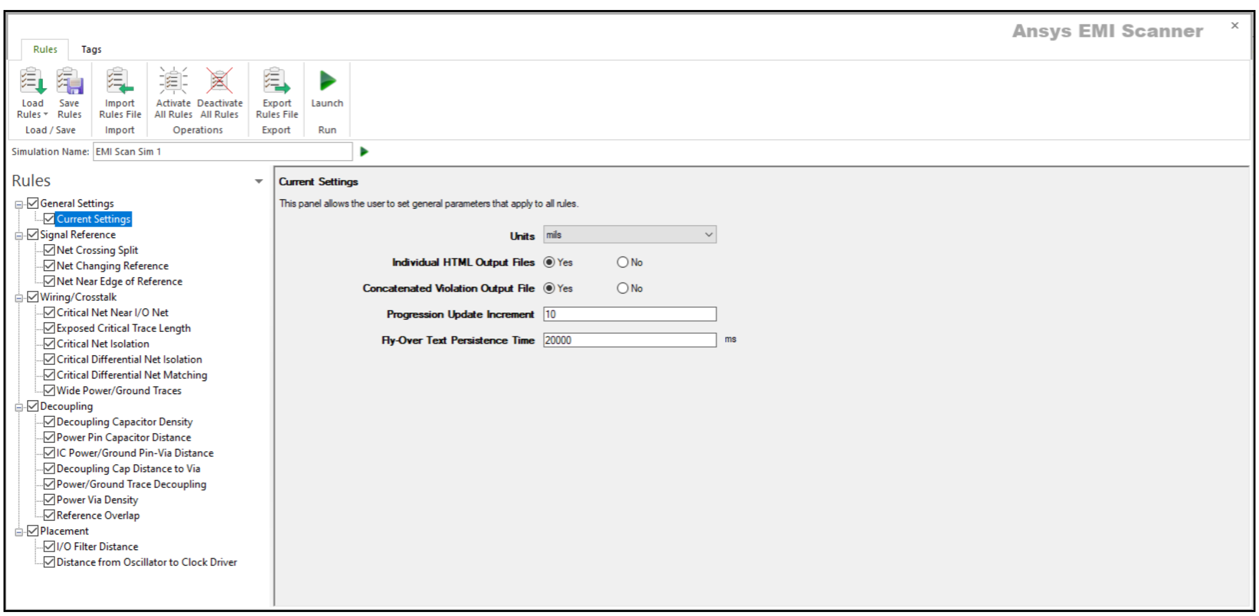
3. From this screen, set the following:
   - **Units** – selects the unit in which other rules are measured.
   - **Individual HTML Output Files** – selects whether to export EMI Scanner results as one large, combined HTML file or as individual HTML files. See Exporting EMI Scanner Results.
4. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

Applying Signal Reference Rules

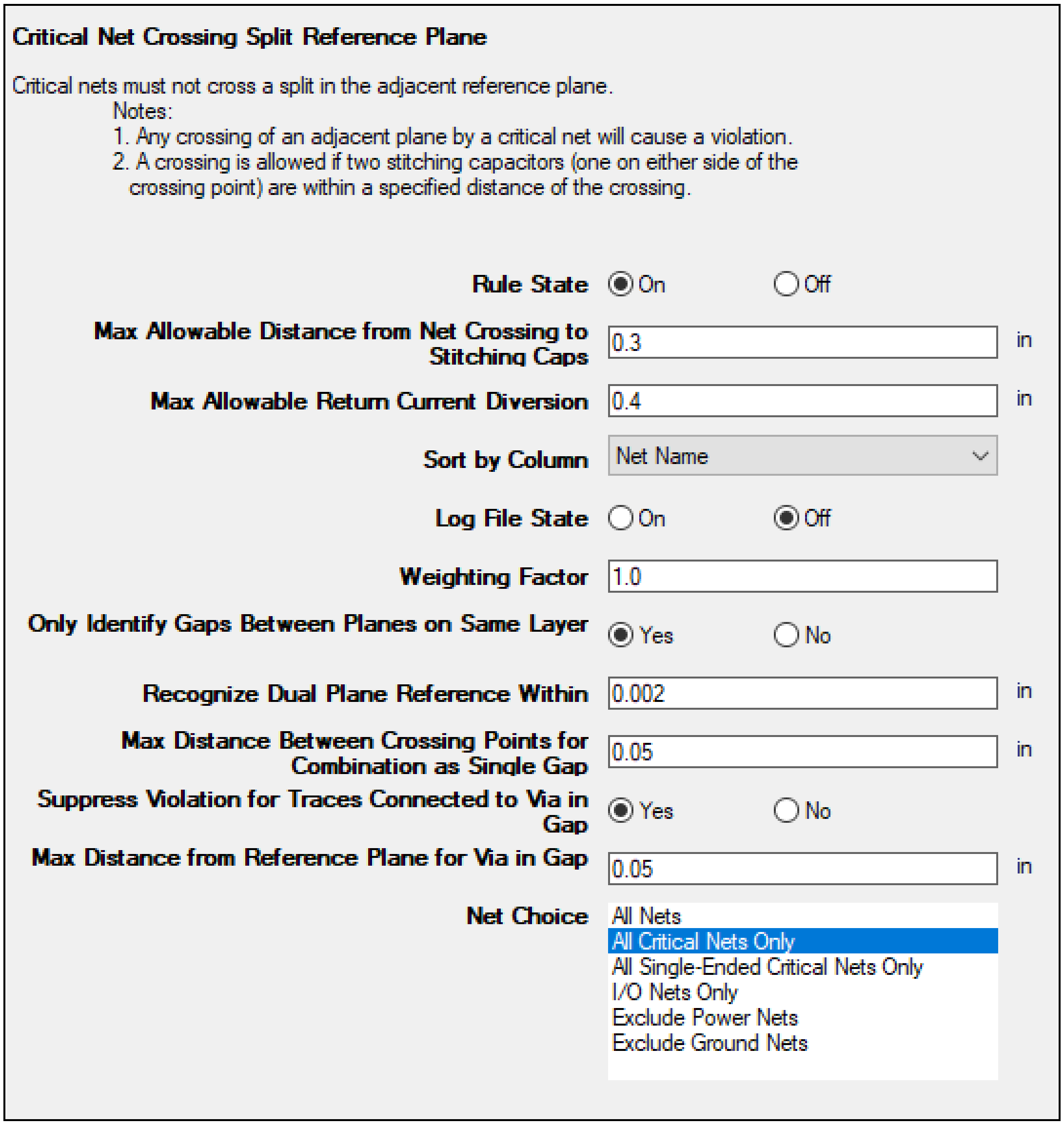
To change EMI Scanner Signal Reference rules:

1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



2. From the **Rules** pane, select one of the three subcategories under **Signal Reference**:
   - **Net Crossing Split** – displays **Critical Net Crossing Split Reference Plane** rule configuration.

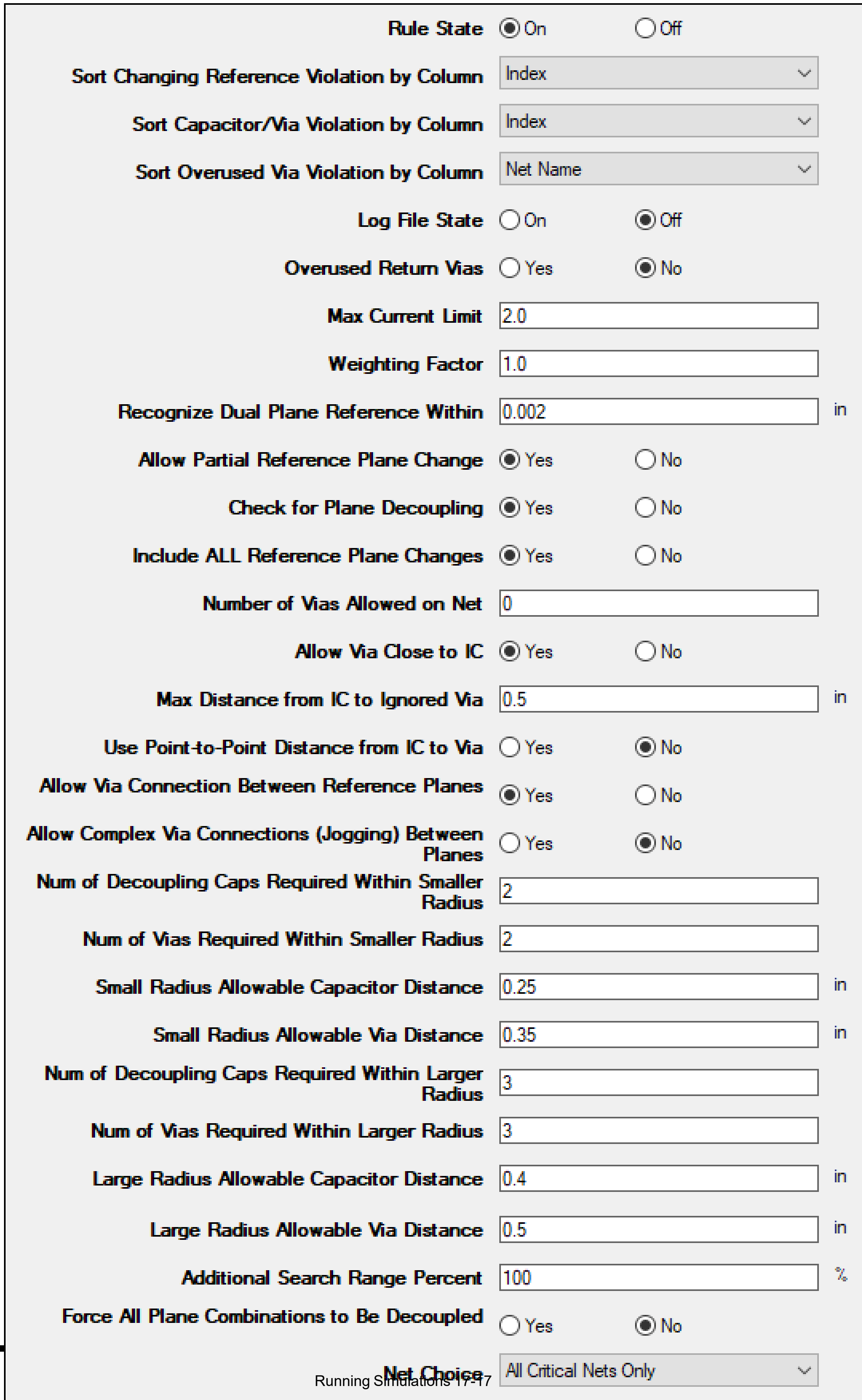


This rule states that critical nets must not cross a split in the adjacent reference plane. A cross is allowed only if two stitching capacitors are within a specified distance of the crossing.

From here, set the following:

- **Rule State** – toggles Net Crossing Split rules on or off.
- **Max Allowable Distance from Net Crossing to Stitching Caps** – sets the maximum distance on the point where the net crosses a split in its reference plane. Two stitching capacitors must be found within this distance.
- **Max Allowable Return Current Diversion** – sets the maximum distance for a return current diversion. If a signal's return current can divert itself around the gap in the reference plane, the total distance of the diversion must be less than this value.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Only Identify Gaps Between Planes on Same Layer** – determines whether gaps are only between planes on the same layer or not. When the rule tries to pair up nearby plane boundary crossing points and identify gaps, this parameter enables or disables a check for layer commonality.
- **Recognize Dual Plane Reference Within** – instructs EMI Scanner to consider both planes as a reference when checking for gap crossing when the distance difference to each plane is less than the specified distance. This allows recognition of both planes as references for strip line traces when the trace is offset on the exact center by a small amount (normally for manufacturing reasons).
- **Max Distance Between Crossing Points for Combination as a Single Gap** – sets the maximum distance for combining crossing points into a single violation (e.g., the entrance and exit of a gap between planes). Too large a value can cause the algorithm to pair up unrelated crossing points, and too small a value can prevent crossing points from being paired up, resulting in more violations. It is recommended to set the value to a reasonable distance based on the board design technology being used (e.g., slightly larger than the typical plane separation within a layer).
- **Suppress Violation for Traces Connected to Via in Gap** – determines whether to ignore trace crossing split/gap when connected to a via within that gap.
- **Max Distance from Reference Plane for Via in Gap** – sets the maximum distance for which via violations within a gap are suppressed. If the trace extends greater than this distance on the edge of the reference plane to the via, its violation are not suppressed.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Net Changing Reference** – displays **Critical Net Changing Reference Plane** rule configuration.


Rule State
On
Off
Sort Changing Reference Violation by Column
Index
Sort Capacitor/Via Violation by Column
Index
Sort Overused Via Violation by Column
Net Name
Log File State
On
Off
Overused Return Vias
Yes
No
Max Current Limit
2.0
Weighting Factor
1.0
Recognize Dual Plane Reference Within
0.002
in
Allow Partial Reference Plane Change
Yes
No
Check for Plane Decoupling
Yes
No
Include ALL Reference Plane Changes
Yes
No
Number of Vias Allowed on Net
0
Allow Via Close to IC
Yes
No
Max Distance from IC to Ignored Via
0.5
in
Use Point-to-Point Distance from IC to Via
Yes
No
Allow Via Connection Between Reference Planes
Yes
No
Allow Complex Via Connections (Jogging) Between Planes
Yes
No
Num of Decoupling Caps Required Within Smaller Radius
2
Num of Vias Required Within Smaller Radius
2
Small Radius Allowable Capacitor Distance
0.25
in
Small Radius Allowable Via Distance
0.35
in
Num of Decoupling Caps Required Within Larger Radius
3
Num of Vias Required Within Larger Radius
3
Large Radius Allowable Capacitor Distance
0.4
in
Large Radius Allowable Via Distance
0.5
in
Additional Search Range Percent
100
%
Force All Plane Combinations to Be Decoupled
Yes
No
Net Choice
All Critical Nets Only

This rule states that critical nets must not change reference planes.

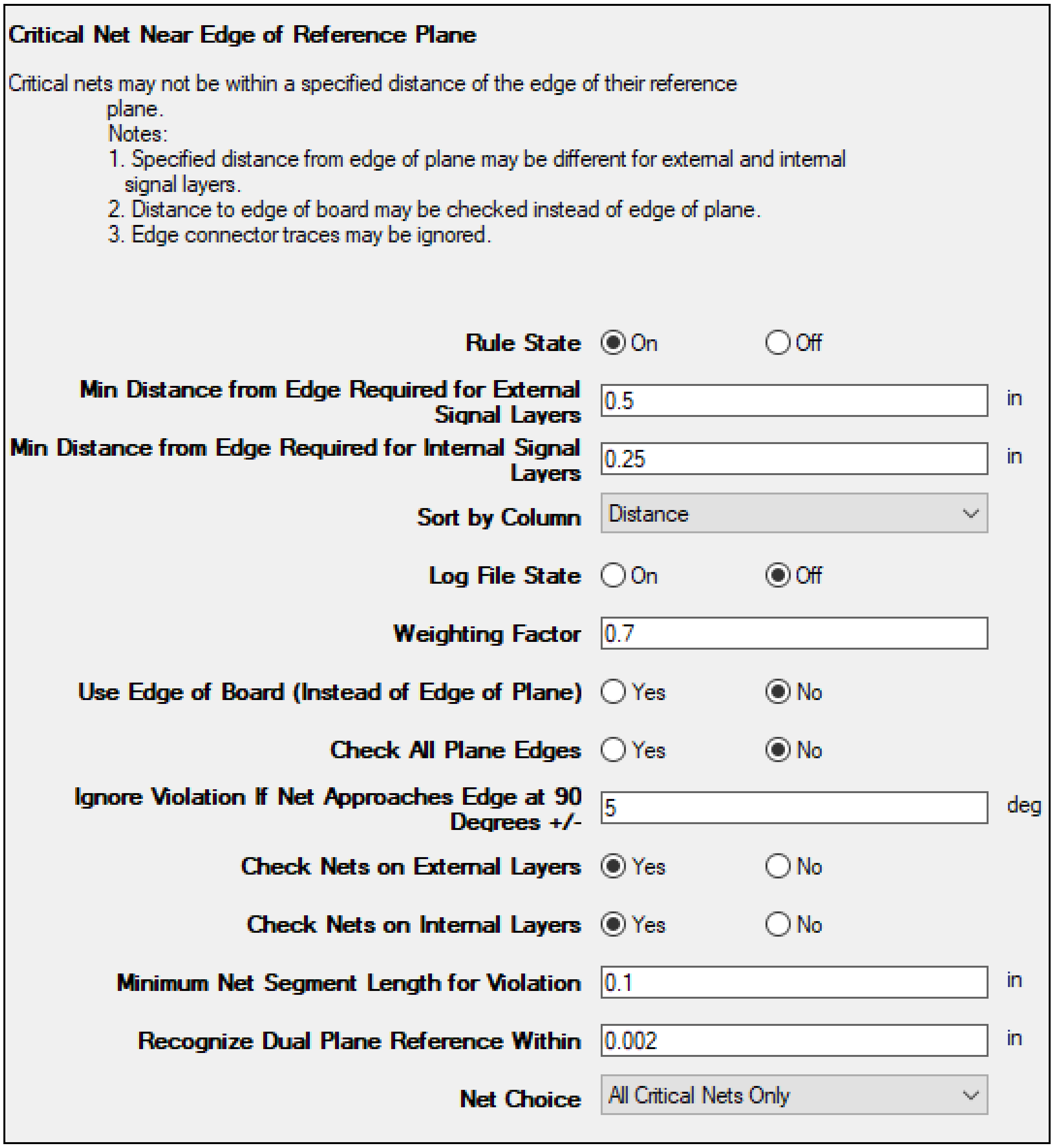
> **Note:**
> 1. Any change in the reference plane for a critical net causes a violation.
> 2. Reference plane changing is allowed if two decoupling capacitors (or vias if the planes share the same net name) are within a specified radial distance on the signal via where the signal changes reference planes.
> 3. Reference plane changing is allowed if three decoupling capacitors (or vias if the planes share the same net name) are within a specified radial distance on the signal via where the signal changes reference planes.
> 4. Reference plane changing can be ignored when the initial via is within a specified distance from its IC pin (transmit or receive).

From here, set the following:

- **Rule State** – toggles Net Changing Reference rules on or off.
- **Sort Changing Reference Violation by Column** – specifies the column by which Changing Reference Violation results are sorted.
- **Sort Capacitor/Via Violation by Column** – specifies the column by which Capacitor/Via Violation results are sorted.
- **Sort Overused Via Violation by Column** – specifies the column by which Overused Via Violation results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Overused Return Vias** – determines whether EMI Scanner checks for overused return vias.
- **Max Current Limit** – specifies the maximum number of signal currents that a via should carry. Use in conjunction with **Overused Return Vias**.
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Recognize Dual Plane Reference Within** – instructs EMI Scanner to consider both planes as a reference when checking for gap crossing when the distance difference to each plane is less than the specified distance. This allows recognition of both planes as references for strip line traces when the trace is offset on the exact center by a small amount (normally for manufacturing reasons).
- **Allow Partial Reference Plane Change** – instructs EMI Scanner to ignore a violation if one of the two reference planes remains a reference plane on both sides of the via transition.

- **Check for Plane Decoupling** – instructs EMI Scanner to ignore a violation if a decoupling capacitor exists between the two reference planes (entering reference plane and leaving reference plane) and is within a specified distance of the signal via.
- **Include ALL Reference Plane Changes** – instructs EMI Scanner whether to display all reference plane changes. When there are two associated reference planes for the entry to the via, and two different associated reference planes for the trace leaving the via, selecting 'yes' shows two violations for this transition. If 'no' is selected and a decoupling capacitor exists between one of the entering reference planes and one of the leaving reference planes (in the specified distance), no violation are reported, even though the two other reference planes do not have any connection.
- **Number of Vias Allowed on Net** – specifies the number of via reference plane changes that are allowed on critical nets. If no vias are chosen except the 'escape' via close to the IC, set this parameter to zero, and set **Allow Via Close to IC** to 'yes'.
- **Allow Via Close to IC** – determines whether to ignore vias that are close to the IC and are required to get a trace on the surface to an inner layer.
- **Max Distance from IC to Ignored Via** – specifies the maximum distance between IC and via at which vias is ignored.
- **Use Point-to-Point Distance from IC to Via** – if 'yes' is selected, the actual trace path length is NOT calculated. This option allows EMI Scanner to check rules faster, but may cause violations to be ignored incorrectly.
- **Allow Via Connection Between Reference Planes** – allows a via to be used to connect reference planes (to provide return current path) when the two reference planes share the same net name.
- **Allow Complex Via Connections (Jogging) Between Planes** – allows a set of vias (e.g., 'blind' vias that do not extend through all layers) to be used to connect reference planes (to provide return current path) when the two reference planes share a net name. This advanced checking requires extra processing time because all vias found in the larger via radius are considered as potential paths.
- **Num of Decoupling Caps Required Within Smaller Radius** – when **Allow Partial Reference Plane Change** is selected, a return current path (when a critical net changes reference planes) may be provided with either capacitors or vias. This setting specifies the number of capacitors required in the smaller radius.
- **Num of Vias Required Within Smaller Radius** – when **Allow Partial Reference Plane Change** is selected, a return current path (when a critical net changes reference planes) may be provided with either capacitors or vias. This setting specifies the number of vias required in the smaller radius. A via between planes is a lower impedance/inductance connection than a decoupling capacitor, so you may want to specify a different setting.

  - **Small Radius Allowable Capacitor Distance** – specifies the smaller radius used with the **Num of Decoupling Caps Required Within Smaller Radius** setting.
  - **Small Radius Allowable Via Distance** – specifies the smaller radius used with the **Num of Vias Required Within Smaller Radius** setting.
  - **Num of Decoupling Caps Required Within Larger Radius** – if a smaller radius is not specified, this allows specification of a larger radius for a larger number of decoupling capacitors.
  - **Num of Vias Required Within Larger Radius** – if a smaller radius is not specified, this allows specification of a larger radius for a larger number of vias. A via between planes is a lower impedance/inductance connection then a decoupling capacitor, so you may want to specify a different setting.
  - **Large Radius Allowable Capacitor Distance** – specifies the larger radius used with the **Num of Decoupling Caps Required Within Larger Radius** setting.
  - **Large Radius Allowable Via Distance** – specifies the larger radius used with the **Num of Vias Required Within Larger Radius** setting.
  - **Additional Search Range Percent** – specifies an additional search range beyond the larger radius for additional capacitors (e.g., if the larger radius is set to 500mils and the Additional Search Range Percent is 100%, EMI Scanner reports capacitors within 1000mils when a violation occurs).
  - **Force All Plane Combinations to Be Decoupled** – determines whether to decouple all reference plane combinations. See Technical Note.
  - **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.
- **Net Near Edge of Reference** – displays **Critical Net Near Edge of Reference Plane** rule configuration.



This rule states that critical nets must not be within a specified distance of the edge of their reference plane.

From here, set the following:

- **Rule State** – toggles Net Near Edge of Reference rules on or off.
- **Min Distance from Edge Required for External Signal Layers** – specifies the minimum distance that must exist between the edge of the reference plane and a critical signal net when the net is on an external layer.

- **Min Distance from Edge Required for Internal Signal Layers** – specifies the minimum distance that must exist between the edge of the reference plane and a critical signal net when the net is on an internal layer.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Use Edge of Board (Instead of Edge of Plane)** – allows you to check distance from critical nets to the edge of the board instead of the edge of the reference plane. This can be useful if the board is only partially complete and all reference planes are not yet completed.
- **Check All Plane Edges** – checks the distance from nets to any edge of their reference plane, rather than limiting the check to reference plane edges near the edge of the board
- **Ignore Violation If Net Approaches Edge at 90 Degrees +/-** – EMI scanner ignores edge connectors that approach the edge of the reference plane or board at a 90 degree angle. This option specifies a tolerance that allows the approach to be slightly off a perfect 90 degrees.
- **Check Nets on External Layers** – specifies whether nets on external layers should be checked.
- **Check Nets on Internal Layers** – specifies whether nets on internal layers should be checked.
- **Minimum Net Segment Length for Violation** – specifies the minimum net segment length before EMI Scanner reports a violation if the net is in the specified distance on the edge of the reference plane or board.
- **Recognize Dual Plane Reference Within** – instructs EMI Scanner to consider both planes as a reference when checking for traces too close to the edge of the reference planes when the distance difference to each plane is less than the specified distance. This allows recognition of both planes as references for strip line traces when the trace is offset on the exact center by a small amount (normally for manufacturing reasons).
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

3. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

Note:

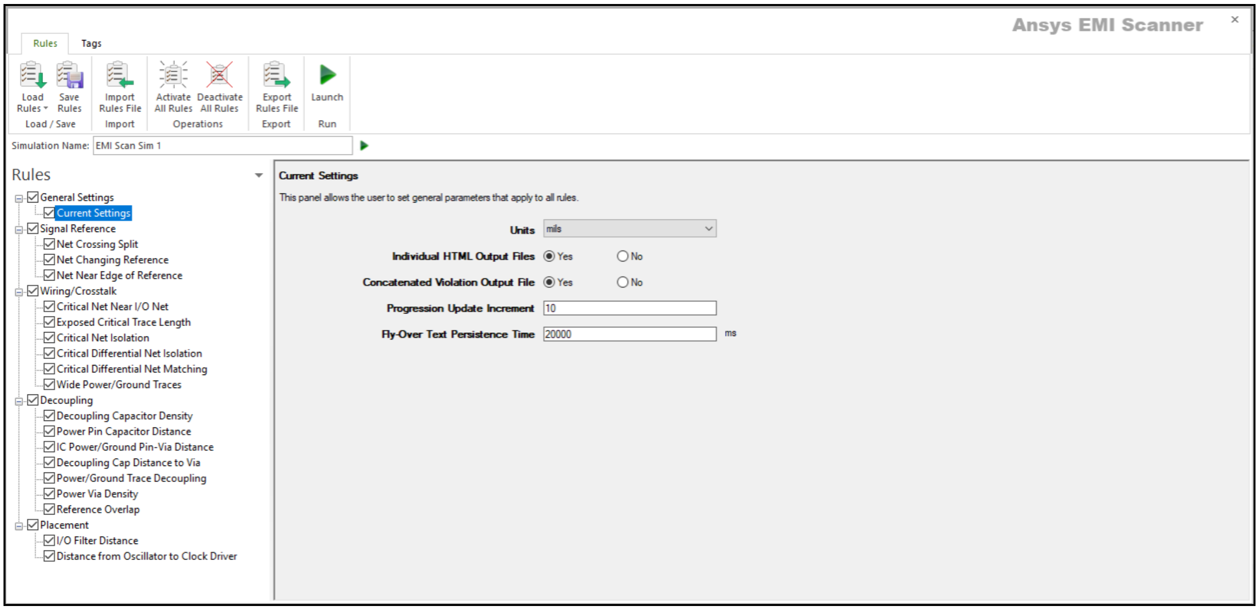
- You can display a description of a rule parameter at any time as with the cursor over it.
- Units of measure are set in General Settings.

Applying Wiring/Crosstalk Rules

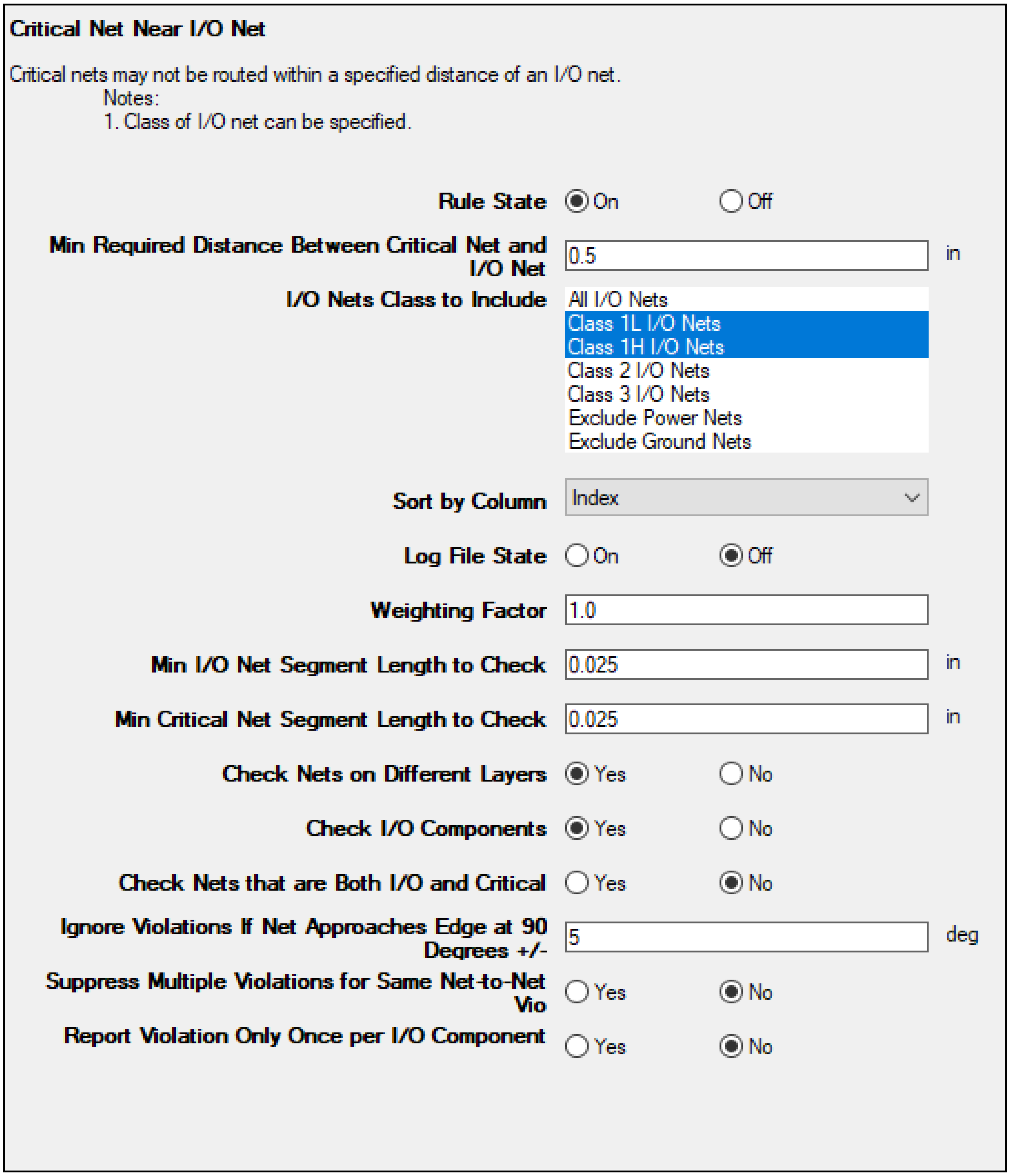
To change EMI Scanner Wiring/Crosstalk rules:

1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



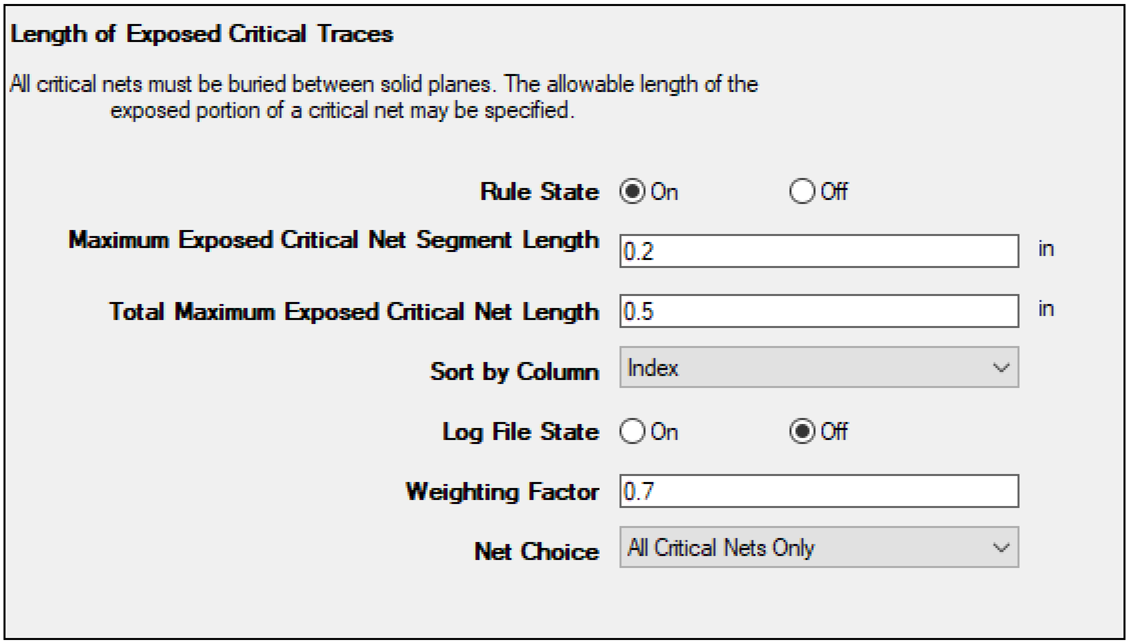
2. From the **Rules** pane, select one of the six subcategories under **Wiring/Crosstalk**:
   - **Critical Net Near I/O Net** – displays **Critical Net Near I/O Net** rule configuration.



This rule states that critical nets may not be routed within a specified distance from an I/O net.

From here, set the following:

  - **Rule State** – toggles Critical Net Near I/O Net rules on or off.
  - **Min Required Distance Between Critical Net and I/O Net** – specifies the minimum distance between all critical nets and all I/O nets.
  - **I/O Nets Class to Include** – specifies the type(s) of I/O Nets to include. Select one, or **Ctrl**+click to select multiple types. See Technical Note.
  - **Sort by Column** – specifies the column by which results are sorted.
  - **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
  - **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
  - **Min I/O Net Segment Length to Check** – specifies the smallest I/O net segment length to check for violations. EMI Scanner ignores All I/O net segments smaller than this length.
  - **Min Critical Net Segment Length to Check** – specifies the smallest critical net segment length to check for violations. EMI Scanner ignores all critical net segments smaller than this length.
  - **Check Nets on Different Layers** – determines whether to check the distance between I/O nets and critical nets when these nets are on different layers and between the same pair of planes.
  - **Check I/O Components** – determines whether any I/O component that is in the specified distance from a critical net causes a violation.
  - **Check Nets that are Both I/O and Critical** – determines whether nets labeled as both critical and I/O should be checked for distance from other I/O nets.
  - **Ignore Violations If Net Approaches Edge at 90 Degrees +/-** – EMI scanner ignores edge connectors that approach the edge of the reference plane or board at a 90 degree angle. This option specifies a tolerance that allows the approach to be slightly off a perfect 90 degrees.
  - **Suppress Multiple Violations for Same Net-to-Net Ratio** – determines whether to suppress multiple violation reporting when multiple segments on a critical net infringe on the same I/O net.
  - **Report Violation Only Once per I/O Component** – reduces run time by checking only for violations between critical nets and I/O components when no other violation has been found for an I/O component.
- **Exposed Critical Trace Length** – displays **Length of Exposed Critical Traces** rule configuration.

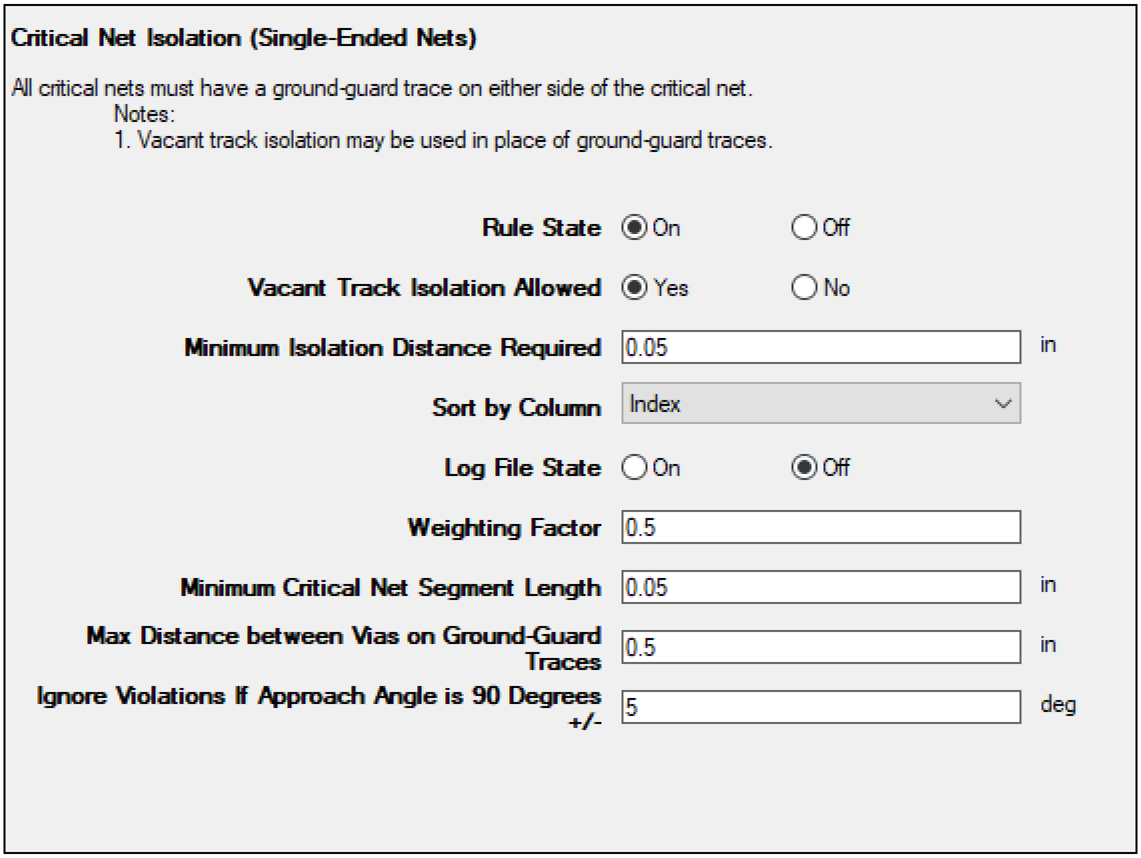


This rule states that all critical nets must be buried between solid planes.

From here, set the following:

- **Rule State** – toggles Exposed Critical Trace Length rules on or off.
- **Maximum Exposed Critical Net Segment Length** – specifies the maximum critical net segment length allowed on external (exposed) layers.
- **Total Maximum Exposed Critical Net Length** – specifies the total maximum critical net length allowed on external (exposed) layers (e.g., multiple segments on a critical net may be exposed, but short enough not to cause a violation individually).
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Critical Net Isolation** – displays **Critical Net Isolation (Single-Ended Nets)** rule configuration.



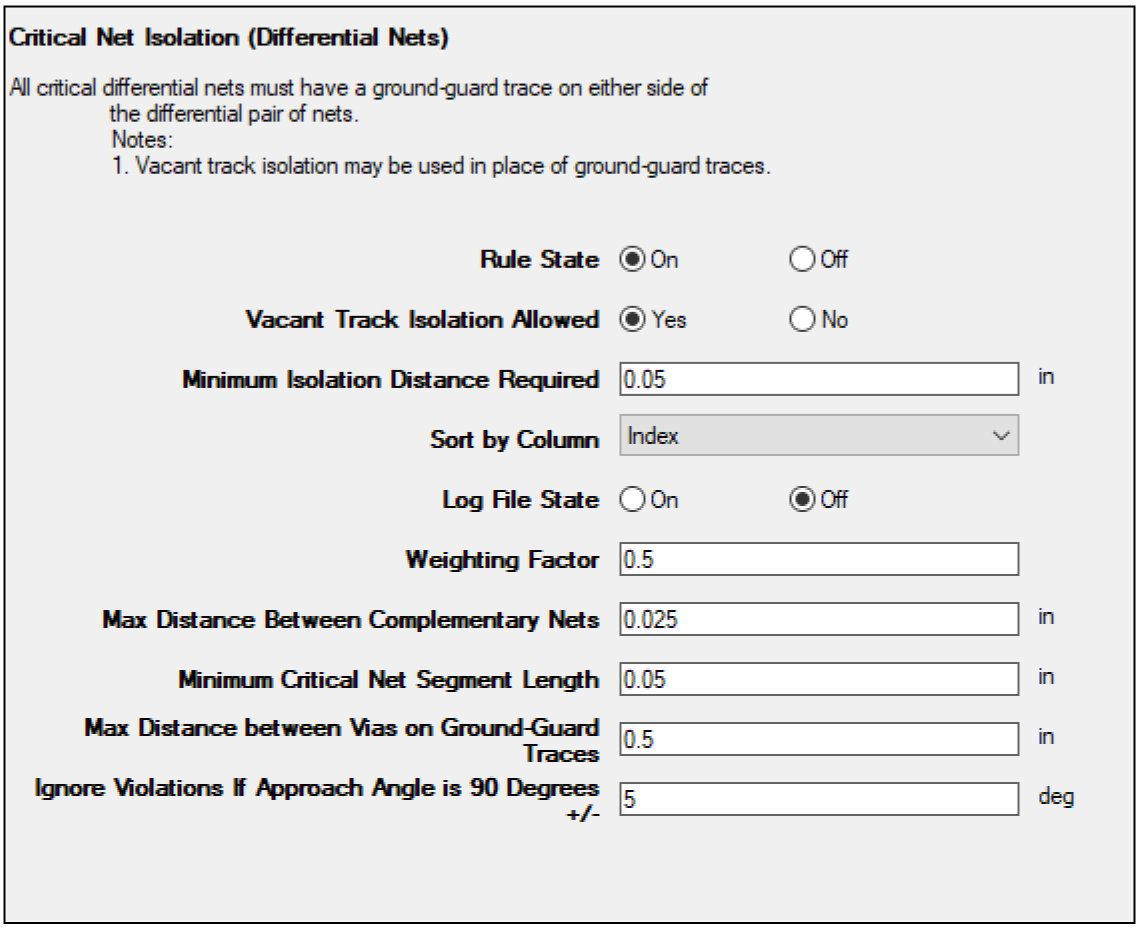
This rule states that all critical nets must have empty tracks or a "ground-guard" trace on either side of the critical net.

From here, set the following:

- **Rule State** – toggles Critical Net Isolation rules on or off.
- **Vacant Track Isolation Allowed** – specifies whether vacant tracks (empty space) may be used in place of ground-guard traces.
- **Minimum Isolation Distance Required** – selects the minimum edge-to-edge distance required between critical nets and all other nets before EMI Scanner reports a violation. Used in coordination with **Vacant Track Isolation Allowed**.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

- **Minimum Critical Net Segment Length** – specifies the minimum critical net segment length before EMI Scanner reports a violation. Segments shorter than this length is ignored.
- **Max Distance Between Vias on Ground-Guard Traces** – specifies the maximum distance between vias on ground-guard traces.
- **Ignore Violations If Approach Angle is 90 Degrees +/-** – EMI scanner ignores edge connectors that approach the edge of the reference plane or board at a 90 degree angle. This option specifies a tolerance that allows the approach to be slightly off a perfect 90 degrees.

- **Critical Differential Net Isolation** – displays **Critical Net Isolation (Differential Nets)** rule configuration.

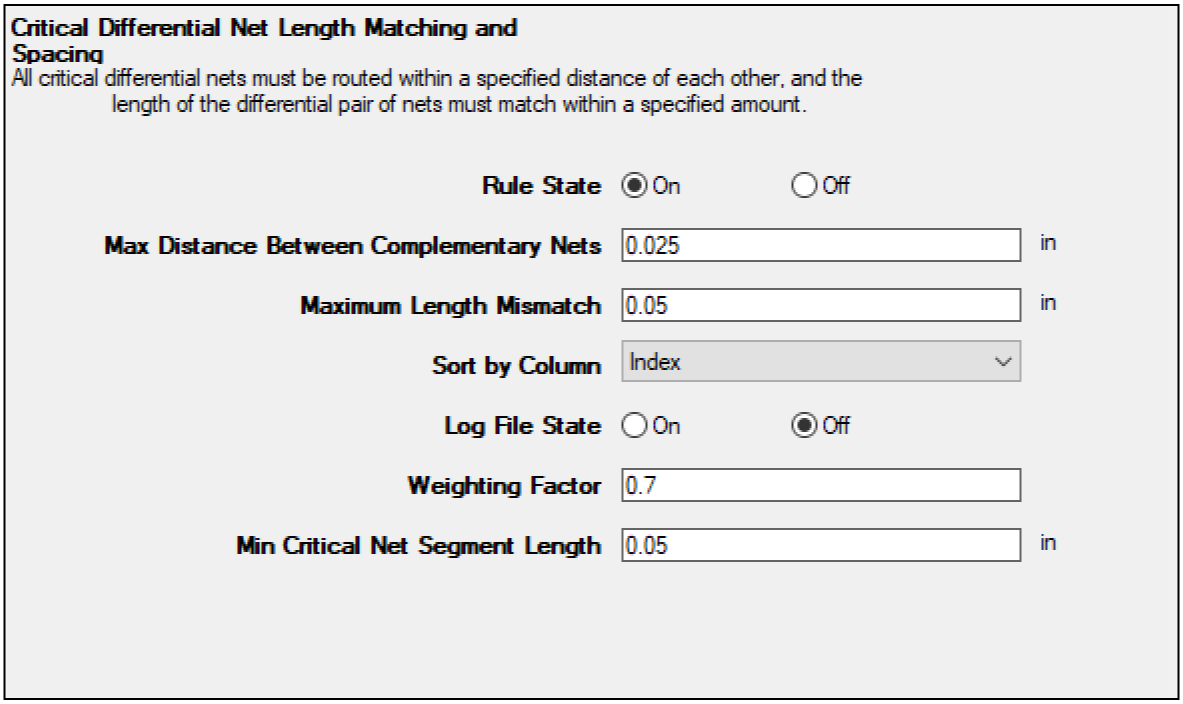


This rule states that all differential critical nets must have empty tracks or a "ground-guard" trace on either side of the differential pair.

From here, set the following:

- **Rule State** – toggles Critical Differential Net Isolation rules on or off.
- **Vacant Track Isolation Allowed** – determines whether vacant tracks (empty space) may be used in place of ground-guard traces.
- **Minimum Isolation Distance Required** – when **Vacant Track Isolation Allowed** is selected, this setting determines the minimum edge-to-edge distance required between critical nets and all other nets before EMI Scanner reports a violation.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Max Distance Between Complementary Nets** – sets the maximum spacing between a pair of critical nets (creating a critical differential pair).
- **Minimum Critical Net Segment Length** – allows you to ignore short critical net segments by specifying the minimum critical net segment length before EMI Scanner reports a violation.
- **Max Distance between Vias on Ground-Guard Traces** – specifies the maximum distance between vias on ground-guard traces.
- **Ignore Violations If Approach Angle is 90 Degrees +/-** – EMI scanner ignores edge connectors that approach the edge of the reference plane or board at a 90 degree angle. This option specifies a tolerance that allows the approach to be slightly off a perfect 90 degrees.

- **Critical Differential Net Matching** – displays **Critical Differential Net Length Matching and Spacing** rule configuration.

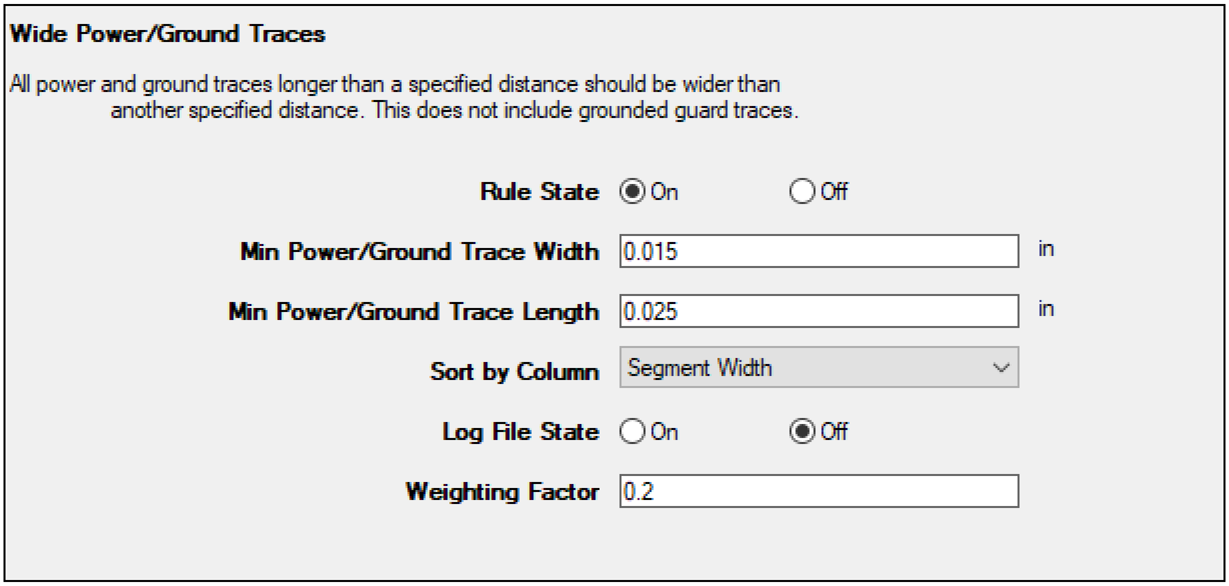


This rule states that all critical nets must be routed within a specified distance of each other, and the length of the differential pair must match within a specified amount.

From here, set the following:

- **Rule State** – toggles Critical Differential Net Matching rules on or off.
- **Max Distance Between Complementary Nets** – sets the maximum spacing between a pair of critical nets (creating a critical differential pair).
- **Maximum Length Mismatch** – specifies the maximum difference in length between a critical differential pair of nets.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Min Critical Net Segment Length** – allows you to ignore short critical net segments by specifying the minimum critical net segment length before EMI Scanner reports a violation.

- **Wide Power/Ground Traces** – displays **Wide Power/Ground Traces** rule configuration.



This rule states that all power and ground traces longer than a specified distance must be wider than a specified amount.

From here, set the following:

- **Rule State** – toggles Wide Power/Ground Traces rules on or off.
- **Min Power/Ground Trace Width** – sets a minimum width limit for all power/ground traces (except grounded guard traces).
- **Min Power/Ground Trace Length** – specifies the smallest trace length to check for violations. EMI Scanner ignores traces smaller than this length.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

3. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

Note:

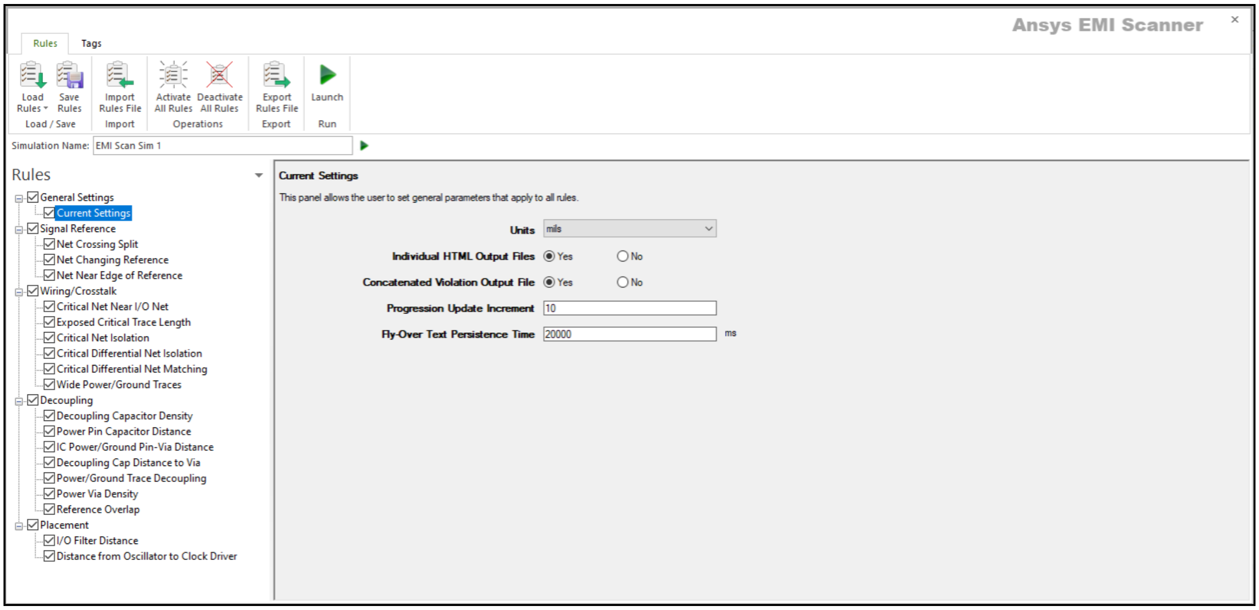
- You can display a description of a rule parameter at any time as with the cursor over it.
- Units of measure are set in General Settings.

Applying Decoupling Rules

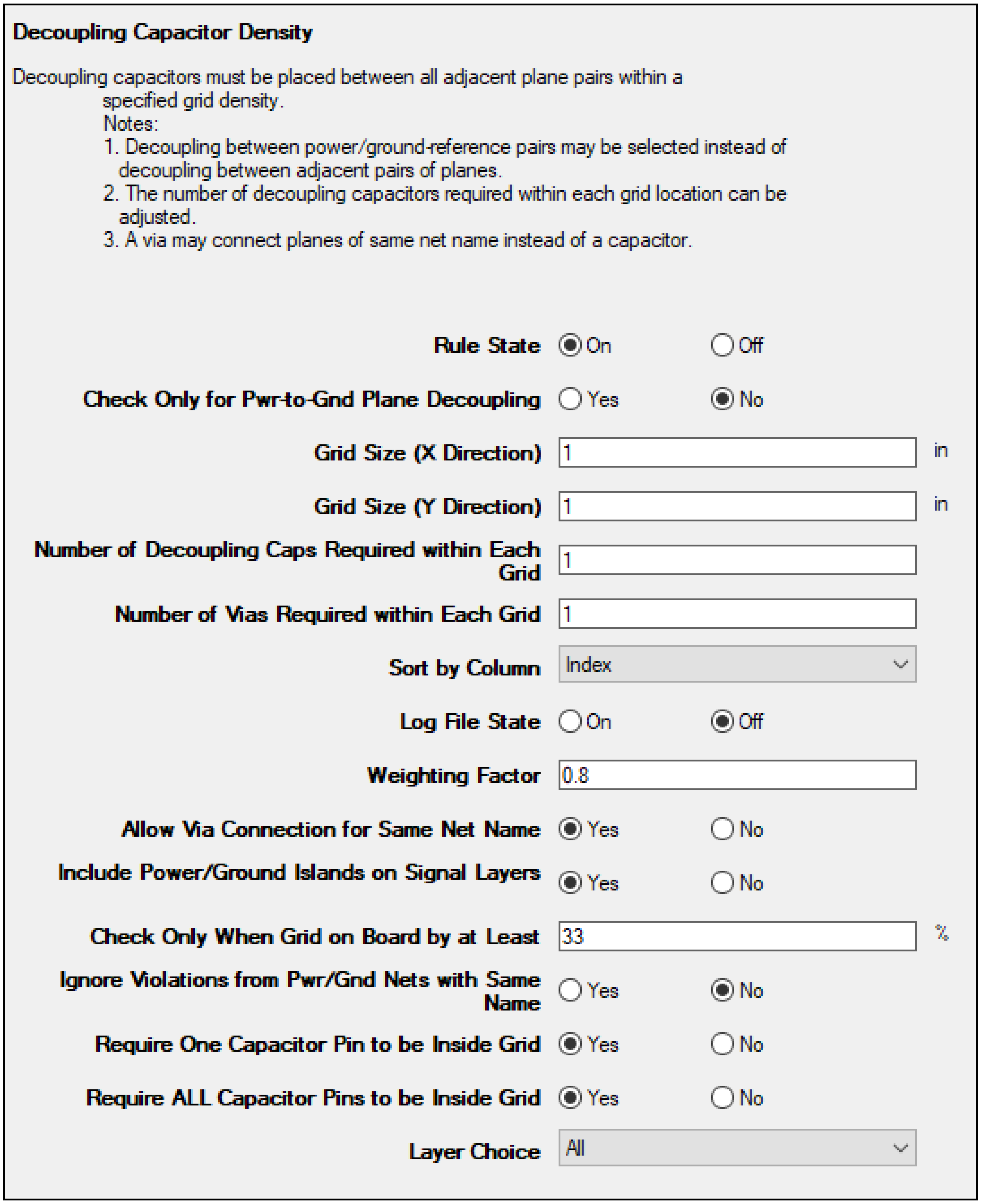
To change EMI Scanner Decoupling rules:

1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



2. From the Rules pane, select one of the six subcategories under **Decoupling**:
   - **Decoupling Capacitor Density** – displays **Decoupling Capacitor Density** rule configuration.

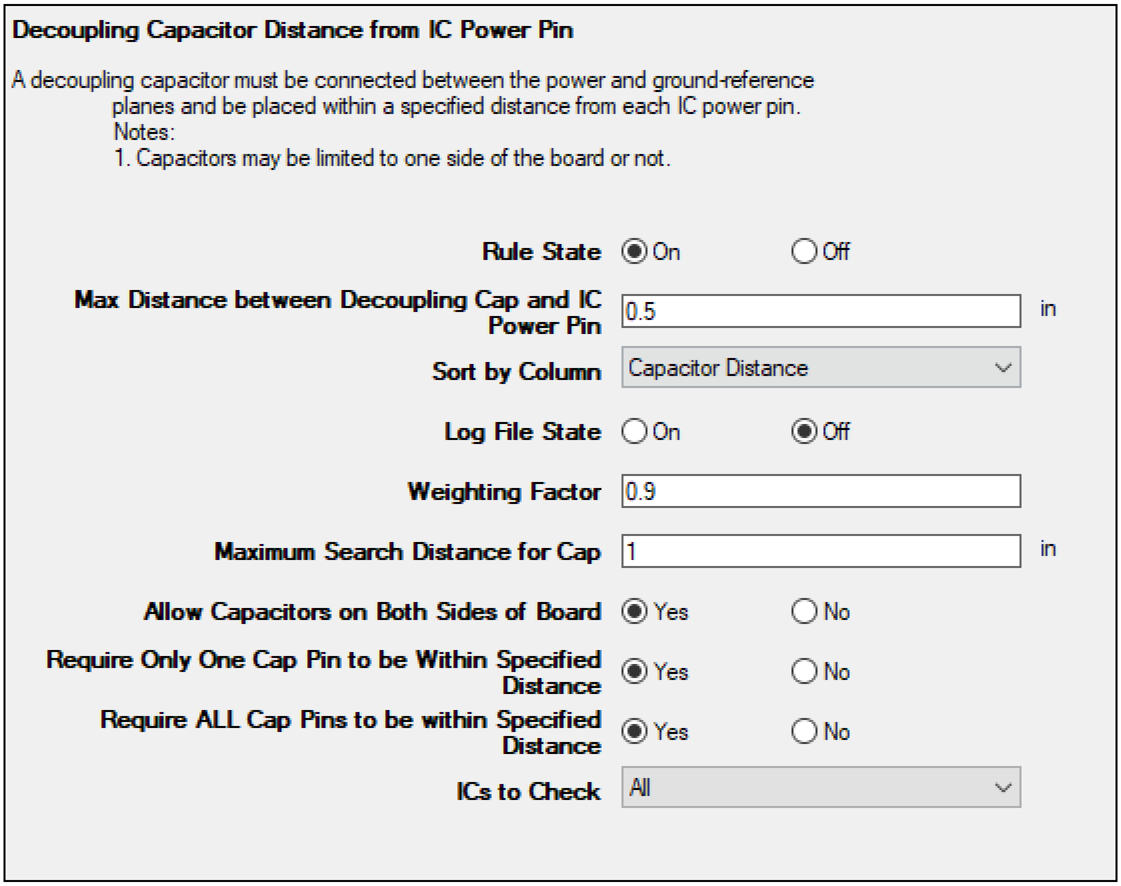


This rule states that decoupling capacitors must be placed between all adjacent plane pairs within a specified grid density.

From here, set the following:

- **Rule State** – toggles Decoupling Capacitor Density rules on or off.
- **Check Only for Pwr-to-Gnd Plane Decoupling** – sets whether to ignore plane-to-plane decoupling and check only for decoupling capacitors between power and ground planes.
- **Grid Size (X Direction)** – specifies the size of the grid used to check for decoupling capacitors. Used in conjunction with **Grid Size (Y Direction)**.
- **Grid Size (Y Direction)** – specifies the size of the grid used to check for decoupling capacitors. Used in conjunction with **Grid Size (X Direction)**.
- **Number of Decoupling Caps Required within Each Grid** – specifies the number of decoupling capacitors connecting between planes that are required within each grid location.
- **Number of Vias Required within Each Grid** – specifies the number of vias connecting between planes that are required within each grid location. You must select **Allow Via Connection for Same Net Name** to use this option.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Allow Via Connection for Same Net Name** – determines whether EMI Scanner allows a via in place of a decoupling capacitor when the two planes are the same net name.
- **Include Power/Ground Islands on Signal Layers** – specifies whether to check for decoupling on power/ground islands.
- **Check Only When Grid on Board by at Least** – specifies the percentage of a grid that must be on the board in order for it to be checked.
- **Ignore Violations from Pwr/Gnd Nets with Same Name** – determines that only one plane of a given name is checked for violations (e.g., if two planes are "gnd" and a violation occurs between gnd and another layer, only one violation is reported).
- **Require One Capacitor Pin to be Inside Grid** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
- **Require ALL Capacitor Pins to be Inside Grid** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
- **Layer Choice** – specifies the layers to include in this rule analysis.

- **Power Pin Capacitor Distance** – displays **Decoupling Capacitor Distance from IC Power Pin** rule configuration.

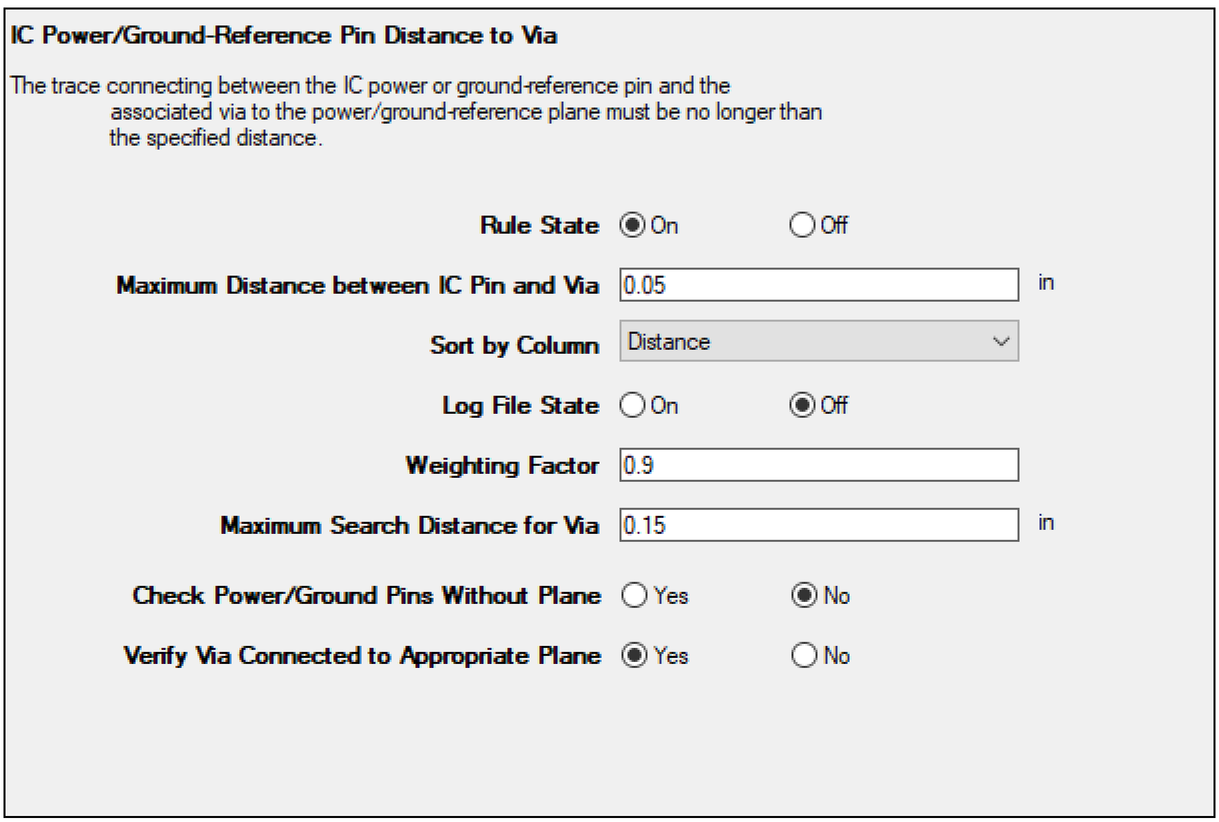


This rule states that a decoupling capacitor must be connected between the power and ground-reference planes and be placed within a specified distance of each IC power pin.

From here, set the following:

- **Rule State** – toggles Power Pin Capacitor Distance rules on or off.
- **Max Distance between Decoupling Cap and IC Power Pin** – specifies the maximum distance between an IC power pin and its decoupling capacitor.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

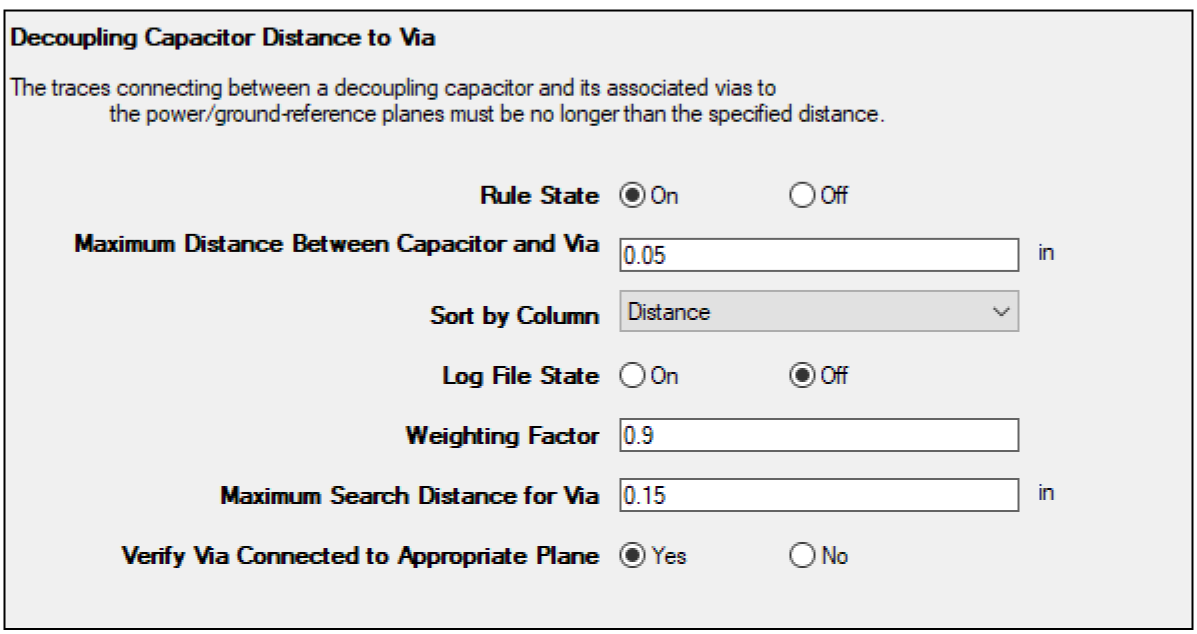
  - **Maximum Search Distance for Cap** – specifies a maximum search distance for a capacitor. A larger search box increases the chances of EMI Scanner finding capacitors, but can slow processing time.
  - **Allow Capacitors on Both Sides of Board** – specifies whether to include capacitors on both sides of the board.
  - **Require Only One Cap Pin to be Within Specified Distance** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
  - **Require ALL Cap Pins to be within Specified Distance** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
  - **ICs to Check** – selects the types of components to include in this rule analysis.
- **IC Power/Ground Pin-Via Distance** – displays **IC Power/Ground-Reference Pin Distance to Via** rule configuration.



This rule states the trace connecting the IC power/ground reference pin to its associated via to the power/ground-reference plane must be no longer than a specified distance.

From here, set the following:

  - **Rule State** – toggles IC Power/Ground Pin-Via Distance rules on or off.
  - **Maximum Distance between IC Pin and Via** – specifies the maximum distance between the center of the IC power or ground-reference pin and the center of the via to the power or ground-reference plane.
  - **Sort by Column** – specifies the column by which results are sorted.
  - **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
  - **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
  - **Maximum Search Distance for Via** – specifies a maximum search distance for a via when a violation is found. This allows you to spot vias that are barely outside the original requirement and therefore not serious violations.
  - **Check Power/Ground Pins Without Plane** – checks the distance between the IC power/ground pin and its via when the power/ground plane has not been completed yet.
  - **Verify Via Connected to Appropriate Plane** – selecting 'no' speeds up EMI Scanner by not verifying the via connects to the chosen plane (shape).
- **Decoupling Cap Distance to Via** – displays **Decoupling Capacitor Distance to Via** rule configuration.

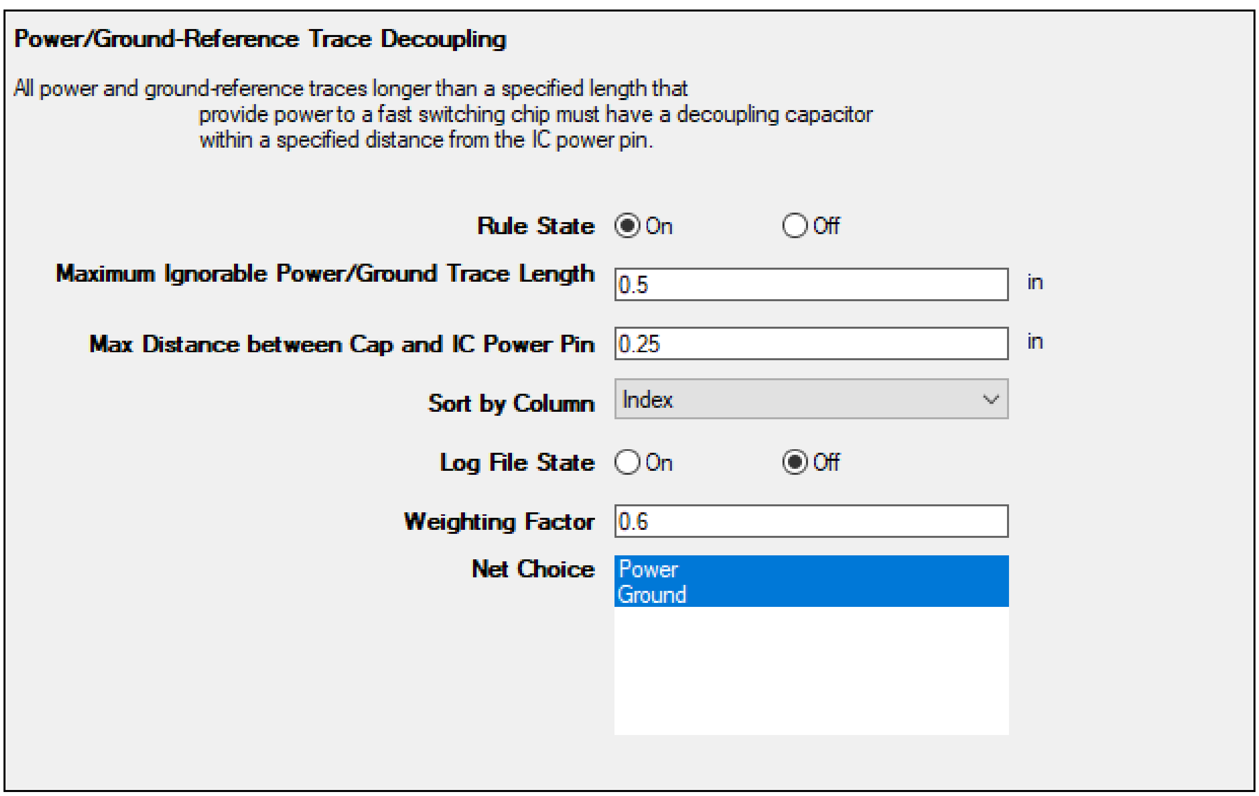


This rule states the trace connecting a decoupling capacitor to its associated via to the power/ground-reference plane must be no longer than the specified distance.

From here, set the following:

- **Rule State** – toggles Decoupling Cap Distance to Via rules on or off.
- **Maximum Distance Between Capacitor and Via** – specifies the maximum distance between the capacitor pin and the via to the power or ground-reference plane.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Maximum Search Distance for Via** – specifies a maximum search distance for a via when a violation is found. This allows you to spot vias that are barely outside the original requirement and therefore not serious violations.
- **Verify Via Connected to Appropriate Plane** – selecting 'no' speeds up EMI Scanner by not verifying the via connects to the chosen plane (shape).

- **Power/Ground Trace Decoupling** – displays **Power/Ground-Reference Trace Decoupling** rule configuration.

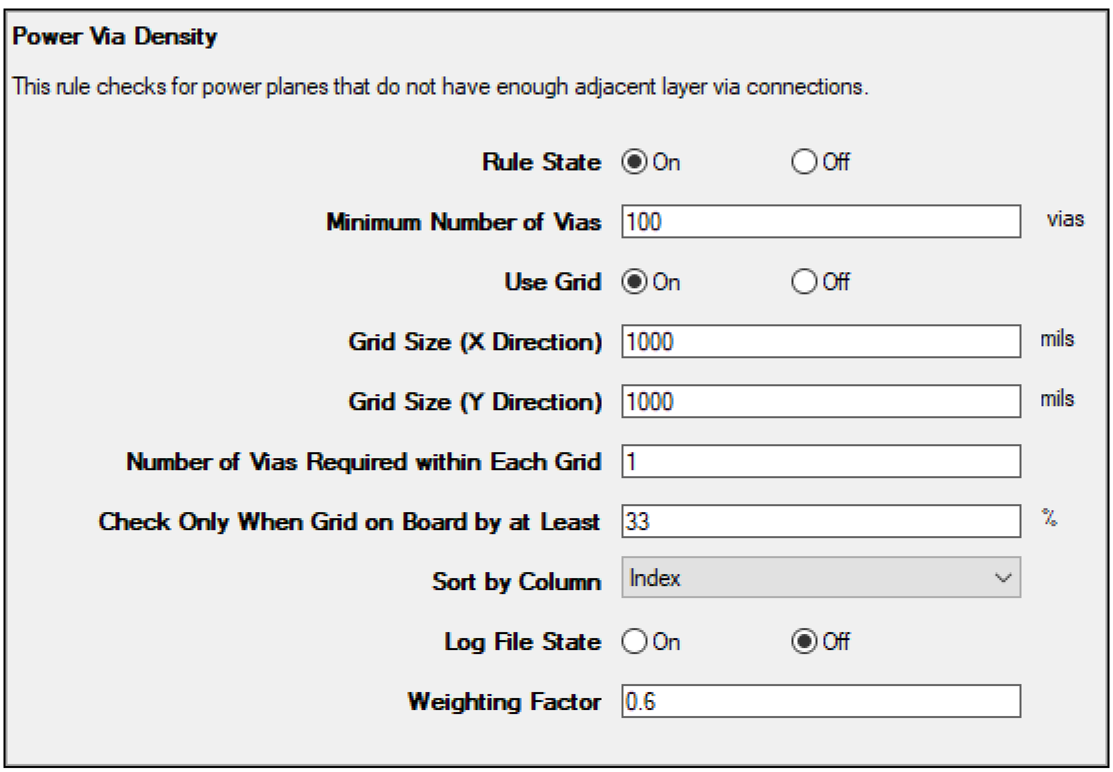


  This rule states that all power and ground-reference traces longer than a specified length must have a decoupling capacitor within a specified distance of the IC power pin.

  From here, set the following:

  - **Rule State** – toggles Power/Ground Trace Decoupling rules on or off.
  - **Maximum Ignorable Power/Ground Trace Length** – specifies the maximum power/ground trace length that may be ignored. All traces longer than this length must meet this rule's requirement.
  - **Max Distance between Cap and IC Power Pin** – specifies the maximum distance between the capacitor pin and the power or ground-reference pin.
  - **Sort by Column** – specifies the column by which results are sorted.
  - **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
  - **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See: Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
  - **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Power/Via Density** – displays **Power Via Density** rule configuration.

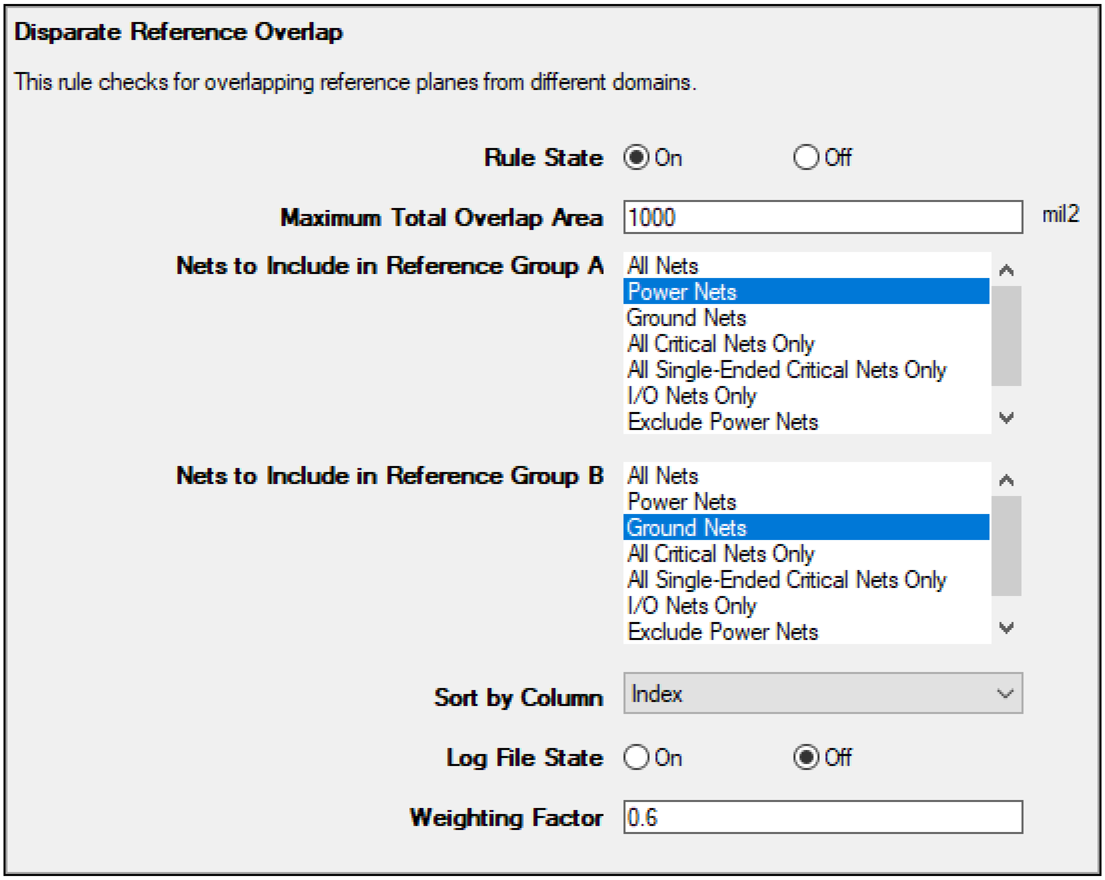


This rule checks for power planes that do not have enough adjacent layer via connections.

From here, you can set the following:

- **Rule State** – toggles Power Via Density rules on or off.
- **Minimum Number of Vias** – sets the minimum number of vias required to connect adjacent power planes.
- **Use Grid** – toggles the use of a grid on or off.
- **Grid Size (X Direction/Y Direction)** – when Use Grid is enabled, these fields specify the size of the grid.
- **Number of Vias Required within Each Grid** – when Use Grid is enabled, specifies the number of vias required within each grid.
- **Check Only When Grid on Board by at Least** – specifies the percentage of the grid that must be on the board for analysis to occur.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See: Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

- **Reference Overlap** – displays **Disparate Reference Overlap** rule configuration.



The Disparate Reference Overlap rule checks for overlapping reference planes from different domains.

From here, you can set the following:

- **Rule State** – toggles Disparate Reference Overlap on or off.
- **Maximum Total Overlap Area** – specifies the maximum overlap area that is acceptable. Overlap over this maximum will report a violation.
- **Nets to Include in Reference Group A** – selects a net type.
- **Nets to Include in Reference Group B** – selects a net type.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of

badness" multiplied by the weighting factor. See: Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

>

3. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

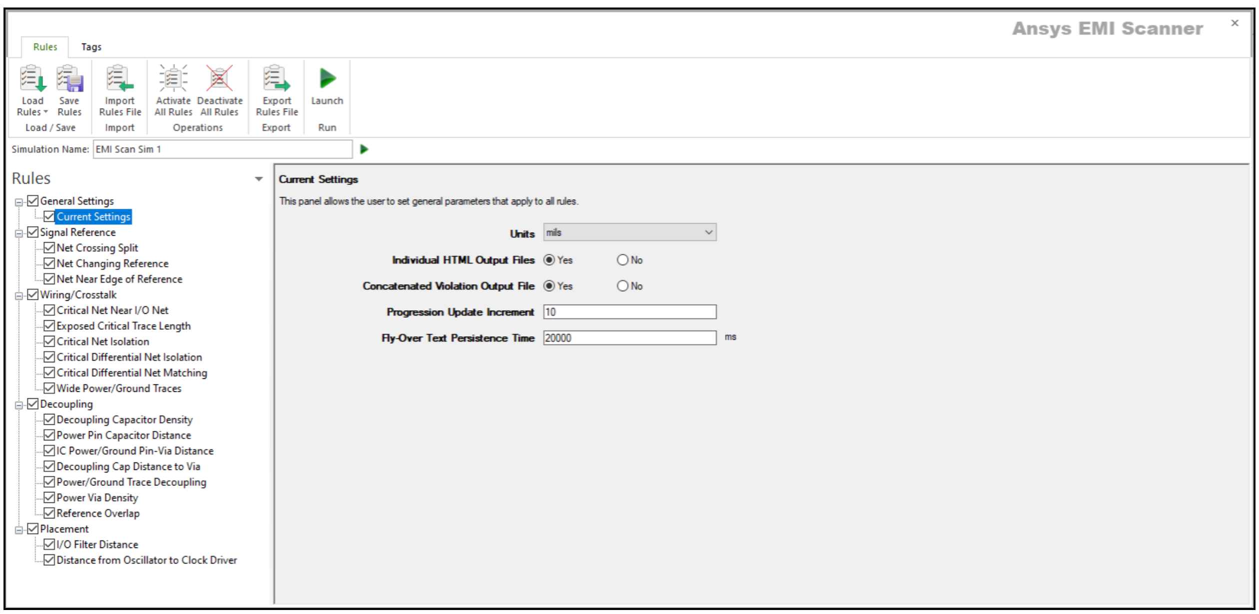
> **Note:**
>
> - You can display a description of a rule parameter as with the cursor over it.
> - Units of measure are set in General Settings.

Applying Placement Rules

To change EMI Scanner Placement rules:

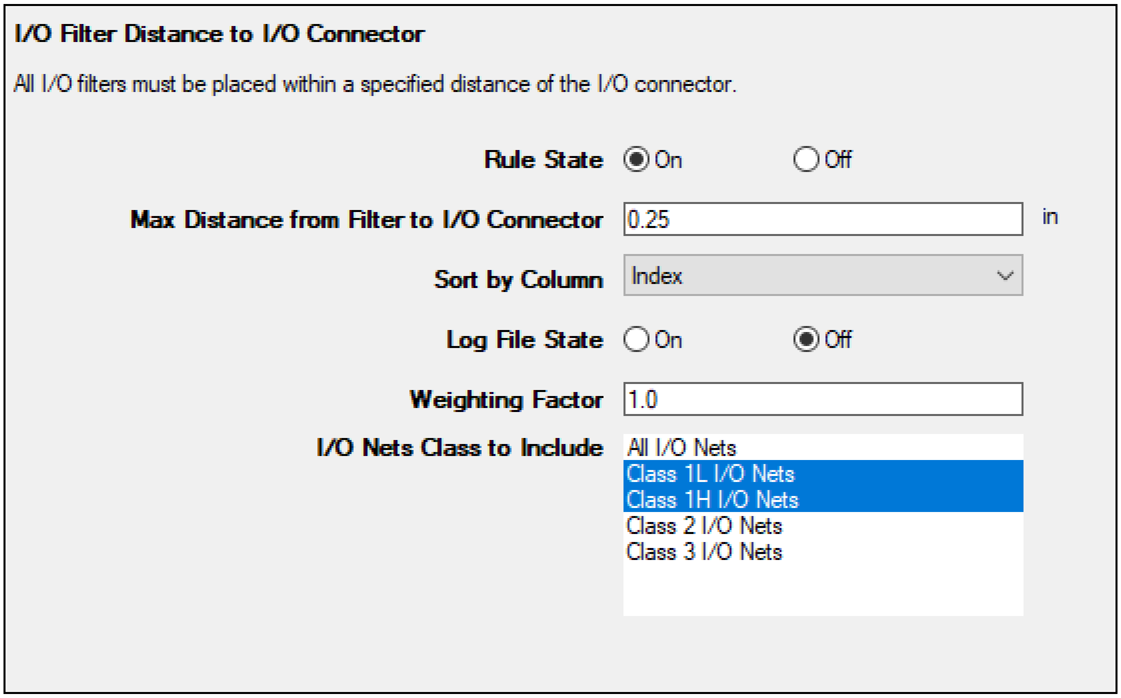
1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



2. From the Rules pane, select one of the two subcategories under **Placement**:

   - **I/O Filter Distance** – displays the **I/O Filter Distance to I/O Connector** rule configuration.

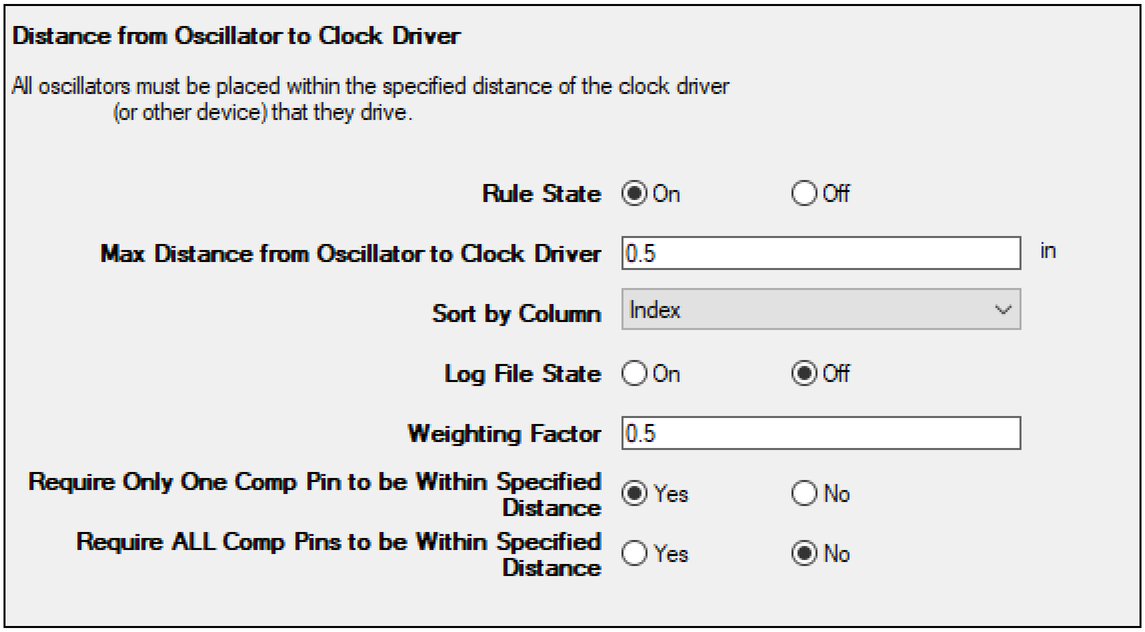


This rule states that all I/O filters must be placed within a specified distance of the I/O connector.

From here, set the following:

- **Rule State** – toggles I/O Filter Distance rules on or off.
- **Max Distance from Filter to I/O Connector** – specifies the maximum distance a filter may be placed on the I/O connector.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **I/O Nets to Include** – specifies the type(s) of I/O Nets to include. Select one, or **Ctrl**+click to select multiple types. See Technical Note.

- **Distance from Oscillator to Clock Driver** – displays the **Distance from Oscillator to Clock Driver** rule configuration.



This rule states that all oscillators must be placed within a specified distance on the clock driver or other device they drive.

From here, set the following:

- **Rule State** – toggles Distance from Oscillator to Clock Driver rules on or off.
- **Max Distance from Oscillator to Clock Driver** – specifies the distance within which all oscillators must be placed of the clock driver (or other device) they drive.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Require Only One Comp Pin to be Within Specified Distance** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
- **Require ALL Comp Pins to be Within Specified Distance** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.

3. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

Note:

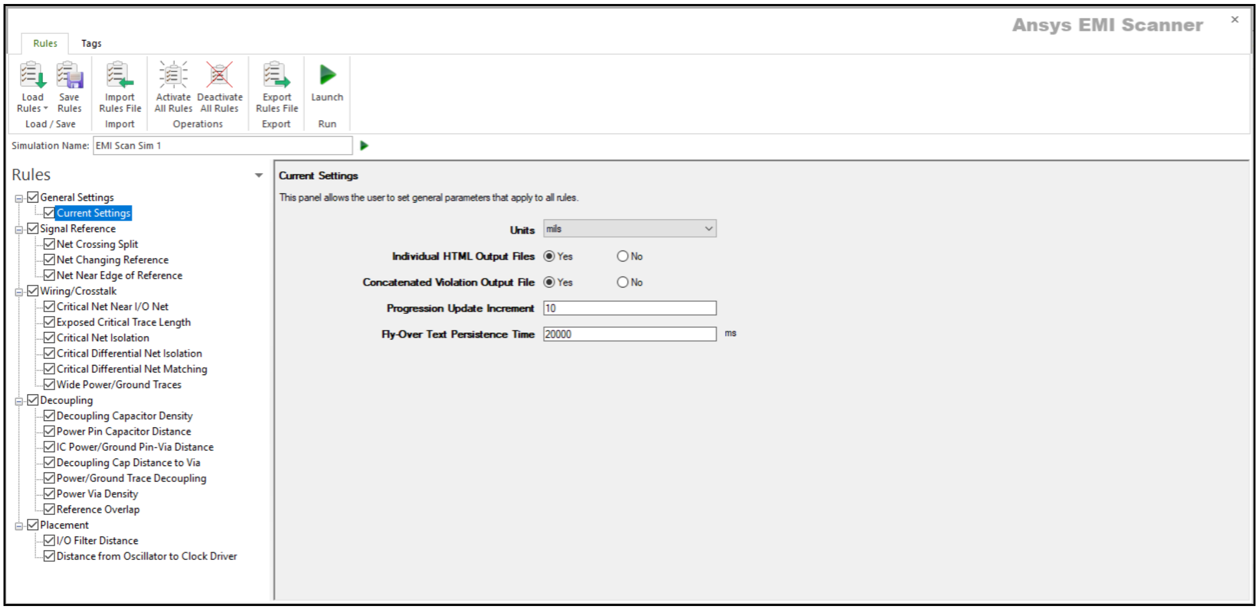
- You can display a description of a rule parameter at any time as with the cursor over it.
- Units of measure are set in General Settings.

Applying Net Integrity Rules

To change EMI Scanner Net Integrity rules:

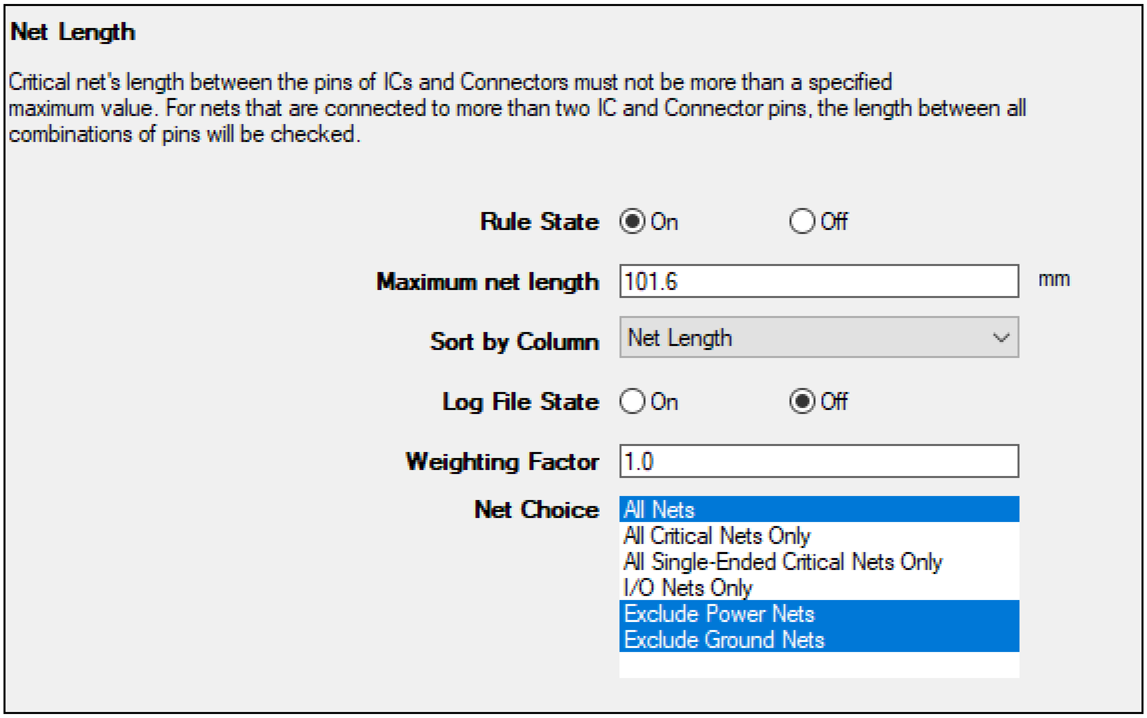
1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.

2. Click **Load Rules** and select a profile containing Signal Integrity rules (e.g., Default SI Rules Profile).

3. From the Rules pane, select one of the eight subcategories under **Net Integrity**:
   - **Net Length** – displays the **Net Length** rule configuration.

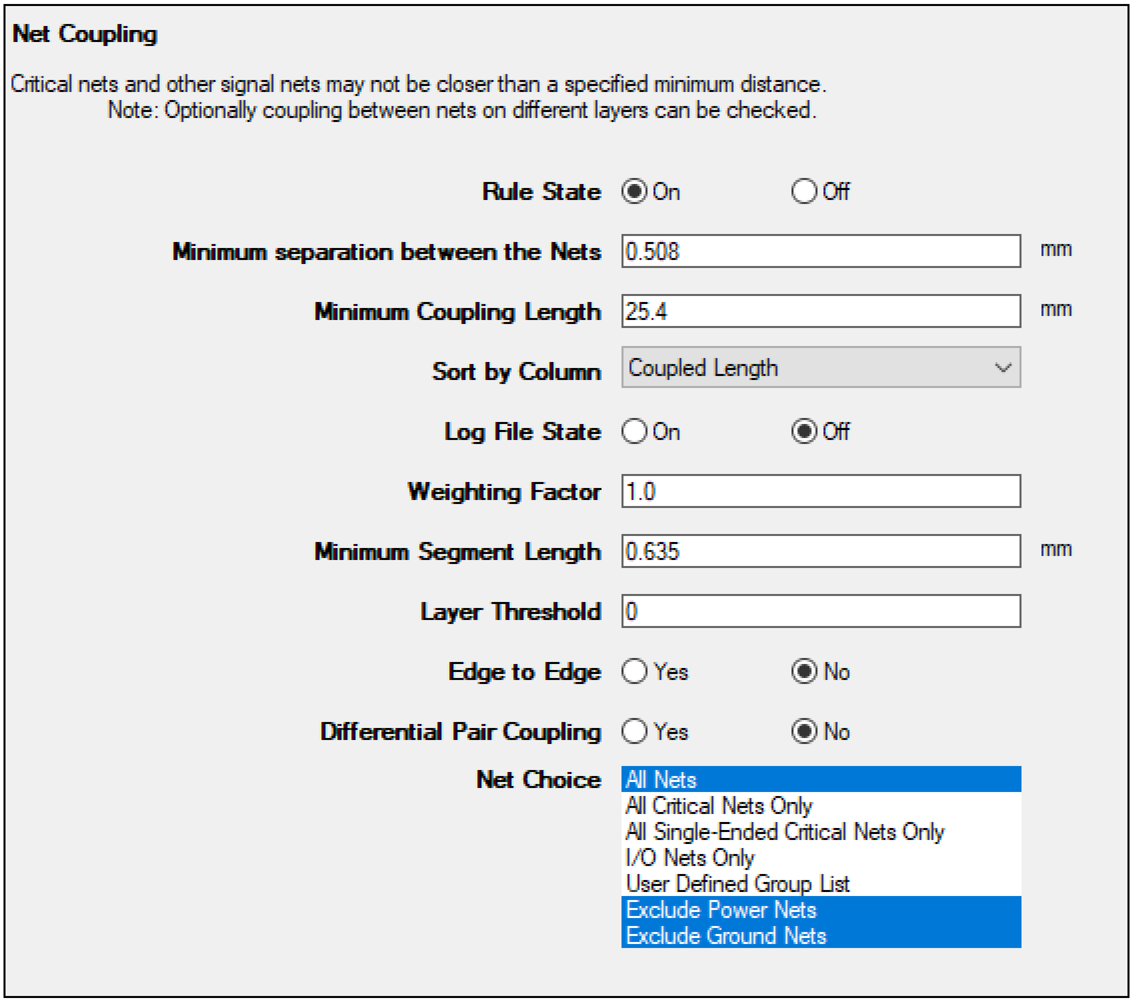


This rule states that a critical net's length between pins of ICs and Connectors must not be more than a specified value.

From here, you can set the following:

- **Rule State** – toggles Net Length rules on or off.
- **Maximum Net Length** – specifies the maximum net length.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Net to Net Coupling** – displays the **Net Coupling** rule configuration.



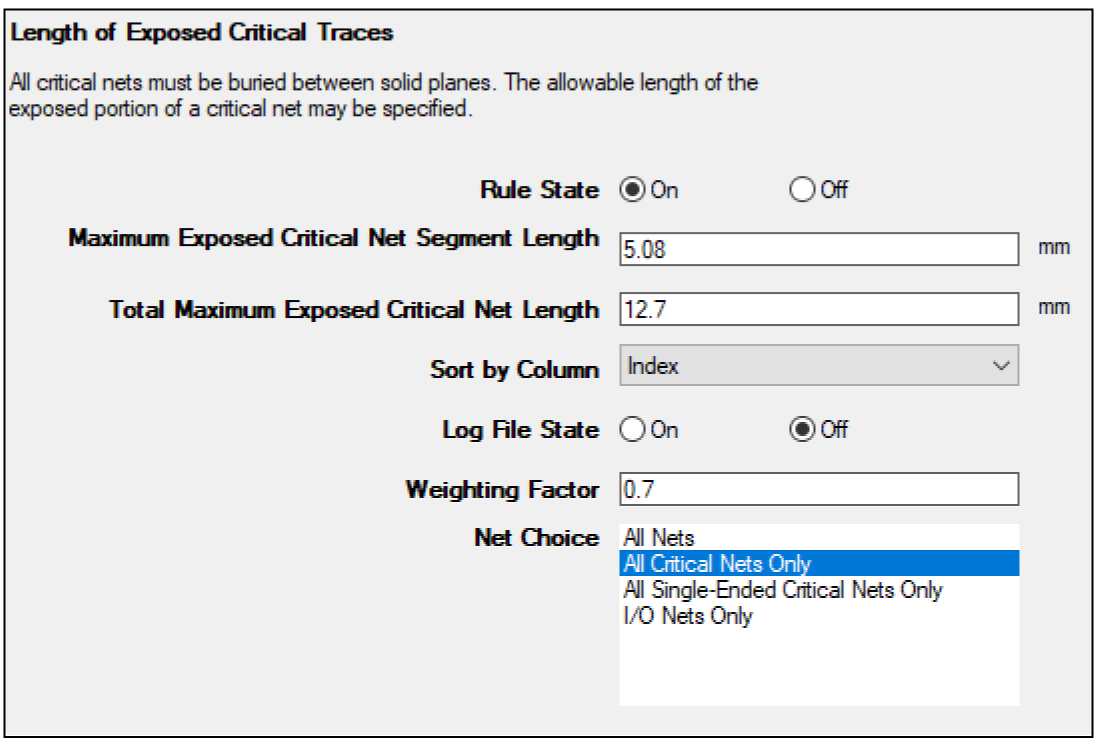
This rule states that critical nets and other signal nets may not be closer than a specified minimum distance.

From here, you can set the following:

- **Rule State** – toggles Net Coupling rules on or off.
- **Minimum Separation Between the Nets** – controls how close the nets can be before they are considered coupled.
- **Maximum Coupling Length** – specifies how much coupling must exist before a violation is reported.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).

  - **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
  - **Minimum Segment Length** – specifies the smallest net segment length to check for coupling. Any segments smaller than this length will be ignored.
  - **Layer Threshold** – sets the number of layers above and below to check for coupling (e.g., 0 = only the selected layer, 1 = the selected layer plus one layer above and one layer below).
  - **Edge to Edge** – by default, the minimum distance between nets is based on center-to-center measurement. Selecting this option changes the measurement to edge-to-edge.
  - **Differential Pair Coupling** – determines whether to allow or ignore coupling between the nets that make up a differential pair.
  - **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Exposed Critical Trace Length** – displays the **Length of Exposed Critical Traces** rule configuration.

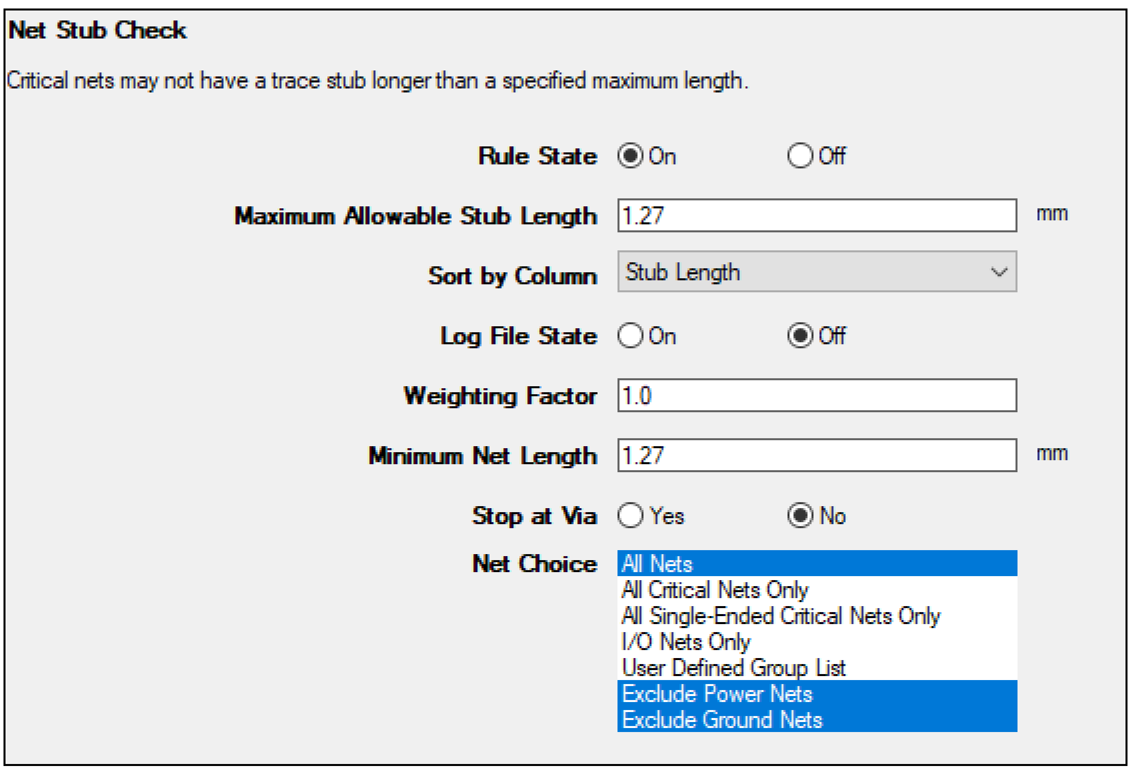


This rule states that all critical nets must be buried between solid planes.

From here, you can set the following:

  - **Rule State** – toggles Net Length rules on or off.
  - **Maximum Exposed Critical Net Segment Length** – specifies the maximum critical net segment length allowed on external (exposed) layers.
  - **Total Maximum Exposed Critical Net Length** – specifies the total maximum critical net length allowed on external (exposed) layers (e.g., multiple segments on a critical net may be exposed, but short enough to not cause a violation individually).
  - **Sort by Column** – specifies the column by which results are sorted.
  - **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
  - **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
  - **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Net Stub** – displays the **Net Stub Check** rule configuration.

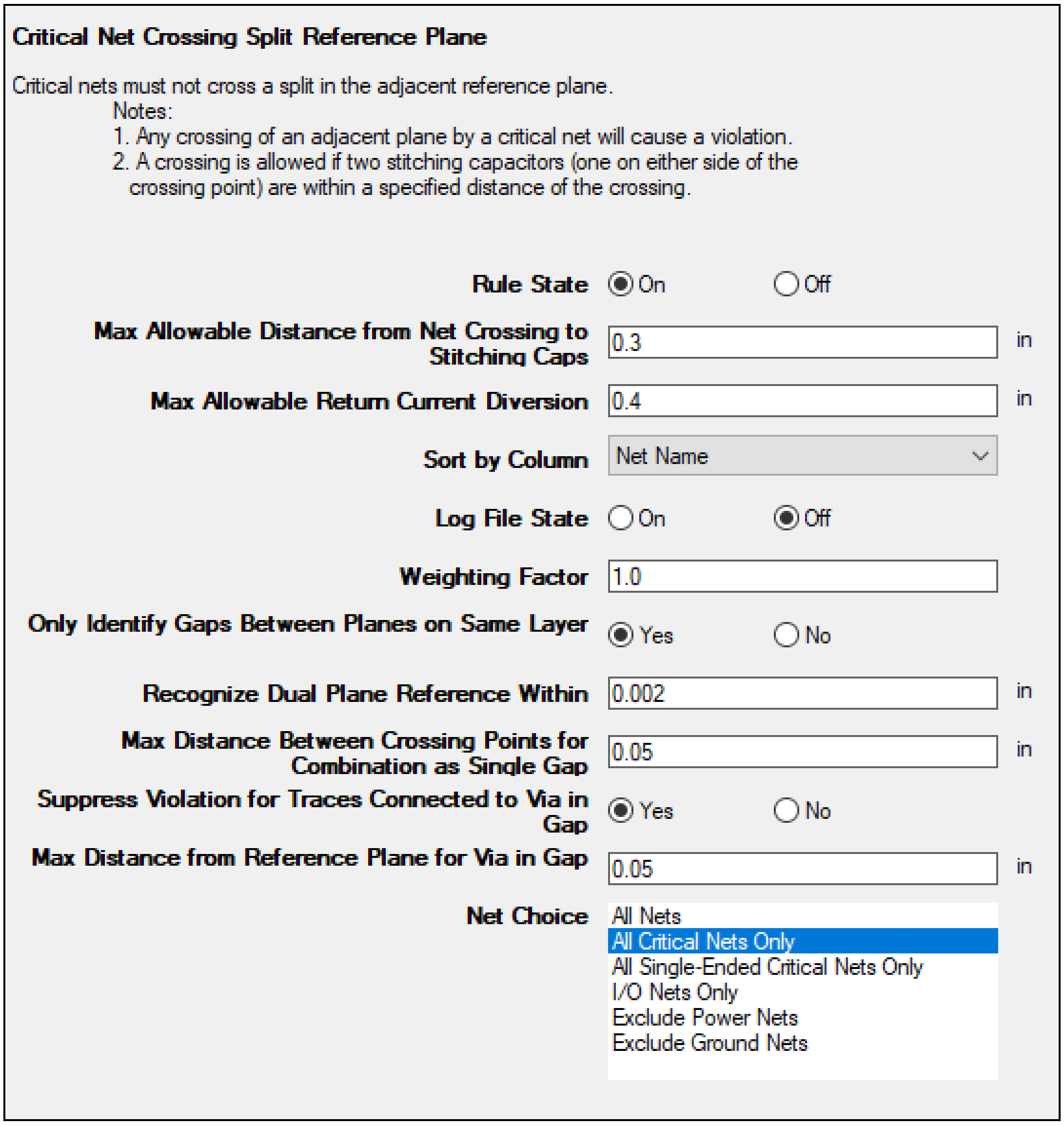


This rule states that critical nets may not have a trace stub longer than the specified maximum length.

From here, you can set the following:

- **Rule State** – toggles Net Stub Check rules on or off.
- **Maximum Allowable Stub Length** – specifies the maximum stub length. Stubs longer than this are violations.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Minimum Net Length** – sets the minimum length at which nets are included. Nets must be longer than this length to be considered.
- **Stop at Via** – determines whether vias count as a stub termination condition. By defauly, the termination of a stub is a pin or a fork in the net, whichever is found first. Setting this parameter to 'yes' adds vias (to pins and forks) as a stub termination condition.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Net Crossing Split** – displays the **Critical Net Crossing Split Reference Plane** rule configuration.



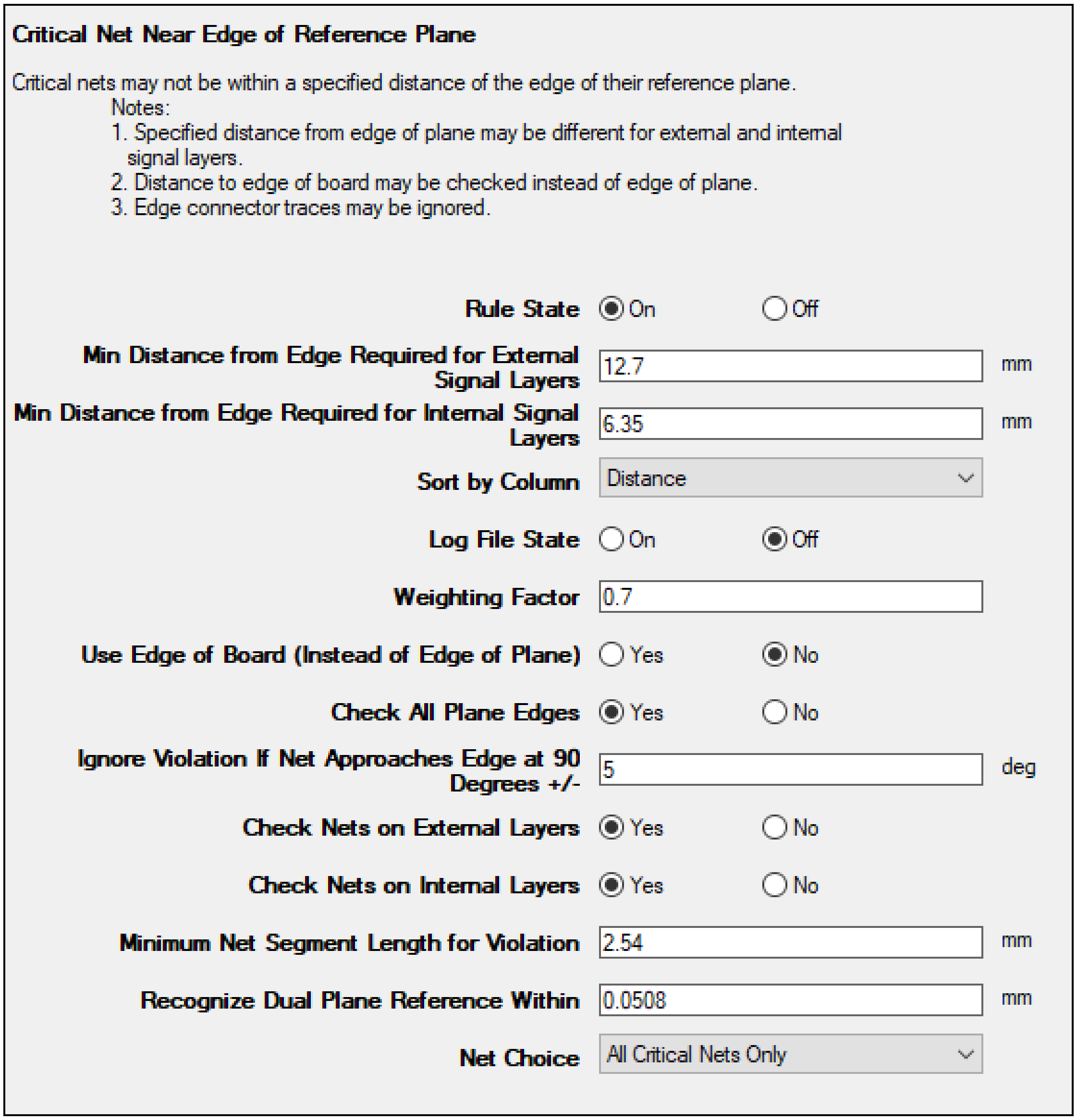
This rule states that critical nets must not cross a split in the adjacent reference plane.

From here, you can set the following:

- **Rule State** – toggles Net Crossing Split rules on or off.
- **Max Allowable Distance from Net Crossing to Stitching Caps** – sets the distance on the point where the net crosses a split in its reference plane at which two stitching capacitors must be found.
- **Max Allowable Return Current Diversion** – determines whether a signal's return current can divert itself around the gap in the reference plane. The total distance of the diversion must be less than this value.

- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Only Identify Gaps Between Planes on Same Layer** – chooses whether or not gaps are only between planes on the same layer. When the rule tries to pair up nearby plane boundary crossing points and identify gaps, this parameter will enable or disable a check for layer commonality.
- **Recognize Dual Plane Reference Within** – specifies a distance and instructs EMI Scanner to consider both planes as a reference when checking for gap crossing when the distance difference to each plane is less than the specified distance. This allows recognition of both planes as references for strip line traces when the trace is offset on the exact center by a small amount (normally for manufacturing reasons).
- **Max Distance Between Crossing Points for Combination as a Single Gap** – specifies the distance under which nearby crossing points are combined into a single violation. Setting too large a value can cause the algorithm to pair up unrelated crossing points, and setting too small a value can prevent crossing points from being paired up, resulting in more violations. It is recommended to set the value slightly larger than the typical plane separation within a layer, based on the board design technology being used.
- **Suppress Violation for Traces Connected to Via in Gap** – instructs EMI Scanner whether to ignore trace crossing split/gap when connected to a via within that gap.
- **Max Distance from Reference Plane for Via in Gap** – specifies the maximum distance from edge of the reference plane to the via. Violations will not be suppressed (from above) for vias within gap if the trace extends greater than the specified distance.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Net Near Edge of Reference** – displays the **Critical Net Near Edge of Reference Plane** rule configuration.



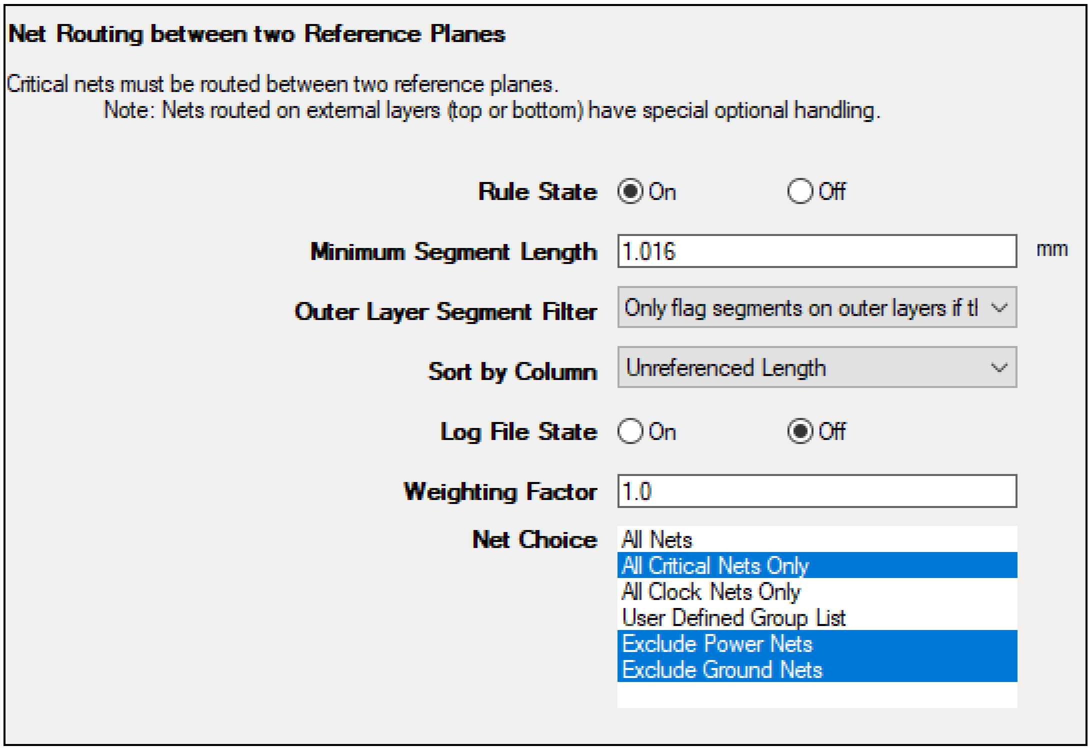
This rule states that critical nets may not be within a specified distance of the edge of their reference plane.

From here, you can set the following:

- **Rule State** – toggles Net Near Edge of Reference rules on or off.
- **Min Distance from Edge Required for External Signal Layers** – specifies the minimum distance that must exist between the edge of the reference plane and a critical signal net when the net is on an external layer.

- **Min Distance from Edge Required for Internal Signal Layers** – specifies the minimum distance that must exist between the edge of the reference plane and a critical signal net when the net is on an internal layer.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Use Edge of Board (Instead of Edge of Plane)** – allows EMI Scanner to check distance from critical nets to the edge of the board instead of the edge of the reference plane. This can be useful if the board is only partially complete and all reference planes are not yet completed.
- **Check All Plane Edges** – checks the distance from nets to any edge of their reference plane, rather than limiting the check to reference plane edges near the edge of the board.
- **Ignore Violation If Net Approaches Edge at 90 Degrees +/-** – EMI scanner ignores edge connectors that approach the edge of the reference plane or board at a 90 degree angle. This option specifies a tolerance that allows the approach to be slightly off a perfect 90 degrees.
- **Check Nets on External Layers** – specifies whether nets on external layers should be checked.
- **Check Nets on Internal Layers** – specifies whether nets on internal layers should be checked.
- **Minimum Net Segment Length for Violation** – specifies the minimum net segment length before EMI Scanner will report a violation if the net is in the specified distance on the edge of the reference plane (or board).
- **Recognize Dual Plane Reference Within** – specifies a distance and instructs EMI Scanner to consider both planes as a reference when checking for traces too close to the edge of reference planes when the distance difference to each plane is less than the specified distance. This allows recognition of both planes as references for strip line traces when the trace is offset on the exact center by a small amount (normally for manufacturing reasons).
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Between Ref Plane Routing** – displays the **Net Routing Between Two Reference Planes** rule configuration.

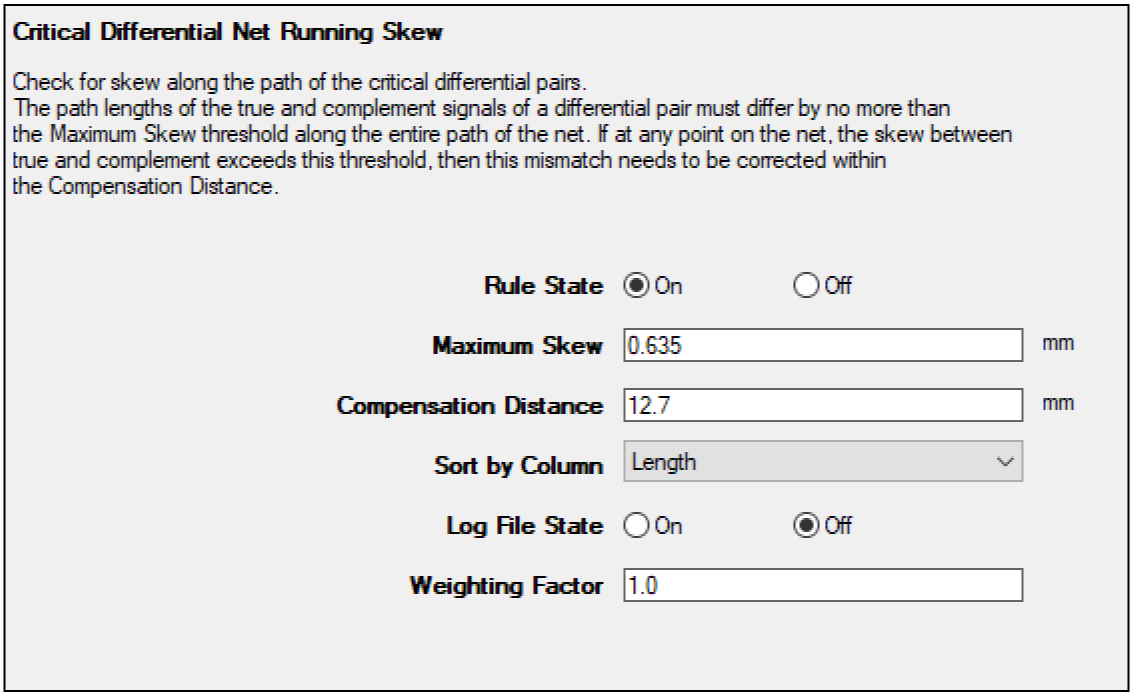


This rule states that critical nets must be routed between two reference planes.

From here, you can set the following:

- **Rule State** – toggles Net Routing Between Two Reference Planes rules on or off.
- **Minimum Segment Length** – specifies the minimum net segment length before EMI Scanner will report a violation.
- **Outer Layer Segment Filter** – specifies how segments on outer layers are to be treated.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Diff Running Skew** – displays the **Critical Differential Net Running Skew** rule configuration.



This rule checks for skew along the path of critical differential pairs.

From here, you can set the following:

- **Rule State** – toggles Diff Running Skew rules on or off.
- **Maximum Skew** – sets the maximum allowable skew.
- **Compensation Distance** – skew that exceeds the maximum allowable skew must be corrected within this distance.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

4. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

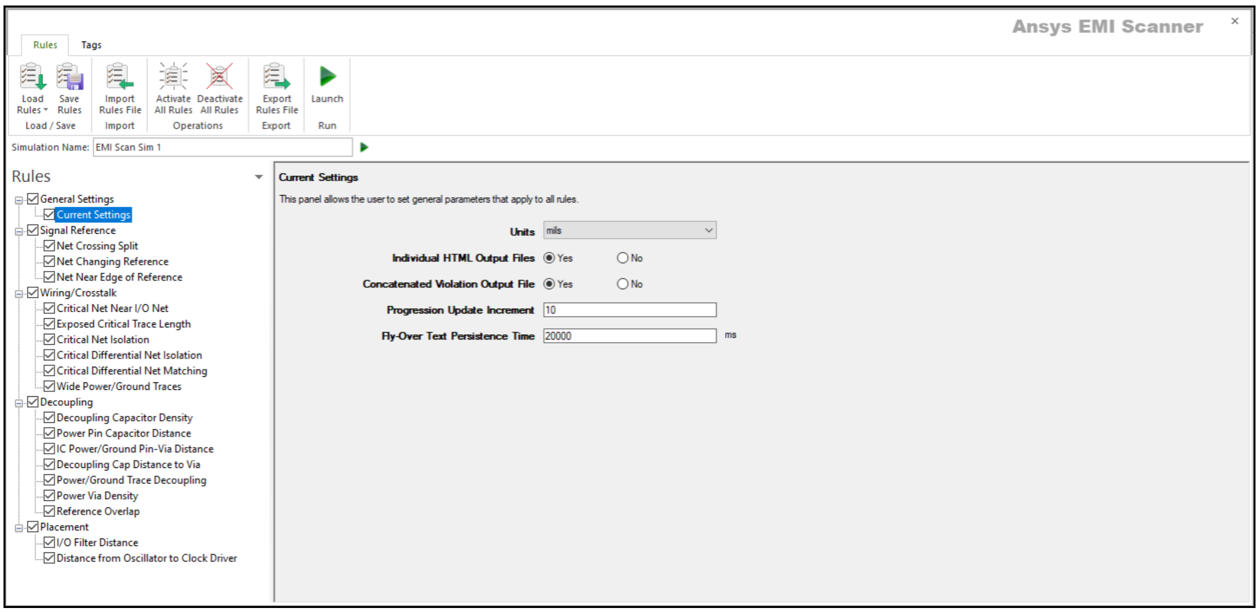
> **Note:**
>
> - You can display a description of a rule parameter at any time by hovering the cursor over it.
> - Units of measure are set in General Settings.

Applying Via Integrity Rules

To change EMI Scanner Via Integrity rules:

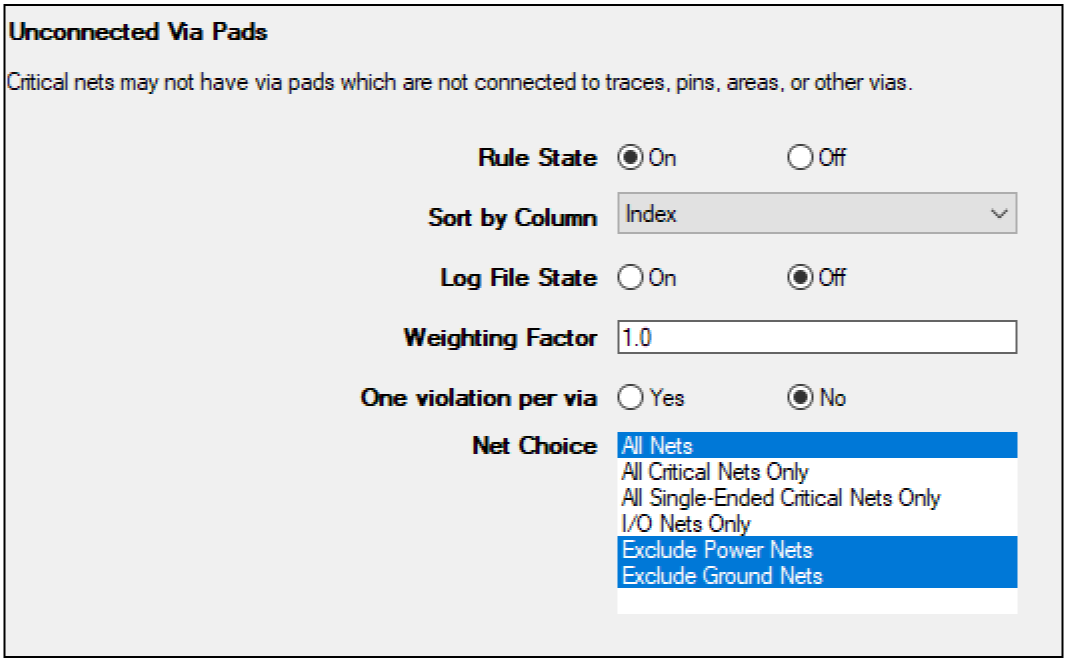
1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



2. Select a profile containing Via Integrity rules (e.g., Default SI Rules Profile) on the **Load Rules** drop-down menu.

3. From the Rules pane, select one of the four subcategories under **Via Integrity**:
   - **Unconnected Via Pads** – displays the **Unconnected Via Pads** rule configuration.

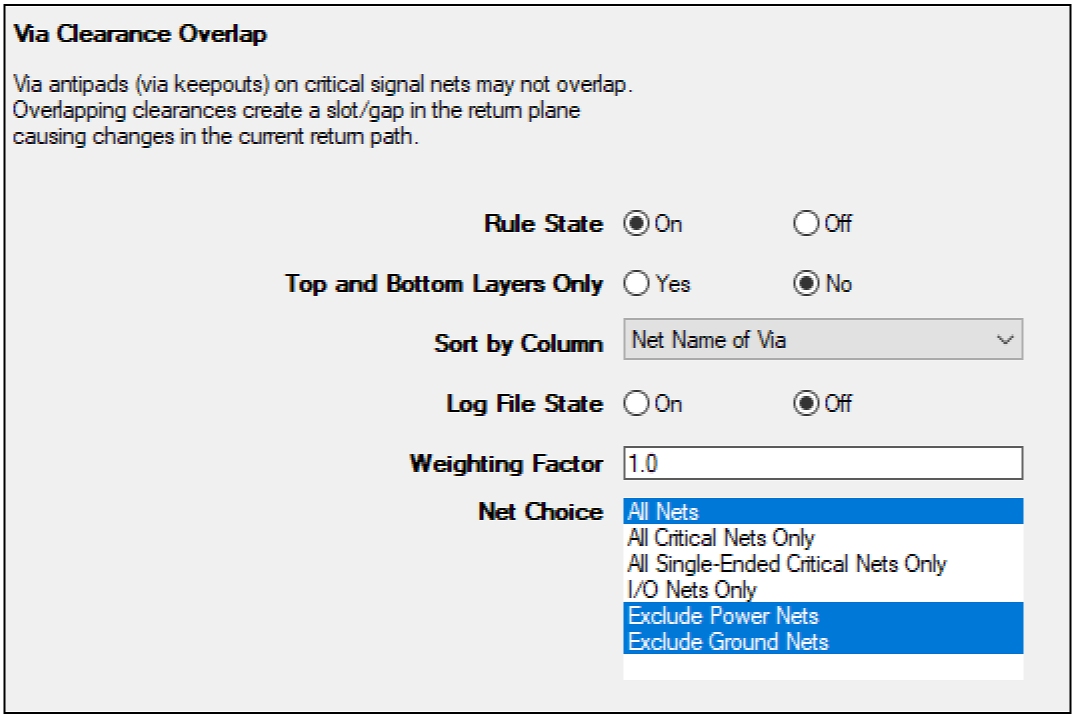


This rule states that critical nets may not have via pads that are not connected to traces, pins, areas, or other vias.

From here, set the following:

- **Rule State** – toggles Unconnected Via Pads rules on or off.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **One Violation Per Via** – when **Yes** is selected, multiple violations on a via are counted as one violation.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Via Clearance Overlap** – displays the **Via Clearance Overlap** rule configuration.

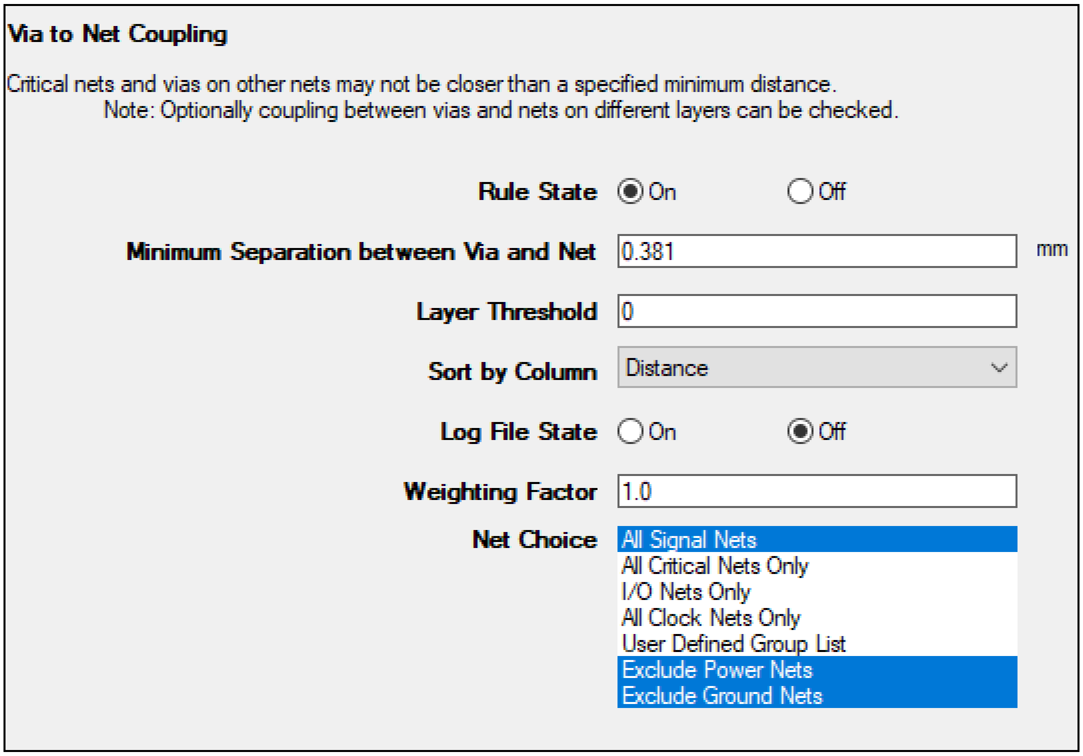


This rule states that via antipads on critical signal nets may not overlap.

From here, set the following:

- **Rule State** – toggles Via Clearance Overlap rules on or off.
- **Top and Bottom Layers Only** – limits analysis to the top and bottom layers only.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Via to Net Coupling** – displays the **Via to Net Coupling** rule configuration.



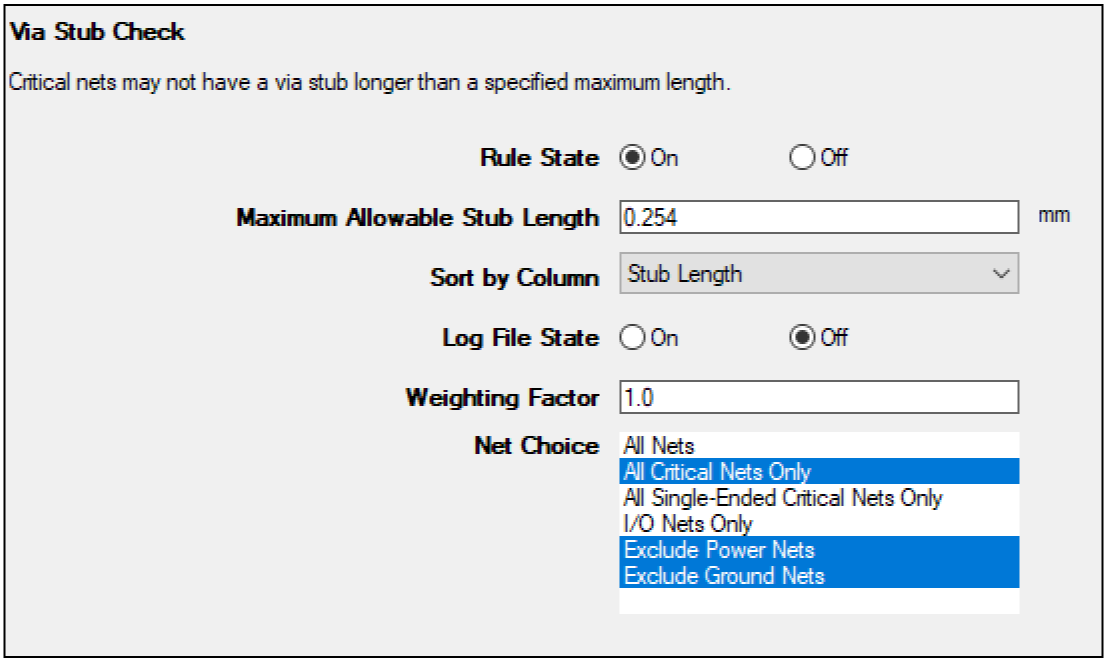
This rule states that critical nets and vias on other nets may not be closer than a specified minimum distance.

From here, set the following:

- **Rule State** – toggles Via to Net Coupling rules on or off.
- **Minimum Separation Between Via and Net** – controls how close the net and the via can be before they are considered coupled.
- **Layer Threshold** – sets the number of layers above and below to check for coupling (e.g., 0 = only the selected layer, 1 = the selected layer plus one layer above and one layer below).
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

  - **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Via Stub** – displays the **Via Stub Check** rule configuration.



  This rule states that critical nets may not have a via stub longer than a specified maximum length.

  From here, set the following:

  - **Rule State** – toggles Via Stub rules on or off.
  - **Maximum Allowable Stub Length** – specifies the maximum stub length. Any stub longer than this setting is a violation.
  - **Sort by Column** – specifies the column by which results are sorted.
  - **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
  - **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
  - **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

4. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

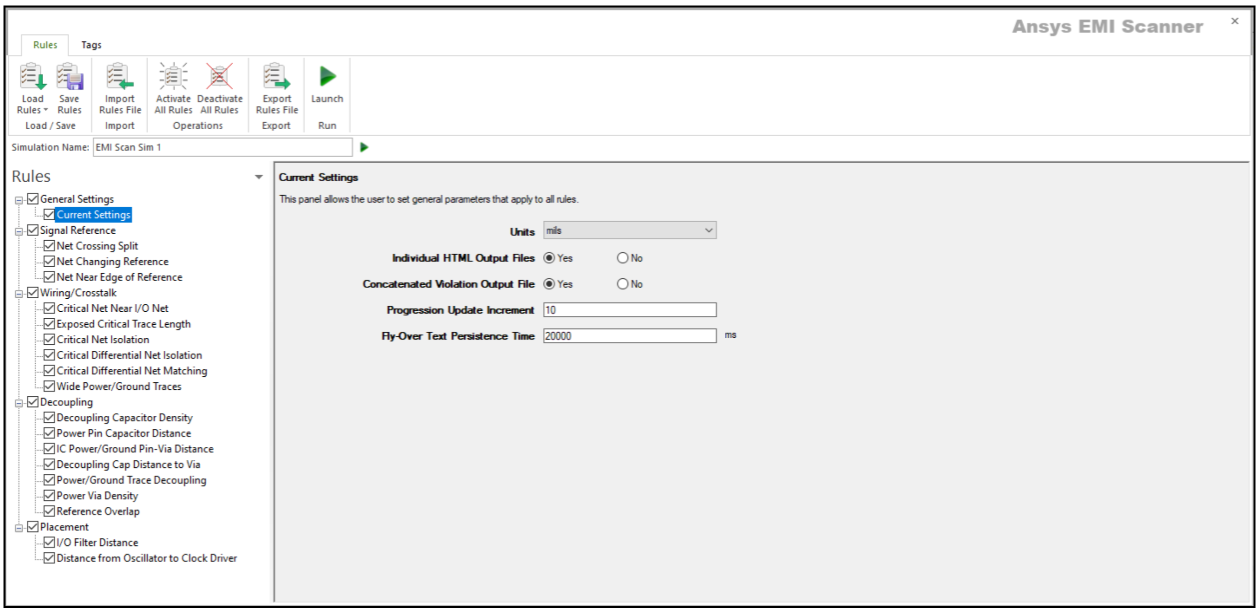
> **Note:**
>
> - You can display a description of a rule parameter at any time as with the cursor over it.
> - Units of measure are set in General Settings.

Applying Power Integrity Rules

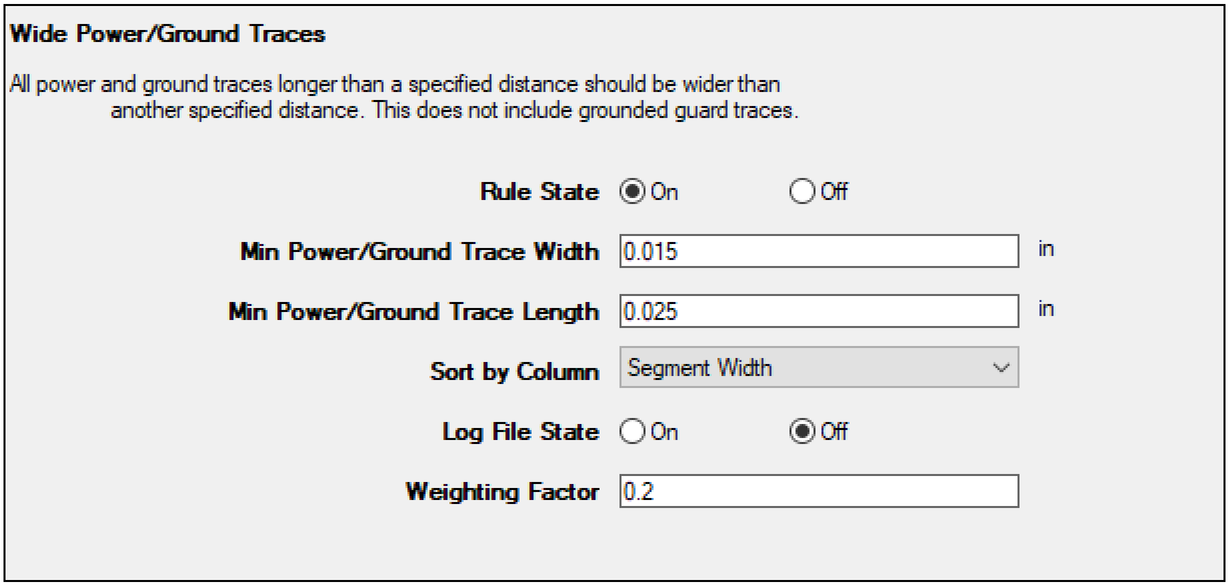
To change EMI Scanner Power Integrity rules:

1. Launch EMI Scanner.

   The **EMI Scanner** window opens, on the **Rules** tab.



2. Click **Load Rules** and select a profile containing Power Integrity rules (e.g., Default SI Rules Profile).
3. From the Rules pane, select one of the seven subcategories under **Power Integrity**:
   - **Wide Power/Ground Traces** – displays the **Wide Power/Ground Traces** rule configuration.

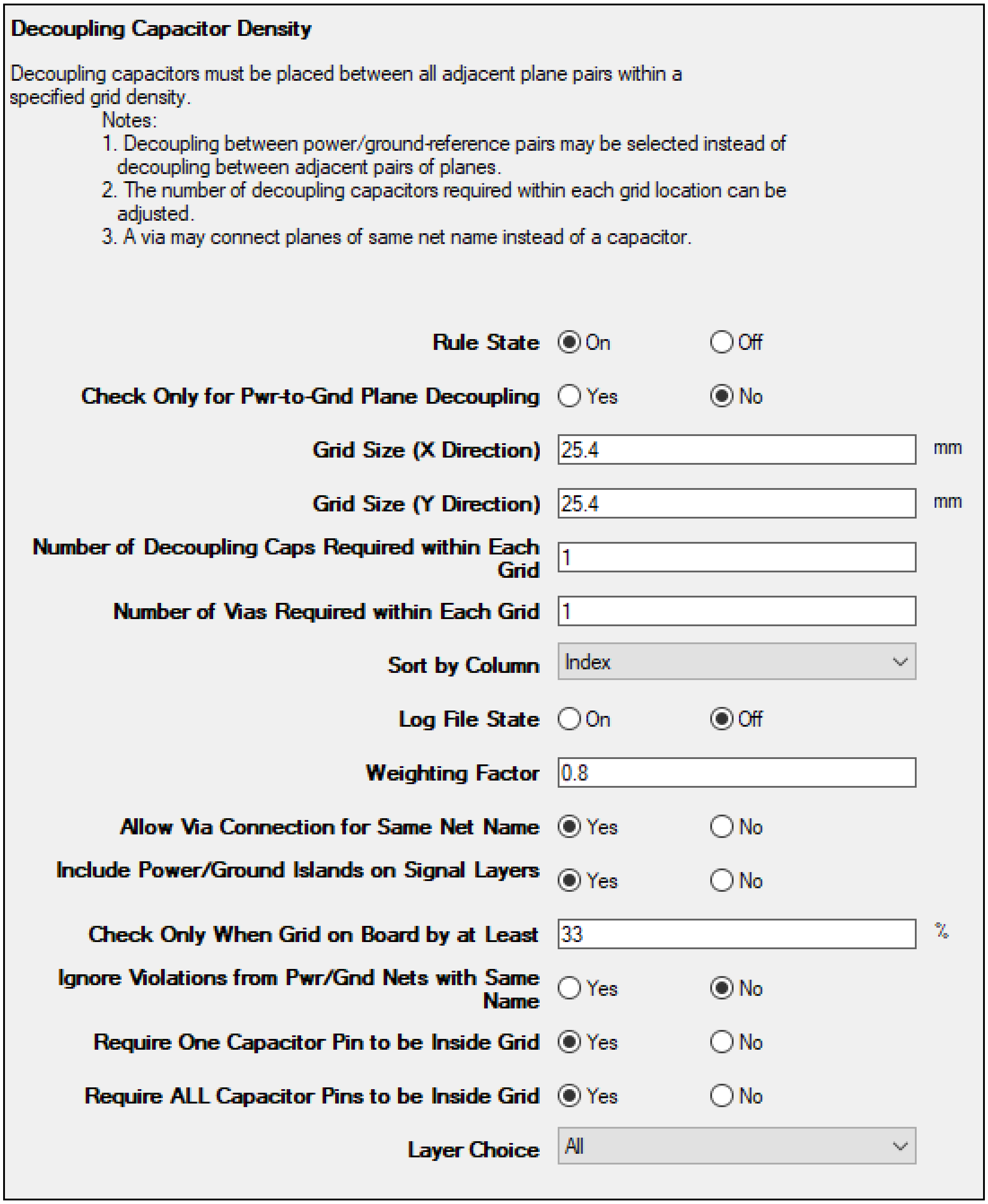


This rule states that all power and ground traces longer than a specified distance should be wider than another specified distance.

From here, set the following:

- **Rule State** – toggles Wide Power/Ground Traces rules on or off.
- **Min Power/Ground Trace Width** – sets the minimum width limit for all power/ground traces (except grounded guard traces).
- **Min Power/Ground Trace Length** – sets the minimum length limit for all power/ground traces (except grounded guard traces).
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

- **Decoupling Capacitor Density** – displays the **Decoupling Capacitor Density** rule configuration.



Decoupling Capacitor Density

Decoupling capacitors must be placed between all adjacent plane pairs within a specified grid density.

Notes:

1. Decoupling between power/ground-reference pairs may be selected instead of decoupling between adjacent pairs of planes.
2. The number of decoupling capacitors required within each grid location can be adjusted.
3. A via may connect planes of same net name instead of a capacitor.

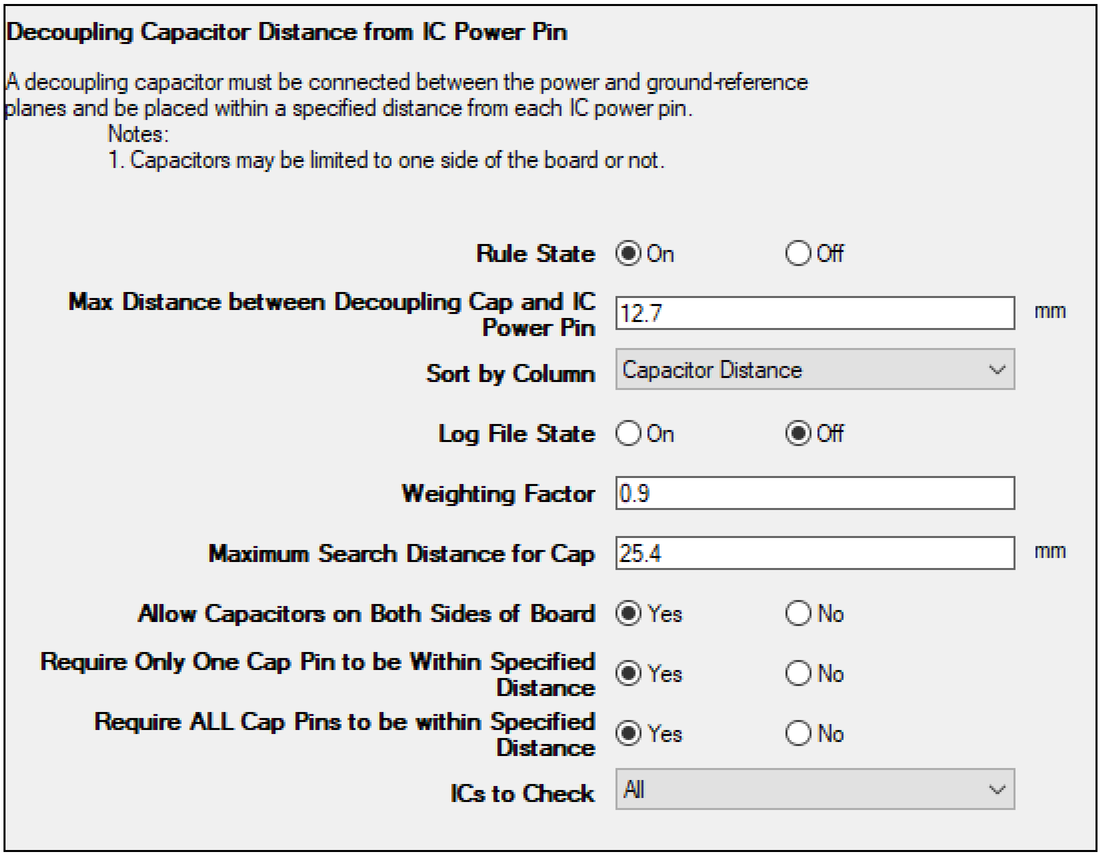
Rule State	(•) On	( ) Off	
Check Only for Pwr-to-Gnd Plane Decoupling	( ) Yes	(•) No	
Grid Size (X Direction)	25.4		mm
Grid Size (Y Direction)	25.4		mm
Number of Decoupling Caps Required within Each Grid	1		
Number of Vias Required within Each Grid	1		
Sort by Column	Index		
Log File State	( ) On	(•) Off	
Weighting Factor	0.8		
Allow Via Connection for Same Net Name	(•) Yes	( ) No	
Include Power/Ground Islands on Signal Layers	(•) Yes	( ) No	
Check Only When Grid on Board by at Least	33		%
Ignore Violations from Pwr/Gnd Nets with Same Name	( ) Yes	(•) No	
Require One Capacitor Pin to be Inside Grid	(•) Yes	( ) No	
Require ALL Capacitor Pins to be Inside Grid	(•) Yes	( ) No	
Layer Choice	All		

This rule states that decoupling capacitors must be placed between all adjacent plane pairs within a specified grid density.

From here, set the following:

- **Rule State** – toggles Decoupling Capacitor Density rules on or off.
- **Check Only for Pwr-to-Gnd Plane Decoupling** – sets whether to ignore plane-to-plane decoupling and check only for decoupling capacitors between power and ground planes.
- **Grid Size (X Direction)** – specifies the size of the grid used to check for decoupling capacitors. Used in conjunction with **Grid Size (Y Direction)**.
- **Grid Size (Y Direction)** – specifies the size of the grid used to check for decoupling capacitors. Used in conjunction with **Grid Size (X Direction)**.
- **Number of Decoupling Caps Required within Each Grid** – specifies the number of decoupling capacitors connecting between planes that are required within each grid location.
- **Number of Vias Required within Each Grid** – specifies the number of vias connecting between planes that are required within each grid location. You must select **Allow Via Connection for Same Net Name** to use this option.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Allow Via Connection for Same Net Name** – determines whether EMI Scanner allows a via in place of a decoupling capacitor when the two planes are the same net name.
- **Include Power/Ground Islands on Signal Layers** – specifies whether to check for decoupling on power/ground islands.
- **Check Only When Grid on Board by at Least** – specifies the percentage of a grid that must be on the board in order for it to be checked.
- **Ignore Violations from Pwr/Gnd Nets with Same Name** – determines that only one plane of a given name is checked for violations (e.g., if two planes are "gnd" and a violation occurs between gnd and another layer, only one violation is reported).
- **Require One Capacitor Pin to be Inside Grid** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
- **Require ALL Capacitor Pins to be Inside Grid** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
- **Layer Choice** – specifies the layers to include in this rule analysis.

- **Power Pin Capacitor Distance** – displays the **Decoupling Capacitor Distance from IC Power Pin** rule configuration.

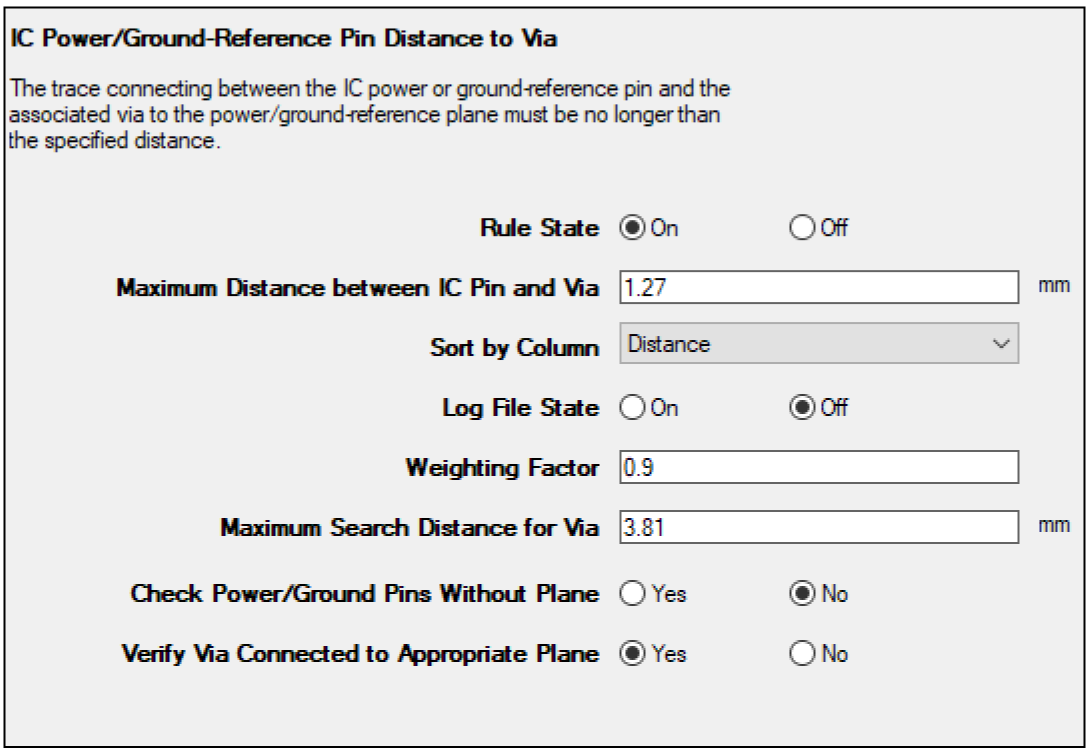


This rule states that a decoupling capacitor must be connected between the power and ground-reference planes and be placed within a specified distance of each IC power pin.

From here, set the following:

- **Rule State** – toggles Power Pin Capacitor Distance rules on or off.
- **Max Distance Between Decoupling Cap and IC Power Pin** – specifies the maximum distance between an IC power pin and its decoupling capacitor.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

   - **Maximum Search Distance for Cap** – specifies the maximum search distance for a capacitor. A larger search box increases the chances of EMI Scanner finding capacitors, but can slow processing time.
   - **Allow Capacitors on Both Sides of Board** – allows EMI Scanner to include capacitors on both sides of the board to satisfy this rule.
   - **Require Only One Cap Pin to be within Specified Distance** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
   - **Require ALL Cap Pins to be within Specified Distance** – changes how EMI Scanner determines whether a component is inside or outside the grid. See Technical Note.
   - **ICs to Check** – specifies the types of components to include in this rule analysis.
- **IC Power/Ground Pin-Via Distance** – displays the **IC Power/Ground-Reference Pin Distance to Via** rule configuration.

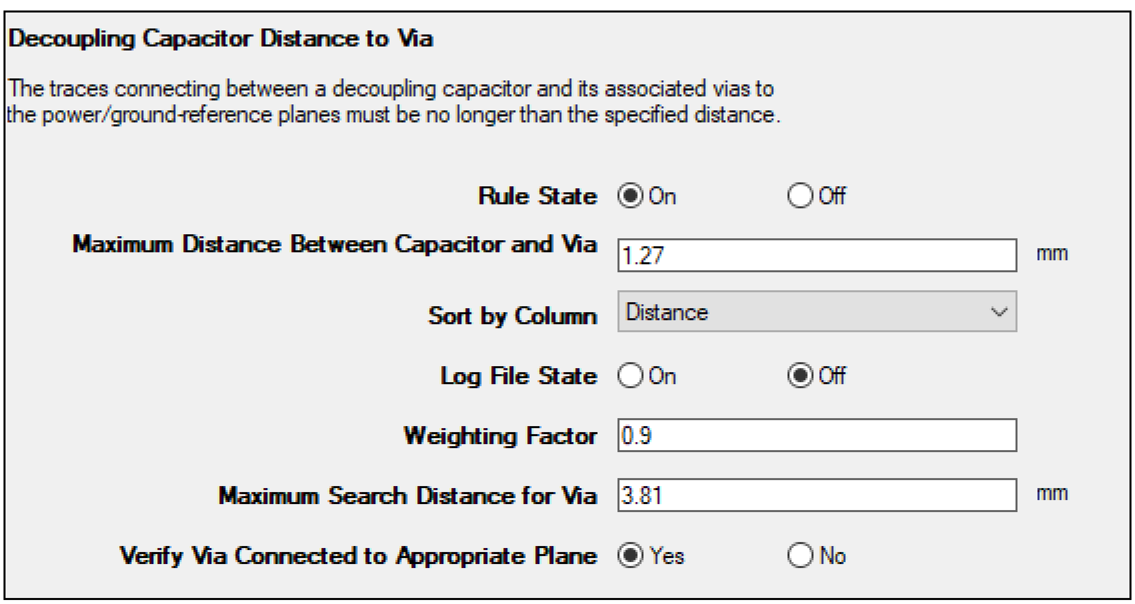


This rule states the trace connecting the IC power or ground-reference pin and the associated via to the ground/reference plane must be no longer than a specified distance..

From here, set the following:

- **Rule State** – toggles IC Power/Ground Pin-Via Distance rules on or off.
- **Maximum Distance Between IC Pin and Via** – specifies the maximum distance between the IC power or ground-reference pin and the via to the power or ground-reference plane.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Maximum Search Distance for Via** – specifies the maximum search distance for a via when a violation is found. This allows users to quickly determine if the via is barely outside the original requirement and therefore not a serious violation.
- **Check Power/Ground Pins Without Plane** – specifies whether to check the distance between the IC power/ground pin and its via when the power/ground plane has not been completed yet.
- **Verify Via Connected to Appropriate Plane** – specifies whether or not EMI Scanner verifies the via connects to the chosen plane (shape). Selecting 'no' speeds up processing.

- **Decoupling Cap Distance to Via** – displays the **Decoupling Capacitor Distance to Via** rule configuration.

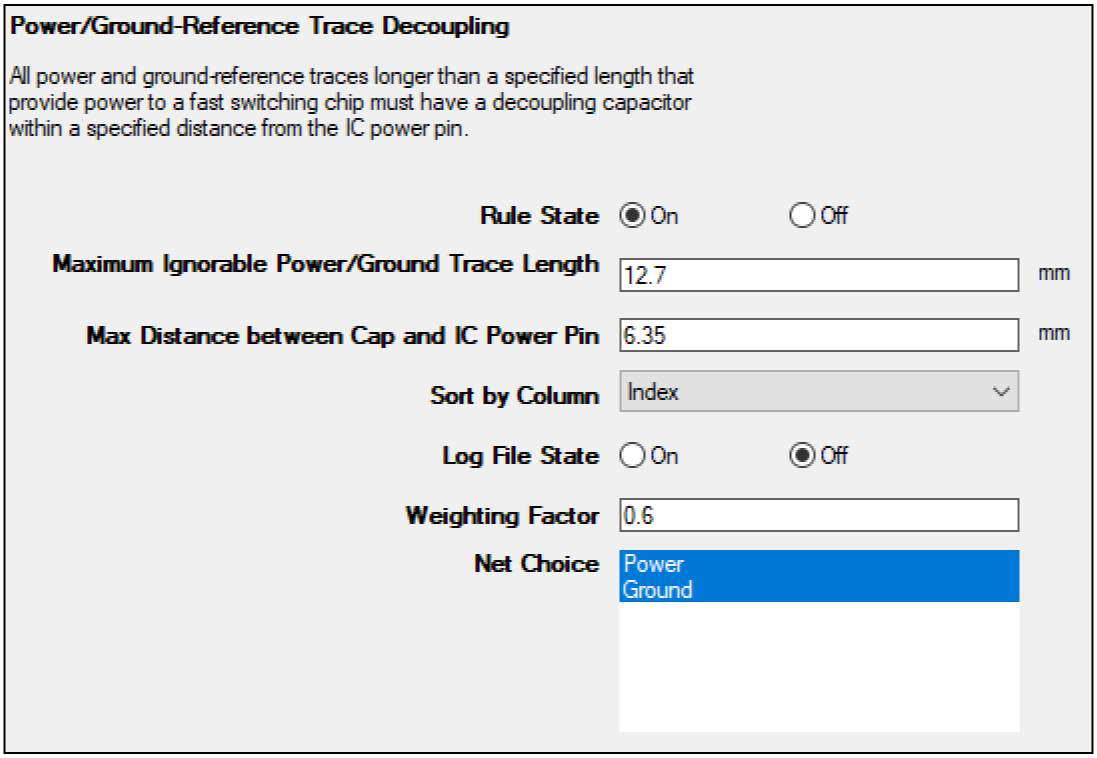


This rule states the traces connecting a decoupling capacitor and its associated vias to the power/ground-reference planes must be no longer than a specified distance.

From here, set the following:

- **Rule State** – toggles Decoupling Capacitor Distance to Via rules on or off.
- **Maximum Distance Between Capacitor and Via** – specifies the maximum distance between the capacitor pin and the via to the power or ground-reference plane.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Maximum Search Distance for Via** – specifies the maximum search distance for a via when a violation is found. This allows the user to quickly determine if the via is barely outside the original requirement and therefore not a serious violation.
- **Verify Via Connected to Appropriate Plane** – specifies whether or not EMI Scanner verifies the via connects to the chosen plane (shape). Selecting 'no' speeds up processing.

- **Power/Ground Trace Decoupling** – displays the **Power/Ground-Reference Trace Decoupling** rule configuration.

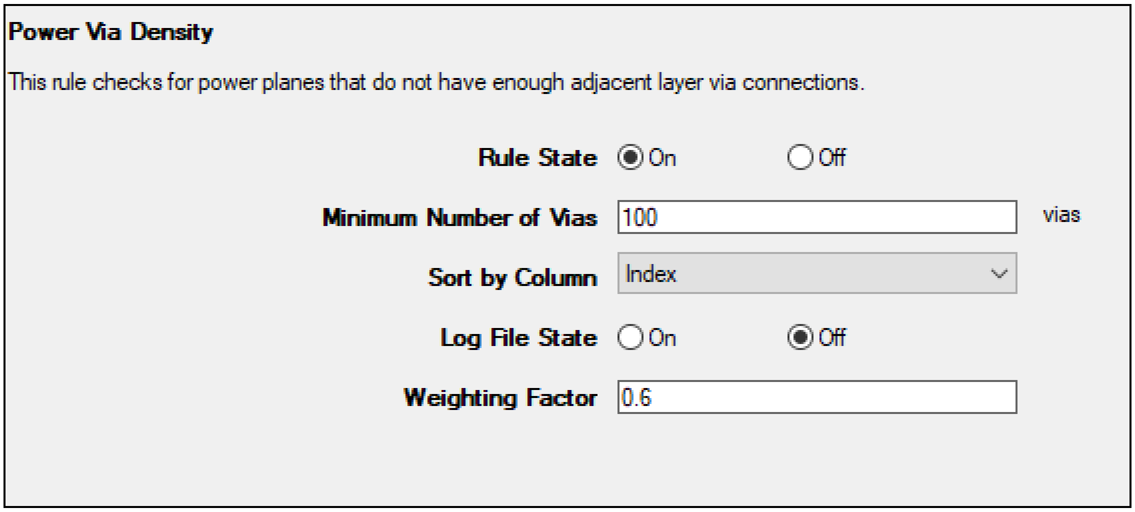


This rule states that all power and ground-reference traces longer than a specified length that provide power to a fast switching chip must have a decoupling capacitor within a specified distance on the IC power pin.

From here, set the following:

- **Rule State** – toggles Power/Ground-Reference Trace Decoupling rules on or off.
- **Maximum Ignorable Power/Ground Trace Length** – specifies the maximum allowable power/ground trace length that may be ignored. All traces longer than this length must meet this rule's requirement.
- **Max Distance Between Cap and IC Power Pin** – specifies the maximum distance between the capacitor pin and the power or ground-reference pin.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.
- **Net Choice** – specifies the nets to include in this rule analysis. Select one, or **Ctrl**+click to select multiple nets.

- **Power Via Density** – displays the **Power Via Density** rule configuration.



This rule checks for power planes that do not have enough adjacent layer via connections.

From here, set the following:

- **Rule State** – toggles Power Via Density rules on or off.
- **Minimum Number of Vias** – sets the minimum number of vias required to connect adjacent power planes.
- **Sort by Column** – specifies the column by which results are sorted.
- **Log File State** – determines whether EMI Scanner creates a log file with additional details (intended for use in debugging).
- **Weighting Factor** – specifies the weighting factor (0<factor<=1) that is applied to violations of this rule when assigning a rank. A violation's rank is its own "degree of badness" multiplied by the weighting factor. See Exporting EMI Scanner Results and Analyzing EMI Scanner Results Using iQ-Harmony.

4. Rules are automatically saved for the current EMI scan. To save them for a future scan, click **Save Rules**.

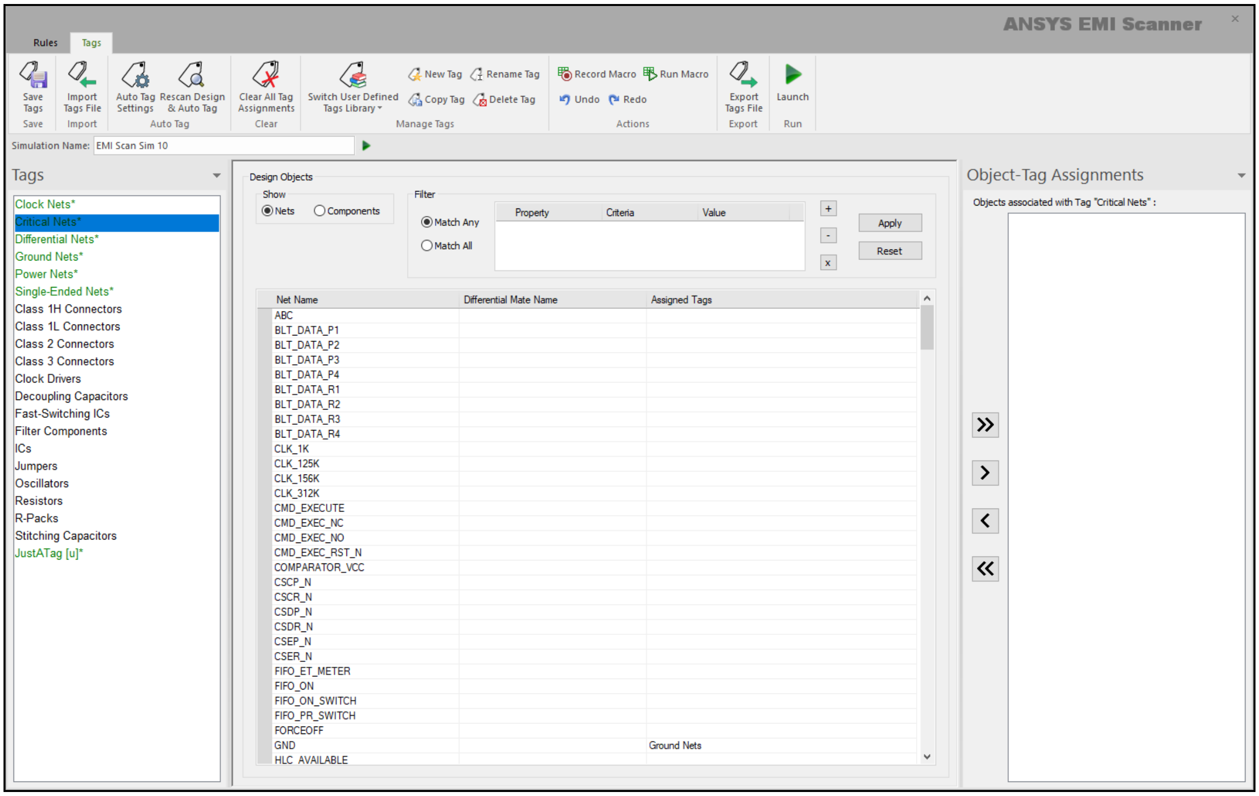
Note:

- You can display a description of a rule parameter at any time as with the cursor over it.
- Units of measure are set in General Settings.

Using Tags in EMI Scanner

The EMI Scanner **Tags** tab contains tag assignment panes and a ribbon with the following buttons:

- **Save Tags** – saves tags to the **Tags** pane.
- **Import Tags File** – allows you to import tags from a *.tgs file.
- **Auto Tag Settings** – allows you to change default tagging behavior.
- **Rescan Design & Auto Tag** – applies new tagging behavior.
- **Clear All Tag Assignments** – unassigns all nets and components from tags.
- **Switch User-Defined Tags Library** – allows you to browse to a different tagset.
- **New Tag** – creates a new tag.
- **Rename Tag** – renames a user-created tag.
- **Copy Tag** – copies a tag's properties into a new tag.
- **Delete Tag** – removes a tag on the **Tags** pane.
- **Record/Stop Macro** – starts or stops macro recording.
- **Run Macro** – runs a previously recorded macro.
- **Undo** – used to undo the previous action (only for manual assignments).
- **Redo** – used to restore an undone action (only for manual assignments).
- **Export Tags File** – allows you to export tags in a *.tgs file for later use.
- **Launch** – begins the EMI scan.



From the Tags pane (located on the left), Net tags appear in green, while Component tags appear in black. User-created tags are denoted by **[u]**.

Tags can be assigned automatically or manually.

Also add custom tags or create a tagging macro.

Assigning Tags Automatically

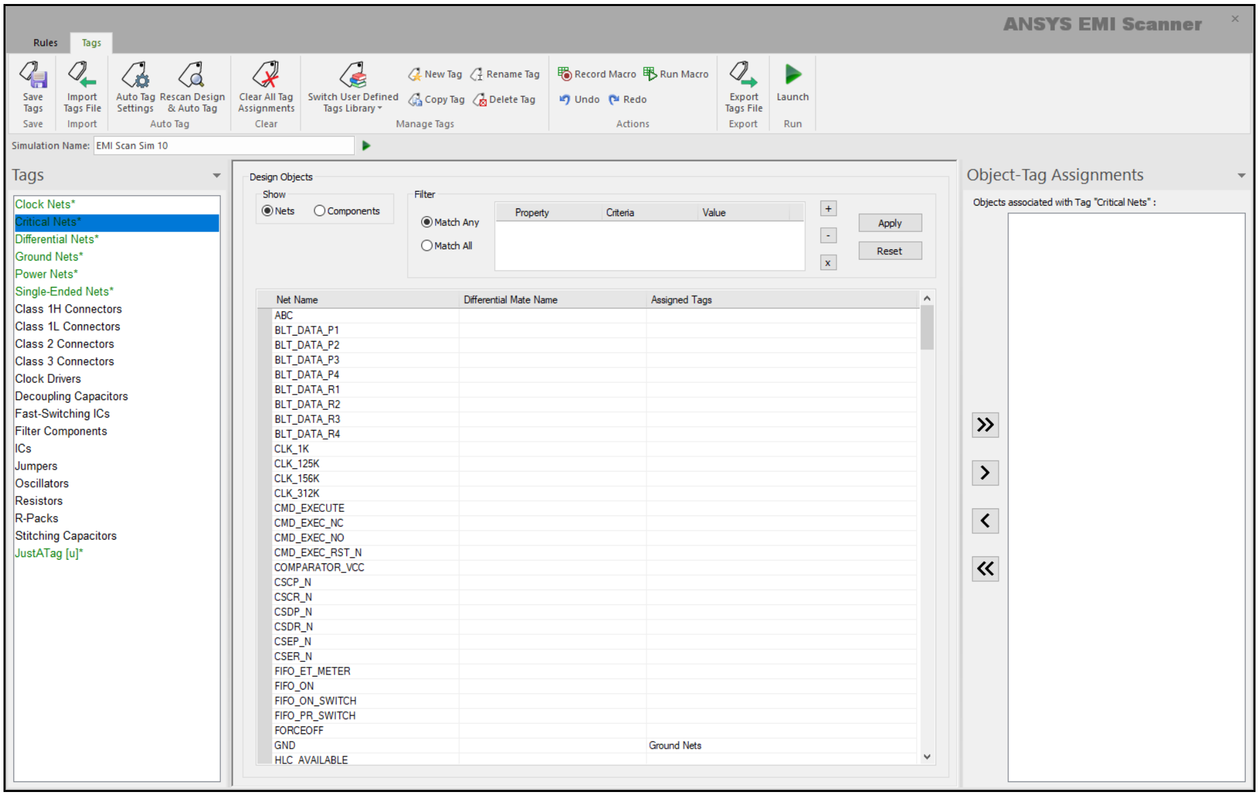
Upon loading, some tags are automatically assigned nets or components. By default, Auto Tag makes the following selections:

- **Critical Nets** – non power-ground nets that have a port.
- **Differential Nets** – those set up in the Differential Nets workspace.
- **Power/Ground Nets** – those set up in the Power/Ground workspace. Otherwise, distinguished by net name.
- **Decoupling Capacitors** – capacitors with names starting from C or EMC, with a value less than 10uF.
- **Stitching Capacitors** – capacitors connecting two power planes or two ground planes.

To change Auto Tag settings:

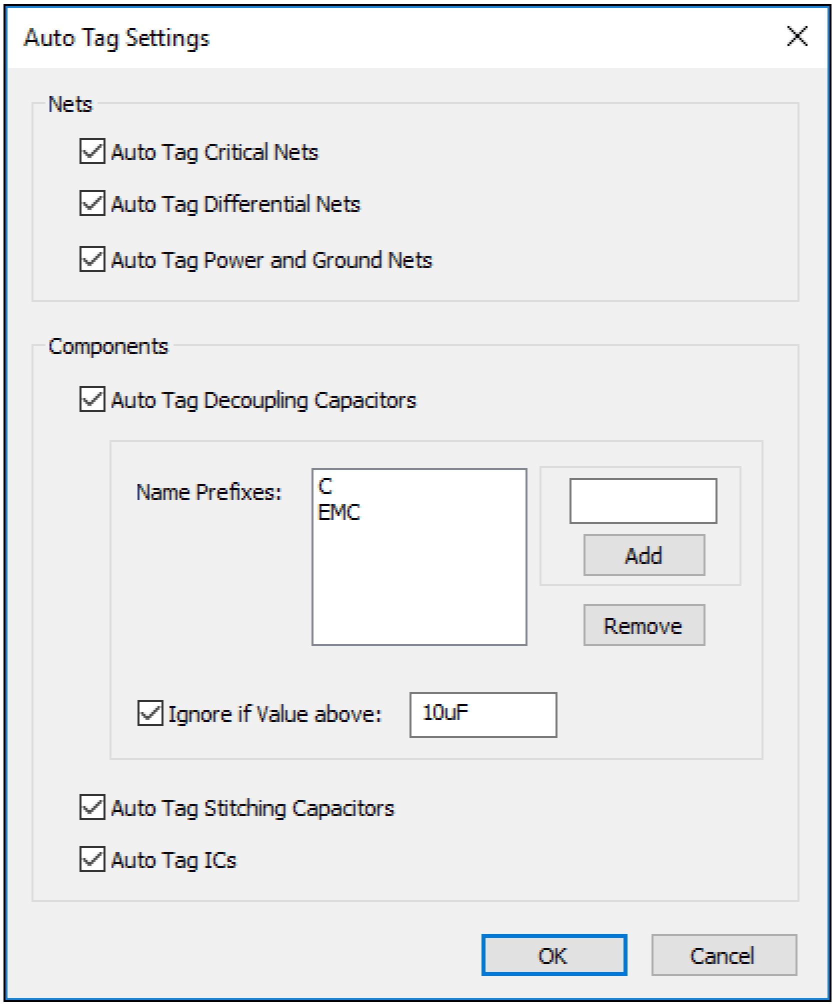
1. Launch EMI Scanner.
2. Click **Tags**.

   The **EMI Scanner** displays the **Tags** tab.



Net tags appear in green, while Component tags appear in black.

3. From the **Auto Tag** area, click **Auto Tag Settings** to open the **Auto Tag Settings** window.

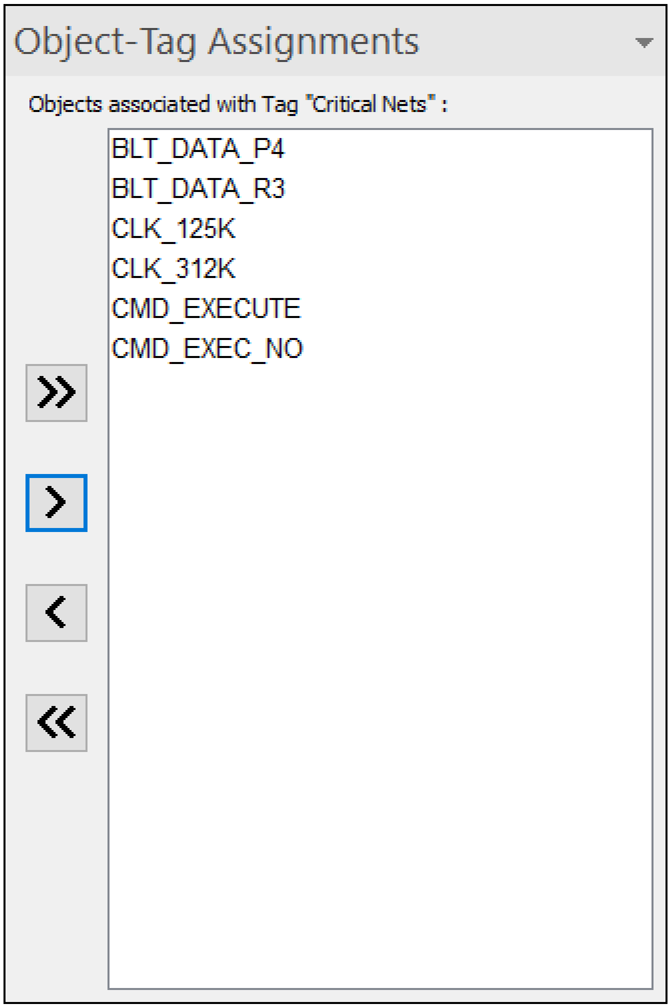


4. From the **Nets** area, determine the types of nets you want automatically tagged.
5. From the **Components** area, use the check boxes to determine whether to automatically tag Decoupling Capacitors, Stitching Capacitors, or ICs.

   If you select **Auto Tag Decoupling Capacitors**, select to **Add** or **Remove** name prefixes using the list and field. Also select **Ignore if Value above** to ignore capacitors above the value you specify.
6. Click **OK** to apply settings and return to the **EMI Scanner** window.
7. Click **Rescan Design & Auto Tag**.

   The design is scanned and tagged according to your settings.

   The **EMI Scanner** window displays the new tags in the **Object-Tag Assignments** pane for each tag.

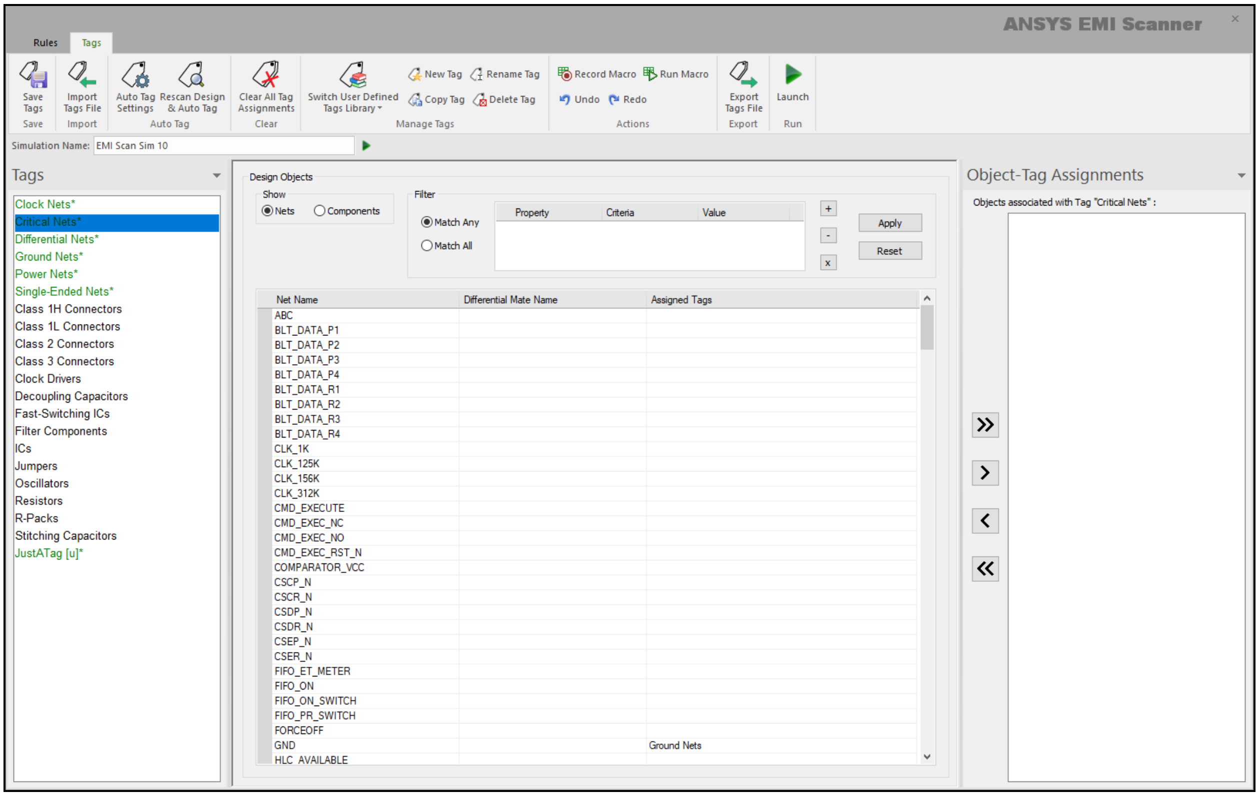


Assigning Tags Manually

To assign tags manually:

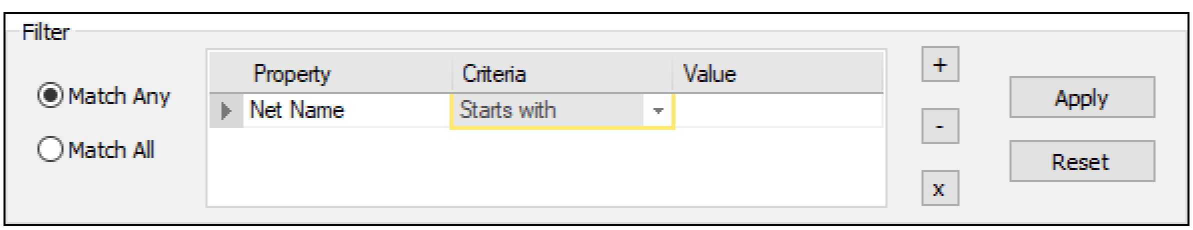
1. Launch EMI Scanner.
2. Click **Tags**.

   The **EMI Scanner** displays the **Tags** tab.



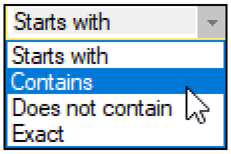
Net tags appear in green, while Component tags appear in black.

3. From the **Tags** pane, select a tag (e.g., Clock Nets, Decoupling Capacitors, Resistors) to which you want to assign nets or components. Also add custom tags.
4. The center pane updates to display a list of eligible nets or components. If chosen, apply filters to view only the nets or components you're looking for:



   a. Select either **Match Any** (nets or components matching any one criterion displays) or **Match All** (only nets or components meeting all criteria displays).
   b. Add or remove rows on the filter list by clicking **+** and **-**. Also clear all criteria by clicking (**X**).

c. Click the **Criteria** drop-down menu to select the filter type.



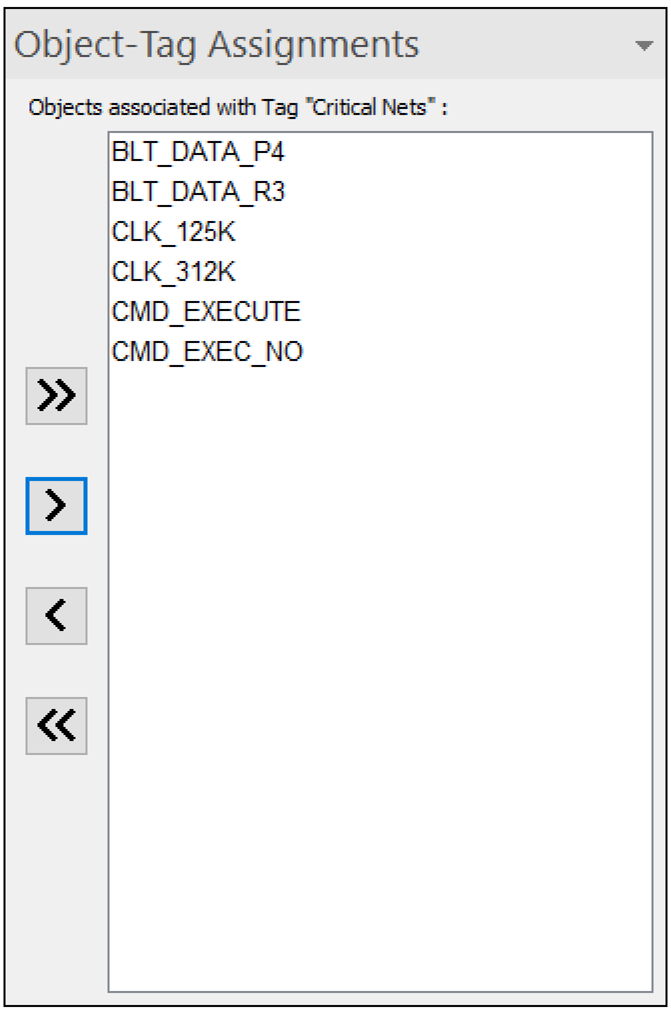
d. Enter filter values in the **Value** field.

e. Click **Apply** to apply the filters and update the net/component list.

f. Click **Reset** to remove the filter and display the default nets or components.

5. Select nets or components in the list, and use the right arrow button (**>**) to assign them. Use the double arrow (**>>**) to assign all listed nets or components to that tag.

   Assigned nets or components appear in the **Object-Tag Assignments** pane.



You can unassign nets or components individually by selecting them and using the left arrow (**<**), or unassign all nets or components using the double arrow (**<<**).

6. If chosen, click **Export Tags File** to save your tags as a *.tgs file. You can later click **Import Tags File** to use these settings again.

Working with Custom Tags

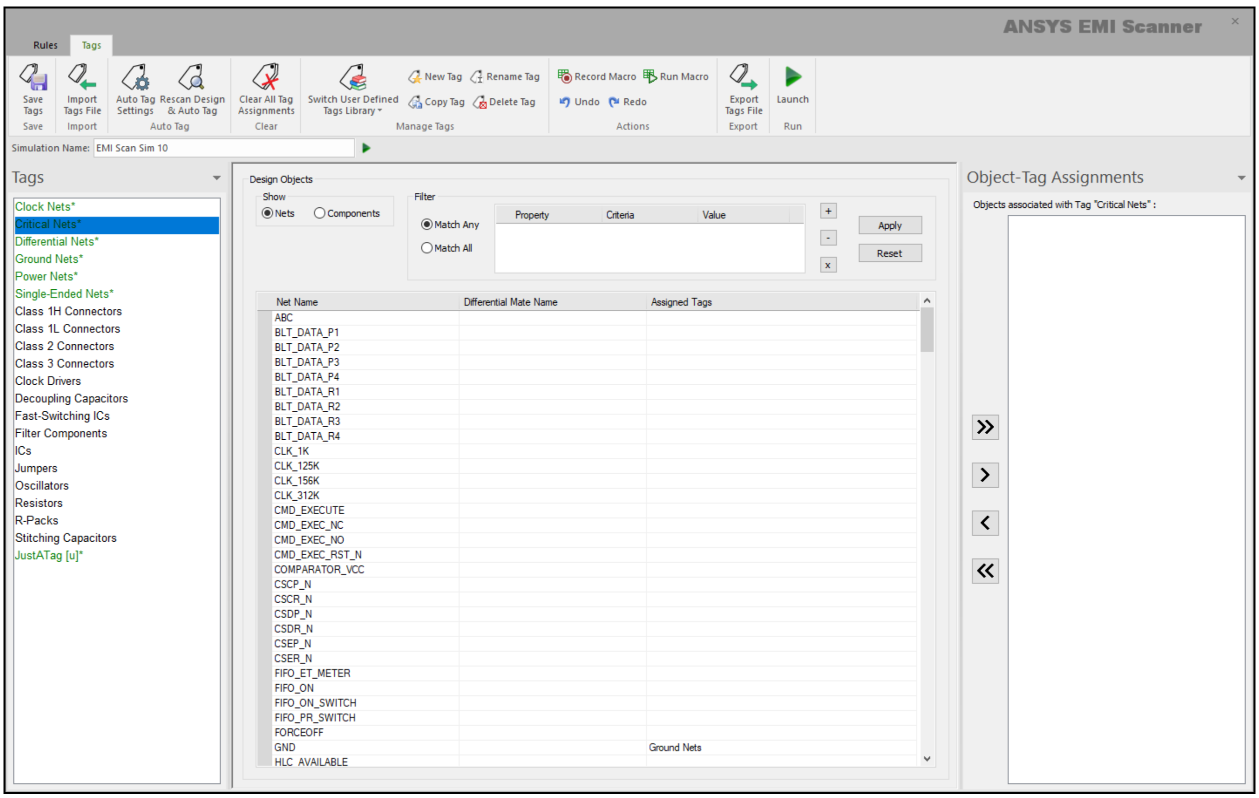
EMI Scanner also allows you to create and assign custom tags. User-created tags are denoted by **[u]** in the Tags pane, and can be assigned manually.

Adding Custom Tags

To add tags:

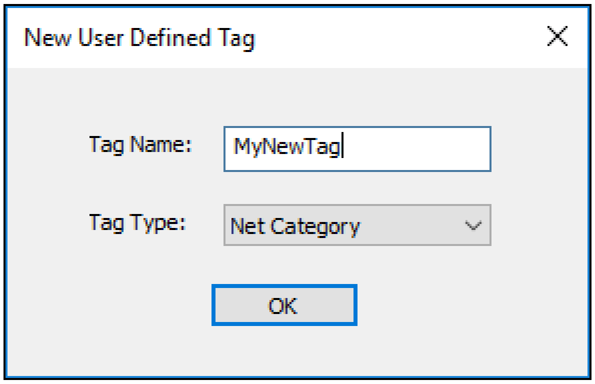
1. Launch EMI Scanner.
2. Click **Tags**.

   The **EMI Scanner** displays the **Tags** tab.



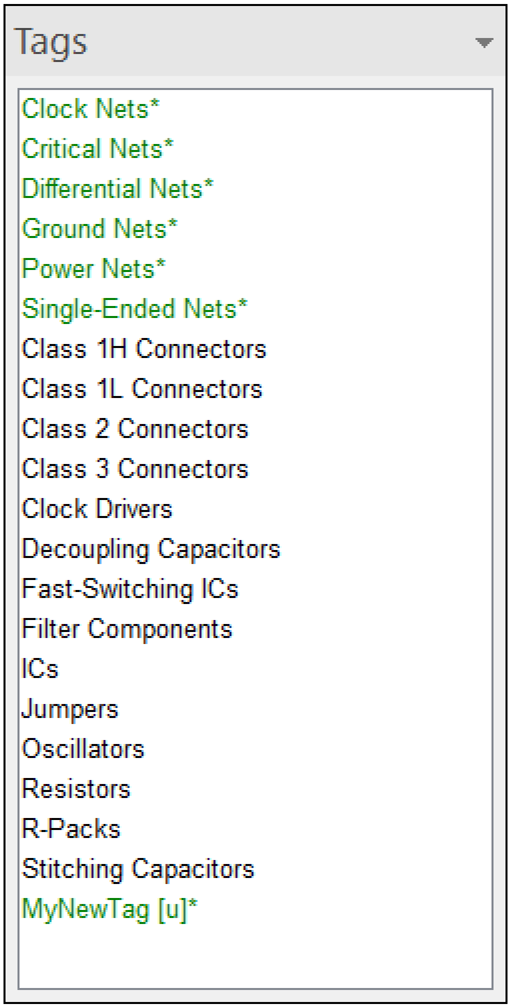
Tags are listed in the Tags pane on the left. Net tags appear in green, while Component tags appear in black.

3. From the **Manage Tags** area of the ribbon, select **New Tag** to open the **New User-Defined Tag** window.



4. Enter a **Tag Name** and select the **Tag Type**. You can create a tag either in the **Net Category** or **Component Category**.
5. Click **OK** to return to the **EMI Scanner** window.

   The new tag appears at the bottom of the **Tags** pane, with a **[u]** indicating that it is a user-created tag.



6. To save your tags, click **Save Tags**.

   Your tags appears in the **Tags** pane the next time you launch EMI Scanner.

Also create a custom tag from an existing tag. To do so:

1. Select a tag on the **Tags** pane.
2. From the **Manage Tags** area, click **Copy Tag**.

   The **Copy User-Defined Tag** window opens, with the **Tag Type** already selected.

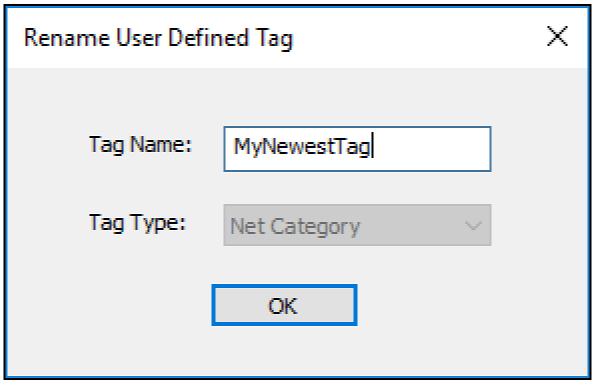
3. Name the new tag and click **OK**.

   The new tag appears in the **Tags** pane.

Renaming a Custom Tag

To rename a custom tag:

1. Select the tag on the **Tags** pane.
2. From the **Manage Tags** area, select **Rename Tag** to open the **Rename User-Defined Tag** window.



3. Rename the tag and click **OK**.

> **Note:**
>
> You can only rename custom tags. You cannot rename a default tag.

Deleting a Custom Tag

To delete a custom tag:

1. Select the tag on the **Tags** pane.
2. Use the double left arrow (**<<**) in the **Object-Tag Assignments** pane to remove any assignments.

> **Important:**
>
> You cannot delete a tag until all object assignments are removed from it.

3. From the **Manage Tags** area, click **Delete Tag**.

   The tag disappears on the **Tags** pane.

> **Note:**
>
> You can only delete custom tags. You cannot delete a default tag.

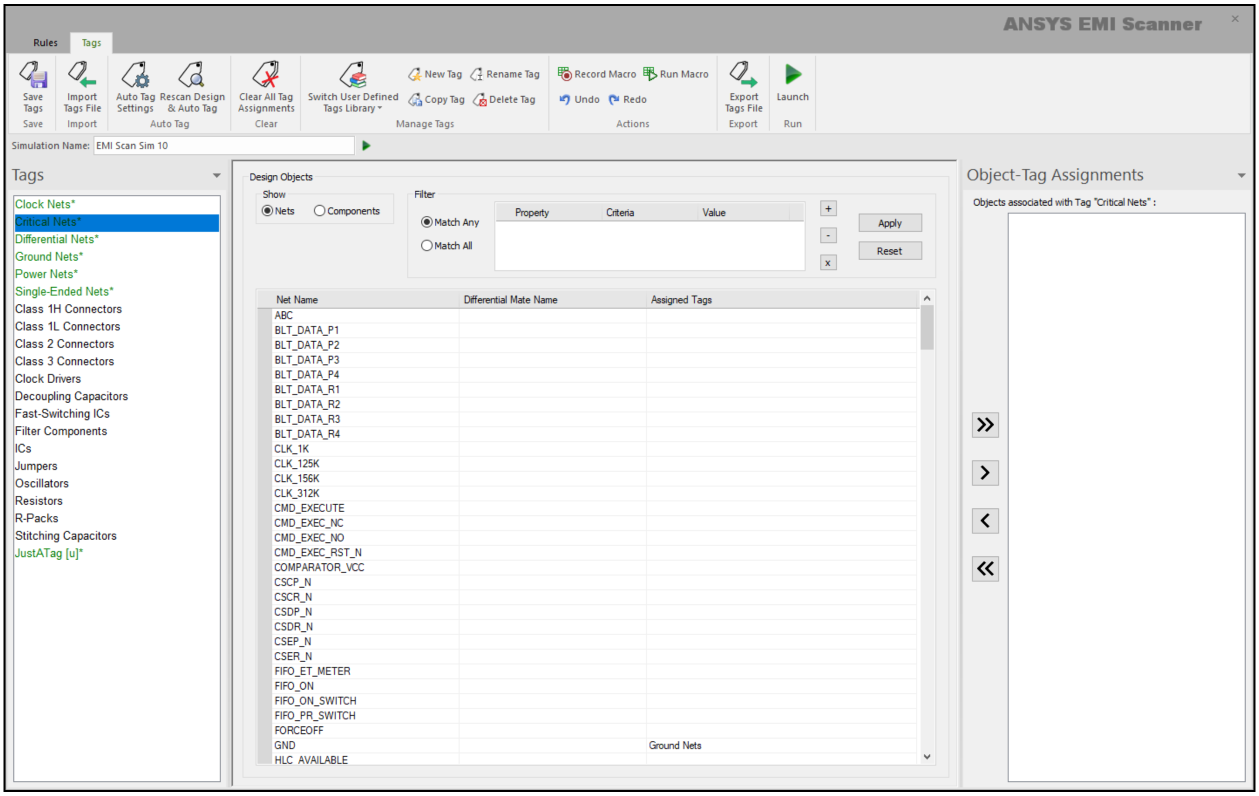
Creating Tagging Macros

You can save your tagging process as a series of steps. Macros are saved as XML-based *.tmx files.

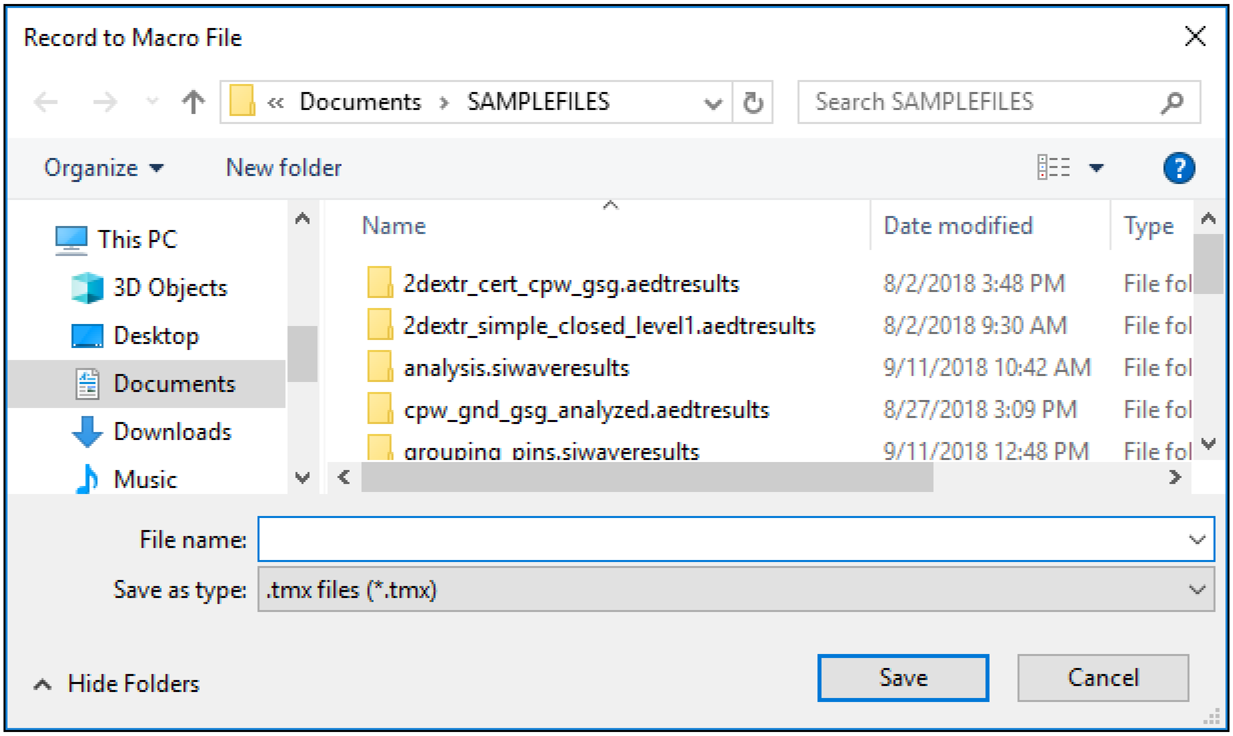
To create a macro:

1. Launch EMI Scanner.
2. Click **Tags**.

   The **EMI Scanner** displays the **Tags** tab.

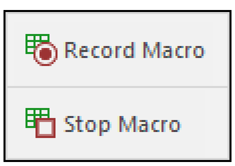


3. From the **Actions** area of the ribbon, select **Record Macro** to open the **Record to Macro File** window.



4. Navigate to your chosen save location, enter a **File name**, and click **Save**.

   Record Macro becomes **Stop Macro**.



5. Complete the actions you wish to save as a macro. You can **Undo** or **Redo** actions using the buttons in the Actions area.
6. When you are finished, click **Stop Macro**.

   The macro *.tmx file is saved at the location you specified earlier.

To run a saved macro, click **Run Macro**.

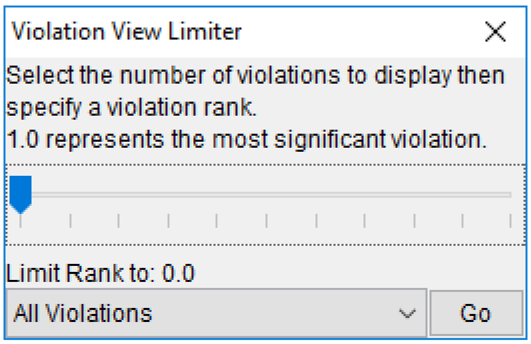
> **Note:**
>
> You cannot record and run a macro at the same time.

Analyzing EMI Scanner Results Using iQ-Harmony

iQ-Harmony receives violations on the EMI Scanner and, based on signal characteristics, makes predictions on the seriousness of the violations. This offers meaningful comparison between rule limits and violations, and allows you to make "what if" comparisons and determine impacts.

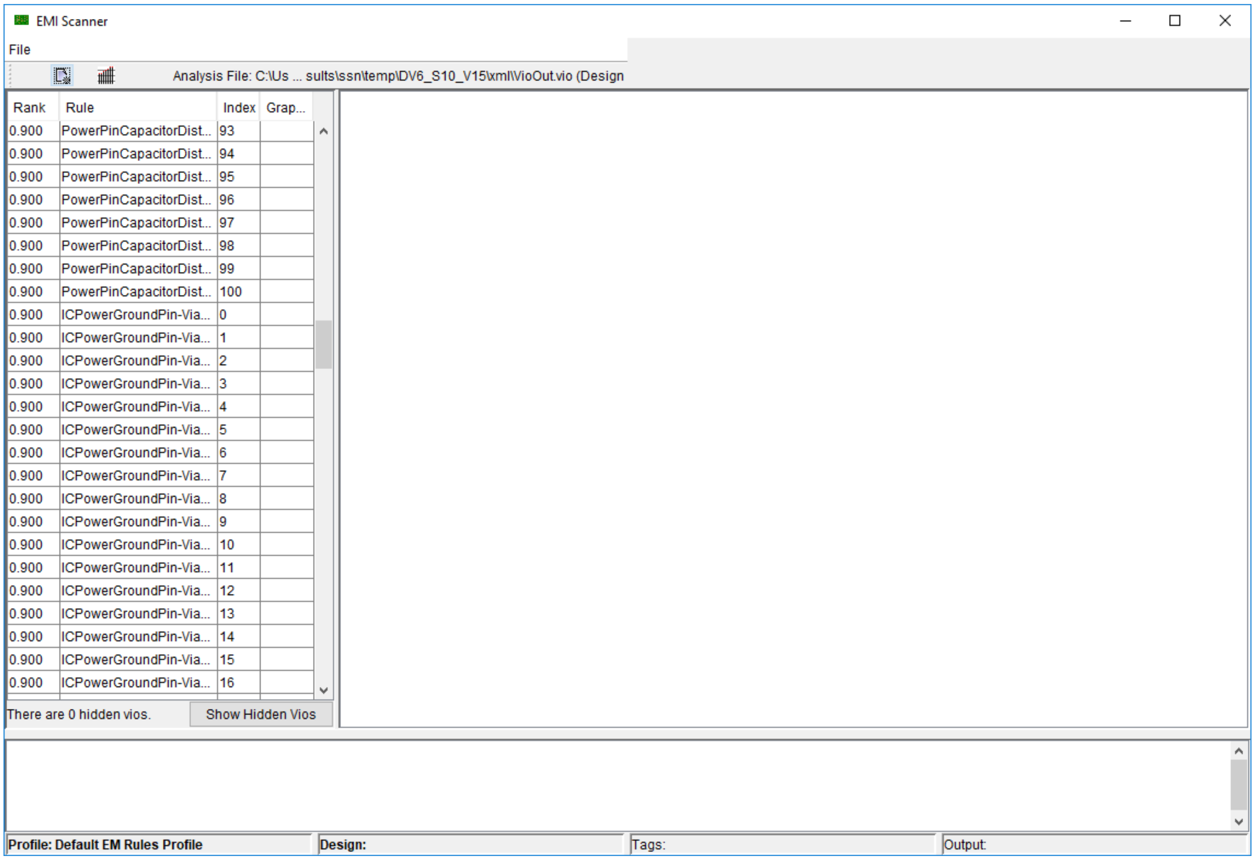
To analyze EMI Scanner results:

1. Run an EMI scan.
2. From the **Project Manager** window, right-click **EMI Setup** and select **Results** > **Analyze with iQ-Harmony** to open the **Violation View Limiter** window.



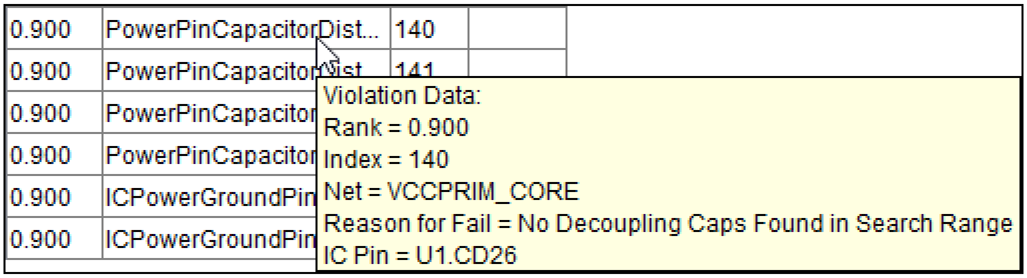
3. Use the drop-down menu to select the number of violations to display. The default is **All Violations**.
4. Use the slider to select a minimum violation rank to display. 0.0 is the least severe and 1.0 is the most severe.

5. Click **Go** to open the **EMI Scanner** window.



Violations are listed in the left-hand pane. By default, they are sorted by rank, with the most severe violations at the top of the list. Click the column headings to change the sorting.

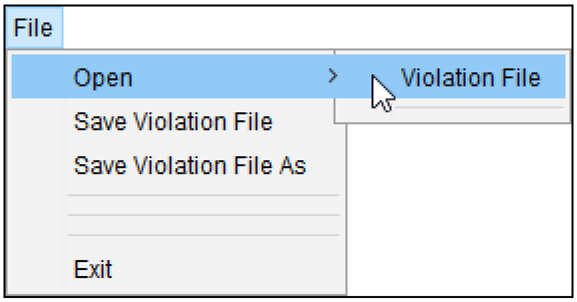
Hovering the cursor over a violation offers more information about the violation.



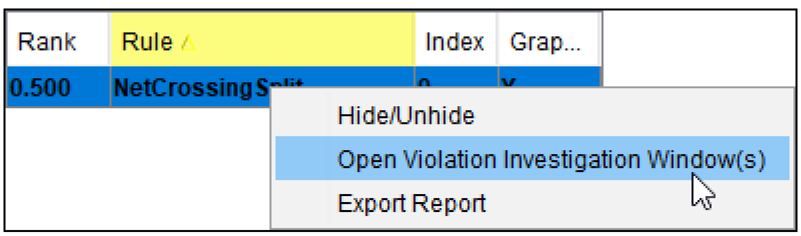
If there are any hidden violations, text at the bottom of the pane notes this, and offers you the option to **Show Hidden Vios**.

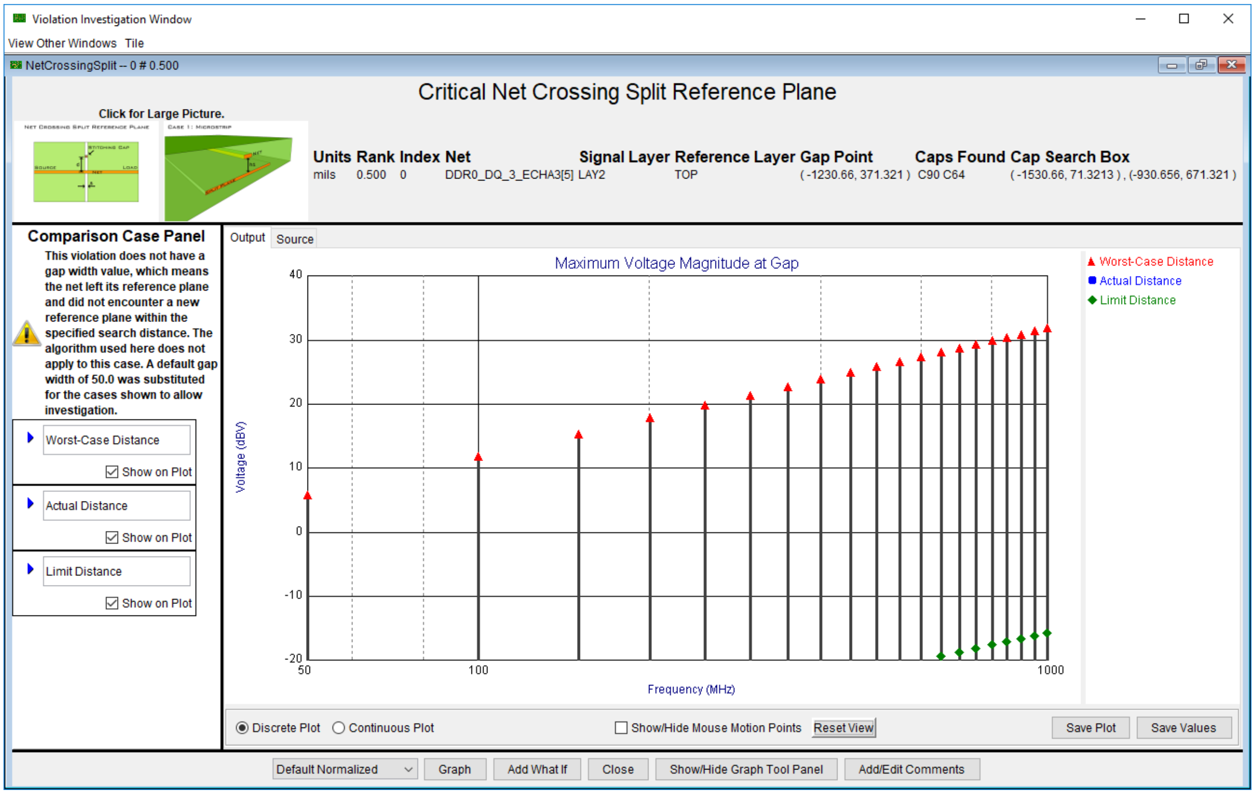
The **File** menu allows you to save the violations list as a *.vio file, or open an existing *.vio file.

Also open an existing *.vio file by clicking the **Open Analysis File** () icon.



The **Current Spectra Library** icon () opens the **Spectra Configuration** window. From the left pane, either double-click a violation, or right-click it and select **Open Violation Investigation Window(s)** to open the **Violation Investigation** window.





The top of the **Violation Investigation** window displays information about the violation's location and severity, and graphic representations of the type of violation. Click a graphic to enlarge it for easier viewing.

The left pane is the **Comparison Case Panel**. The default cases vary depending on the type of violation, and can include: Worst-Case Distance, Actual Distance, Limit Distance, Search Limit Distance. Click the blue triangles to expand the cases and view their display options. You can add a comparison case by clicking **Add What If**.

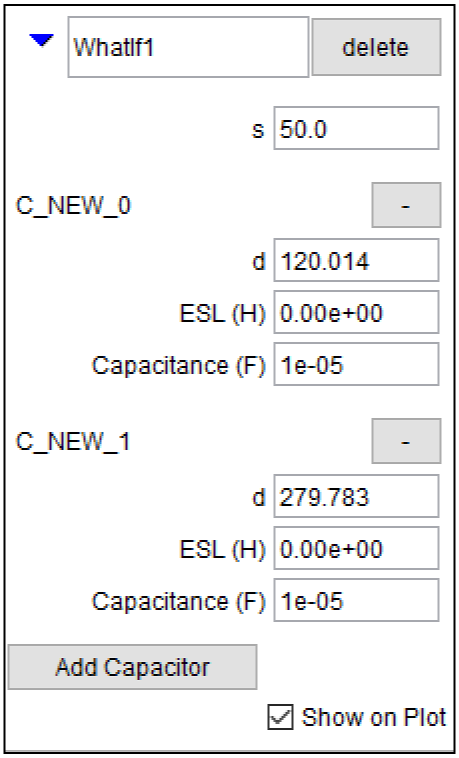
The comparison cases allow you to see how changes in signal characteristics affects the violations. Use the **Show on Plot** check boxes to determine which cases are displayed. You can rename a case by typing into its name field.

The bottom of the **Violation Investigation** window allows you to make additional changes.

Options include:

- **Choose A Spectrum** – allows you to select a default spectrum or one you have added via Spectra Configuration.
- **Graph** – after you change data, click this button to refresh the graph.

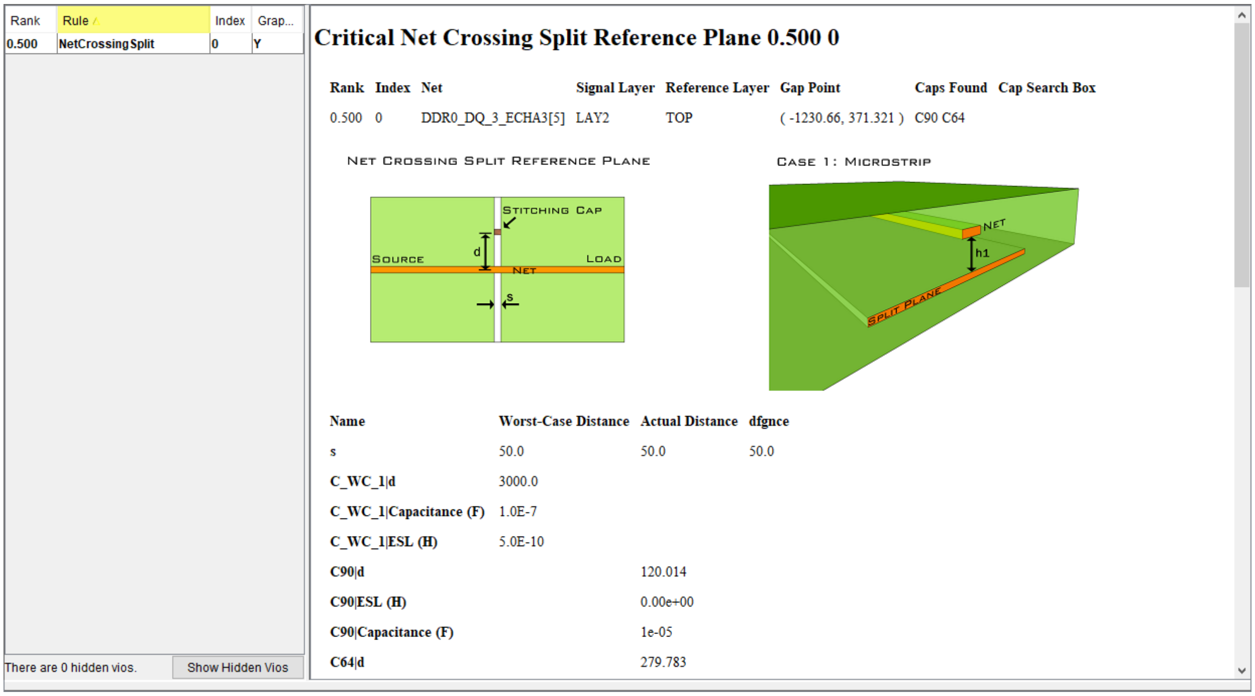
- **Add What If** – adds a new comparison case to the Comparison Case Panel.



  You can change the settings and rename the case just as with default comparison cases. Also delete the custom case.

- **Close** – closes the Violation Investigation Window and returns you to the EMI Scanner window.
- **Show/Hide Graph Tool Panel** – toggles the display of the Graph Tool Panel:
- **Add/Edit Comments** – opens a blank text window to type comments. These are saved with the *.vio file.

6. Analyze the data as chosen.
7. If you wish to save the plot as a *.jpeg or *.png image, click **Save Plot**.
8. If you wish to save the values in plain text format, click **Save Values**.
9. Click **Close** to close the **Violation Investigation** window closes, and the **EMI Scanner** window displays your analysis in the right pane.

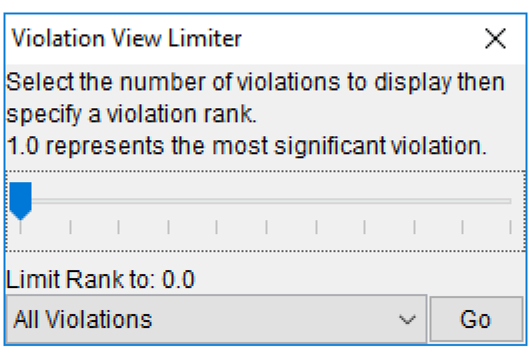


10. When you are finished analyzing, click **File** > **Save Violation File** or **File** > **Save Violation File As** to save your analysis in *.vio format.

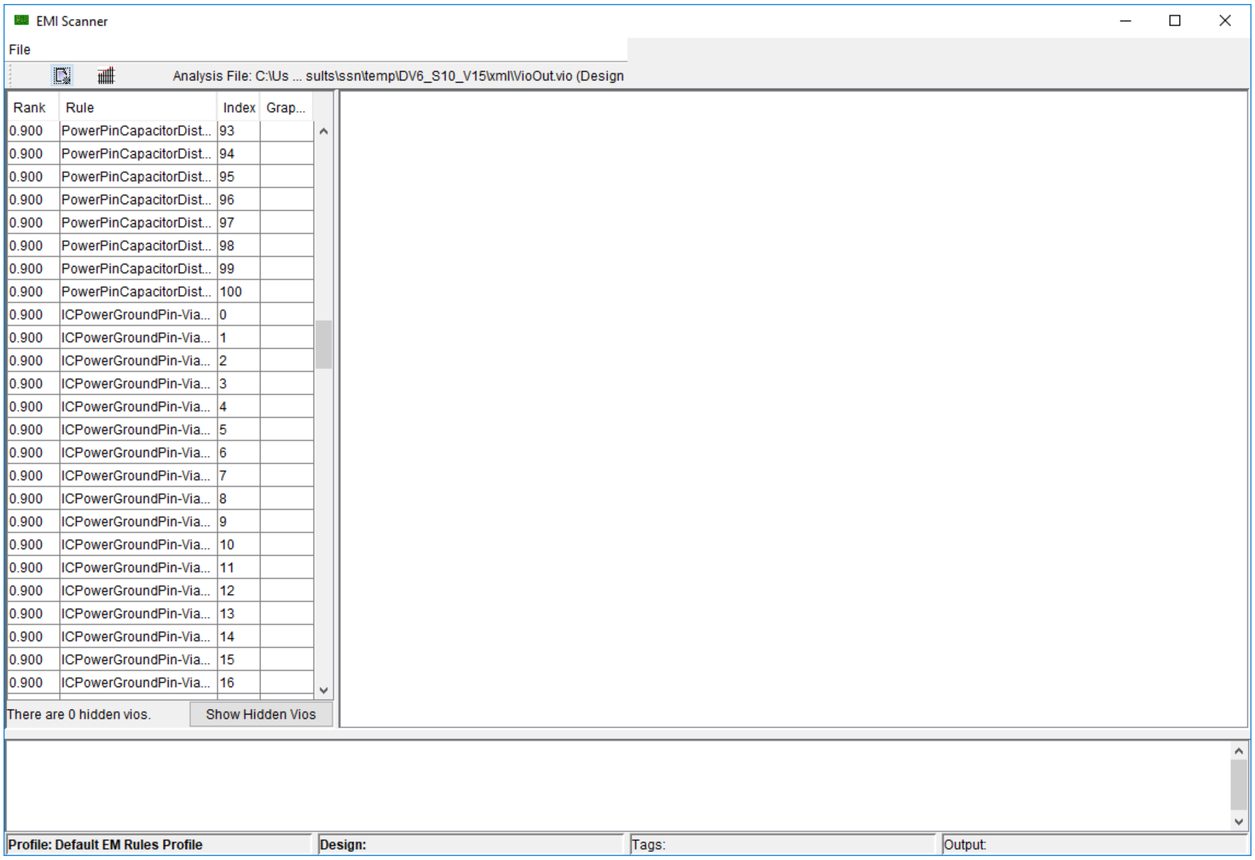
Spectra Configuration

To configure the Spectra library:

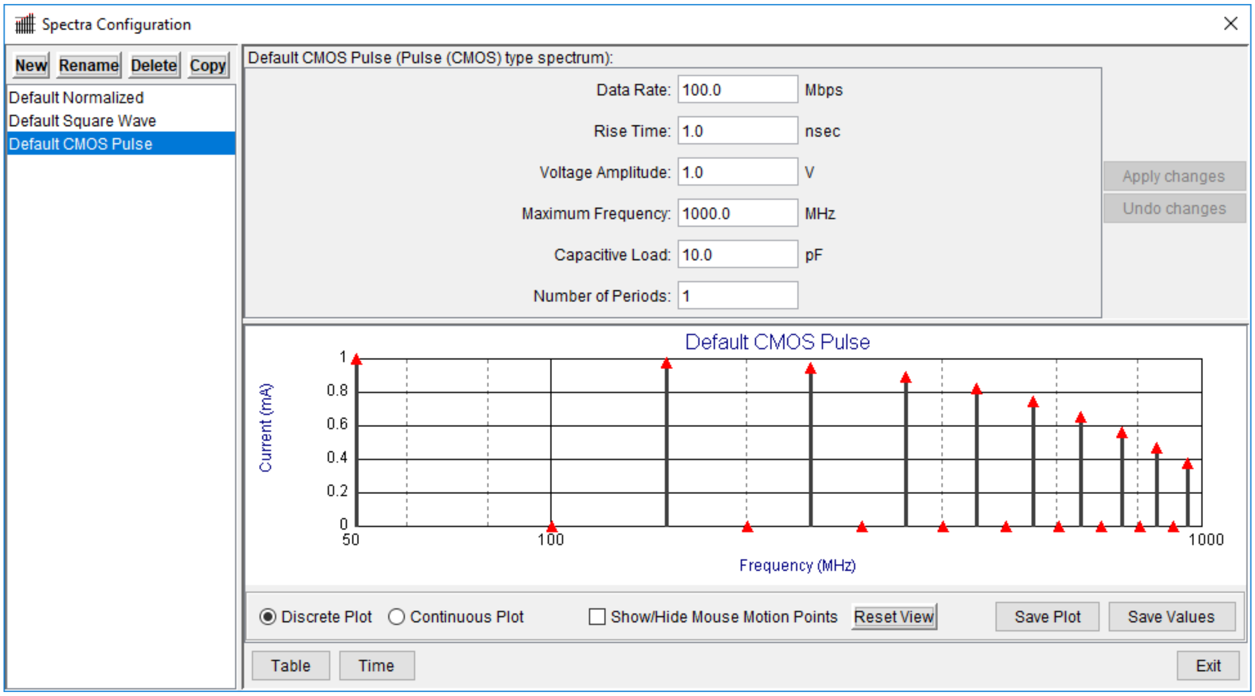
1. Run an EMI scan.
2. Click **Results**.
3. Use the EMI Scan drop-down menu to select **[Simulation Name]** > **Analyze with iQ-Harmony** to open the **Violation View Limiter** window.



4. Use the drop-down menu to select the number of violations to display. The default is **All Violations**.
5. Use the slider to select a minimum violation rank to display. 0.0 is the least severe and 1.0 is the most severe.
6. Click **Go** to open the **EMI Scanner** window.



7. Click the **Current Spectra Library** (icon) icon to open the **Spectra Configuration** window.



Select a spectrum in the left pane to view its settings and plot to the right. You can edit fields directly, then **Apply changes** or **Undo changes**.

By default, a Frequency/Current plot appears. The plot can be viewed as a **Discrete Plot** or a **Continuous Plot**.

You can **Save Plot** as a *.png or *.jpeg image, or **Save Values** in plain text format.

Click **Table** to view information as a table instead of a graph.

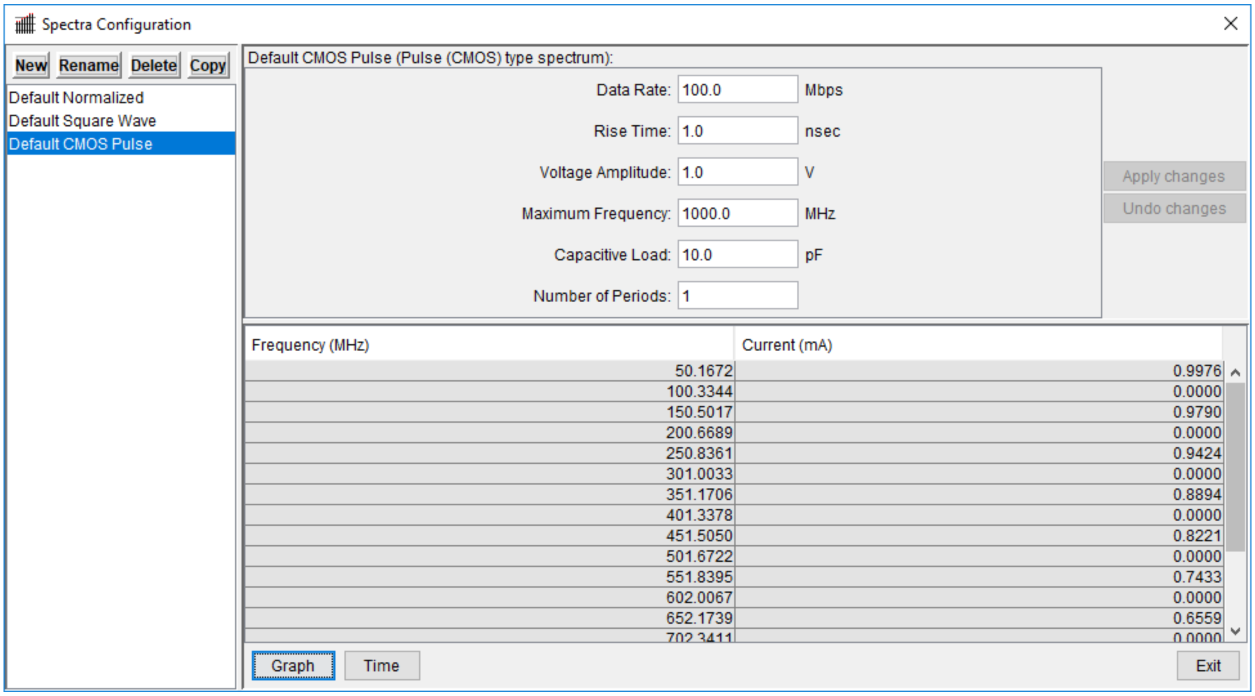
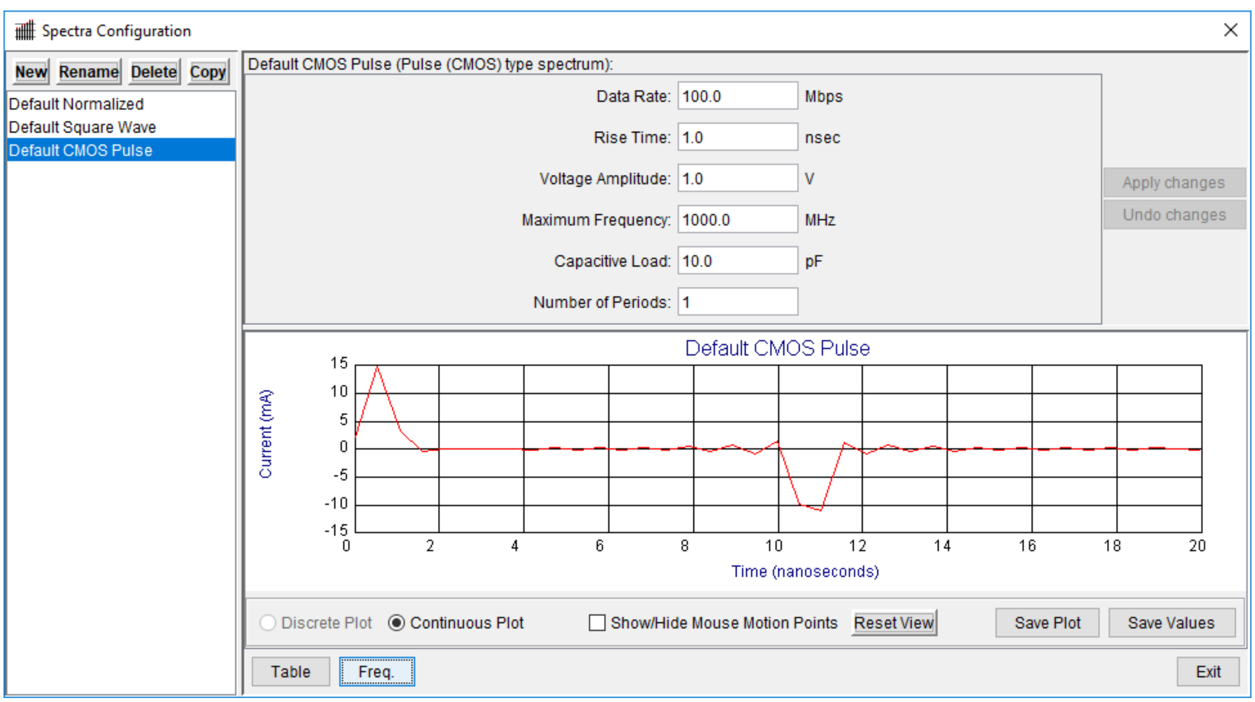


Table becomes **Graph**. Click it again to restore the plot view.

Click **Time** to view Current/Time information instead of Current/Frequency.



Time becomes **Freq.**. Click it again to return to Current/Frequency.

From the Spectra Configuration window, you can also perform the following actions:

- **New** – create a new spectrum.
- **Rename** – change a spectrum name.
- **Delete** – remove a spectrum.
- **Copy** – create a copy of an existing spectrum with a new name.

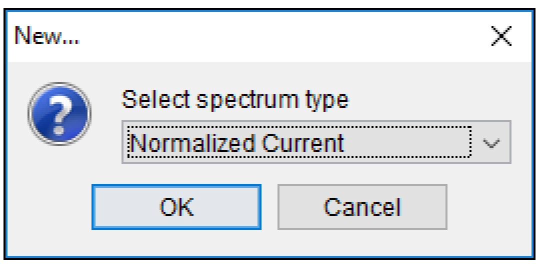
8. Click **Exit** to return to the **EMI Scanner** window.

   Any changes you made are now available in the **Violation Investigation** window.

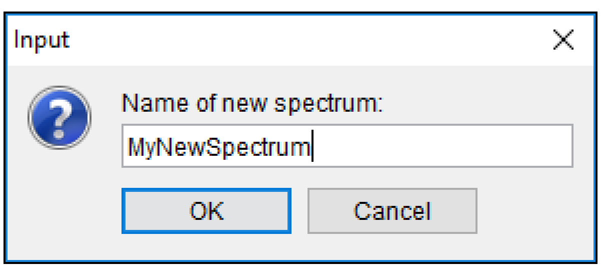
Creating a New Spectrum

To create a new spectrum on the Spectra Configuration window:

1. Click **New** to open the **New** window.

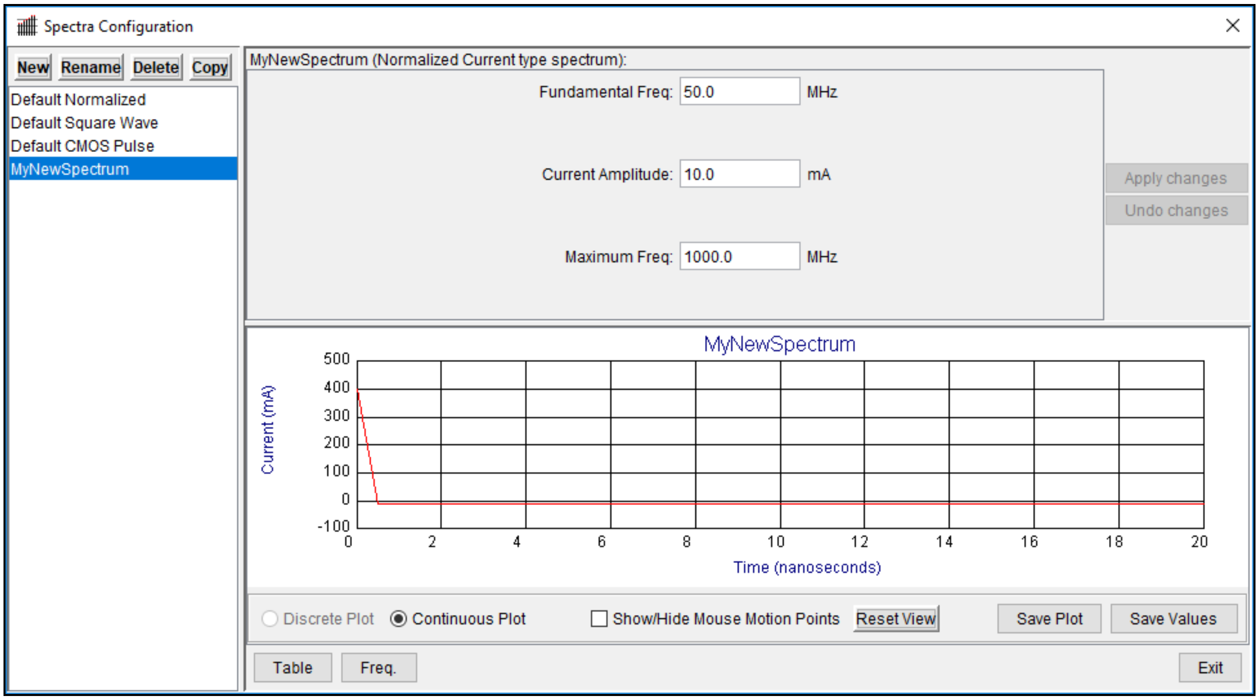


2. Use the drop-down menu to select an existing spectrum type (**Normalized Current**, **Square Wave**, or **Pulse CMOS**), or select **User-defined** to import settings.
3. Click **OK** to open the **Input** window.



4. Name the new spectrum and click **OK**.

   The new spectrum appears in the left pane, and its settings and plot appear to the right.



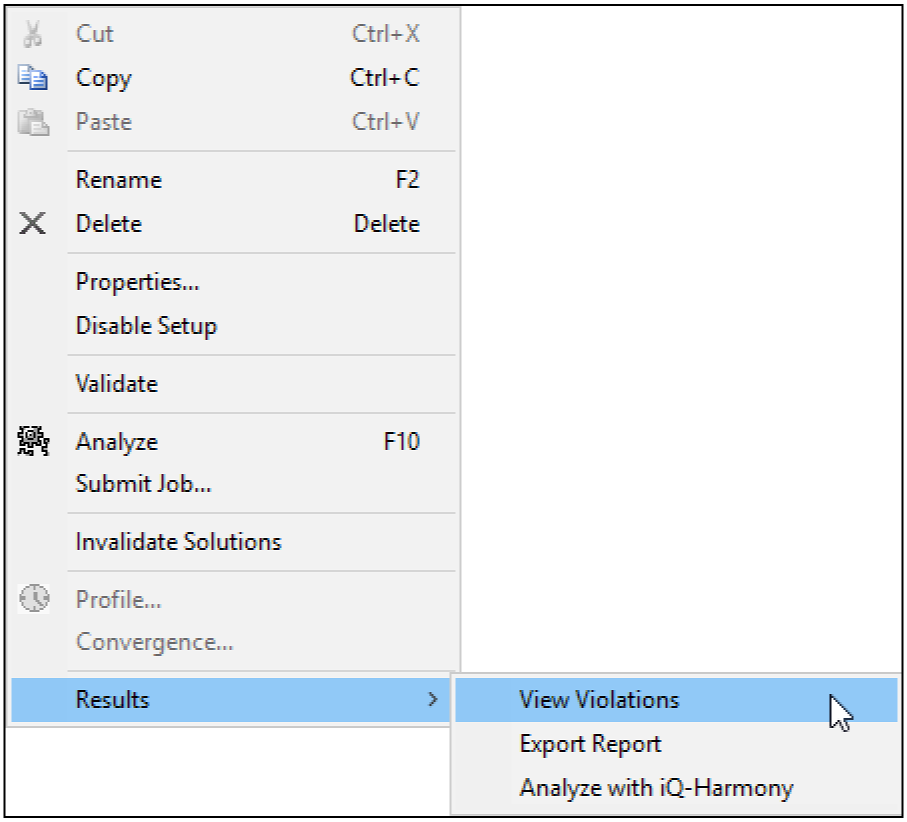
> **Note:**
>
> Configurable settings vary based on the type of spectrum you selected previously.

5. Enter the applicable settings and click **Apply Changes**.
6. Click **Exit** to return to the **EMI Scanner** window.

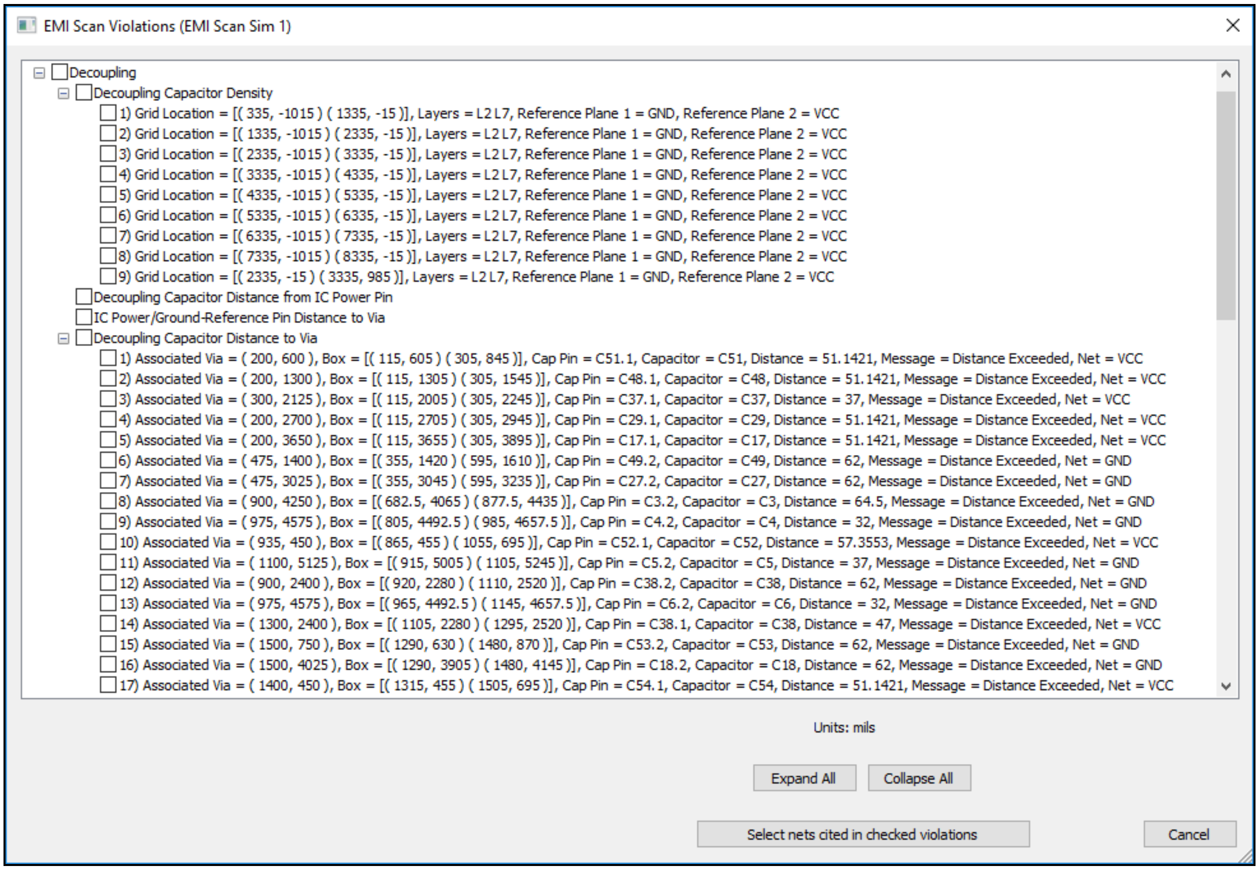
Viewing EMI Scanner Results

Complete these steps to view EMI Scanner results.

1. From the **Project Manager** window, right-click **EMI Setup** and select **Results** > **View Violations**.

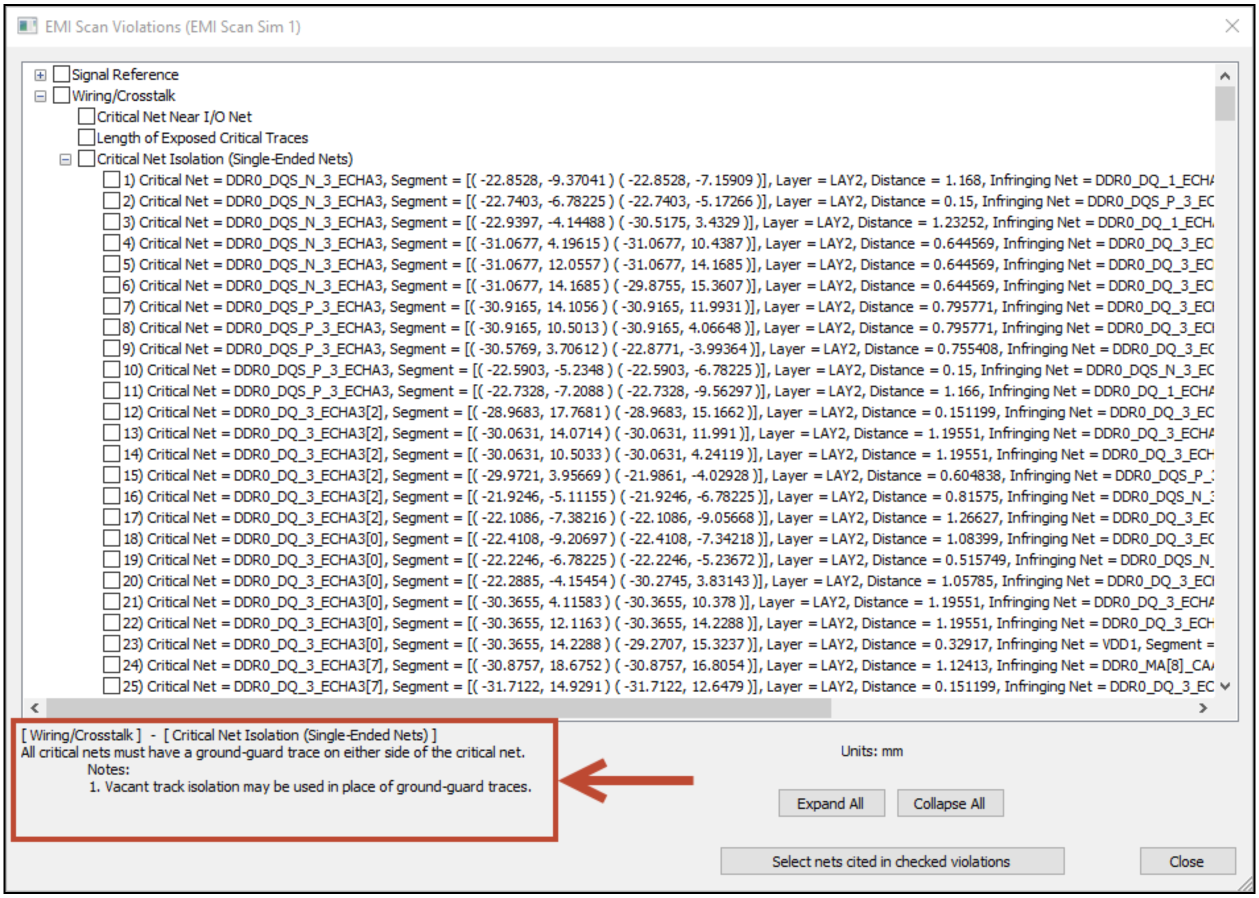


The **EMI Scan Violations** window opens.



2. Violations are listed in a tree, by type. Expand and collapse items in the tree, or click **Expand All**/**Collapse All**.

3. Click a violation to zoom to its location in the **Design Area**. Clicking a violation also populates the **EMI Scan Violations** window with a description of the violation.



4. Use the check boxes in the tree to select multiple violations.
5. Click **Select nets cited in checked violations** to select these nets in the *Modeling* workspace.

Other Results Menu options include:

- **Export Report** – allows you to export an HTML report of EMI rules and violations.
- **Analyze with iQ-Harmony** – opens the Violation View Limiter window, the first step in analyzing EMI Scanner results.

Exporting EMI Scanner Results

To export EMI Scanner results:

1. From the **Project Manager** window, right-click **EMI Setup**.

2. Select **Results** > **Export Report** to open an HTML report in the default Internet browser.

ANSYS EMI Scanner RULES

ANSYS EMI Scanner RULE RESULTS

SIGNAL REFERENCE

Critical Net Crossing Split Reference Plane - Violations: 0	ON
Critical Net Changing Reference Plane - Violations: A - 6, B - 0, C - 0	ON
Critical Net Near Edge of Reference Plane - Violations: 0	ON

WIRING/CROSSTALK

Critical Net Near I/O Net - Violations: 0	ON
Length of Exposed Critical Traces - Violations: 10	ON
Critical Net Isolation (Single-Ended Nets) - Violations: 0	ON
Critical Net Isolation (Differential Nets) - Violations: 0	ON
Critical Differential Net Length Matching and Spacing - Violations: 0	ON
Wide Power/Ground Traces - Violations: 434	ON

DECOUPLING

Decoupling Capacitor Density - Violations: 0	ON
Decoupling Capacitor Distance from IC Power Pin - Violations: 101	ON
IC Power/Ground-Reference Pin Distance to Via - Violations: 178	ON
Decoupling Capacitor Distance to Via - Violations: 146	ON
Power/Ground-Reference Trace Decoupling - Violations: 0	ON
Power Via Density - Violations: 1	ON

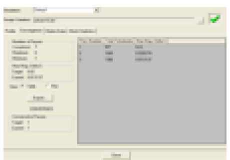
PLACEMENT

I/O Filter Distance to I/O Connector - Violations: 0	ON
Distance from Oscillator to Clock Driver - Violations: 0	ON

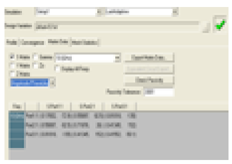
> **Note:**
>
> You can change whether EMI Scanner produces one large, combined HTML report or several individual HTML files in the EMI Scanner General Settings.

18 - Post Processing and Generating Reports

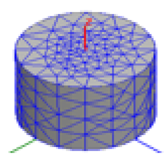
As Ansys Electronics Desktop completes a solution, you can display and analyze the results:



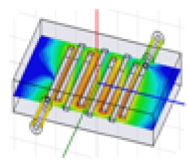
View Solution Data



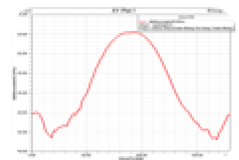
View Optimetrics Results



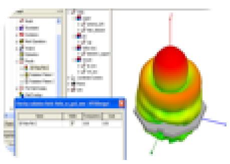
Create Mesh Plots



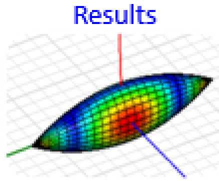
Plot Field Overlays



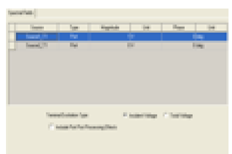
Create or Modify 2D or 3D Reports



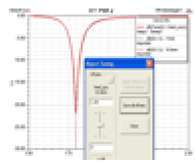
Overlay 3D Field Plot on Model



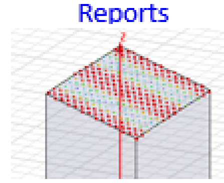
Create Animations



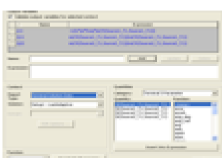
Scale Excitation Mag/Phase



Tune Reports

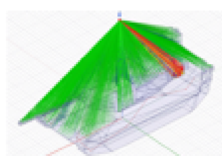


Port Field Display

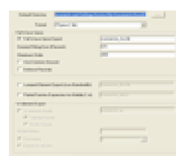


Specify Output Variables

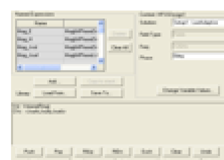
Radiated Fields Observation Regions



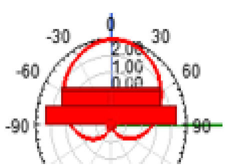
Visual Ray Trace



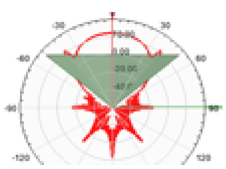
Export Equivalent Circuit Data



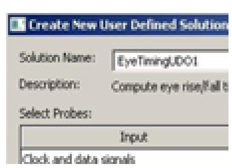
Fields Calculator / Derived Field Quantities



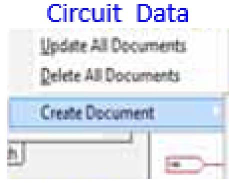
Overlay 2D Radiation Fields



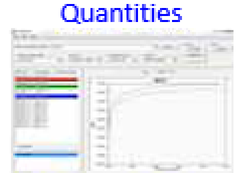
Overlay Monostatic/ Bistatic RCS



User Defined Outputs



User Defined Documents



Network Data Explorer

- View solution data including the following: convergence information, computing resources that were used during the solution process, mesh statistics, and matrices computed for the S-parameters, impedances, and propagation constants during each adaptive, non-adaptive, or sweep solution. For eigenmode solutions, you can view the real and imaginary parts of the frequency and quality factor Q computed for each eigenmode. Solution data can also be viewed while Ansys Electronics Desktop is generating a solution.

- View analysis results for Optimetrics solutions.
- Plot field overlays - representations of basic or derived field quantities - on surfaces or objects.
- Overlay Field Plots on Models
- Create 2D or 3D reports of S-parameters, basic and derived field quantities, and radiated field data.
- Plot the finite element mesh on surfaces or within 3D objects.
- Create animations of field quantities, the finite element mesh, and defined project variables.
- Scale an excitation's magnitude and modify its phase.
- Apply Derivative Tuning to Reports

> **Note:**
>
> Except in the case of non-model boxes drawn in the global coordinate system (CS), non-model objects cannot be used for any fields post processing operation You can use non-model boxes drawn in the global CS for post processing operations, including integration and solution domaining.

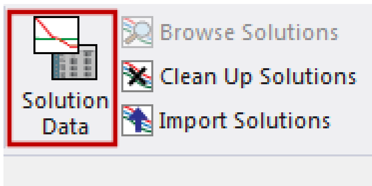
Viewing Solution Data

While Ansys Electronics Desktop is generating a solution, or when it is complete, you can view the following information about the solution:

- Convergence information.
- Computing resources, or profile information, that were used during the solution process.
- Matrices computed for the S-parameters, impedances, and propagation constants during each adaptive, non-adaptive, or sweep solution.
- Mesh statistics
- For eigenmode solutions, view the real and imaginary parts of the frequency and quality factor Q computed for each eigenmode.
- For Characteristic Modes solutions, a CMA Data tab reports the the Number of Modes, Characteristic Angle and current, Modal Significance and Quality Factor, and Voltage per port based in edit sources weighting.
- The state of solved solutions.
- For transient solutions, Transient Data.

To access the **Solution Data** window, in which the information above can be accessed, do one of the following:

- Click **Circuit**> **Results**> **Solution Data**.
- Right-click **Results** in the Project Manager, and then click **Solution Data** on the shortcut menu.
- Select the **Results** tab on the ribbon, and click the **Solution Data** icon.

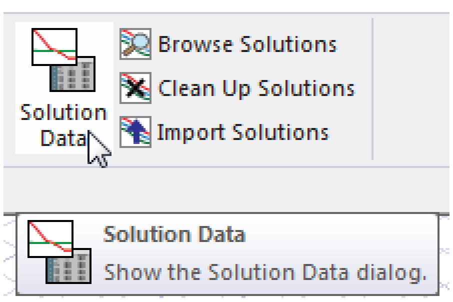


Viewing a Solution Profile

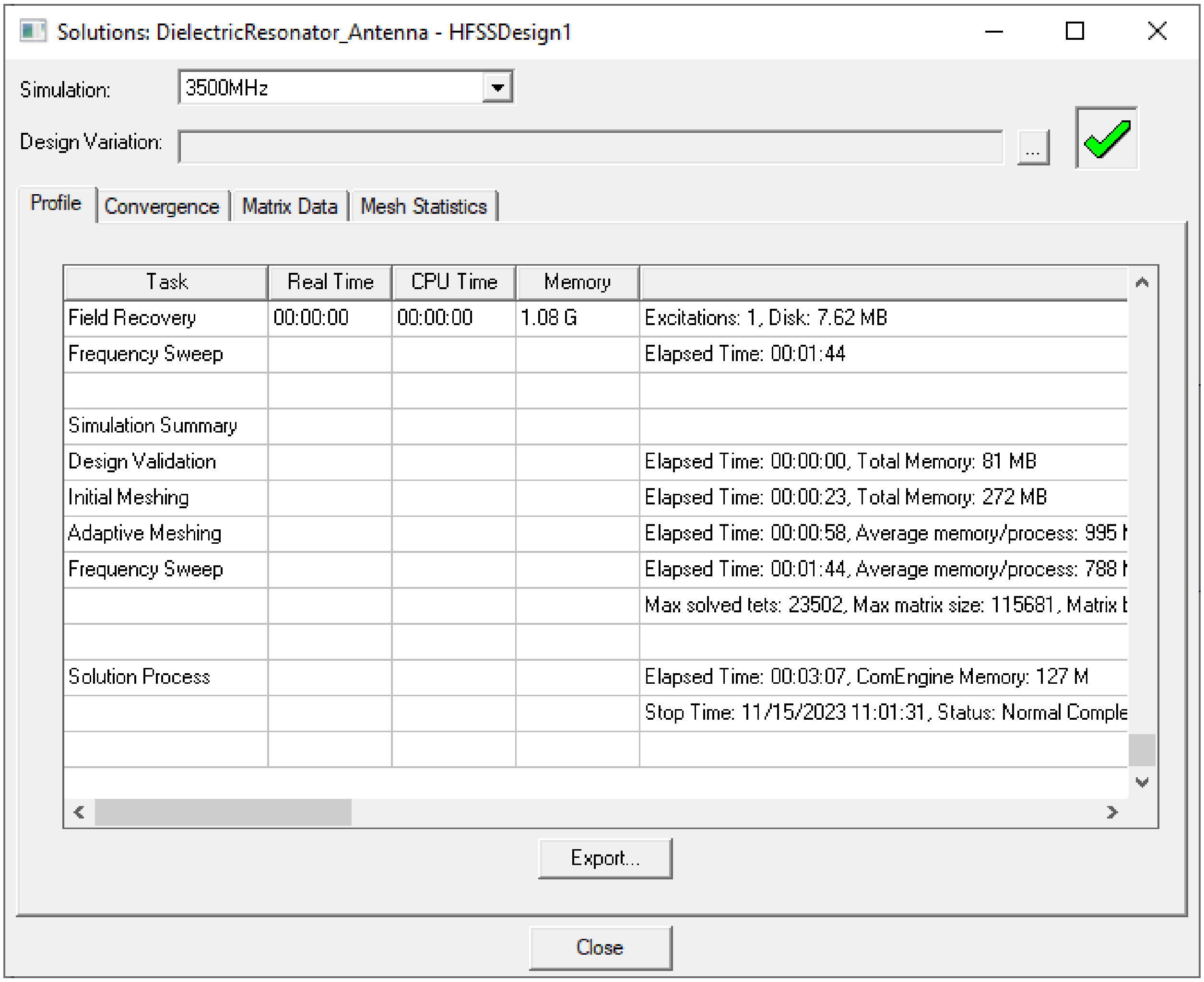
At any time during or after the solution process, you can examine the computing resources or profile data that were used by Ansys Electronics Desktop solvers during the analysis. The profile data is essentially a log of the tasks performed by Ansys Electronics Desktop during analysis. The log indicates the length of time each task took and how much physical and disk memory was required.

1. In the project tree, right-click a solution setup, and select **Profile**.

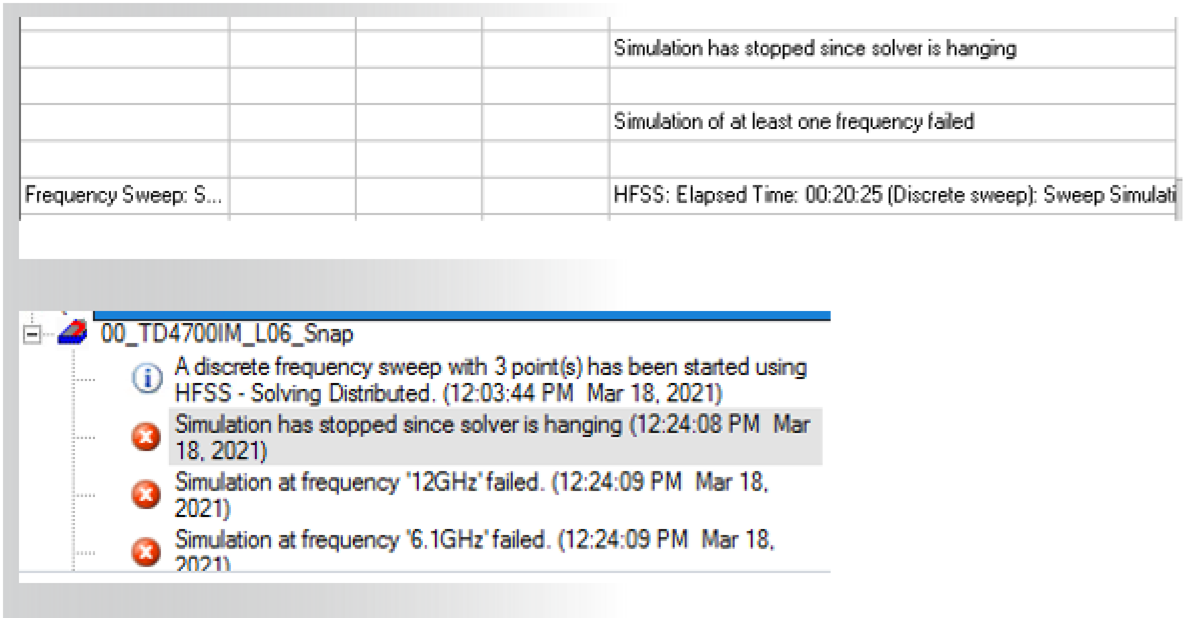
   The **Solutions** dialog box appears, with the **Profile** tab selected. You can also click the **Solution Data** icon on the ribbon to open the **Solutions** dialog box, and then select the **Profile** tab.



   The displayed data depends on the type of problem and solution setup.



If the simulation hangs, equivalent messages appear in the Message window and the Solution **Profile** tab.



2. If one or more dependent setups exist, you can select the desired profile information from the **Simulation** drop-down menu. In general, the information displayed includes:

Task	Lists the type of task that was performed. Tasks include 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 has the form "Solver *pdsn*" (e.g., Solver MRS2, Solver DCS4-L2), where: • *p*, the precision type is: M (mixed for direct solver), or Mxx (for distributed direct solver), or D (double for direct and iterative solver) or E (extended precisions solve) • *d*, the matrix data type is: R (real) or C (complex) • *s*, the symmetry type is: S (symmetric), A (asymmetric), or H (hermitian) • *n*, the number of processors used. You specify the number of available processors on the local machine in the solver options. If a solve does not use all available processors (local or distributed), the number reported may be less than the number available. If a simulation uses the iterative solver, the Solver designation can include a level indicator appended. to an Iterative solver designation (L2 in the example above). The higher the Level number the lower the memory, you will never see L1 (this would be equivalent to direct solver. And a first order solve will only display L2 since it only has one level of order to go down for preconditioning. A second or mixed order solve may display L3 depending on the mesh quality. If the solver switches from the Iterative Solver to the Matrix solver, you see a Matrix solver warning: Switch from Iterative Solver to Direct Solver.
Information	The matrix solver information line includes, for example, Disk = 0 KBytes, matrix size 11137 , matrix bandwidth 20.3. • **Disk** – The amount of hard disk space used during the calculation of the matrix solution. If the disk usage for matrix solver is non-zero in profile, it usually indicates off-core matrix solver. If the matrix solver must solve off-core, smaller blocks of the data to be solved are created on disk, each block is then solved in physical memory, and then the matrix solution is reassembled. As a result of this additional processing, the time required to calculate a solution is higher. • **Matrix size** – The size of the matrix that was solved (the number of unknowns) • **Matrix bandwidth** – An FEM matrix is a sparse matrix. The solver only stores the non-zero entries. The matrix bandwidth is the average number of non-zeros per row. It gives an idea of the sparsity of a FEM matrix.Storage for the sparse matrix is proportional to the total number of nonzeros = #rows x bandwidth. The higher the bases order, the larger the bandwidth. • **Iterations** – For the Iterative Solver.

To export the profile data:

1. Open the **Solutions** dialog box and select the **Profile** tab.
2. Click **Export Profile**.

   This opens a file save dialog box that lets you provide a file name and location.
3. Click **Save**.

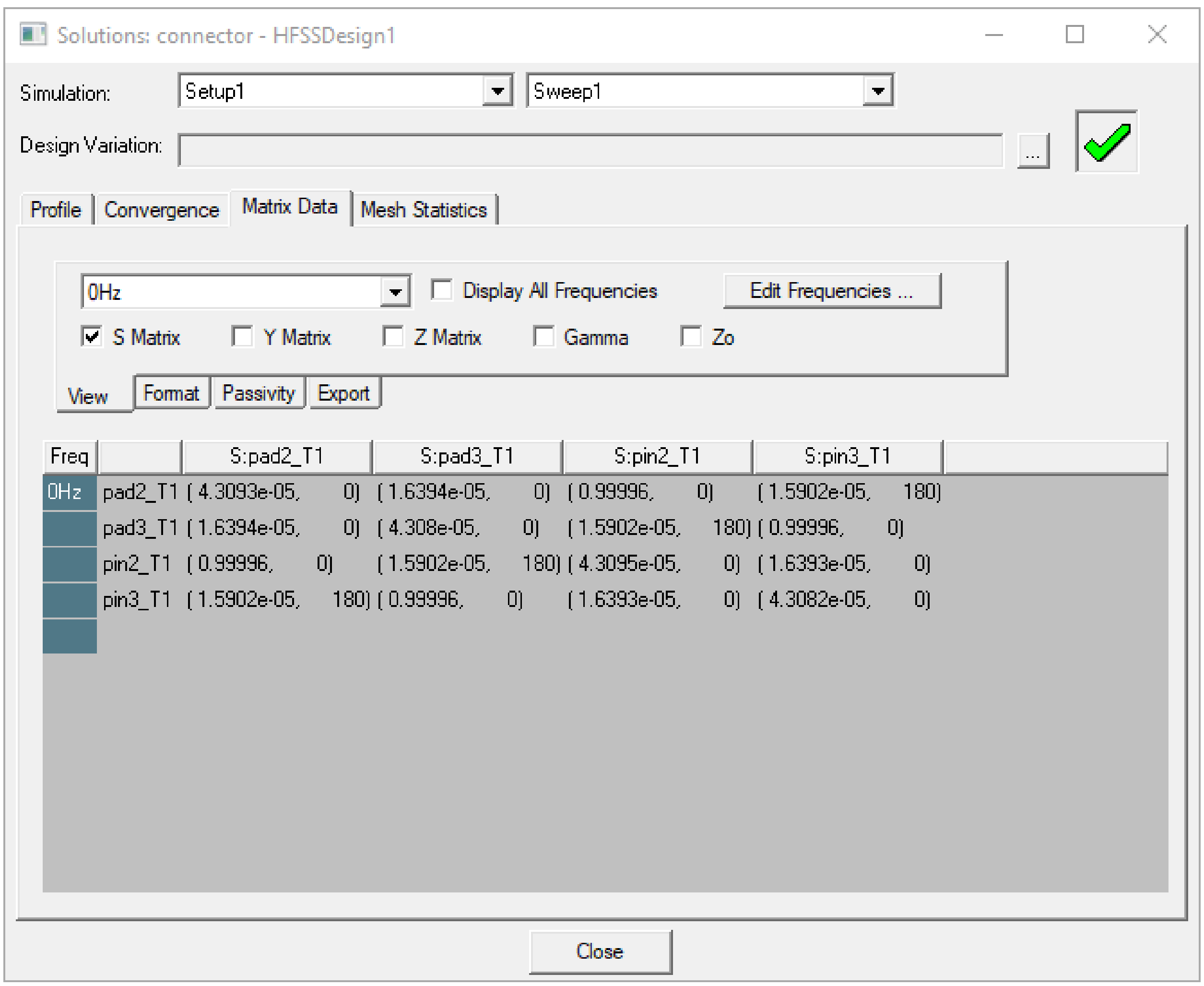
   The data is saved in a text file with a .prof extension.

Viewing Matrix Data

To view matrices computed for the S-parameters, impedance, and propagation constants during each adaptive, non-adaptive, or sweep solution:

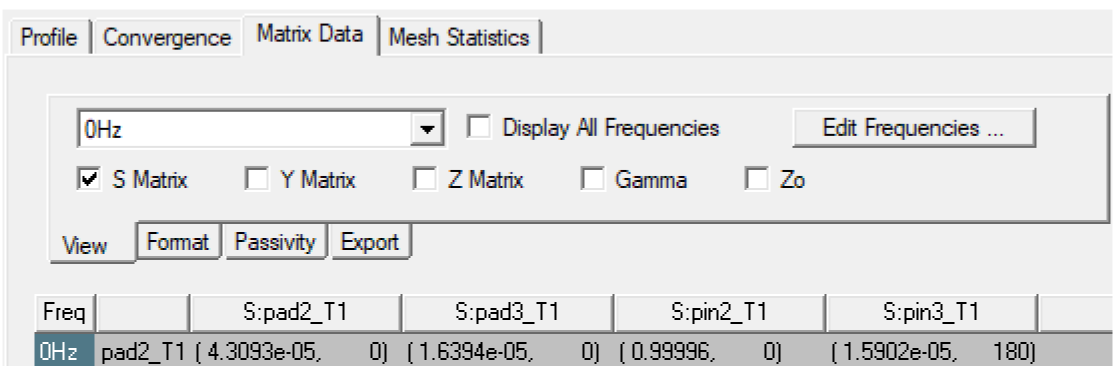
1. In the **Project** tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu, or on the **Results** tab of the ribbon, click **Solution Data**.

   The **Solution Data** dialog box appears. The **Matrix Data** tab is selected.



2. Select which pass to view in the drop down list to the right of the **Simulation** list. To select a specific adaptive pass:
   a. Choose **Adaptive Pass**.
   b. On the **View** tab, select the pass you want from the **Pass** list.
3. If design variations exist, in the **Design Variation** text box, specify the design with the matrices you want to view.

   Optionally, choose a design variation solved during an Optimetrics analysis from the **Set Design Variation** dialog box. This lists all the solved variations in the design. This dialog box is accessible from the **Solution Data** window by clicking the ellipsis button on the right of the Design Variation field, and via the **[solver]** > **Results** > **Apply Solved Variation** command.
4. In the **Simulation** drop-down menu, click the solution setup and solved pass - adaptive, single frequency solution, or frequency sweep - for which you want to view matrices.
5. Select the type of matrix you want to view: **S-matrix**, **Y-matrix**, **Z-matrix**, **Gamma**, or **Zo** (characteristic impedance.) The available types depend on the solution type.



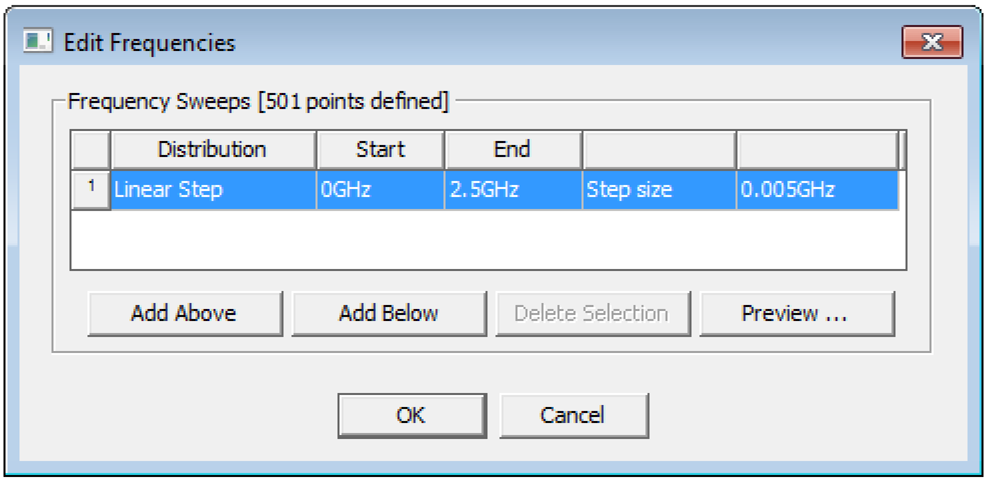
6. Select the solved frequencies to display:
   - To display the matrix entries for all solved frequencies, select **Display All Freqs**.

     If **Display All Freqs** is enabled, the drop-down frequency select list is disabled.
   - To show the matrix entries for a selected solved frequency, ensure that **Display All Freqs** unchecked and use the dropdown list to select the solved frequency for which you want to view matrix entries. You can use a scroll bar for selecting from long frequency lists.

     For adaptive passes, only the solution frequency specified in the **Solution Setup** dialog box is available. For frequency sweeps, the entire frequency range is available.
   - To insert or delete one or more displayed frequencies, click **Edit Frequencies**.

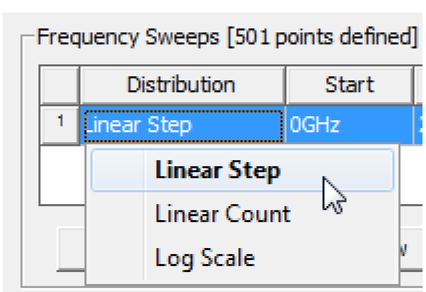
> **Note:**
>
> This command is only available if the sweep type is Fast or Interpolating.

Clicking **Edit Freqs** displays the **Edit Frequencies** dialog box. The current values are displayed in a table.



> **Note:**
>
> Changes to the Start value and End value cannot be outside of the initial range. No message is issued: rather the range is implicitly restricted.

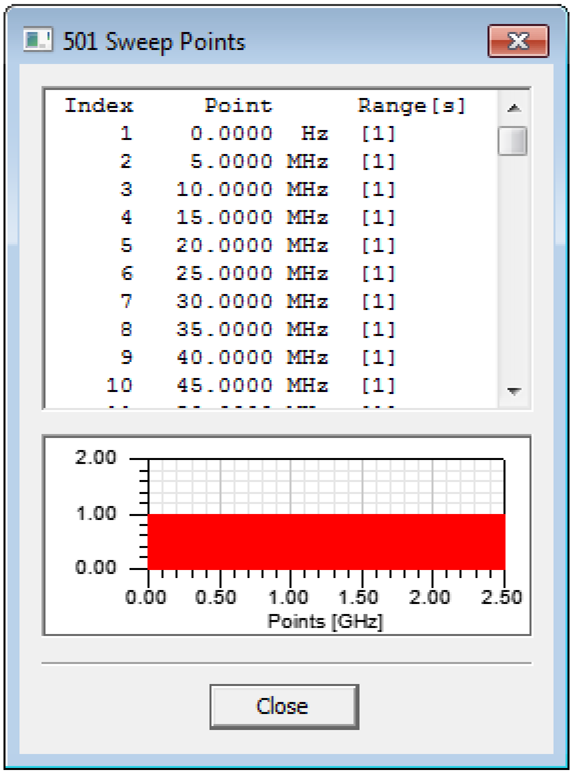
Use the **Add Above** or **Add Below** button to add a new frequency to the table above or below the currently selected value. If no value, or the start value is selected, the new frequency repeats the current Start value and increments the count at the start a new row. You can select the type of distribution for a new Row by selecting from the drop down.



You can then edit the **Start** and **Stop** fields as appropriate for the new row.

The **Delete Selection** button enabled only if a value is selected. **Delete Selection** removes the selected value and decrements the Number of Values field.

The **Preview** button opens a dialog showing the defined sweep points.

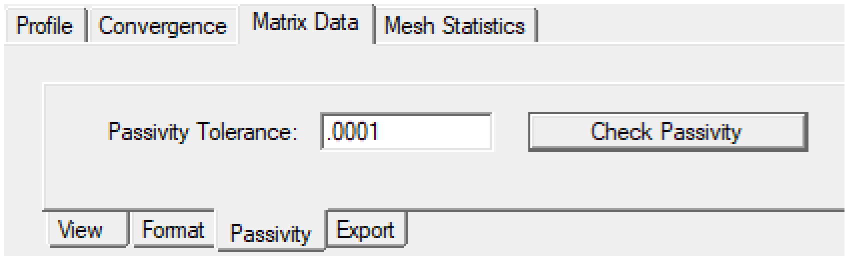


7. Select the display format — On the **Format** tab, select from the drop down menu which has the options **Magnitude/ Phase (deg)**, **Real/ Imaginary**, **dB/Phase (deg)**, **Magnitude**, **Phase (deg)**, **Real**, **Imaginary**, and **dB** — in which to display the matrix information and then in the next drop down menu, select **Terminal Data** or **Differential Pairs**.
8. Click **OK** to apply the changes to the **Solutions** dialog **Matrix Data** tab and close the **Edit Sweep** dialog box, or **Cancel** to close the dialog without applying the changes.

   If you choose to export the matrix data for the Fast or Interpolating sweep after modifying the frequencies in the **Edit Frequencies** dialog box, only those frequencies displayed under the **Matrix Data** tab will be exported.

   The data is displayed in the table. By default, wave ports are listed in alphabetical, then numerical order, just as they appear in the excitation tree. To change the port order, change setting for Default Matrix sort order in the HFSS General options. You may also want see how you can Reorder Matrix Data.

9. Optionally, select the **Passivity** tab and **Check Passivity**.



This passivity check tests whether the S-parameter data from HFSS is passive or not. If the S-Matrix is not passive at one or more frequencies, this check displays a dialog saying that the solution is passive for all frequencies or that identifies the worst frequency violation and identifies the passivity in that case. A uniform renormalization of 50 ohms is performed on the solution data for Passivity checking.

Selecting the Matrix Display Format

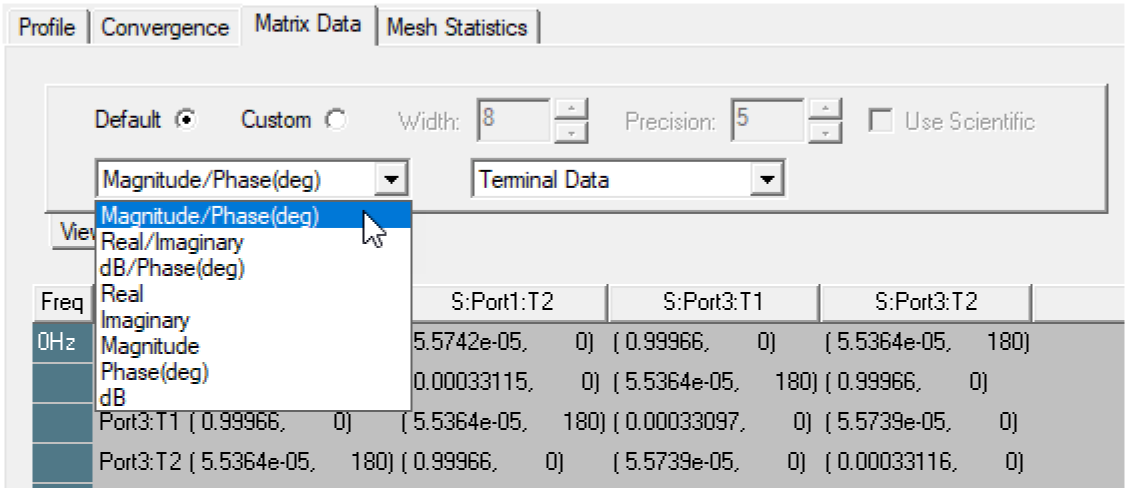
The **Solutions** dialog offers a range of formats for the Matrix display. The available formats depend on the matrix type being displayed. When selected, dB formatting only applies to S - matrix data, even if other matrix types are displayed. The column heads in the display identify the format for the matrix type. You can display matrix data in the following formats.

Magnitude, Phase (deg)	Displays the magnitude and phase (in degrees) of the matrix type.
Real, Imaginary	Displays the real and imaginary parts of the matrix type.
dB, Phase (deg)	Displays the magnitude in decibels and phase in degrees of the matrix type.
Magnitude	Displays the magnitude of the matrix type.
Phase (deg)	Displays the phase in degrees of the matrix type.
Real	Displays the real parts of the matrix type.
Imaginary	Displays the imaginary parts of the matrix type.
dB	Displays the magnitude in decibels of the matrix type.

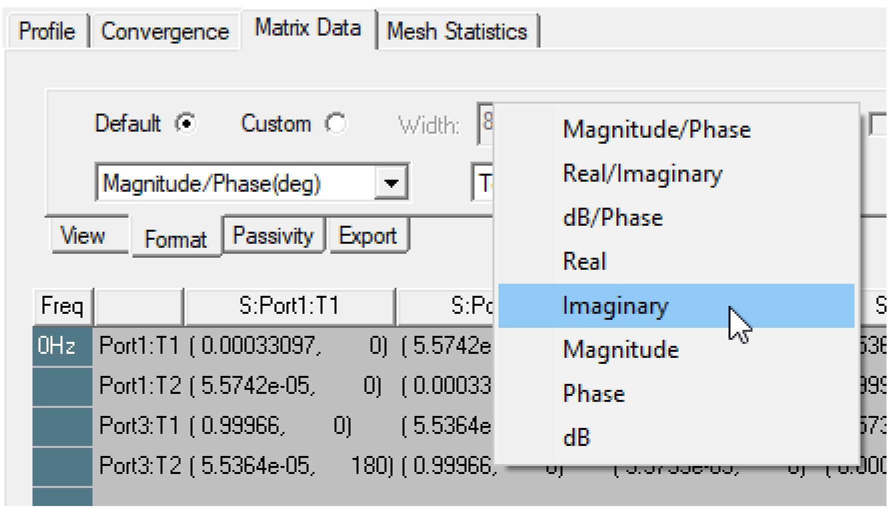
After you select a format, choose whether to display terminal data or differential pairs.

In the **Matrix Data** tab of the **Solutions** window, open the **Format** tab to select the display format in one of two ways:

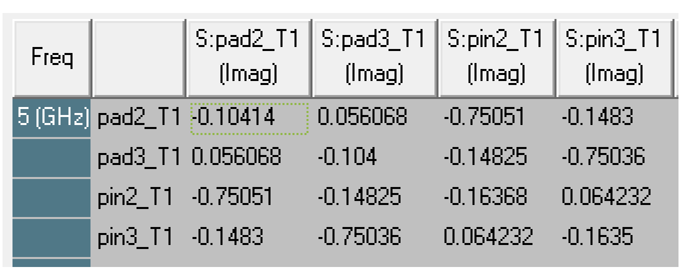
- Select the drop-down menus under the formats:



- right-click the column headings to display a pop-up format menu.



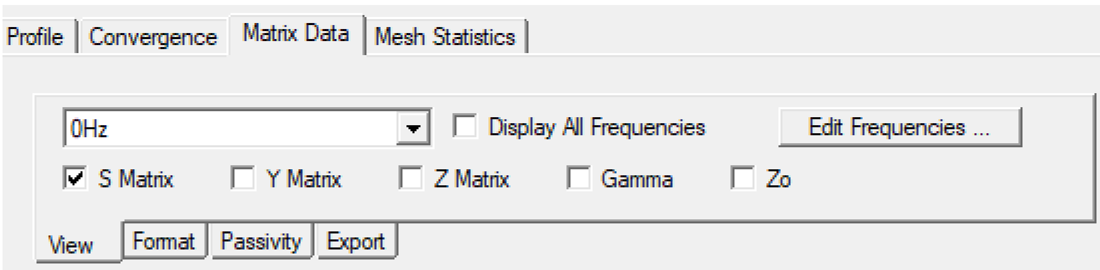
- Selecting from this menu also adds the name of the selected format to the column head, if the format is anything other than dB.



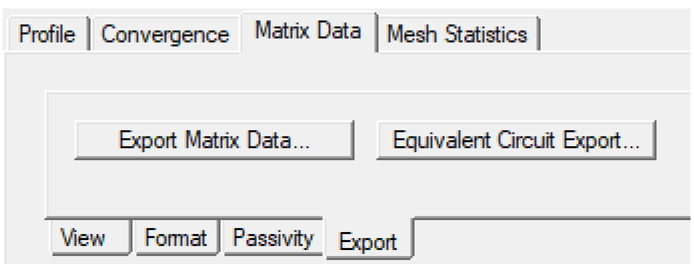
Exporting Matrix Data

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

The **Solution Data** window appears. The **Matrix Data** tab is selected. You can also access the **Solution Data** window from the **Results** tab on the ribbon by clicking the **Solution Data** icon.

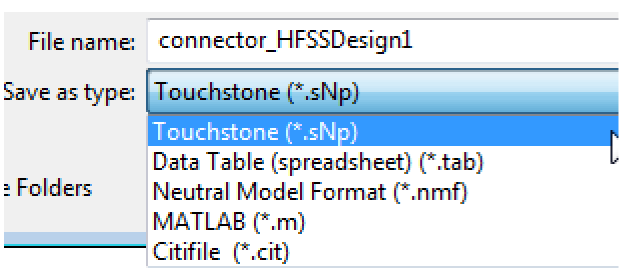


2. Select the type of matrix you want to view: **S-matrix**, **Y-matrix**, **Z-matrix**, **Gamma**, or **Zo** (characteristic impedance.)
3. Select the **Export** tab and click **Export Matrix Data**.

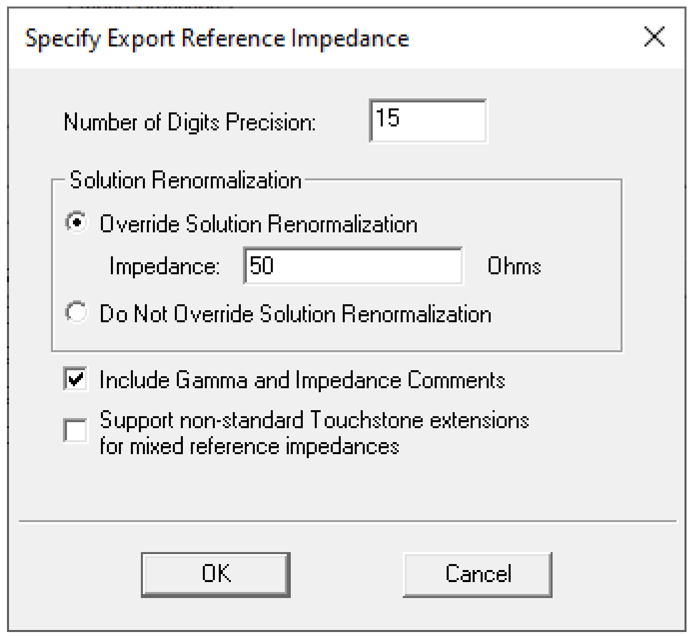


A file browser appears.

4. Type the name of the file you are exporting to in the **File name** text box.
5. Select one of the following file formats from the **Save as type** drop-down menu:



Format	Type	Description
(spreadsheet) *.tab	Data Table	A text file in which the elements of the S-matrix are arranged in a series of columns that are tab-separated and include a first row of headings. The file may be imported into a spreadsheet or similar utility.
***.sNp**	Touchstone	A Touchstone S-parameter file in which the number of ports is indicated by *n*. For example, a Touchstone file with one port would have the file extension .s1p. When you export this format, you are presented with a dialog.



You can specify:

- **Number of Digits Precision** (Default 15)
- **Override Solution Renormalization**. This refers to overriding the renormalization impedances that may have been asked for in the port setup. If so, the export renormalizes impedance
- **Do Not Override Solution Renormalization.** This leaves any port setup renormalization options in place, and grays out the Impedance setting on the dialog.
- **Include Gamma and Impedance Comments**
- **Support for non-standard Touchstone extensions for mixed reference impedances**. Writing gamma and Z_0 is not Touchstone standard, but rather an HFSS "feature" that only HFSS will recognize.

		If you want to export raw S-Parameter data for later use, you may choose to not renormalize the solution. If all ports and associated modes/terminals are normalized to the same impedance and you choose **Do Not Override Solution Renormalization** during export, the Touchstone file header will indicate the normalized impedance. The comment header in the Touchstone file lists the port and mode numbers to show which column contains which port name (in case of confusion between alphabetical and force repriority ordering of ports and associated modes). `! Touchstone file exported from HFSS 2024.2` `! File: D:/Program Files/AnsysEM/v242/Win64/Examples/HFSS/Signal Integrity/package.aedt` `! Generated: 1:56:02 PM Apr 01, 2020` `! Design: HFSSDesign1` `! Project: package` `! Setup: Setup1` `! Solution: Sweep1` `!` `# GHZ S MA R 50.000000` `! Terminal data exported` `! Port[1] = P01__NET179__T1` `! Port[2] = P02__NET178__T1` `! Port[3] = P03__NET178__T1` `! Port[4] = P04__NET179__T1`
***.nmf**	Neutral Model format	Neutral Model file format defined by the MAFET Consortium. When you save to this format, if the design includes variables, you are presented with the NMF Parameters dialog that lets you select which variables to select as parameters in the NMF file. Non-selected variables will be give the constant value shown. You are then presented a dialog that lets you specify the number of digits precision for the renormalizing impedance.
***.m**	MATLAB	The Mathworks' MATLAB file in which the elements of

		the S-, Y-, or Z-matrix are arranged in a series of rows. Matlab only allows creation of complex numbers by specifying real and imaginary parts, not by specifying magnitude and phase.
***.cit**	Citifile	Common Instrumentation Transfer and Interchange file format. It is an ASCII format defined by instrument and CAE designers.

> **Note:**
>
> For Touchstone files, you no longer see a **Combine Sweeps** option on the **Export Network Data** solution dialog box. This is because the current Sweep setup allows you to define multiple sweeps for a single simulation, meaning any needed combining is defined with the sweep setup.

6. Click **Save**.

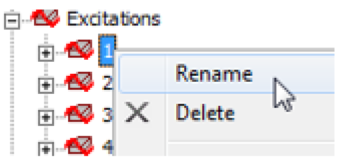
The data is exported to the file.

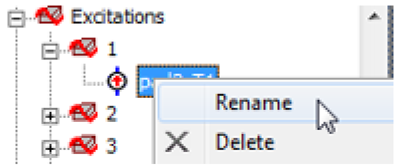
- By default, wave ports are listed in alphabetical, then numerical order, just as they appear in the excitation tree. You can change this order to creation order and back without invalidating the solution on the HFSS Options dialog box.
- If you select Touchstone format, you are first presented with a dialog in which you can specify the Number of Digits precision, and Override the Solution Renormalization. If so, you can specify the export renormalizing impedance (an integer value). Here you also can specify and whether to include **Gamma and Impedance Comments**, and **Number of Digits Precision** (Default 15).
- If you select Neutral File Format, you are presented with a **Specify Export Renormalizing Impedance** dialog that lets you specify the Number of digits precision for the save file.

> **Note:**
>
> If you modify the display of solved frequencies in an Interpolating or Fast sweep under the **Matrix Data** tab (by clicking **Edit Freqs** and then modifying the values in the **Edit Sweep** dialog box,) only those frequencies listed will be exported to the file.

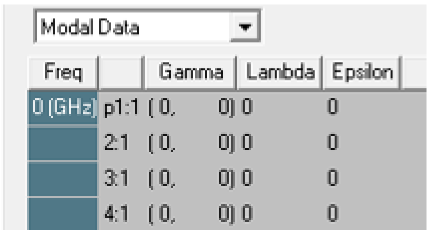
Renaming Matrix Data

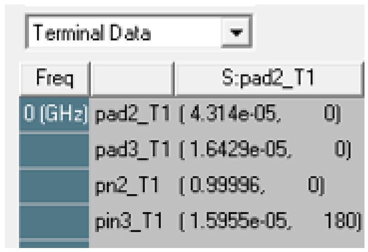
In the project tree, you can right-click a port excitation to rename it.





When you rename a port excitation, the associated Matrix data is reordered so that it can be presented in the same manner. The reordering is done to match the tree-sort order presented for the ports (renamed matrix data is reordered so that alphabetic values appear before numeric values).





Exports of the matrix data are ordered in the same manner. This reordering is conducted as part of post processing and does not force a re-solve.

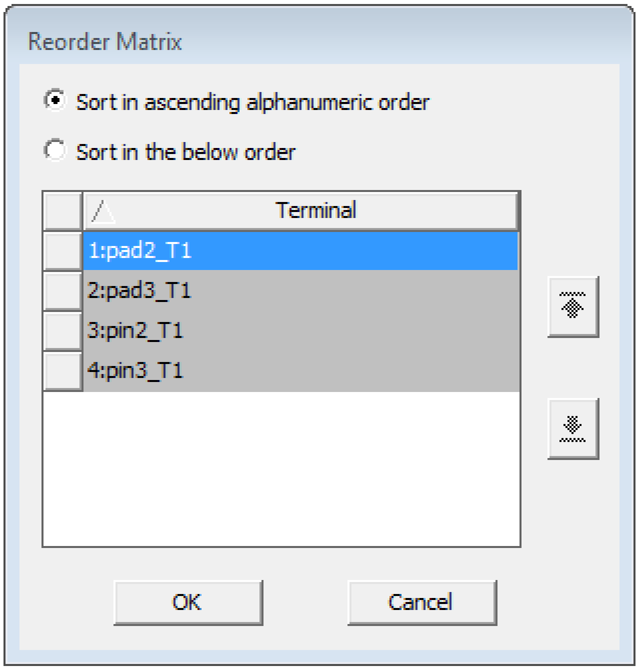
Reordering Matrix Data

HFSS lets you reorder the matrix data as a post-processing step.

1. To re-order the matrix data, either as the default ascending alphanumeric order, or a user specified order, click **HFSS>Excitations>Reorder Matrix**, or right-click Excitations in the Project tree, and click **Reorder Matrix** on the shortcut menu.

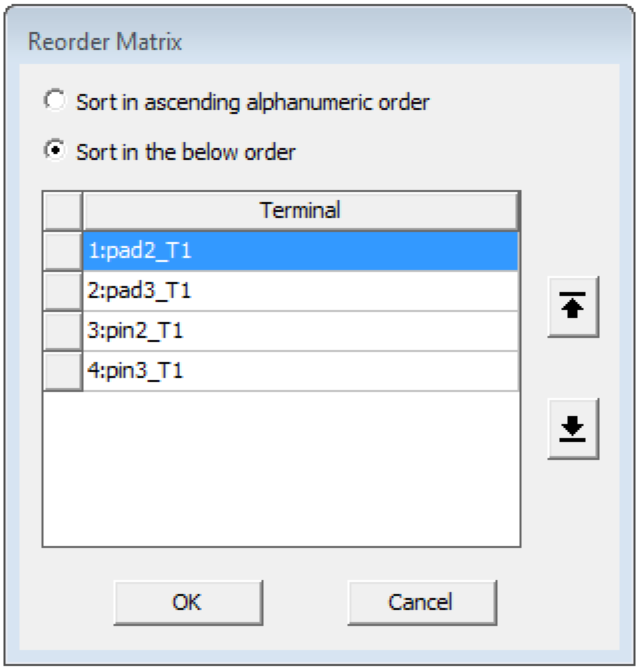
This displays the **Reorder Matrix** dialog. It lists the ports for a modal solution, or the terminals for a terminal solution. For Transient solutions, the dialog displays separate lists for

Active and Passive ports.



2. You can select the radio button for **Sort in ascending alphanumeric order**, or **Sort in the below order**.

If you select **Sort in the below order**, you can select any port or terminal to enable the up and down arrow keys. Use the keys to move the ports into any desired order. In the case of Transient, the arrow keys operate only within a partition, that is only within Active or Passive lists.



You can also use Ctrl+click to select multiple arbitrary ports, or hold Shift to select a range of ports. Clicking outside the list deselects all selections.

> **Note:**
>
> If there are differential pairs, the sort order is still specified) in terms of the underlying terminal names, but the entries that make up the pair should appear in the appropriate sort location for the terminals that are used to define them.

3. Click **OK** to accept the order specified.

For Transient solutions, removing an active source will not affect the solve but might affect the matrix order. Adding an active source will require a resolve if that source has not already been solved.

Exporting Equivalent Circuit Data

You can export S-parameter data from a Driven Terminal solution to PSpice, HSPICE, Spectre or Maxwell Spice format. Importing the new data file to PSpice, HSPICE, Spectre or Maxwell Spice will enable you to include wave effects in the circuit simulations. You can also export a W-Element model for a port.

Note:

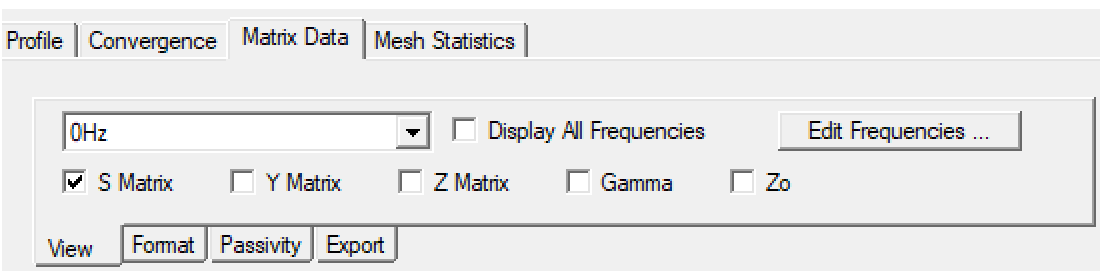
You must have a frequency sweep solution and five or more frequency points to successfully export an equivalent circuit data file. See the *Choosing Frequencies for Full-Wave SPICE* topic of the help for suggestions about the frequency range of the sweep.

The GUI lets you export full-wave Spice for a model that contains differential pairs, but it will silently export the data in its original single-ended form. The full-wave Spice model is a "broadband" equivalent circuit (that is, its S-parameters match those of the solution across the whole frequency sweep range.)

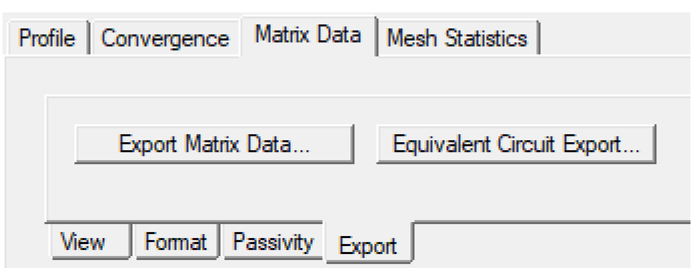
Certain discrete sweeps permit Full-Wave SPICE exports. It is allowed if the discrete data is evenly spaced, includes DC, and has at least 500 frequency points.

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

The **Solution Data** window appears. The **Matrix Data** tab is selected. You can also access the **Solution Data** dialog from the **Results** tab on the ribbon by clicking the **Solution Data** icon.

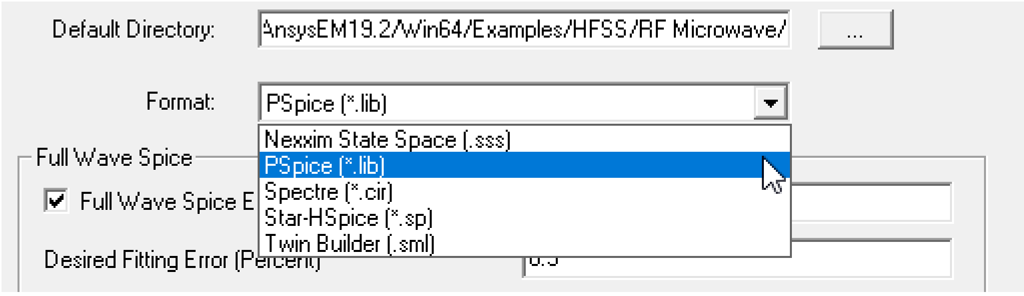


2. Select the **Export** tab and Click **Equivalent Circuit Export**.



The **Equivalent Circuit Export Options** dialog box appears.

3. Type the name or browse to the directory in which you want to store the data.



4. Click one of the following formats in the **Format** list:

PSpice (*.lib)
Nexxim State Space (.sss)
Spectre (*.cir)
Star HSpice (*.sp)
Twin Builder (.sml)

Your format selection affects the options available under **Full Wave Spice Export**. When Twin Builder format is requested, both *.sml and *.png are created. The latter contains the image in GIF format.

Note:

The Export to Twin Builder here does not use the same settings as Nexxim or Network Data Explorer. If you intend to export to Twin Builder, you should use the NdExplorer.

5. If the **Full-Wave Spice Export** check box is enabled, you can select it. Checking the box enables the text field for the file name, and depending on the format selection, other options may be enabled.
6. Desired Fitting Error (percent) has a default value of 0.5. You can edit this.
7. Maximum Order has a default value of 10000.
8. HFSS supports Full Wave Spice Export from a driven modal design as long as all ports have exactly one mode each. However, HFSS does not support definition of differential pairs in a driven modal design.
9. By default **Use Common Ground** is checked and produces circuit models with a "common" (suppressed) ground terminal. Uncheck Use Command Ground to apply one negative reference terminal per port.
10. Optionally, select **Enforce Passivity**. Selecting this enforces passivity in the output file. Passive devices can only dissipate or temporarily store energy, but never generate it.

(You can also check passivity from the **Matrix Data** tab using the Check Passivity button.)

This option is useful in cases where the transient simulation fails due to passivity violations in the circuit model. This circuit model is based on fitting a rational function to the S-parameter data computed by the field solver. Small errors in the data fitting can result in non-passive behavior. Selecting the **Enforce Passivity** option will take more CPU time, but ensures that the resulting model will be passive. The **Enforce Passivity** check box uses "Iterated fitting of passivity violations (IFPV)" method to do the passivity enforcement.

The passivity check tests whether the S-parameter data from HFSS is passive or not. For more information see Passivity.

11. Optionally, select **Lumped Element Export (Low Bandwidth)** if you want to save the data as a low-frequency circuit model using simple lumped elements (resistors, capacitors, inductors, and dependent current sources). The low-bandwidth model is only going to be accurate in a limited frequency range around the adaptive solution frequency

This option is not enabled for Spectre export.

12. Optionally, select **Partial Fraction Expansion for Matlab** if you want to specify a file that expands the partial fractions for use in Matlab. The partial fractions involved describe the frequency response of the low-bandwidth model from the previous step.
13. You can also select **Combine Sweeps** to select and combine available sweeps into a single output file.

By option, for Nexxim State Space, Spectre, and Star-HSpice, you can Export W-Element Data.

14. Click **OK**.

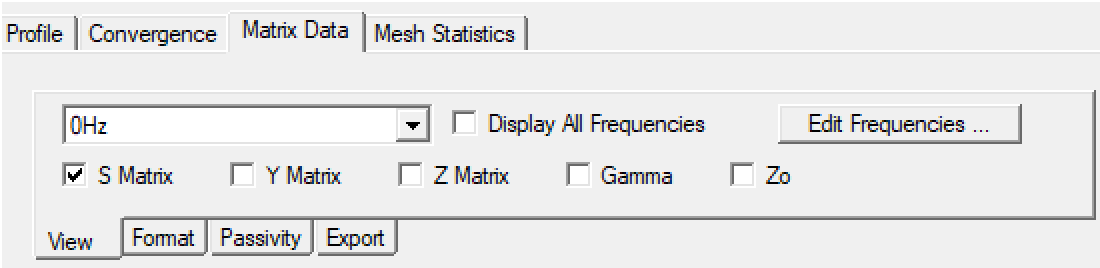
The S-matrices are written to the data file that you specified in the equivalent circuit data format.

Exporting W-element Data

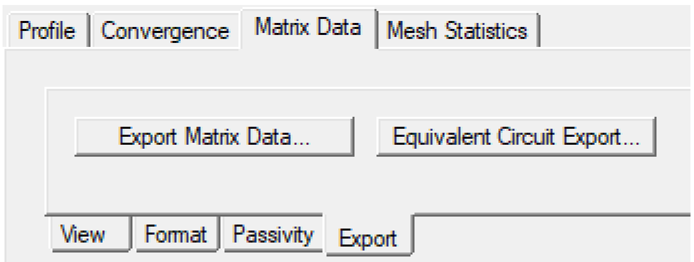
It is possible to extract a W-element model for a port. This W-element model can be used in a SPICE model to represent a length of transmission line of the same cross-section as the port. A W-element model can be extracted for a port only solution and for a full 3D solution.

1. In the project tree, right-click the solution setup of interest, and then click **Matrix Data** on the shortcut menu.

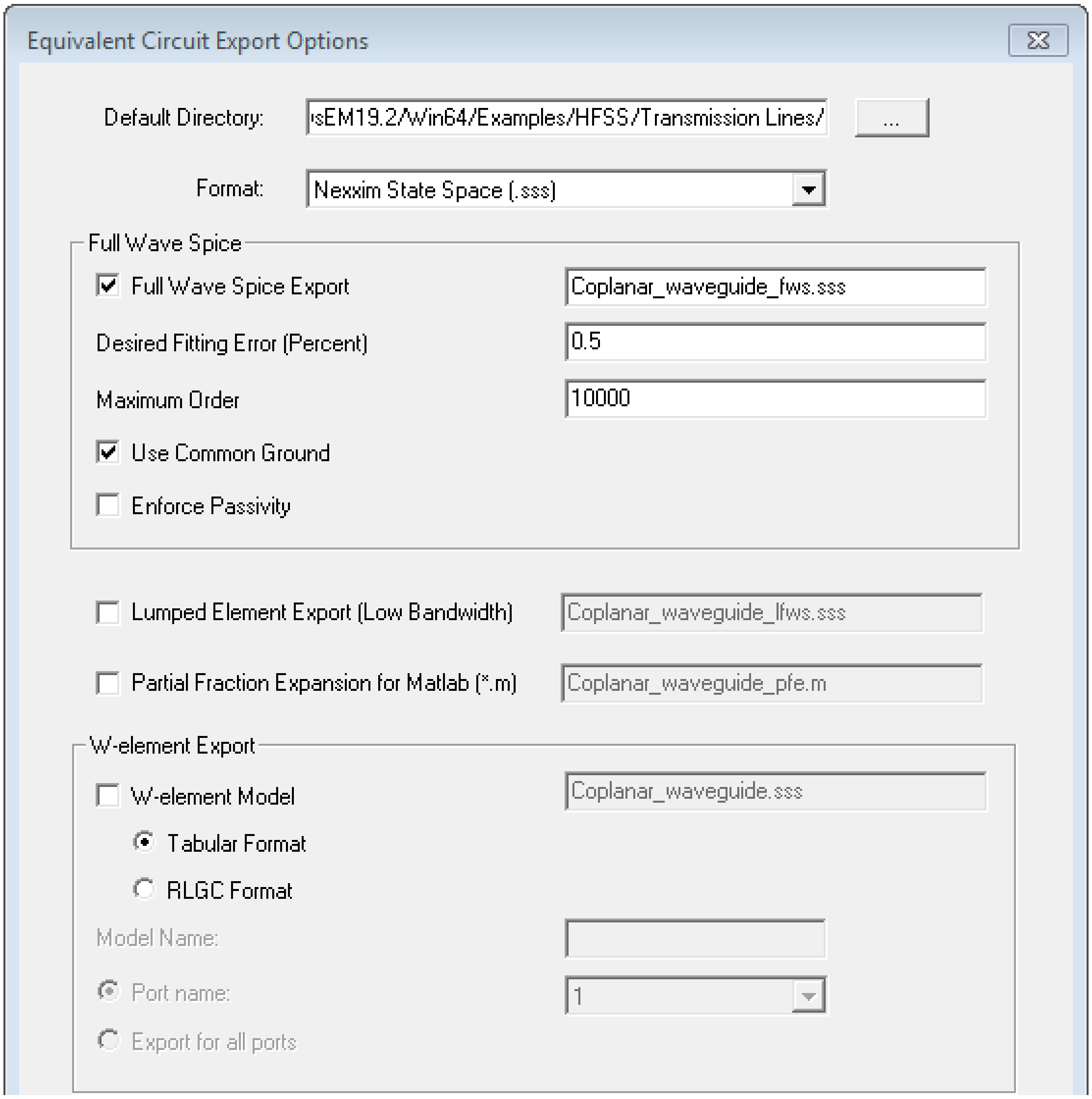
The **Solution Data** window appears. The **Matrix Data** tab is selected. You can also access the **Solution Data** dialog from the **Results** tab on the ribbon by clicking the **Solution Data** icon.



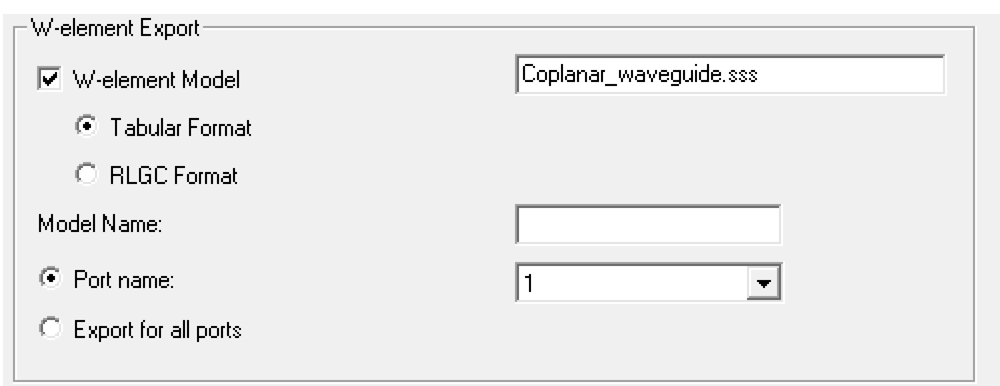
2. Select the **Export** tab and click **Equivalent Circuit Export**.



The **Equivalent Circuit Export Options** dialog box appears. If you select the format as Nexxim State Space, Spectre, or Star-HSpice, at the bottom of the dialog you see the W-element model check box enabled.



3. Click the W-element model check box to enable the W-element fields.



4. The W-element model name field has the project name by default. You can change this if desired.
5. Choose the format as Tabular Format (the default) or RGLC format for W-element export.

Tabular Format: provides a unique RLGC model for each frequency in the solution.

RLGC Format: provides a RLGC fit over a frequency range based on Ro, Lo, Go, Co, Rs and Gd parameters.

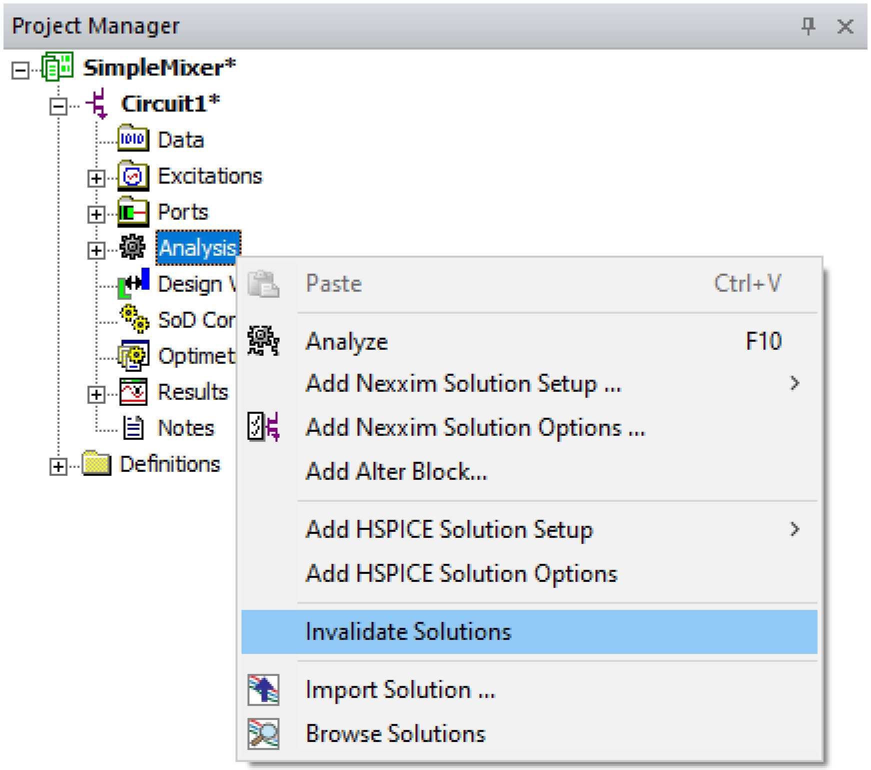
> **Note:**
>
> For the RLGC Format, if only a single frequency solution is selected (e.g., LastAdaptive) then Rs and Gd parameters are ignored.

6. In the **Model name** field, provide a model name.
7. Either select the port from the **Port Name** pull down, or, to export a W-element model for all ports, select the **Export for All Ports** radio button.
8. Click **OK**.

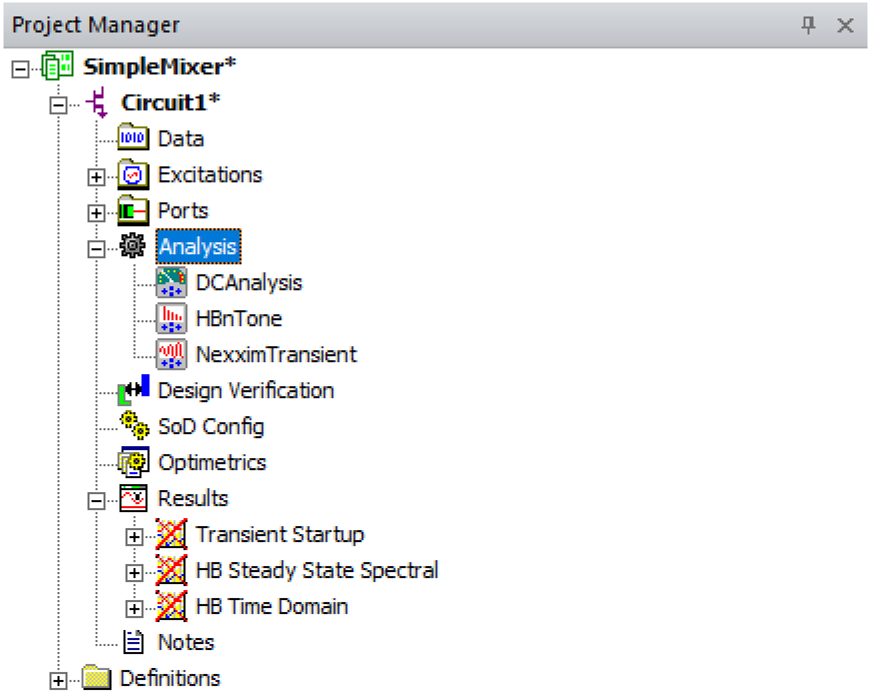
The W-element model is written to the data file that you specified.

Invalidate Solutions

If necessary, the user can erase all solution data, including mesh, matrix, and fields data for all adaptive passes and frequency sweeps. From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Invalidate Solutions**.



Any post-processing results, reports, and/or field overlays that include deleted data will be marked with a red "**X**" in the Project Manager window. Any results marked with an "**X**" is invalid until new solution data is generated.



Deleting Reports

To use **Delete All Reports**:

1. Click **Circuit**> **Results**> **Delete All Reports.** You can also right-click on the Results folder in the Project Manager, to display the shortcut menu, and click **Delete All Reports.**

   All items under the Results folder in the Project Manager are removed.

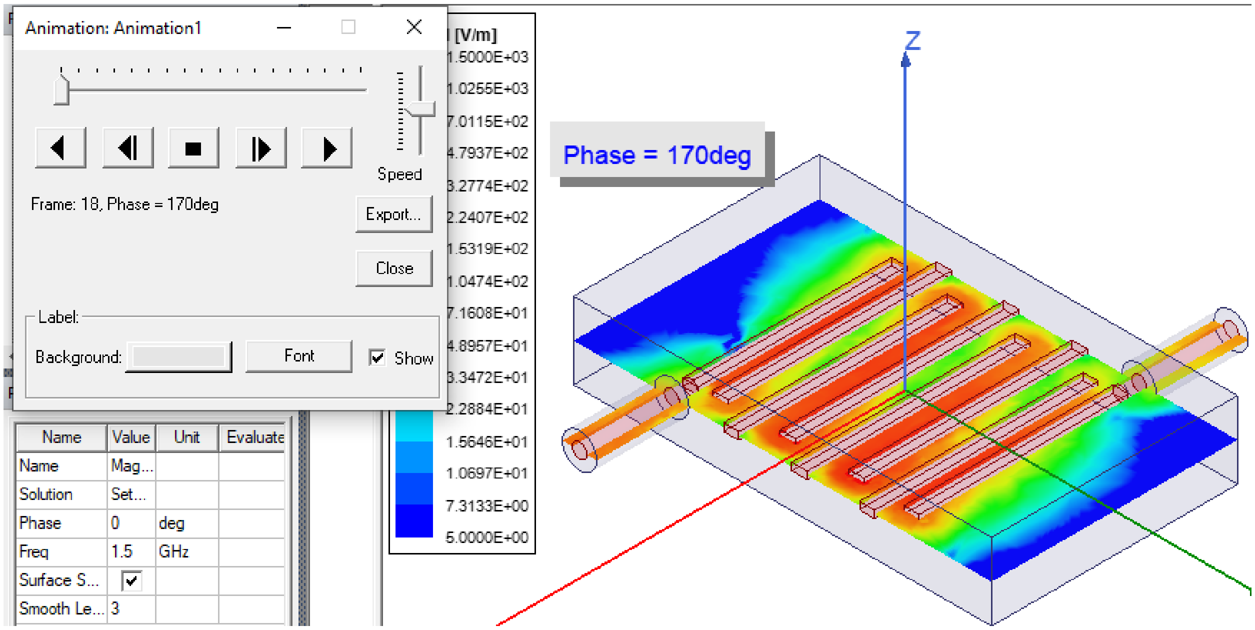
To use **Delete** for a selected report:

1. Right-click a report icon in the Project Manager to display the shortcut menu.
2. Click **Delete** to remove this report.

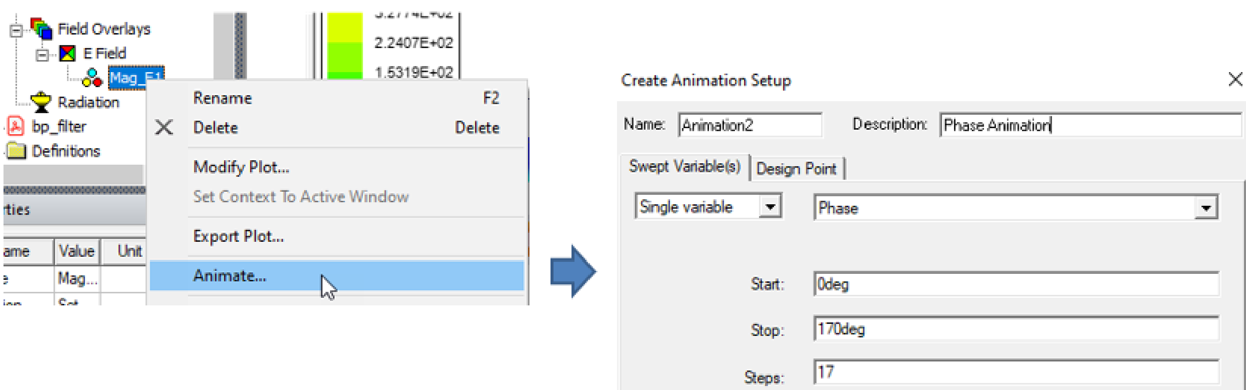
> **Warning:**
>
> Solution data that has been deleted cannot be recovered!

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 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.

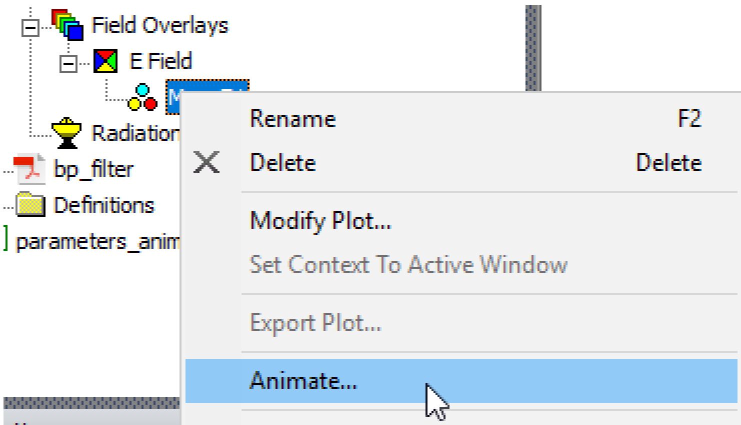
Create a field plot.

Procedure:

To animate a plot with respect to the phase of the plotted field:

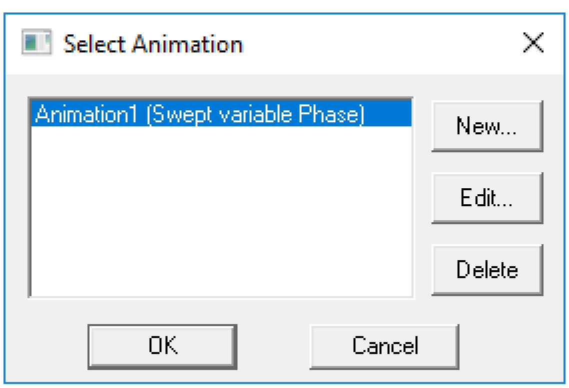
1. Create a field overlay plot to animate.
2. Select the plot in the **Project** tree, and click **HFSS**>**Fields**>**Animate...**, or right-click in the **Modeler** window, and click **View>Animate...**.

Or right-click the field overlay plot of interest and click **Animate...** from the shortcut menu,



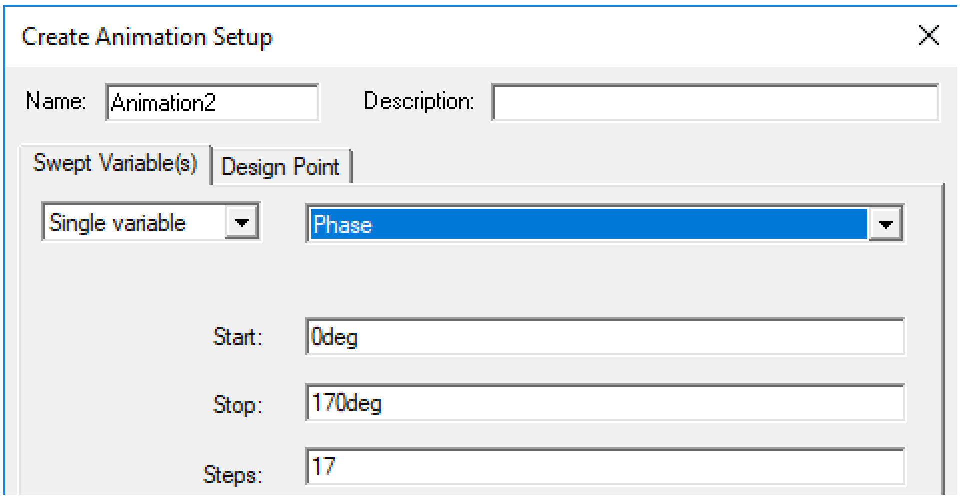
Or select the field overlay plot of interest, select the **View** tab of the ribbon, and click the **Animate** icon.

If you already created an animation, the **Select Animation** dialog box appears. Selecting an existing animation from that list starts it.

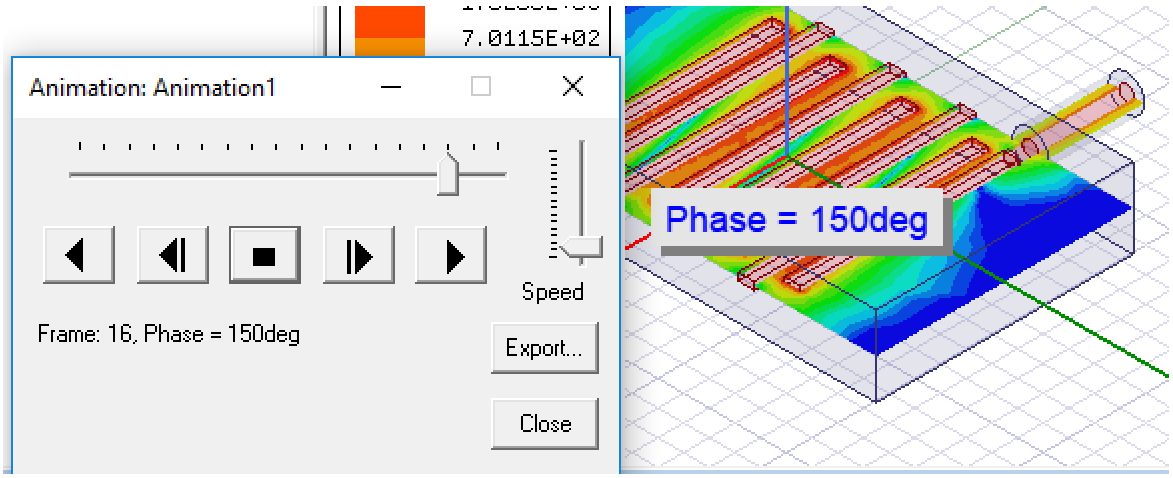


Selecting an existing animation from that list starts it when you click **OK**. Click **Edit...** to open the **Modify Animation Setup** dialog.To create a new animation, click **New**.

The **Create Animation Setup** dialog box appears.

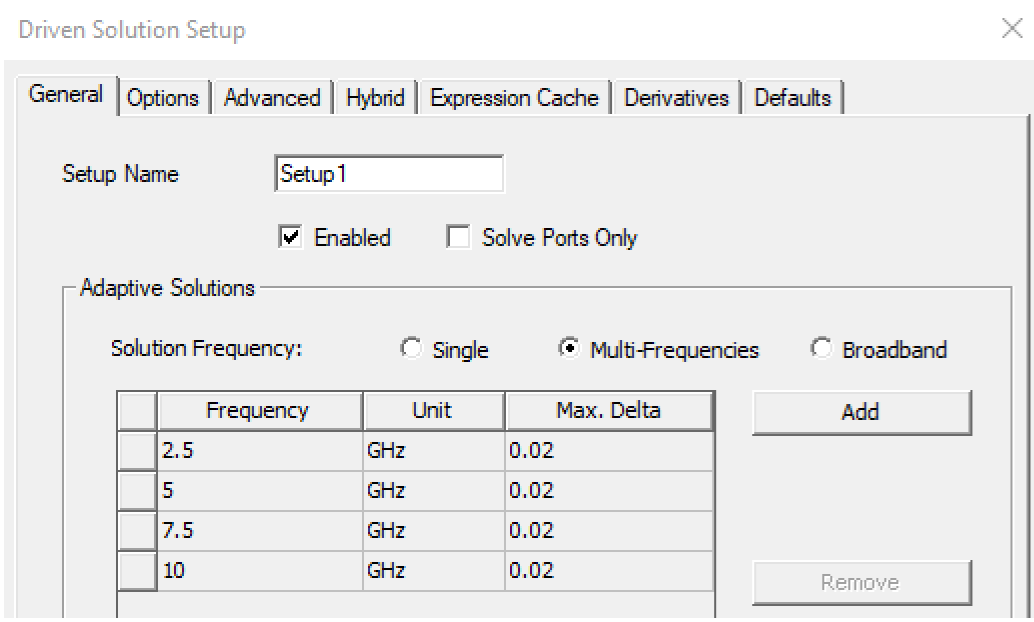


3. Under the **Swept Variable** tab, select **Single Variable** and accept Phase.
4. Specify the phase values you want to include in the animation, as Start, Stop, and Steps to include (e.g., if the **Start** value is **10**, the **Stop** value is **160**, and the number of steps is **10**, the animation 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.



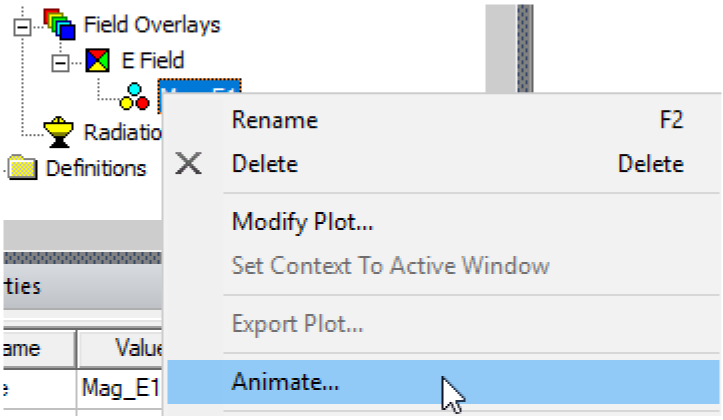
Creating Frequency Animations

1. In order to create Frequency animations, you must define the frequencies of interest in the Solution setup, using the Multi-Frequencies option.



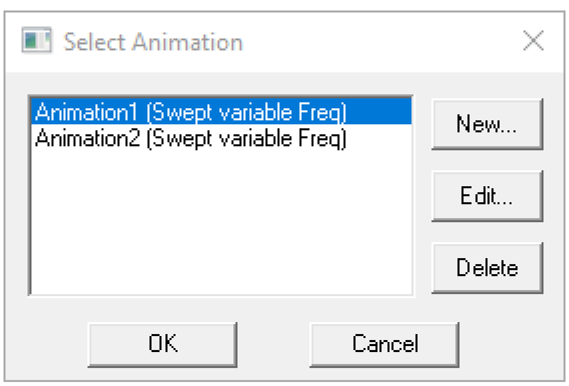
2. After running the analysis, Create a field overlay plot to animate.
3. Do one of the following:

- Click **HFSS**>**Fields**>**Animate**  .
- Right-click the field overlay plot of interest and click **Animate...** from the shortcut menu.

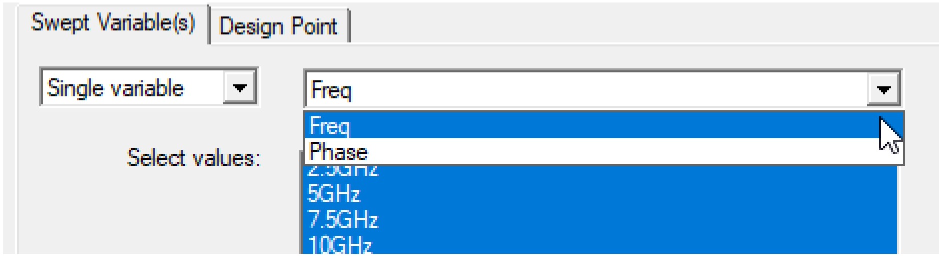


- Select the field overlay plot of interest, select the **View** tab of the ribbon, and click the Animate icon.

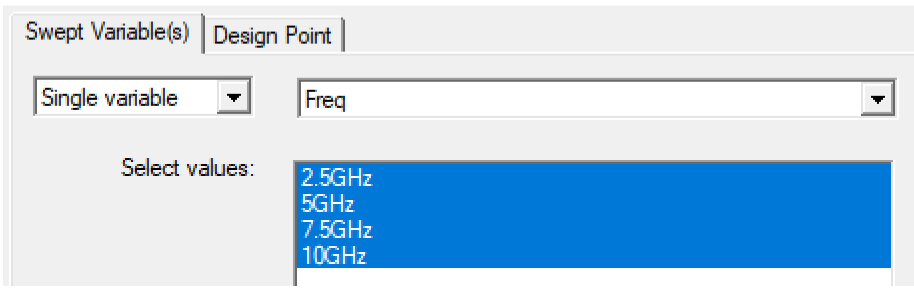
4. If you already created an animation, the **Select Animation** dialog box appears



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.



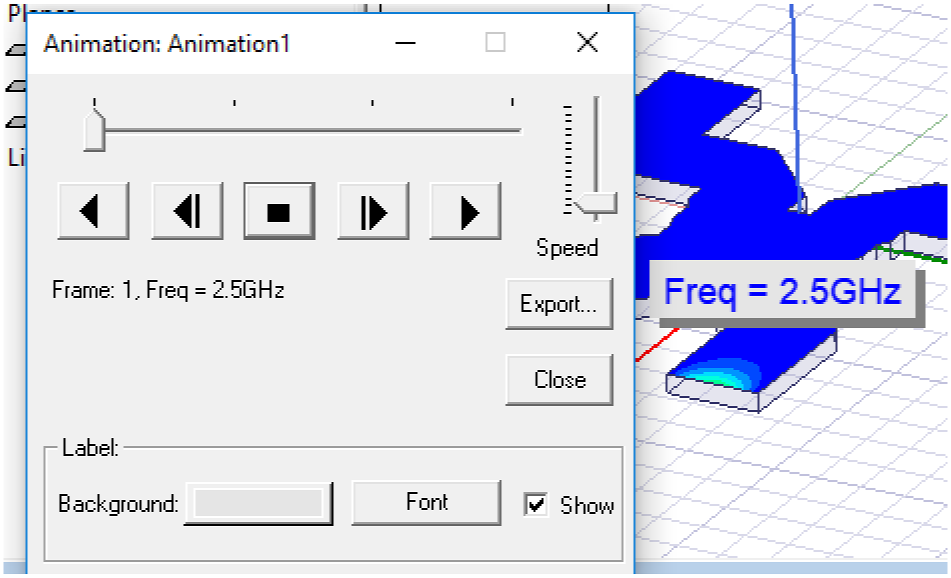
5. Under the **Swept Variable** tab, select **Frequency** from the **Swept Variable** list.
6. Select the frequency values you want to include in the animation from the **Select values** list.



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:

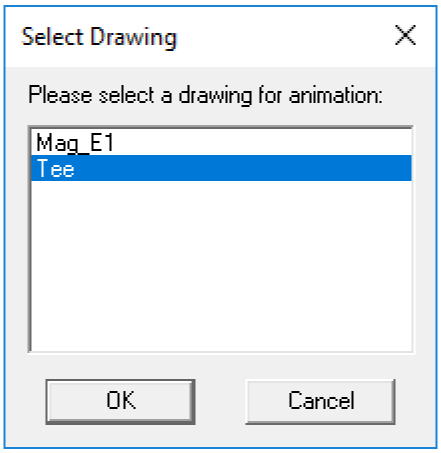
Following is the general procedure for creating an animation that varies a part of the model geometry.

1. Right-click in the *Modeler* window, and click to **View**> **Animate...** or

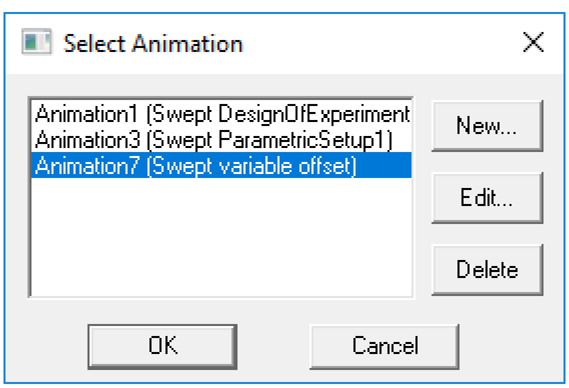
   Select the **View** tab of the ribbon, and click the **Animate** icon.

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.

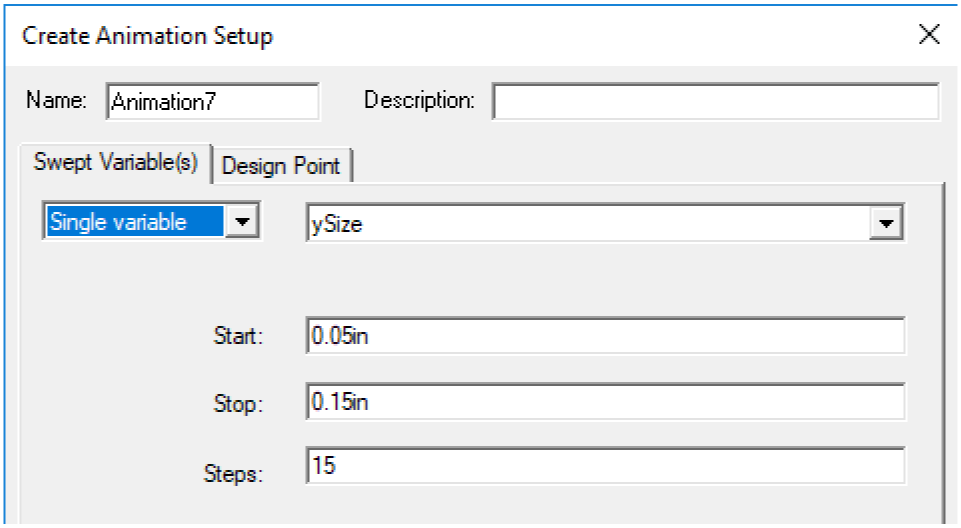
2. In the *Select Drawing* dialog box, select the object that you want to animate.



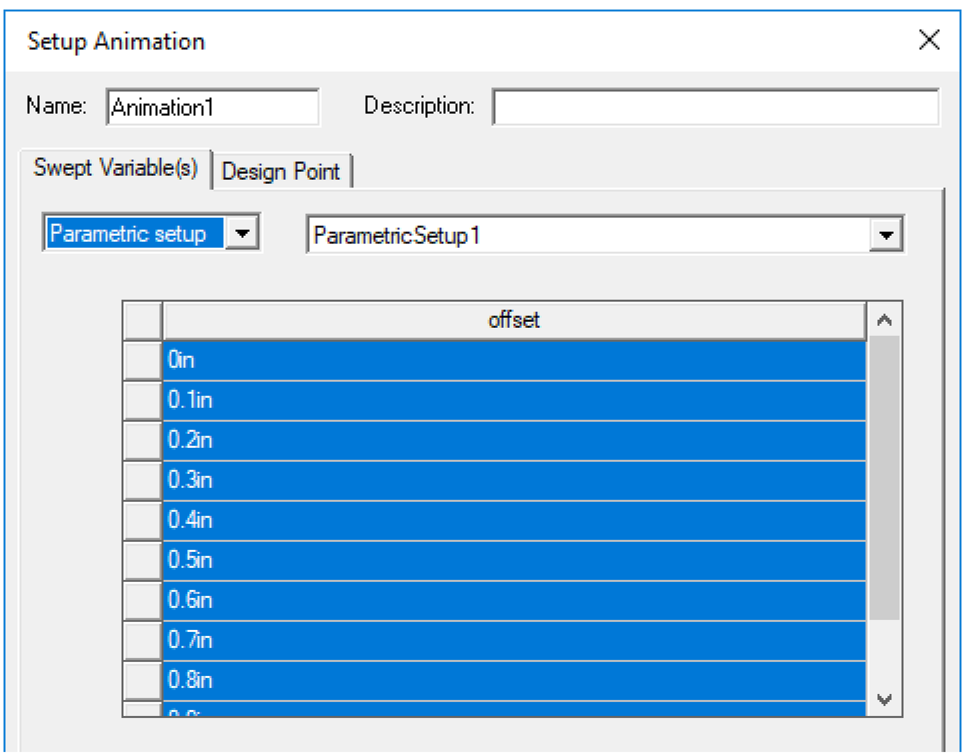
If previous animations have been created for this project, the *Select Animation* dialog box will appear. Selecting an existing animation from that list starts it when you click **OK**. Click **Edit...** to open the *Modify Animation Setup* dialog box. You may choose an animation setup from the list 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.

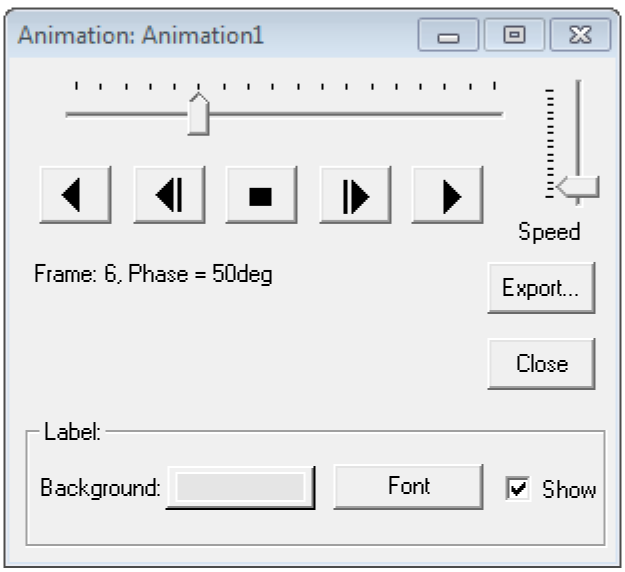


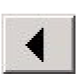
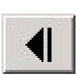
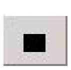
5. If you have created a **Design of Experiments** setup, you can select to define an animation based on that setup and values.
6. If the design has multiple project or intrinsic variables, you use the **Design Point** tab to set the values of the non-animated variables.
7. Click **OK**.

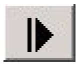
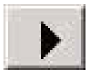
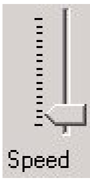
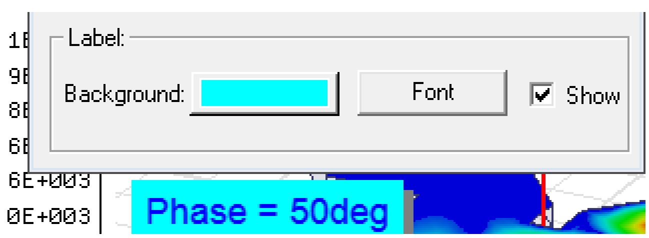
   The animation begins in the view window. It will display one frame for each variable value.

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.

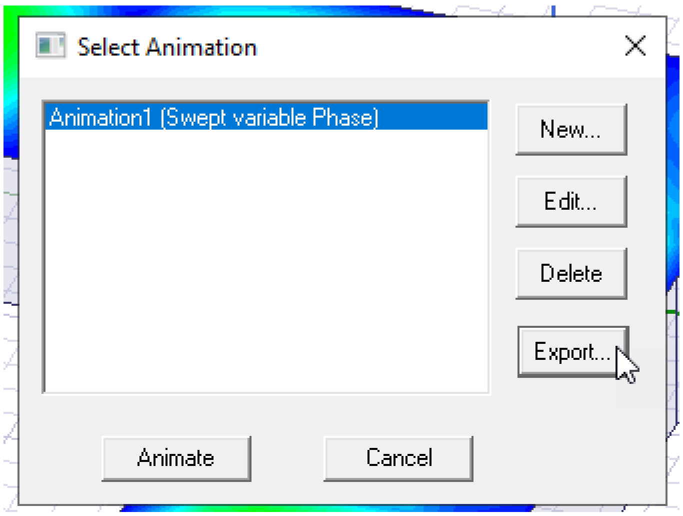
	Steps forward through the animated plot one frame at a time.
	Plays the plot's animation sequence forwards.
	Drag the **Speed** slider to the top to increase the speed of the animation. Drag the **Speed** slider to the bottom to decrease its speed.
Frame information	The current frame and phase at which the plot is being displayed is listed below the control buttons.
Export...	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.
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

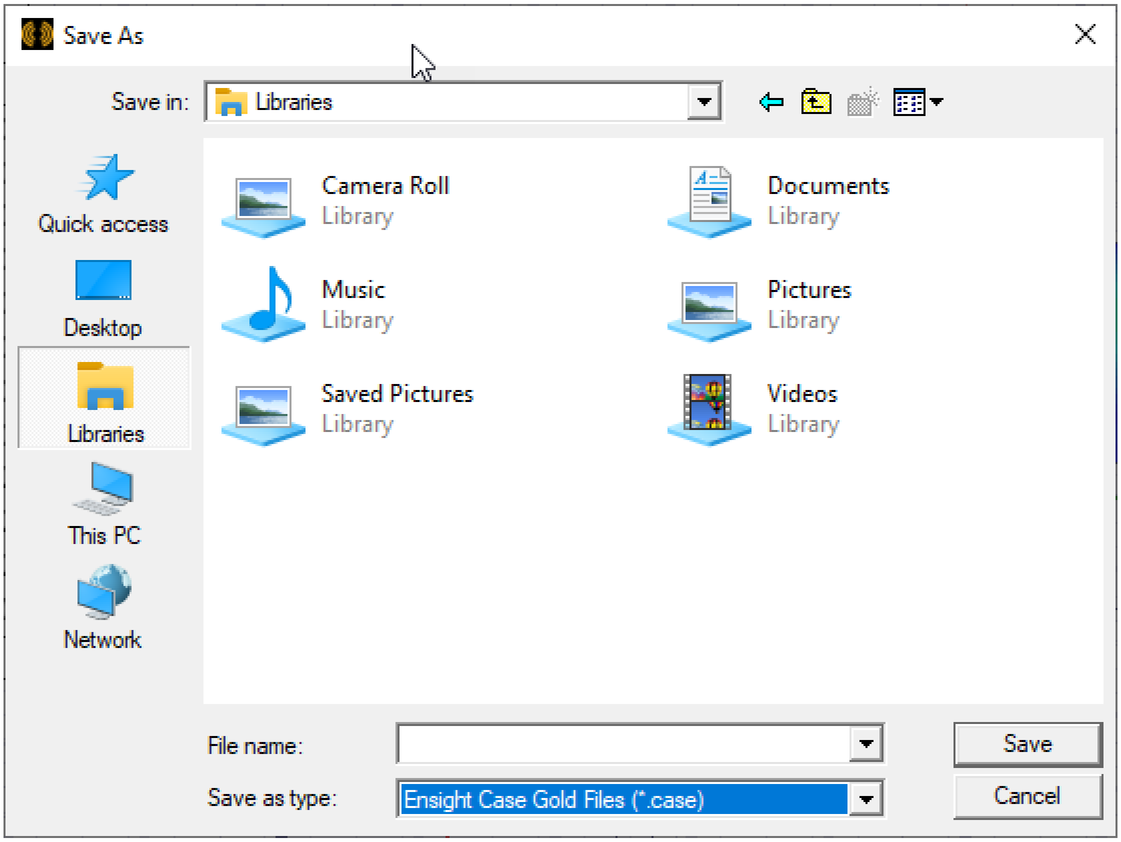
You can export animations from the **Animation** control dialog, or, for Ensight format, from the **Select Animation** dialog.

Exporting Ensight from the Select Animation Dialog

1. Create the animation you want to export.
2. In the **Select Animation** dialog, select **Export...**

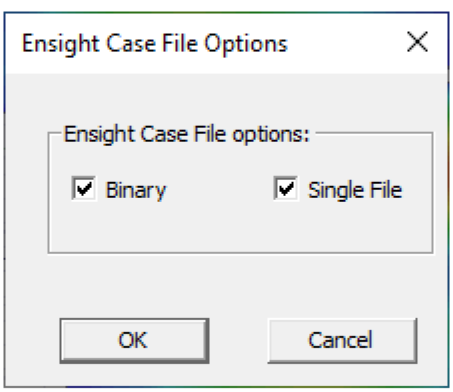


3. This opens the Browser window.



Navigate to the desired location and provide a file name. The format will be Ensight .case files.

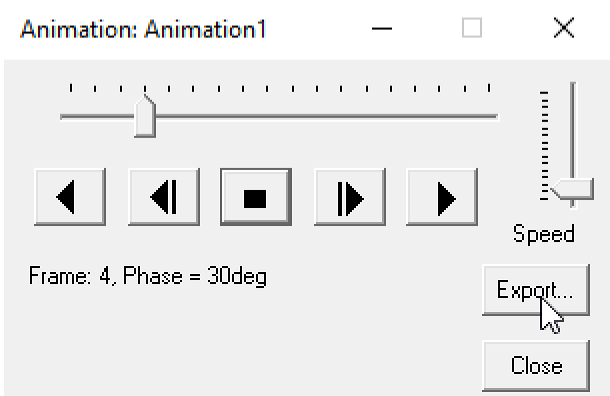
4. When you select **Save**, the **Ensight Case File Options** dialog opens.



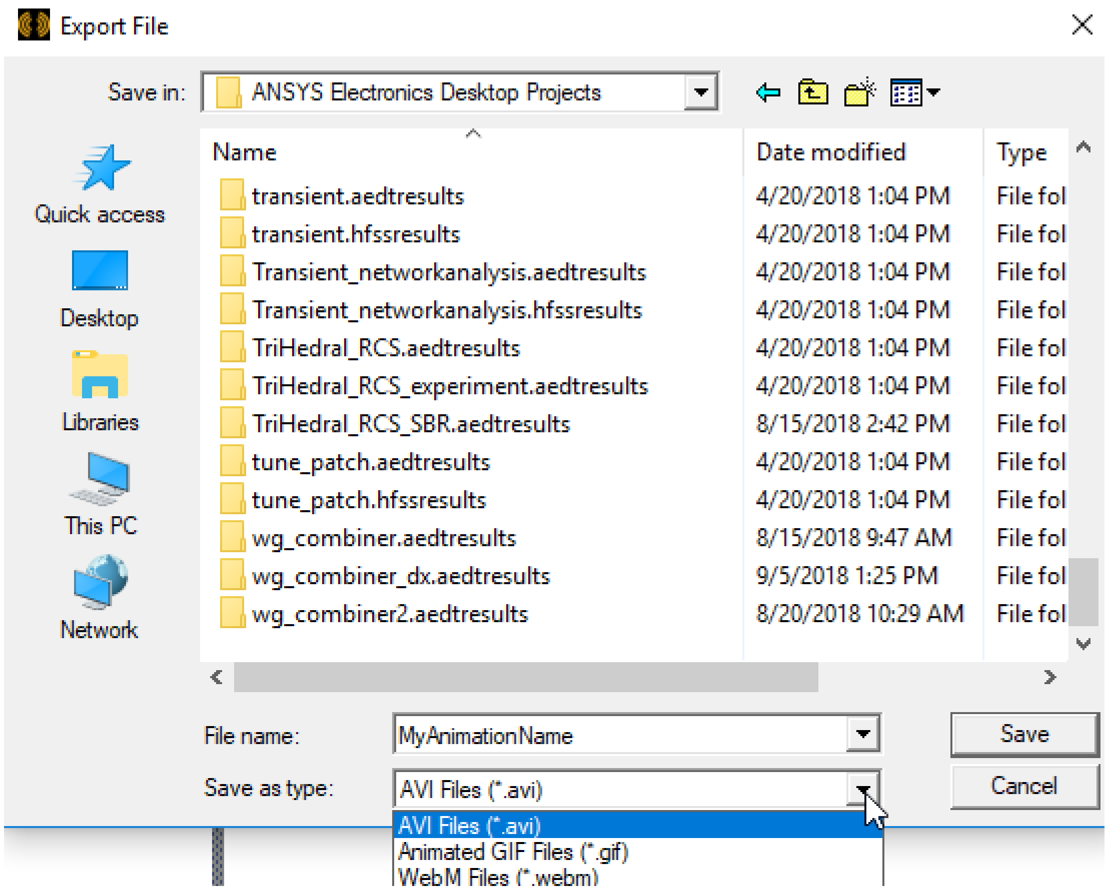
The binary option permits a more compact and efficient animation creation process in Ensight. The Single File options leverages the Ensight transient dataset capability to export all frames of field/geometry data to a single case file. On **OK**, every visible and animatable field plot should generate 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...**.

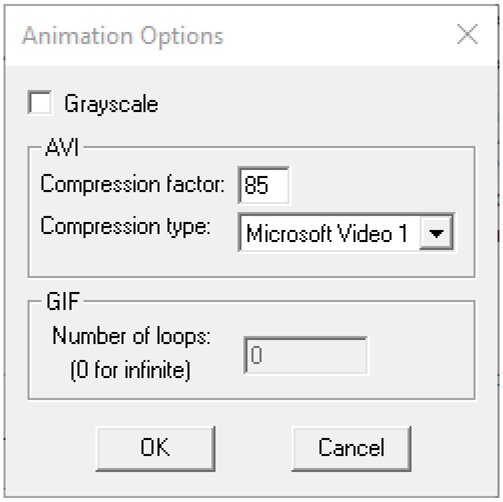


The *Export File* dialog box appears.



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.

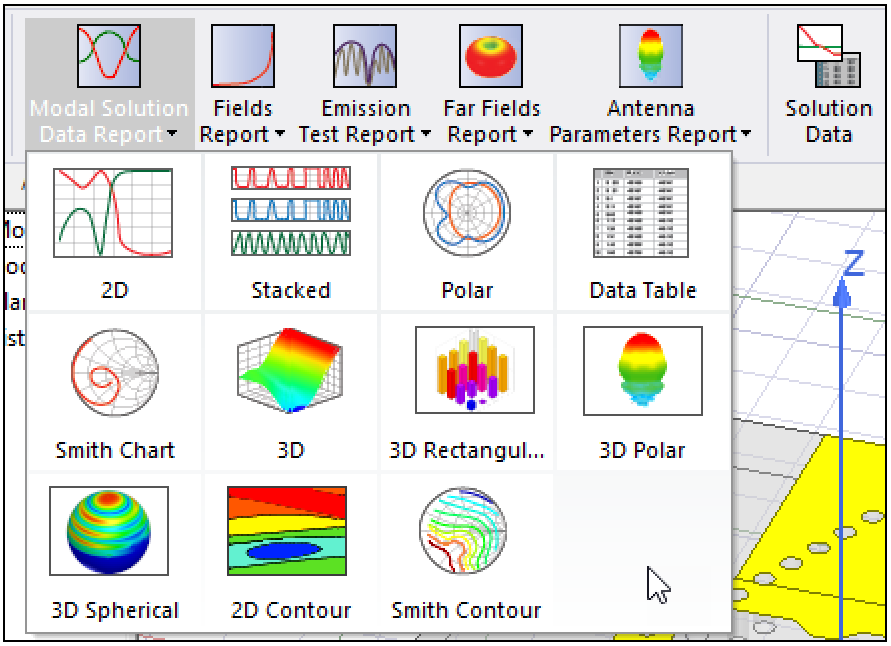
   The animation is exported to the file format you specified.

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>** , 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. For reports for Transient designs, see this discussion.

The **Results** tab for the Ribbon will show icons and drop-down menus for available report types for the active project.



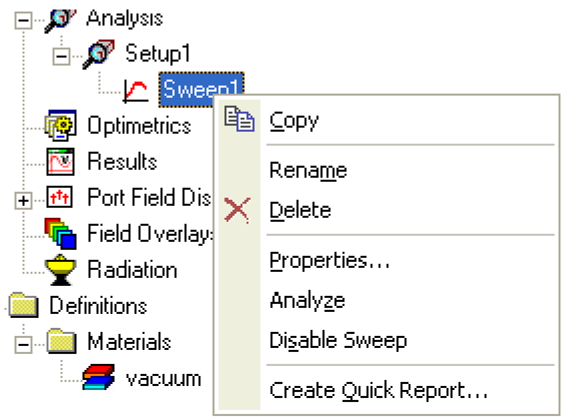
If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **Circuit** > **Results** > **Report Templates** > <***templateName***>. You can also access previously defined 2D templates 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.

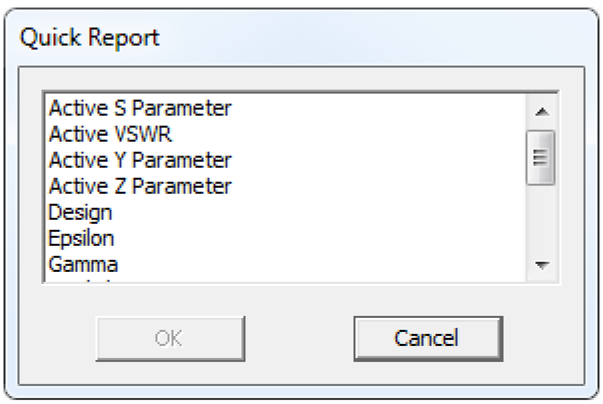
Creating a Quick Report

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.

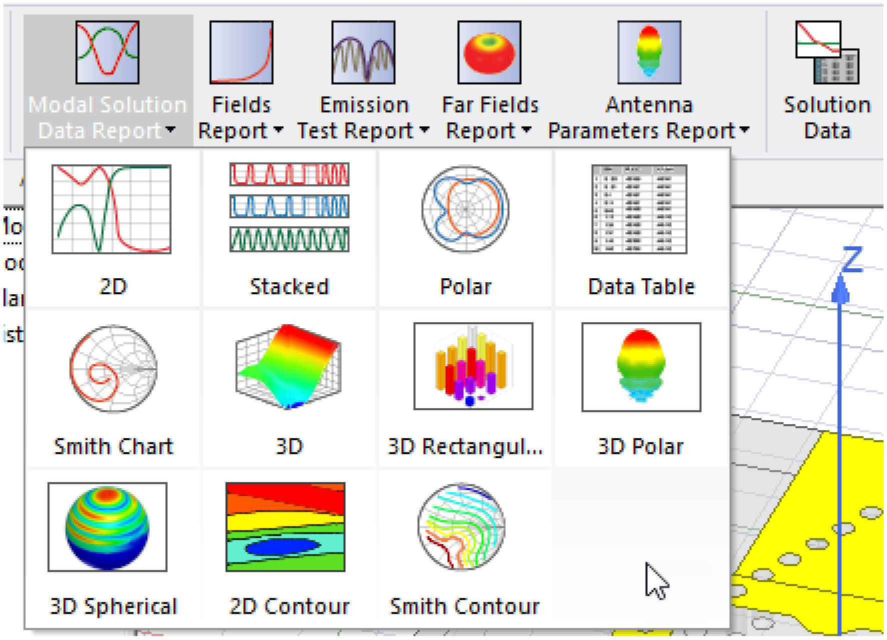
Creating a New Report

Following is the general procedure for creating a new report:

1. On the **Circuit** 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 **Circuit> 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.

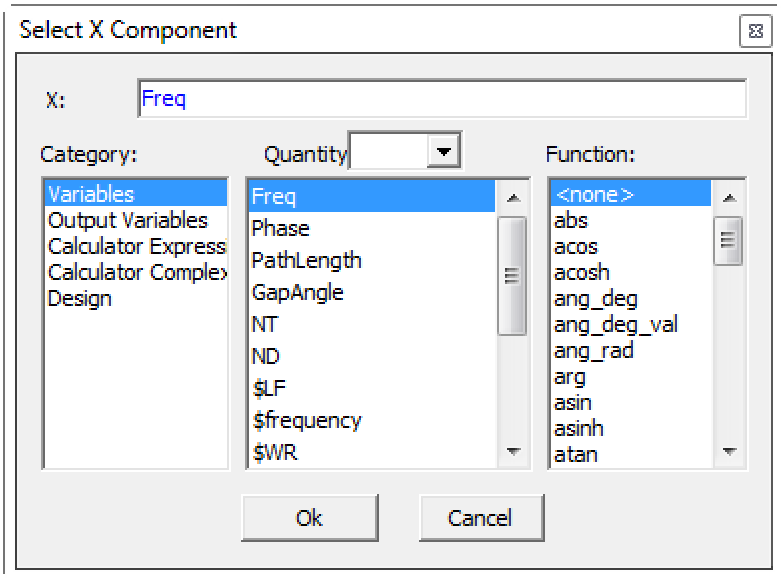
e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies to the currently specified Quantity and Function.

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.

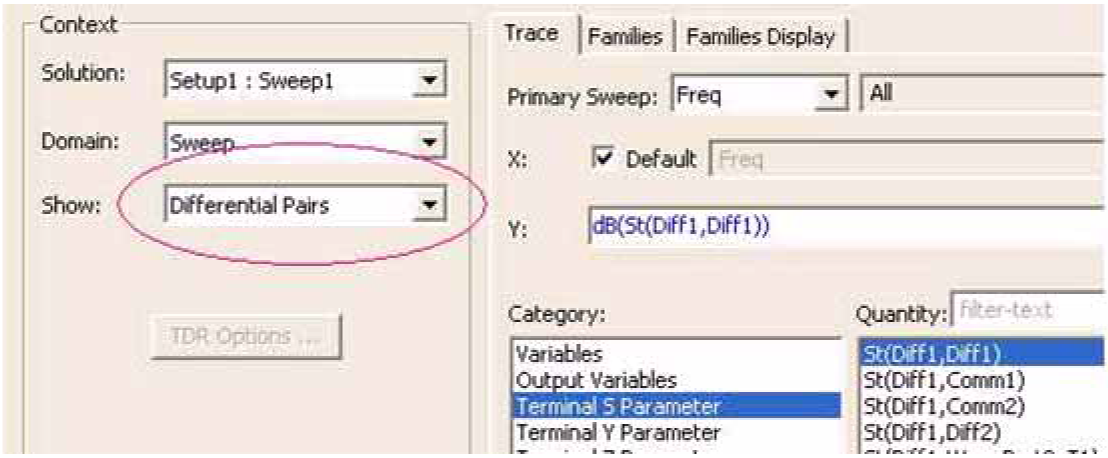
Context Section for Reports

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**.

   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 computs the scattered field in the direction of the plane wave.
4. Show field with a drop-down selection list for Differential pairs or Terminals. This field appears for designs using terminal solutions that have differential pairs defined. It lets you

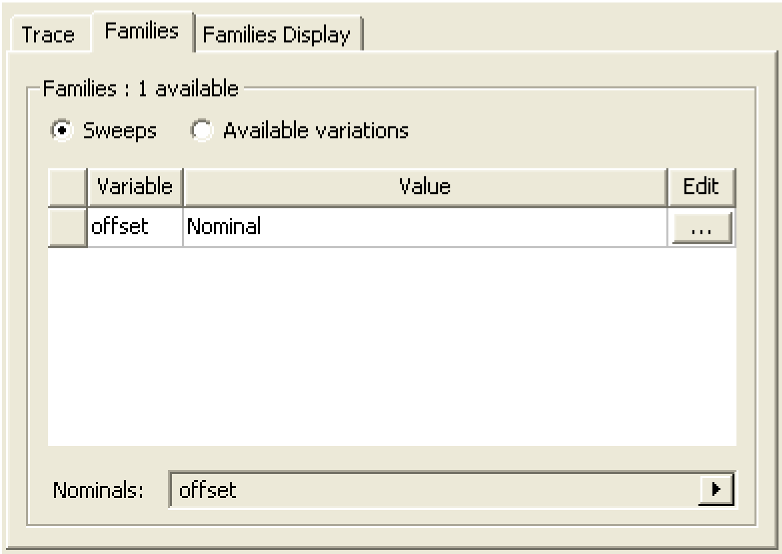
plot either differential pair data, or single-ended terminal data, or both in the same plot without having to disable or enable differential pairs under the Excitations heading in the Project Manager. Note that single-ended quantities are computed as if no differential pairs existed. So in the unlikely case of several terminals where only a subset is combined into pairs, the results may not be as expected.



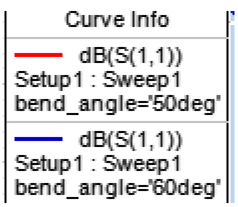
5. Derivative field with a drop-drown selection list of none, all, and specific variables for which you specified **Use** on the Derivatives tab of the solution setup. You can use derivatives in some Optimetrics situations, Far Field reports, non-Port excitations including incident wave, linked field, voltage source, current source, and magnetic bias with the Derivative Tuning feature in the Reporter.
6. A Sources combo box appears in the Reporter when you have specified at least one source in the **Edit Sources** dialog box. 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.

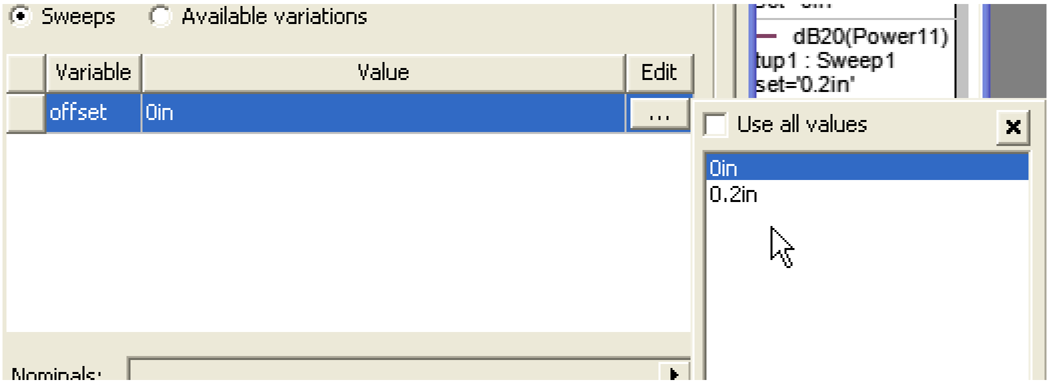


When you select a variable with multiple solved values, a trace for each solved value appears in the Report, with the variable value appended to the trace name in the Report legend.

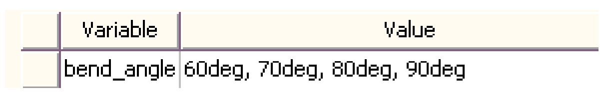


When families are available, you can make selections for the following:

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.

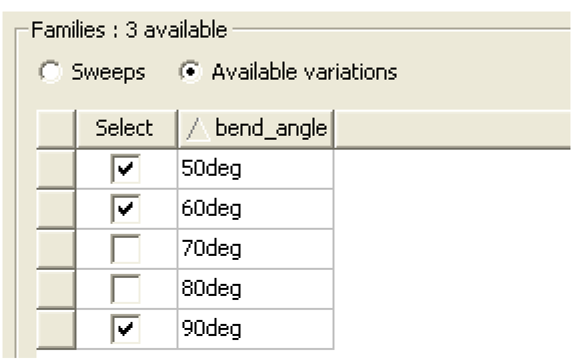


- To select all values, click the check box for **Use all** values. This writes "All" in the value field for that variable. You can also select individual values by clicking on them.
- To select a range of values, hold down the shift key, and click again.
- To select intermittent additional values, hold 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.

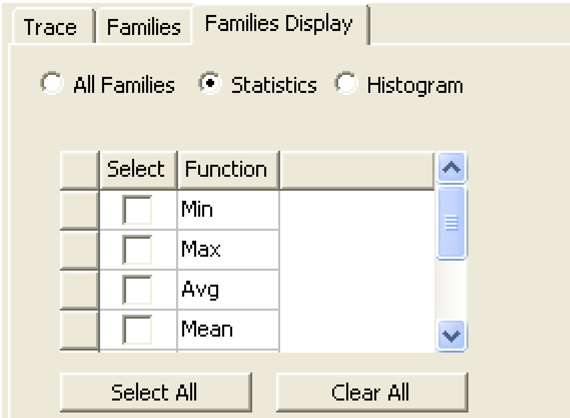
b. To check or clear all variations at once, click the **Select** button at the top of the column.

c. To invert the list order, click the triangle beside the variable name.

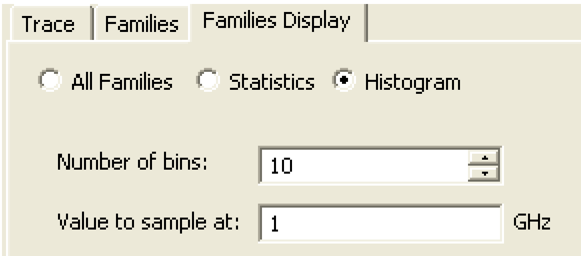
The **Families Display** tab has three radio button selections.

a. **All Families**

b. **Statistics** which lists a table statistical functions that you can select to apply to the plot. The functions include Min, Max, Avg, Mean, Variance, Std Dev, and Sum. You can use the Select check boxes or the Select All and Clear All button.

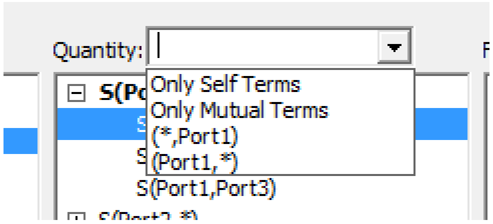


c. **Histogram** which lets you select the number of bins to use for a histogram plot, and the sampling frequency to use.



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.

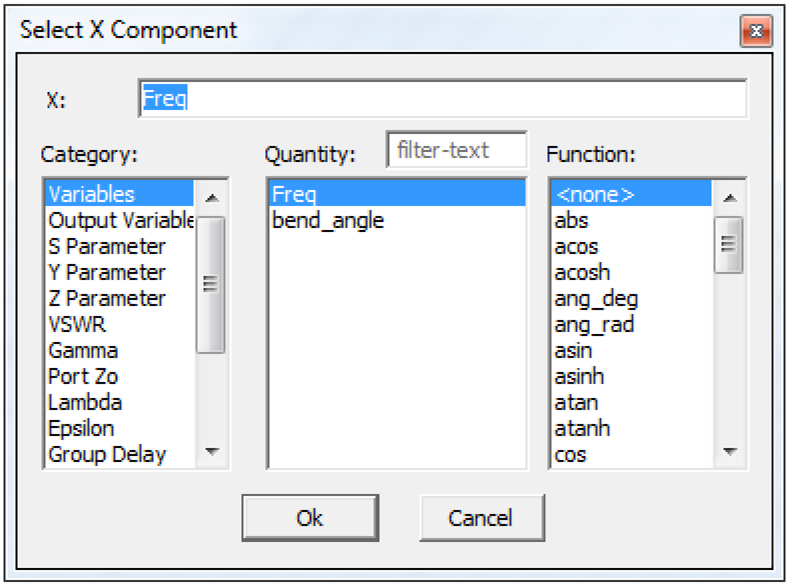
> **Note:**
>
> Color shows valid expression.

   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.



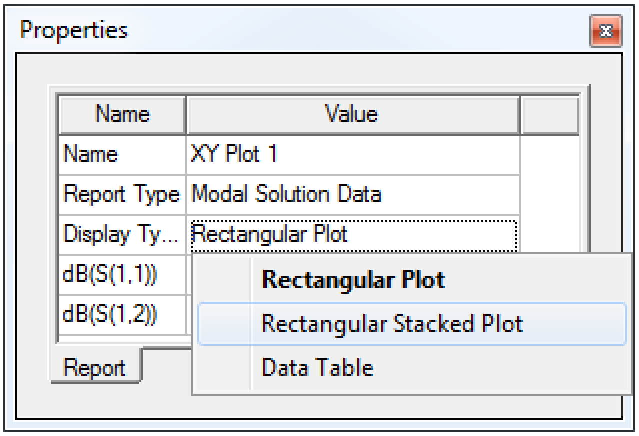
      This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

   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 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.

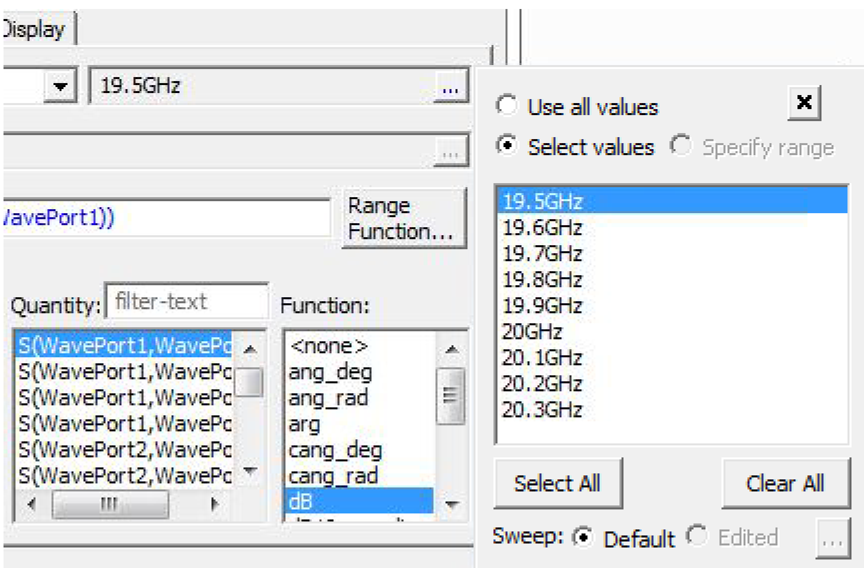


Once you make a selection, the plot display updates for the current selection.

> **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.



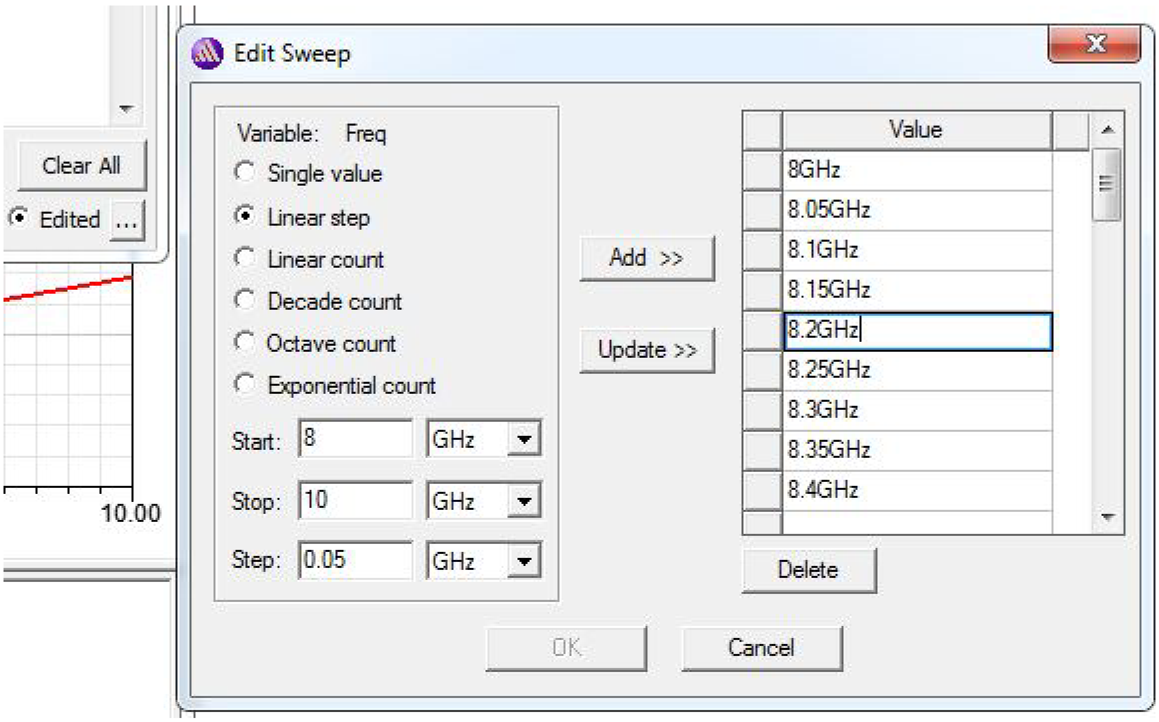
With **Select values** selected, you can select one or more by clicking an individual value, dragging to select multiple values, or using Alt+click to select specific values.

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 **Circuit** > **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.

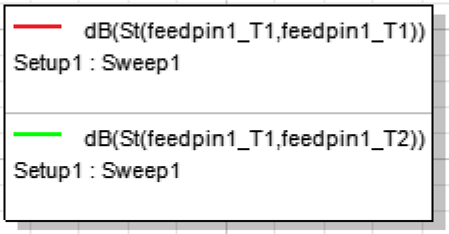
- X or Y or Z **Scaling Tab** – These properties provide control over scaling:
  - Axis Scaling – use the drop-down menu to select scaling as Linear or Log. For the Y axis, all zero or negative values are discarded before log scaling is applied. For 3D plots, scaling is on the Axis tabs
  - Specify Min – 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 Circuittreats 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.

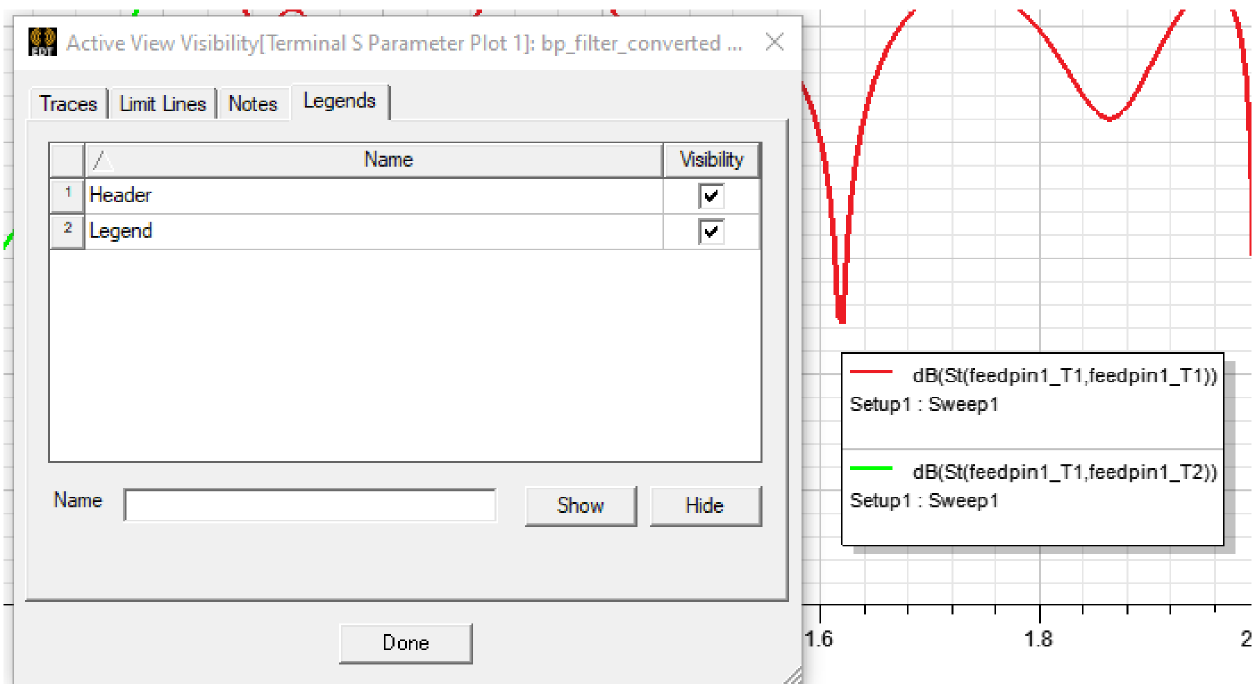


To show or hide a legend in a report:

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.

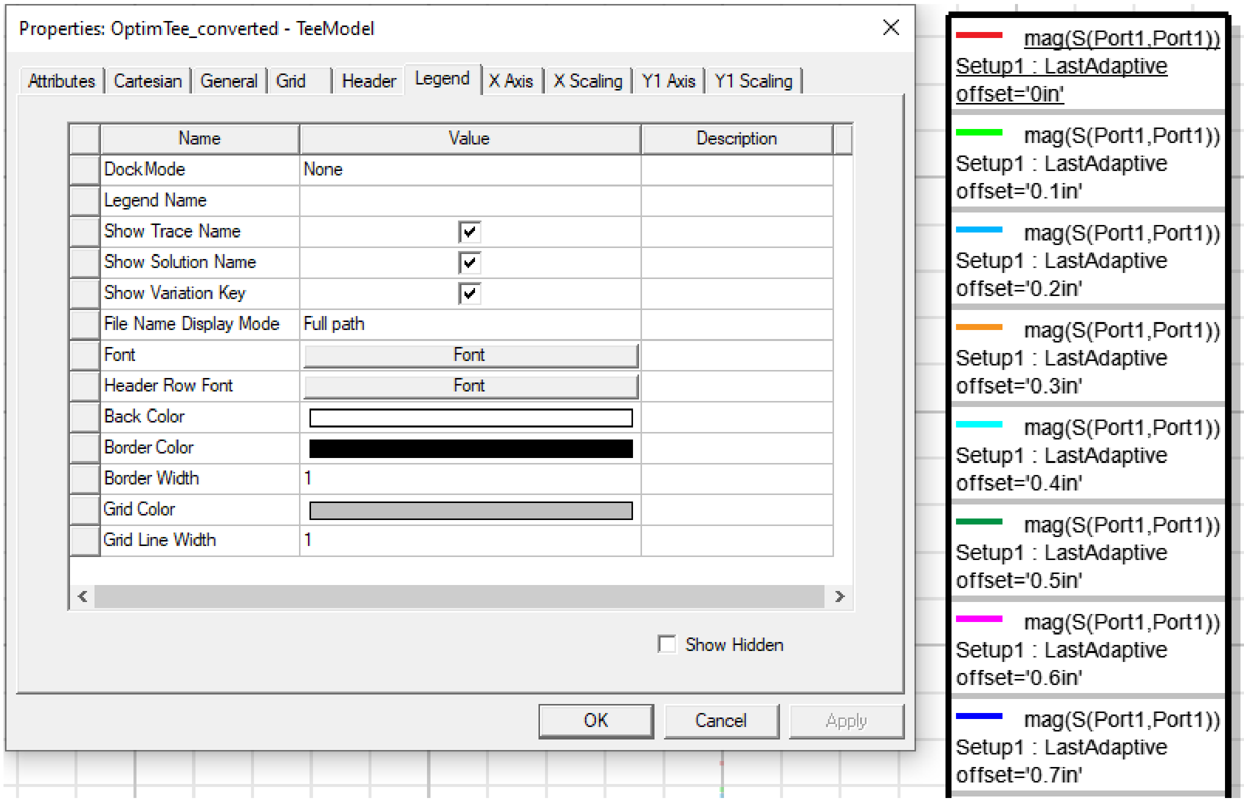


4. Enable the visibility check box, and click **OK** to close the dialog box and apply the change.

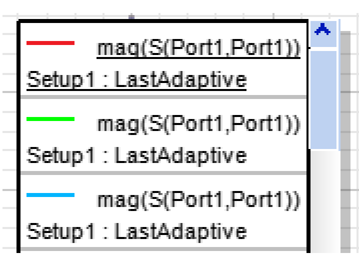
To edit the display properties of a legend:

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.

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:

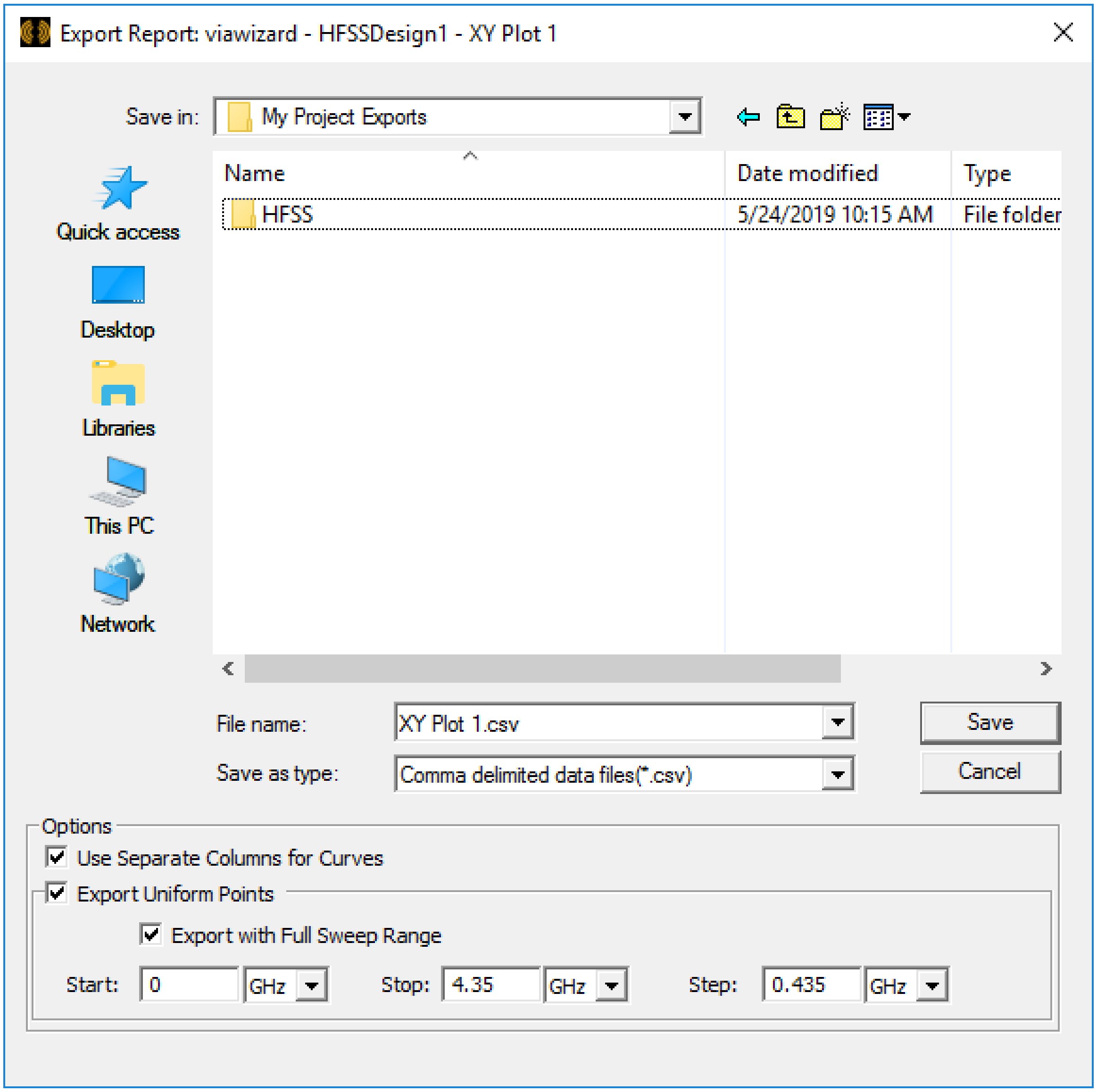
1. In the Project Tree, right-click the report name of interest to display the shortcut menu and click **Report Templates** > **Save Settings as default**. You can also click **Report2D** > **Report templates** > **Save Settings as default** or **Report3D** > **Save Settings as default**.

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).

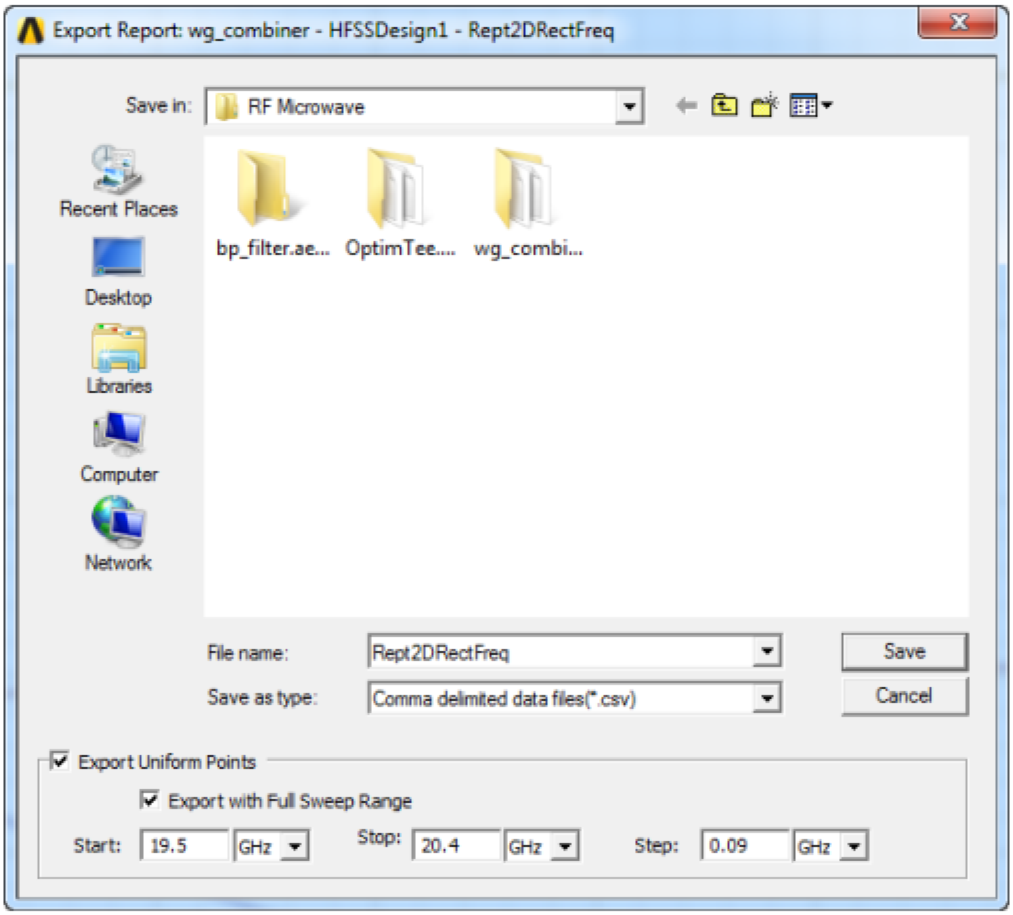
The file will then be available for import using Create Report from file.

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...**

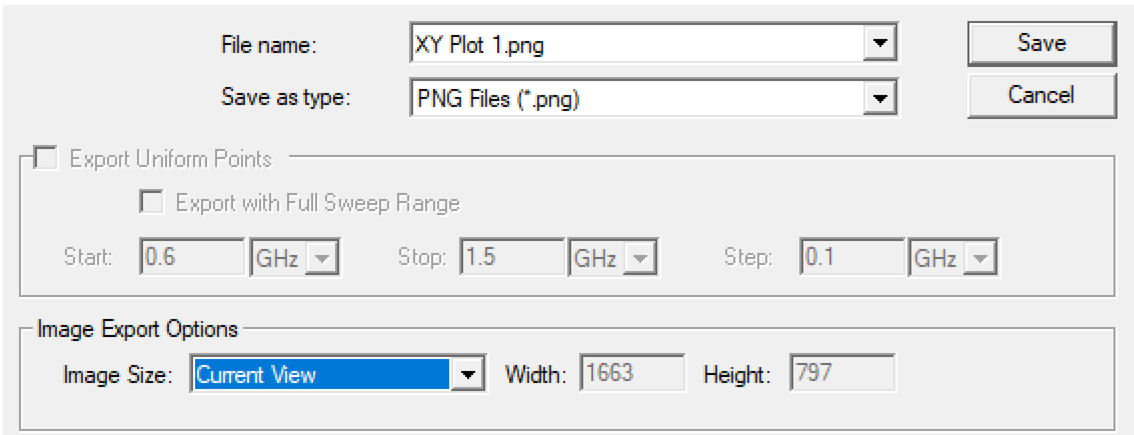
   The **Export Report** dialog box appears:



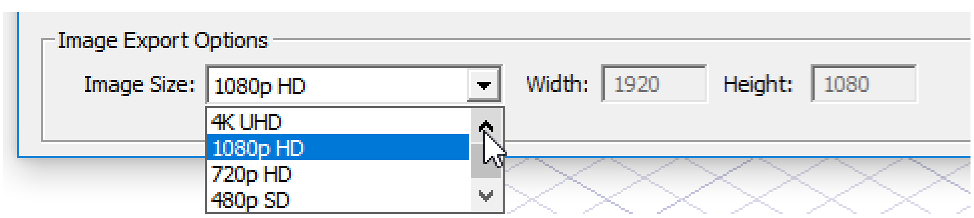
2. Click **Browse...** to open the **Export Report** browser window.
3. Specify the file location and name, and select a graphics format from the drop-down menu:
   - .bmp
   - .gif
   - .jpeg

- .png
- .tiff
- .wrl

4. When you select a graphics 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

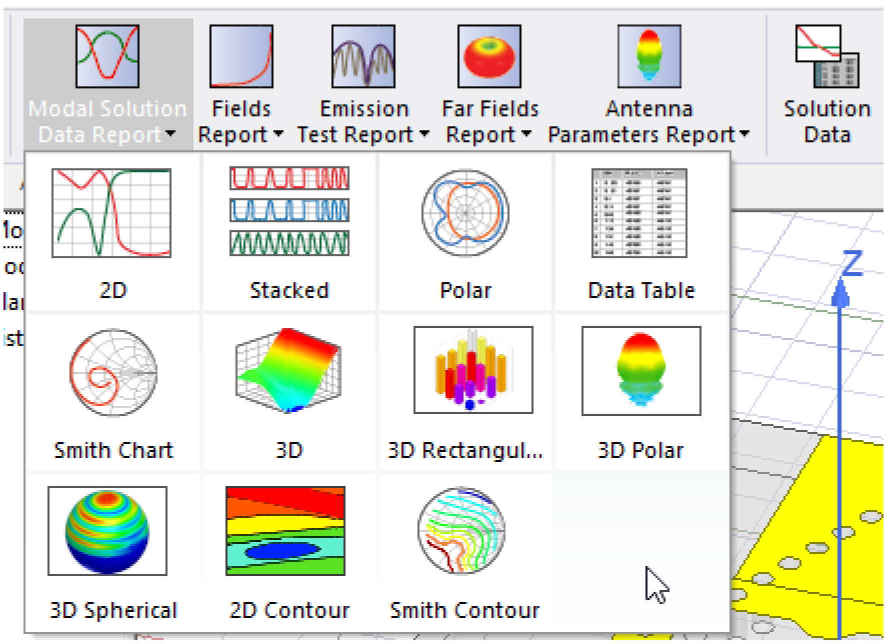
The following table provides information about the file formats that can be imported into plot data reports:

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.

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:

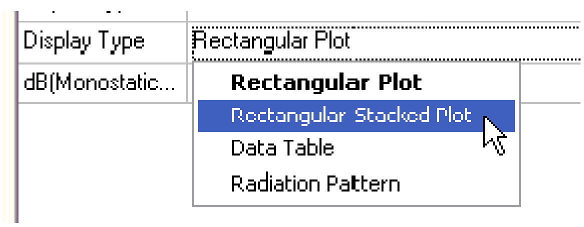


For the initial plot, you can also select from the following **Display Type** formats in the **Create<*type*> Report** submenu:

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	A 3D rectangular (x-y-z) graph.

Rectangular Plot	
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 Contour Plot	A polar chart. Contour plots are useful to visualize surfaces.
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 **Circuit** 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.

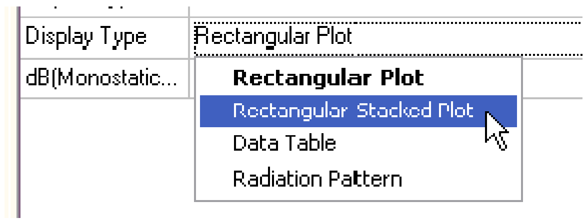
      > **Note:**
      >
      > Color shows valid expression.
   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of 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**.

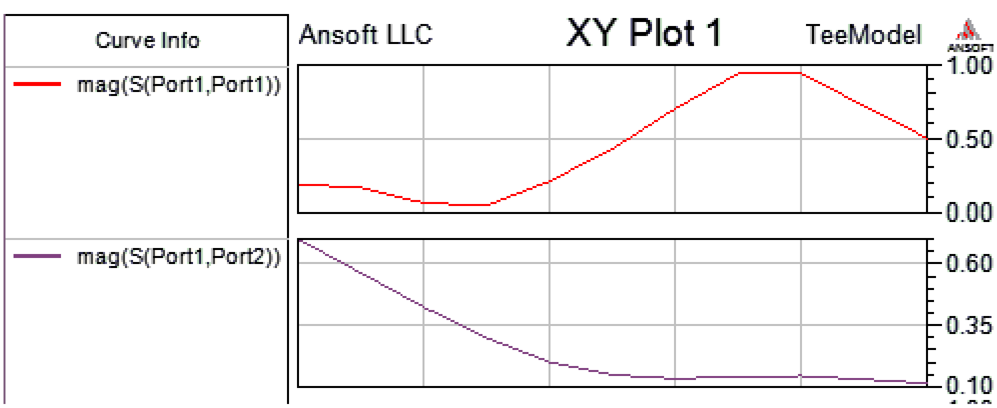
You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Creating 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 **Circuit** 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.

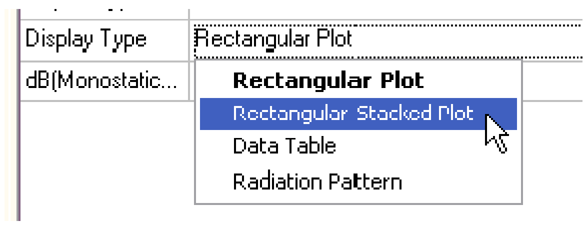
      > **Note:**
      >
      > Color shows valid expression.
   e. **Range Function** button -- opens the *Set Range Function* dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of 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**.

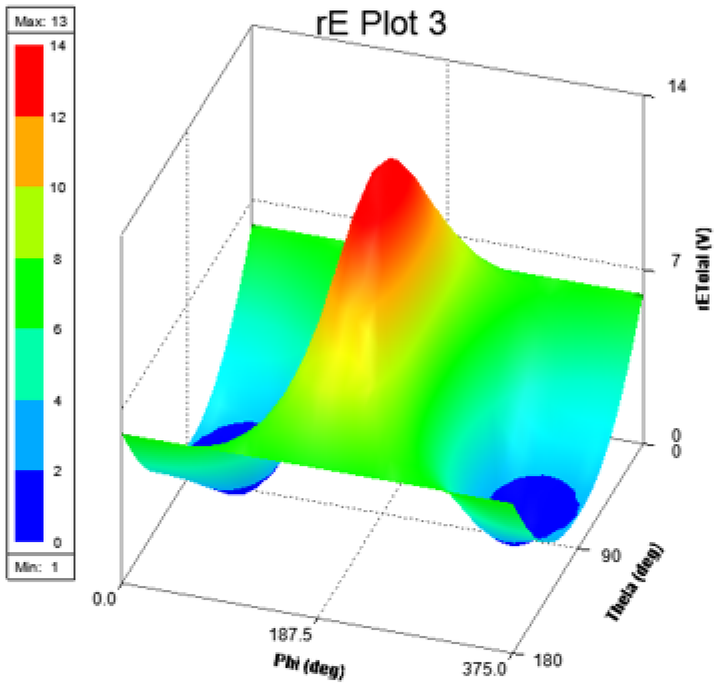
You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



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 **Circuit** 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. Click **New Report.**

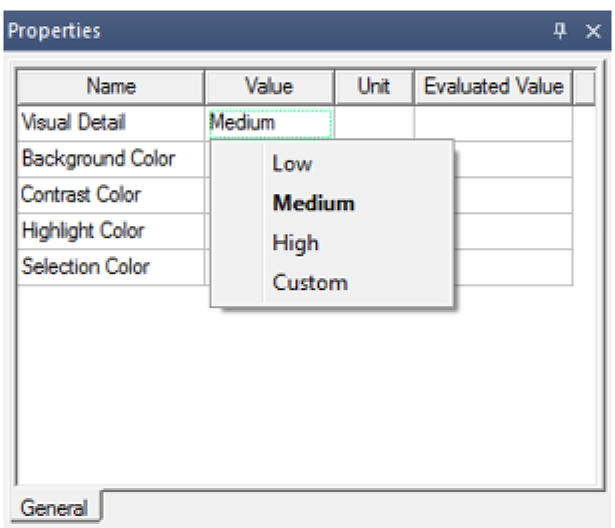
   This creates a new report in Project tree, displays the report with the defined trace, and enables **Add Trace** on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, S Parameter Plot *n* or rE Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the Project Manager. When you select the traces or plots, axis or grid labels, plot header, color key, or variable labels, their properties are displayed in the Properties window. The properties for each plot element can be edited directly to modify the plot content and appearance. See Modifying the Background Properties of a Report.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

Controlling Visual Detail in a 3D Plot

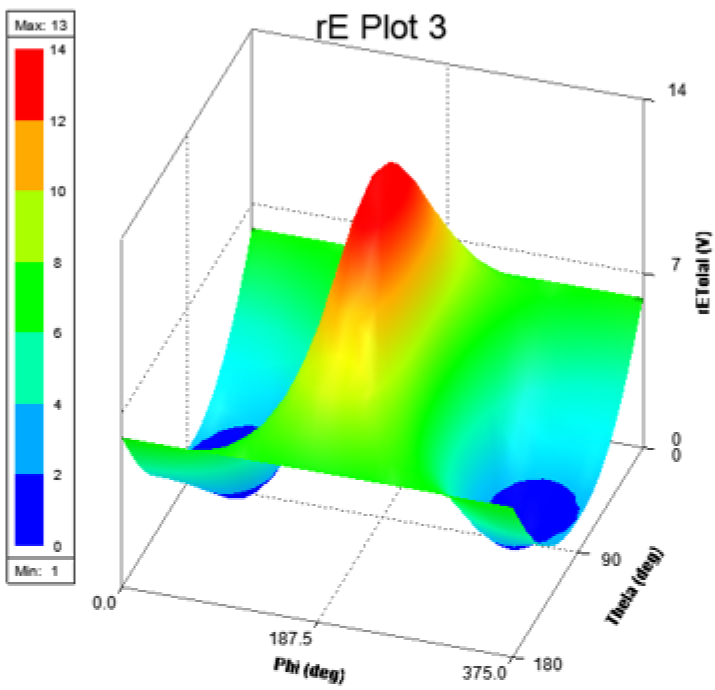
If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides 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).

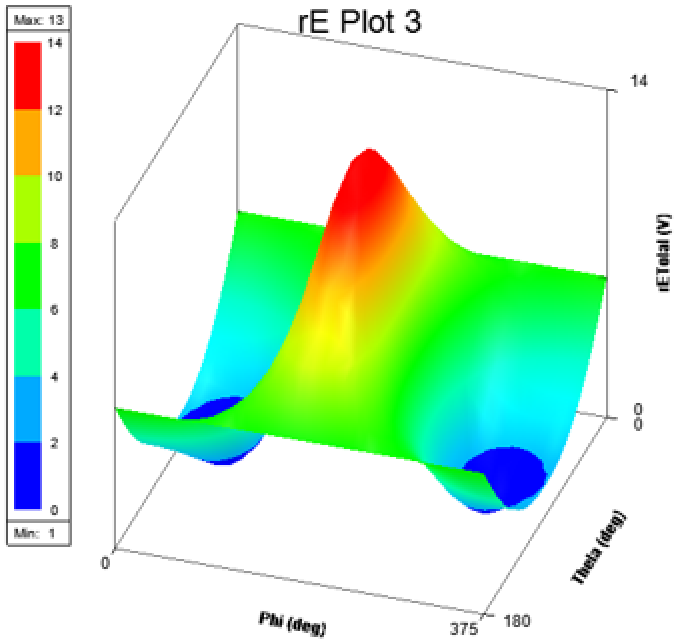
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



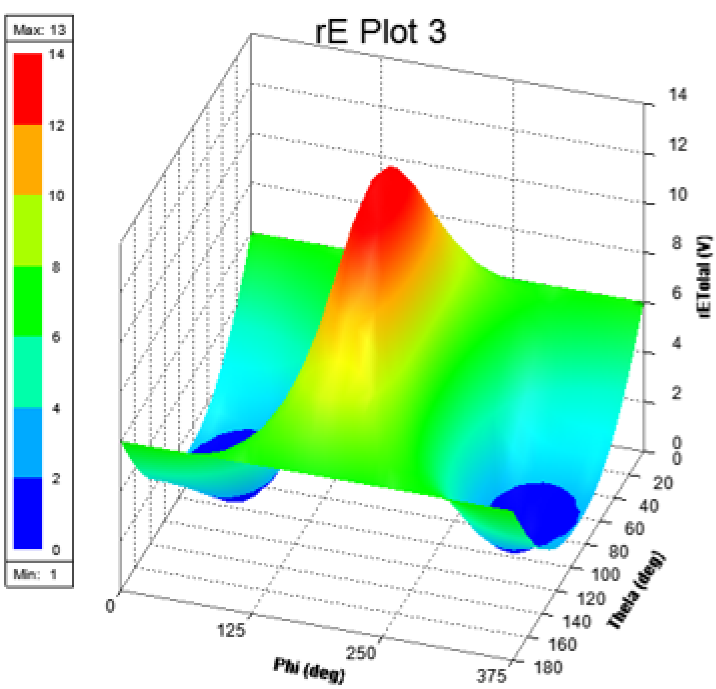
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

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.

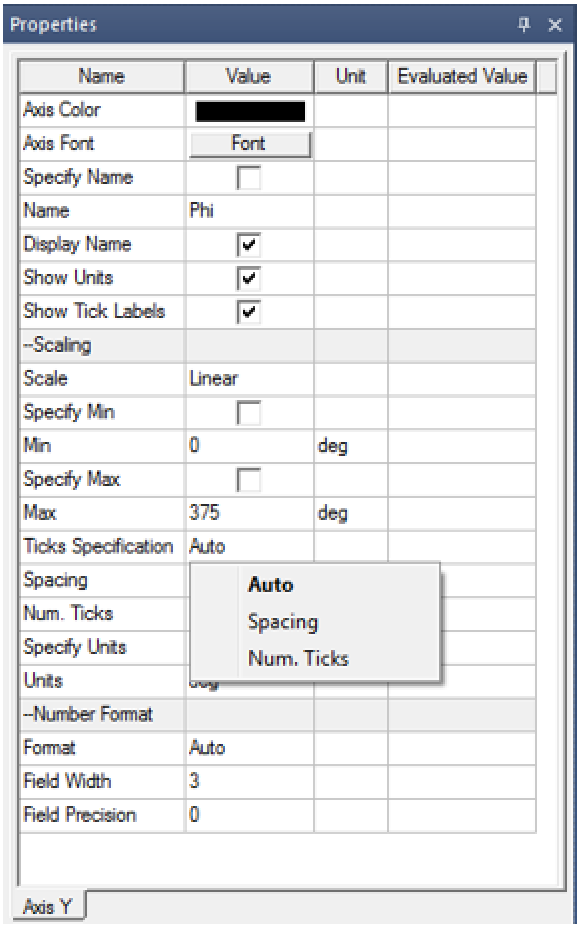
3D Rectangular Plot with High Visual Detail



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 (**Circuit** 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.

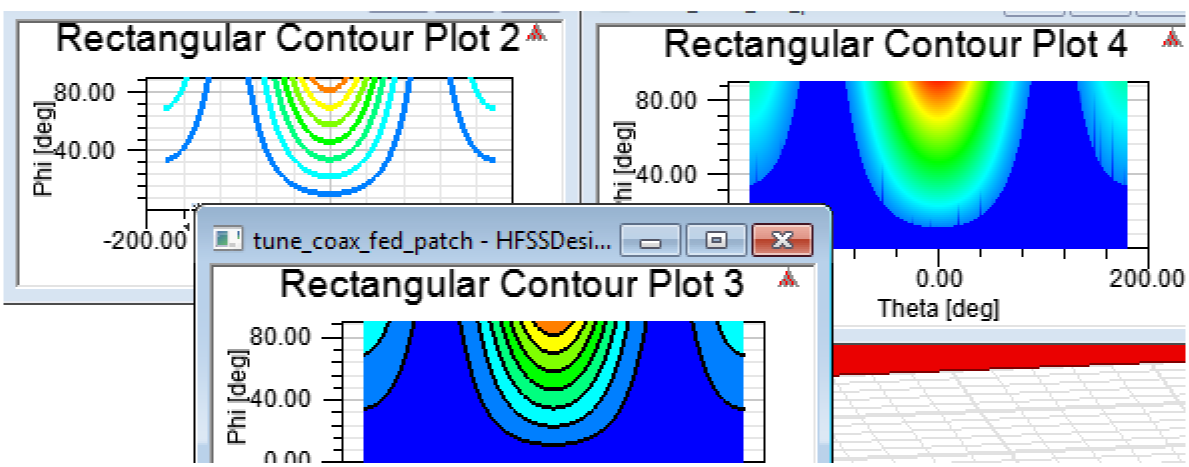
   The Trace Properties for Cartesian plots include tabs for:

   - 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.

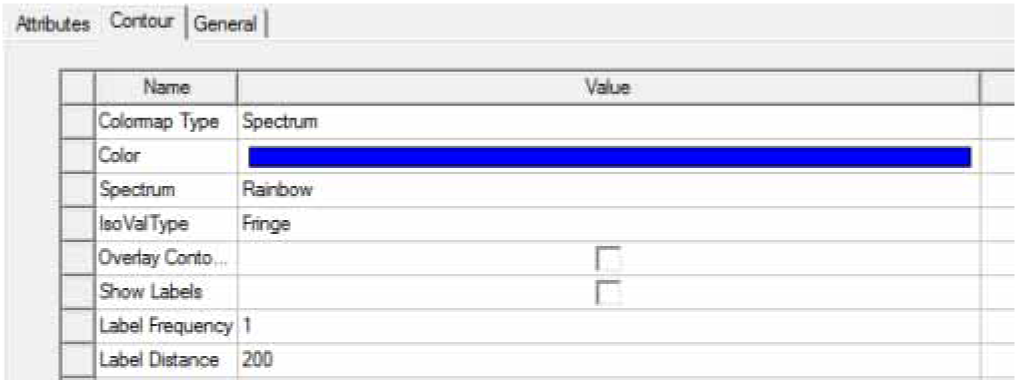
   Spectrum can be Rainbow, Temperature, Magenta, or Gray

   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

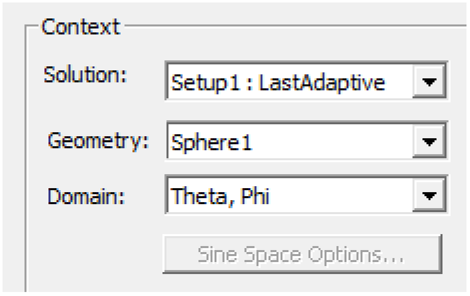
Sine Space plots are implemented as a new Context in Reporter. They will leverage the existing radiated field setups and calculations.

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**

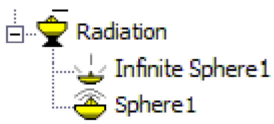
   Results> Create Near Fields Report> Rectangular Contour Plot or **Data Table**.

   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.

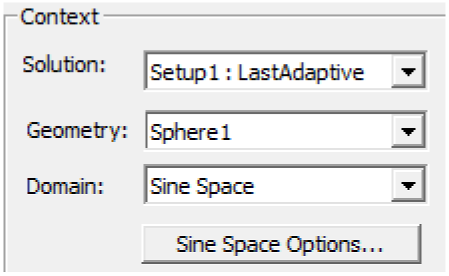
Notice that the Context shown above already includes selection of a Geometry, which corresponds to a Radiation setup.



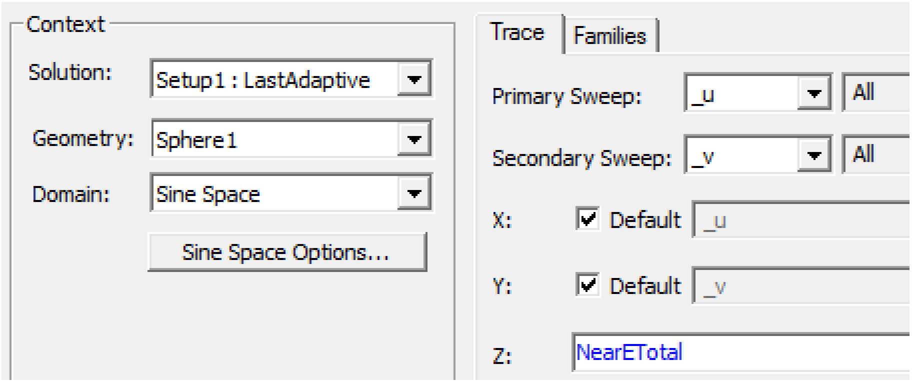
This radiation setup is used to calculate the fields for the Sine Space plot. Internally, this computes the values of the complex E vector based on the theta, phi sampling in the radiation setup. When data is requested for a sine space plot, it will correspond to some arbitrary theta, phi, and the fields 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.

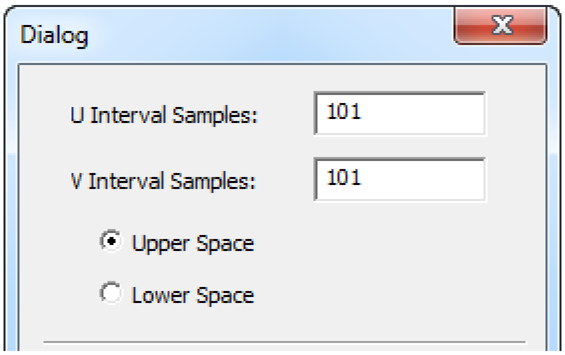
2. Select Sine Space as the Domain. This also enables the **Sine Space** options button.



Selecting "Sine Space" changes the Primary Sweep to U, and the Secondary Sweep to V.

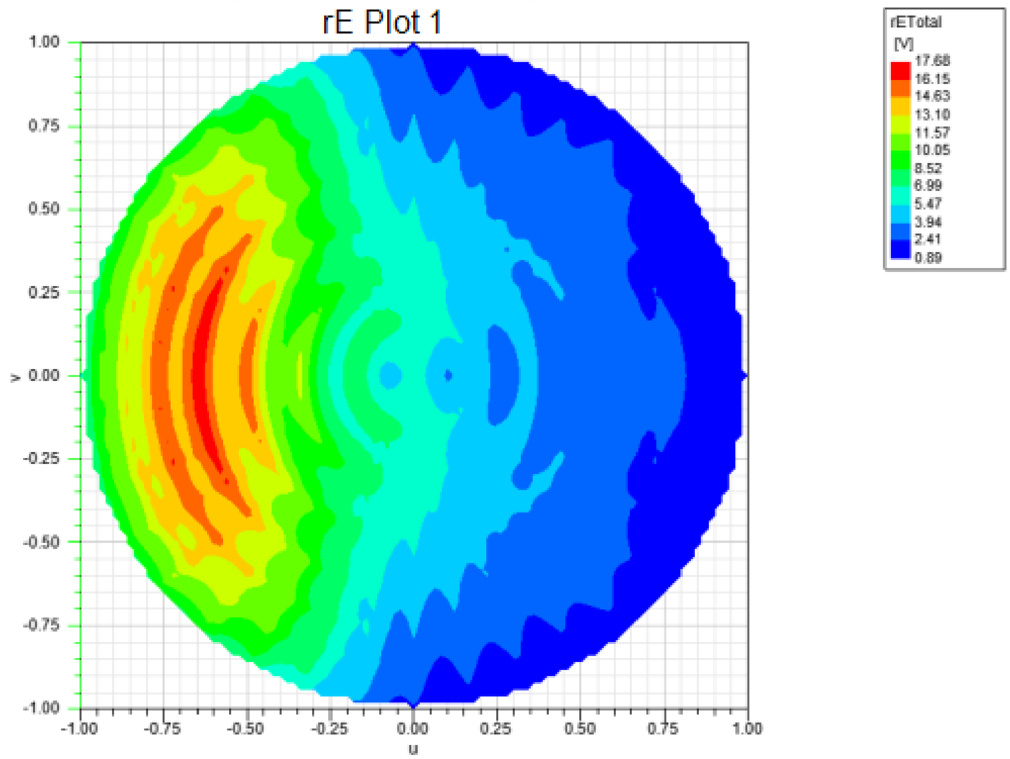


The Sine Space Options provide additional settings, including the sampling interval for the u,v coordinates (default of .05 would provide 40 samples over each axis, -1 to 1) and selection of upper or lower half space (default to upper).

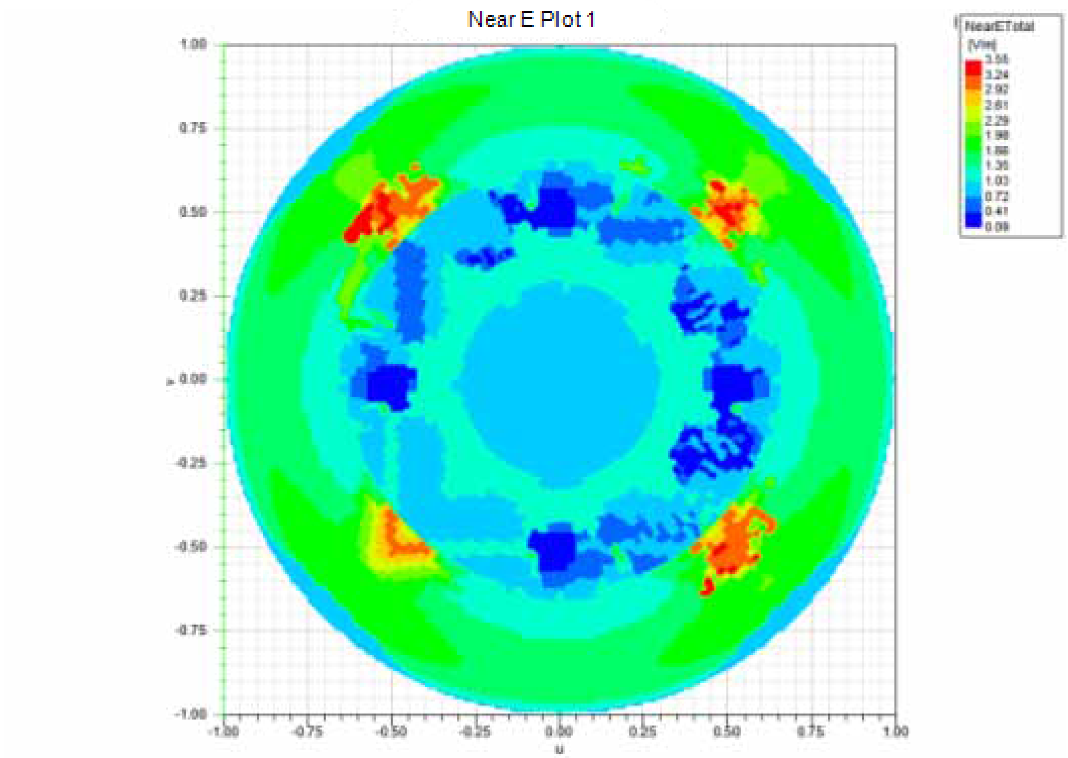


Once you have set up the domain, you define the Sine Space plot, selecting Category, Quantity and Function settings as appropriate. The Report Category you select provides the default report name, for example, S Parameter Plot *n* or Output Variables Plot *n*. Given that the u,v sampling is set in the 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 Circuit, 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 (Circuit 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.

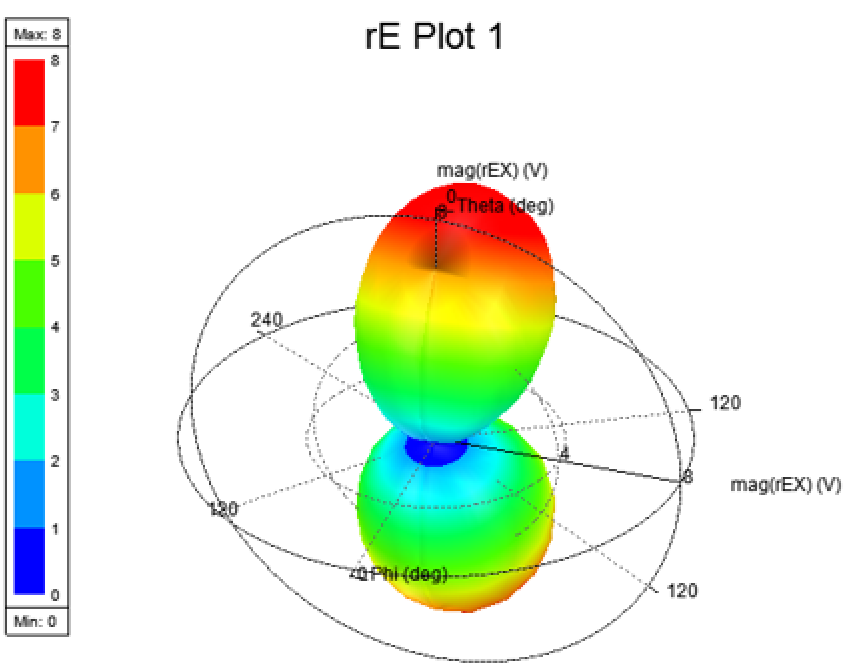
VSWR The voltage standing wave ratio, calculated from the equation

$$\frac{1 + |S_{ij}|}{1 - |S_{ij}|}.$$

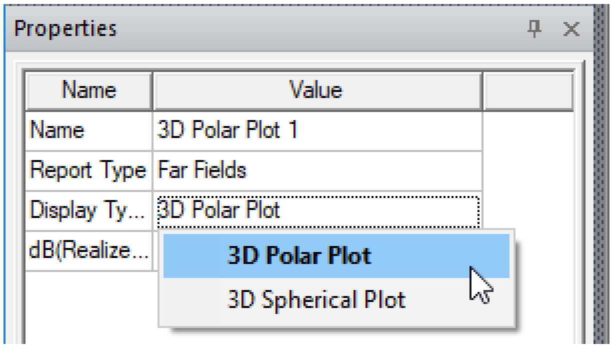
A scale below the plot displays the scale of points along the R-axis.

Creating 3D Polar Plots

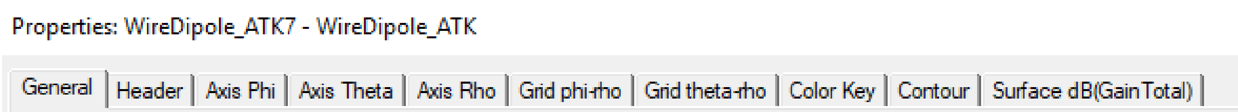
A 3D polar plot is a 3D circular chart divided by the spherical coordinates R, theta, and phi, where R is the radius, or distance from the origin, theta is the angle from the x-axis, and phi is the angle from the origin in the z direction.



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

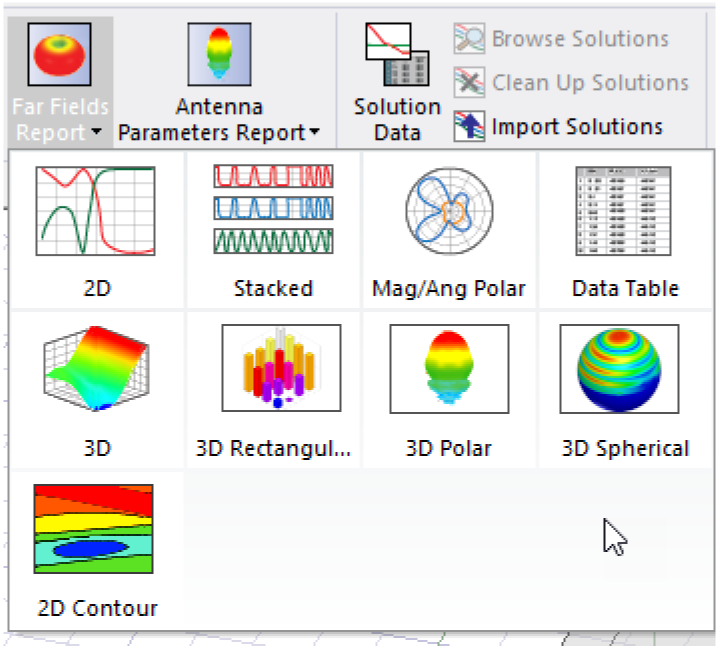
You can edit the properties on these tabs to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Polar Plot" on page 18-97 below.

Creating 3D Polar Plots

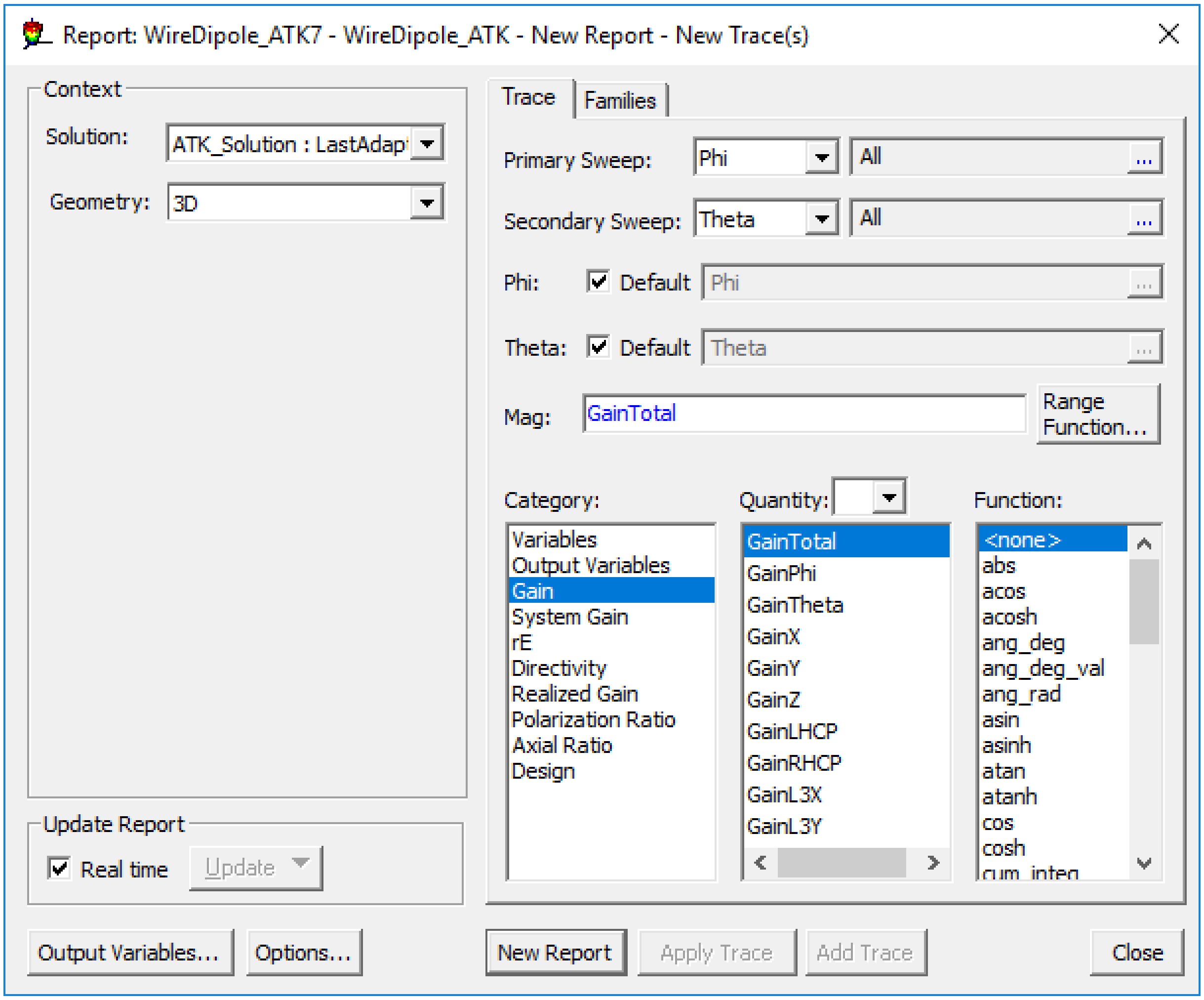
Following is the general procedure for drawing a 3D polar plot of results:

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**.

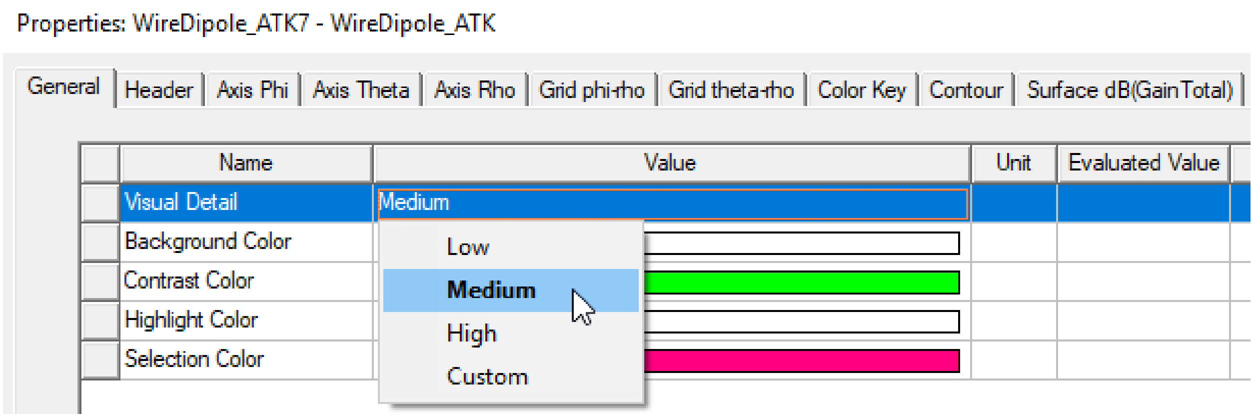
Interacting with 3D Polar Plots

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.

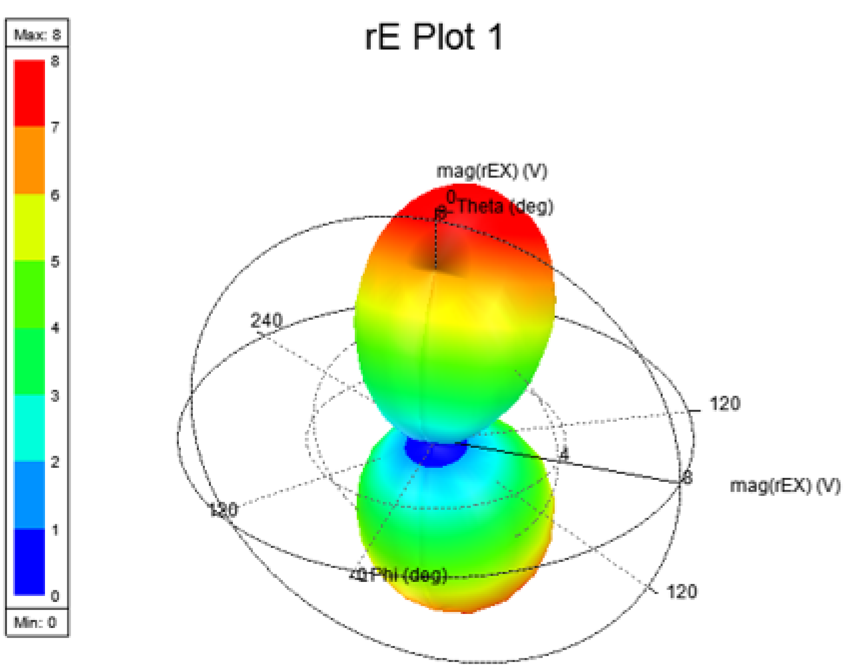
The Visual Detail property on the **General** tab also provides control suited to different screen and plot sizes.



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).

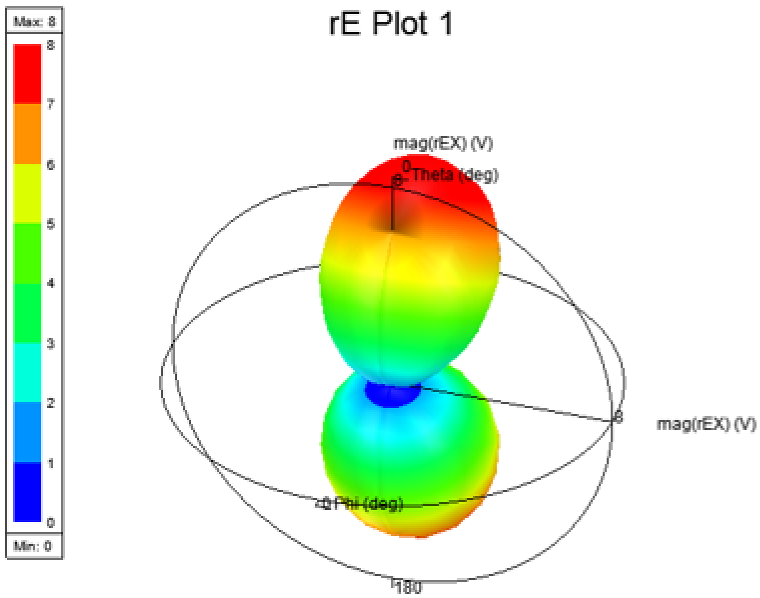
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Polar Plot with Medium Visual Detail



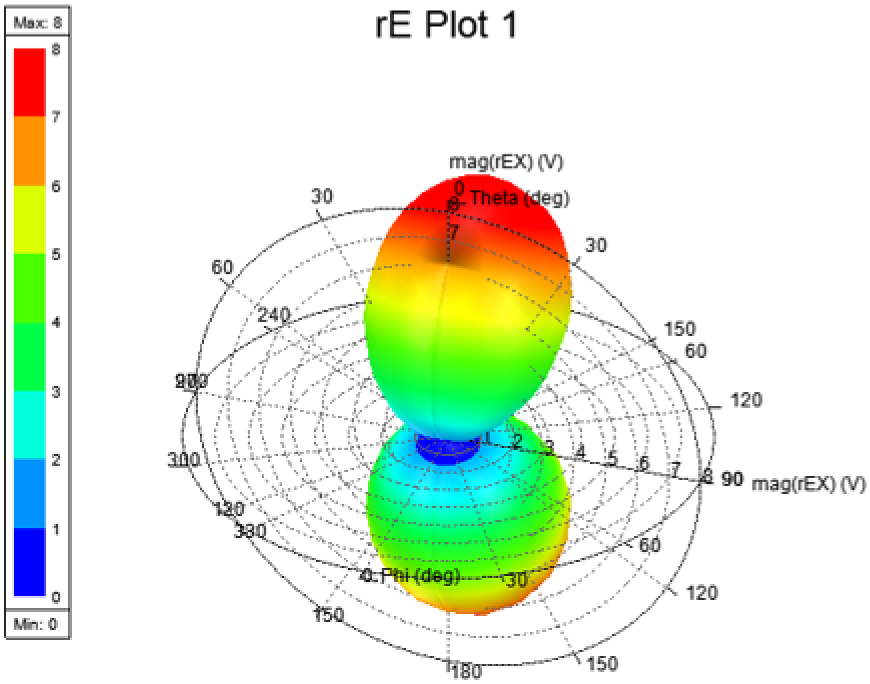
On creation, a 3D Polar Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under the Medium Visual Detail level, 3D Polar Plot has 3 ticks per axis (phi, theta and rho axis) which show min, max and middle value. This setting also shows axes labels.

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.

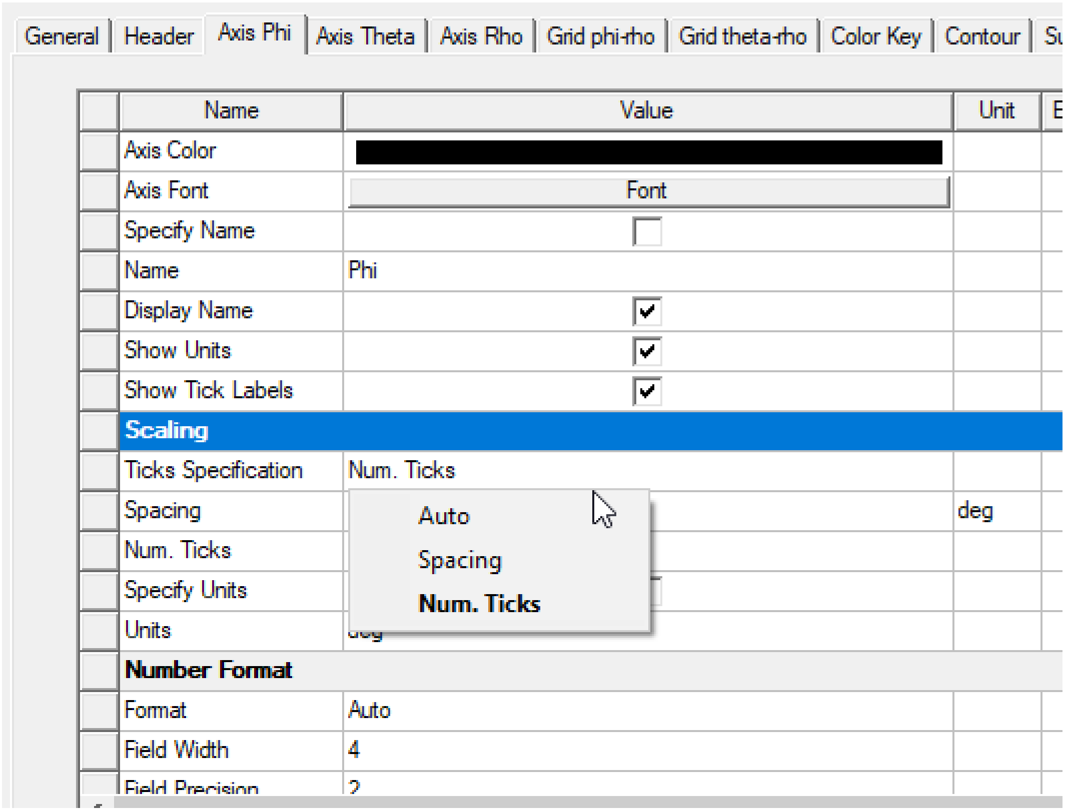
3D Polar Plot with High Visual Detail



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 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 **Circuit**> **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.

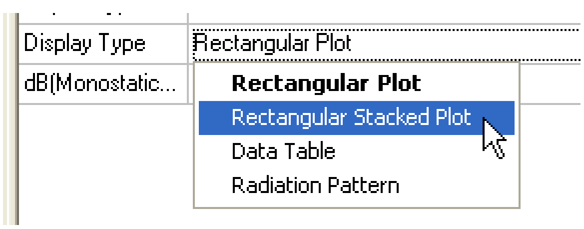
5. On the **Trace** tab **X (Primary sweep)** line, select the sweep variable from the drop-down menu and specify all values or select values.
6. Click **New Report**.

   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

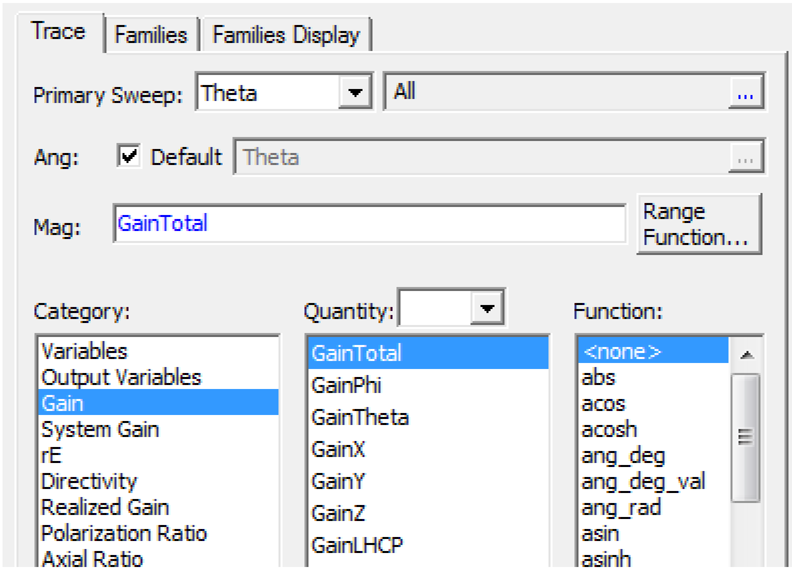
A radiation pattern is a 2D polar plot displaying the intensity of near- or far-field radiation patterns. It is divided by the spherical coordinates R and theta, where R is the radius, or distance from the origin, and theta is the angle from the x-axis. Following is the general procedure for drawing 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.

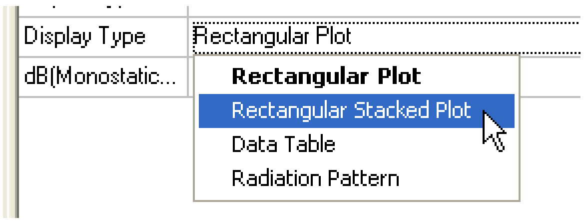
   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
5. The **Families** tab is helpful if you want to use the plot as an overlay and you need to restrict the Phi (default) or Theta values to make the plot appropriate for 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

> **Note:** Time-Domain Reflectometry is supported for Circuit LNA designs only.

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.

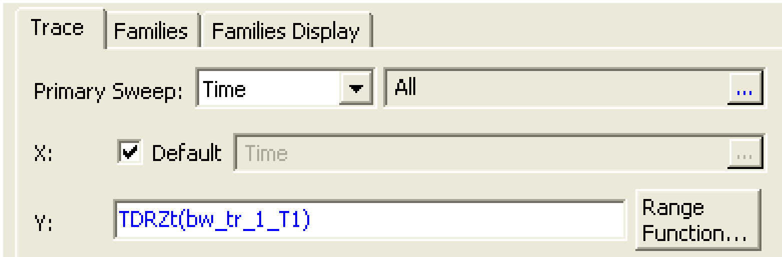
8. Optionally, under **TDR Window**, modify the window type and width.
9. You can use the **Save as Default** to set the current values as a default, and the **Use Defaults** button to use previously saved options. Note that when you select a trace, the initial displayed values are those of the selected 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.

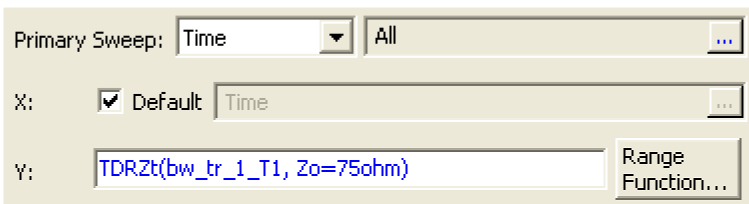
The default impedance (Zo) for the TDRZ quantity is 50 Ohms, unless you specified differently when you Set Renormalizing Impedance for Terminals when you created the terminals in the model. If you need a different impedance value, you can either edit the value in the **Report** dialog (as shown below), or you can create an Output Variable representing Zo × (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.



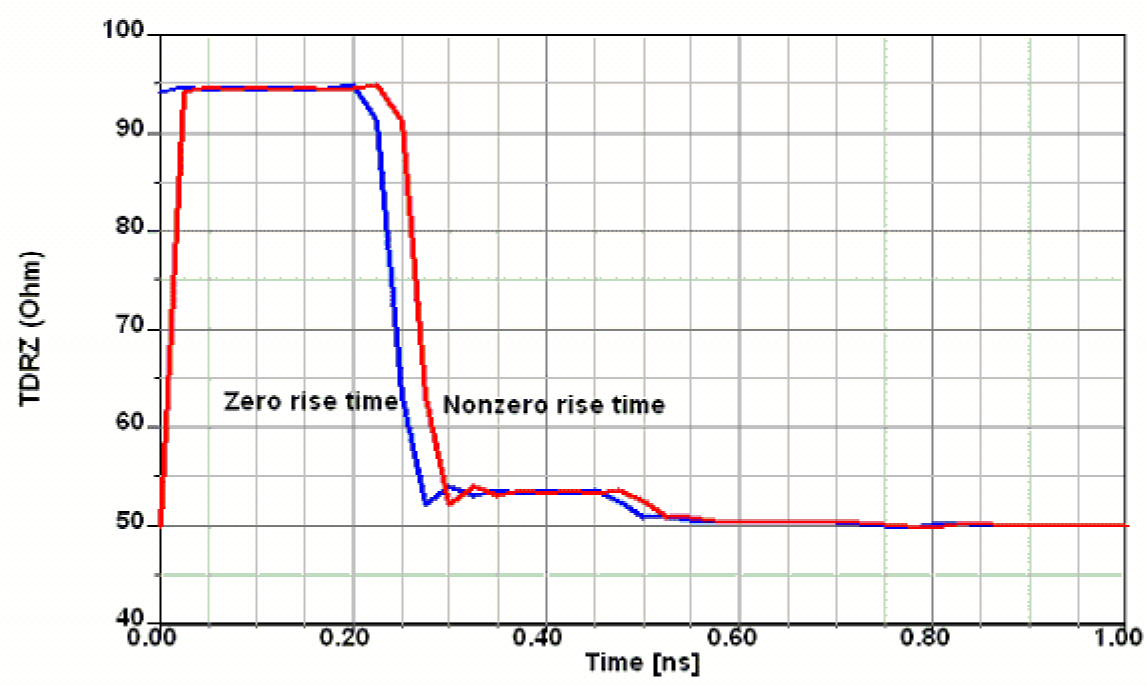
   c. In the **Function** list, click the mathematical function of the quantity to plot.

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.

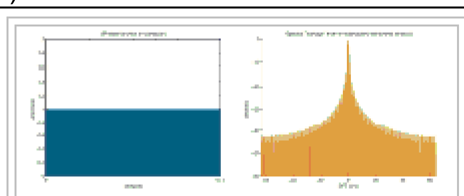
The TDR Options interface also lets you choose a smaller Δt than given by equation (3) above. When you choose a smaller Δt, you increase F_{max} by "zero padding" (adding zero values for S_{11} beyond the calculated frequency sweep). Whether this is justified depends on your judgment. It leads in practice to a smoother TDR signal.

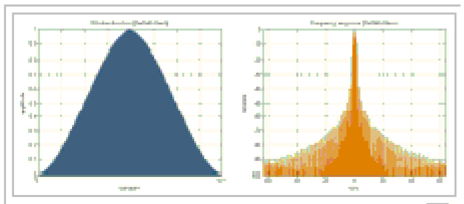
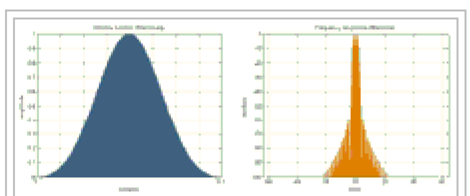
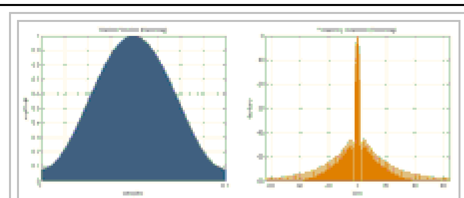
HFSS also lets you set the rise time of your input signal. The rise time should be at least $1/(2F_{max})$. Even this rise time is a bit short for comfort, as it equals the duration of only one time sample. An input signal with a longer rise time has a smaller high-frequency content and will lead to reduced "ringing" in the TDR response.

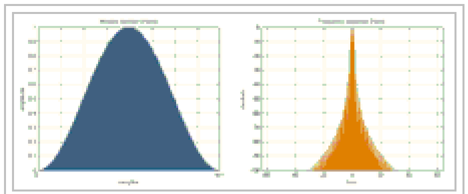
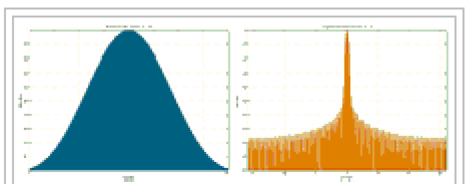
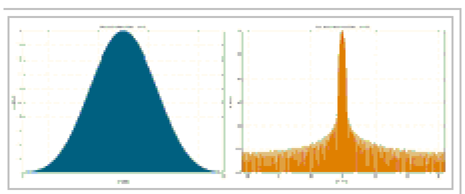
A Hamming or Hann filter will also reduce the high-frequency content and tends to lead to a smoother TDR response. With these filters, one can select a width. A width of 100% is often a good choice.

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:

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. $w(n)=1$.  Rectangular window; B=1.00
Bartlett	A high dynamic range function, with lower resolution, designed for wide band applications.

Window Function	Preferred Use
	$$w(n) = a_0 - a_1\left\|\frac{n}{N-1} - \frac{1}{2}\right\| - a_2\cos\left(\frac{2\pi n}{N-1}\right)$$ where a_0=0.62; a_1=0.48; a_2=0.38  Bartlett-Hann window; B=1.46
Blackman	A high dynamic range function, with lower resolution, designed for wide band applications. $$w(n) = a_0 - a_1\cos\left(\frac{2\pi n}{N-1}\right) + a_2\cos\left(\frac{4\pi n}{N-1}\right)$$ where a_0 =(1-α)/2; α_1=1/2; α_2=α/2  Blackman window; α = 0.16; B=1.73
Hamming	A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe. $$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$  Hamming window; B=1.37
Hanning (default)	A moderate dynamic range function, designed for narrow band applications.

Window Function	Preferred Use
	$w(n) = 0.5\left(1-\cos\frac{2\pi n}{N-1}\right)$  Hann window; B = 1.50
Kaiser	Selecting the Kaiser plot also enables a field to specify an associated Kaiser parameter. The larger the Kaiser parameter, the wider the window. The parameter controls the trade off between width of the central lobe and the area of the side lobes.  Kaiser window, α =2; B=1.5  Kaiser window, α =3; B=1.8
Welch	This approach applies a parabola-shaped window to the frequency domain data. It is based on the Bartlett method but splits the signal into overlapping segments, which are then windowed. The intent is to balance the influence of data in the center of the function.

You can use the **Save as Default** to set the current values as a default, and the **Use Defaults**

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 20 independent Y axis 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 an 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.

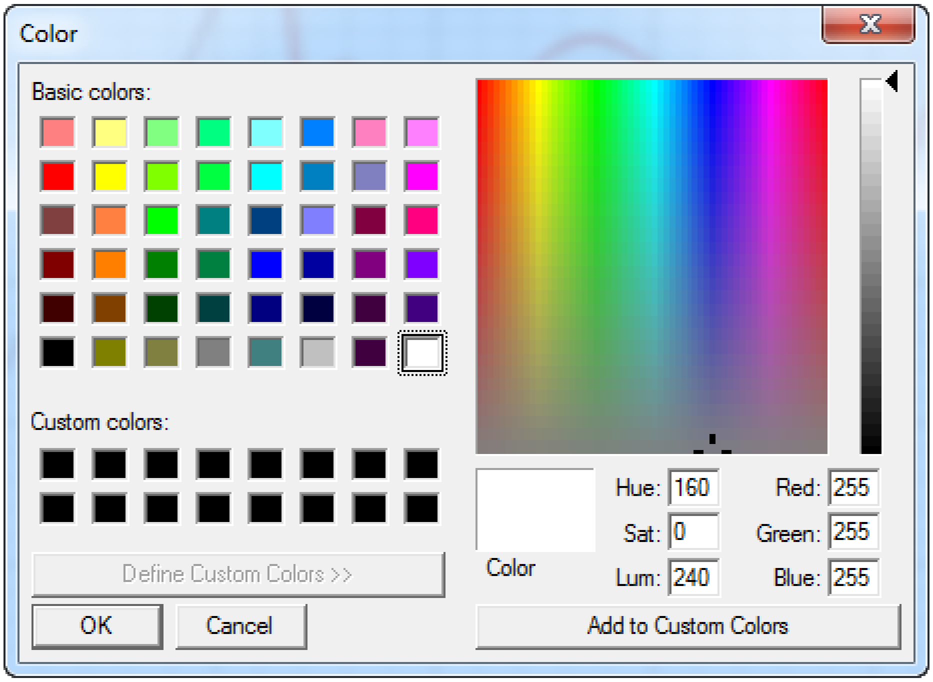
   The *General* tab properties apply to the general appearance of the plot. They include the Background color, Contrast color, Field width, and Whether to use Scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.)

   The *Attributes* tab properties apply specifically to the trace. The defaults are set in the Report2D options. They include:

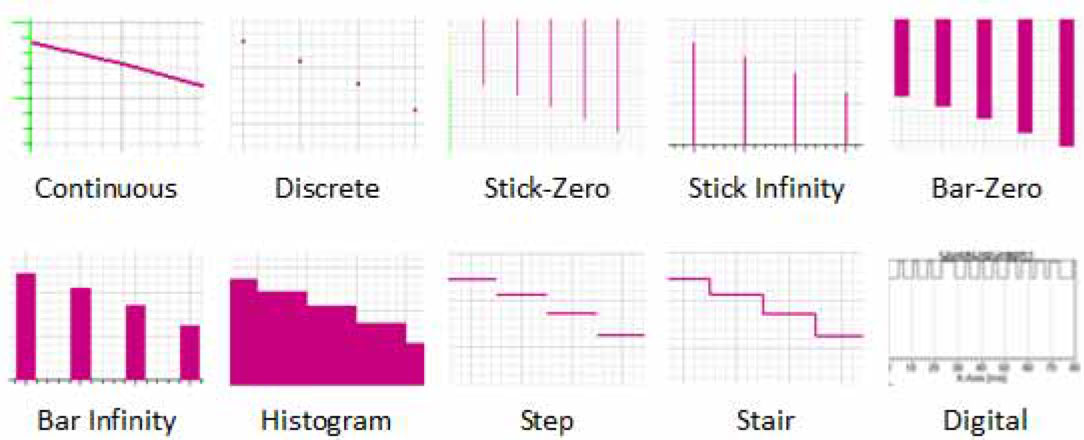
   - Name -- not editable by selecting the trace from the Report. It shows the characteristics of the trace as defined in the *Report* dialog box.

     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.

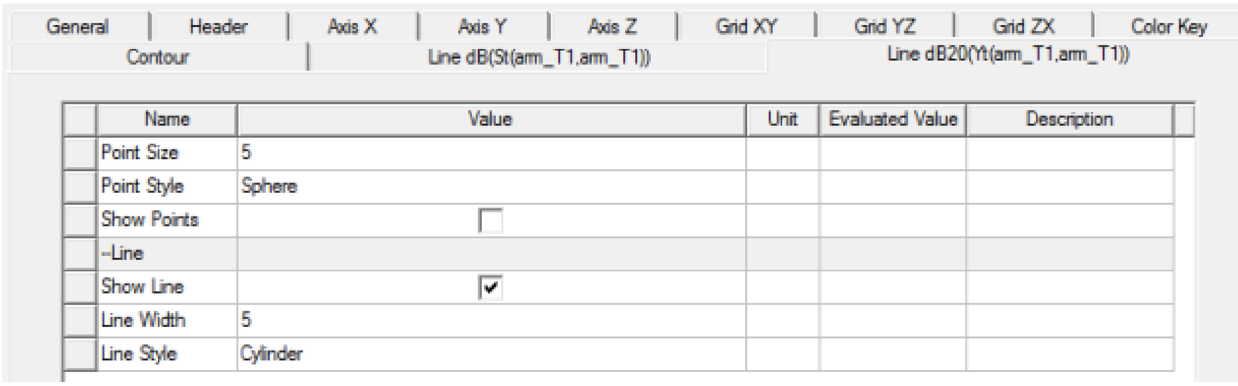
> **Note:**
>
> So that curves with single points always appear, Box is the default symbol.

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:

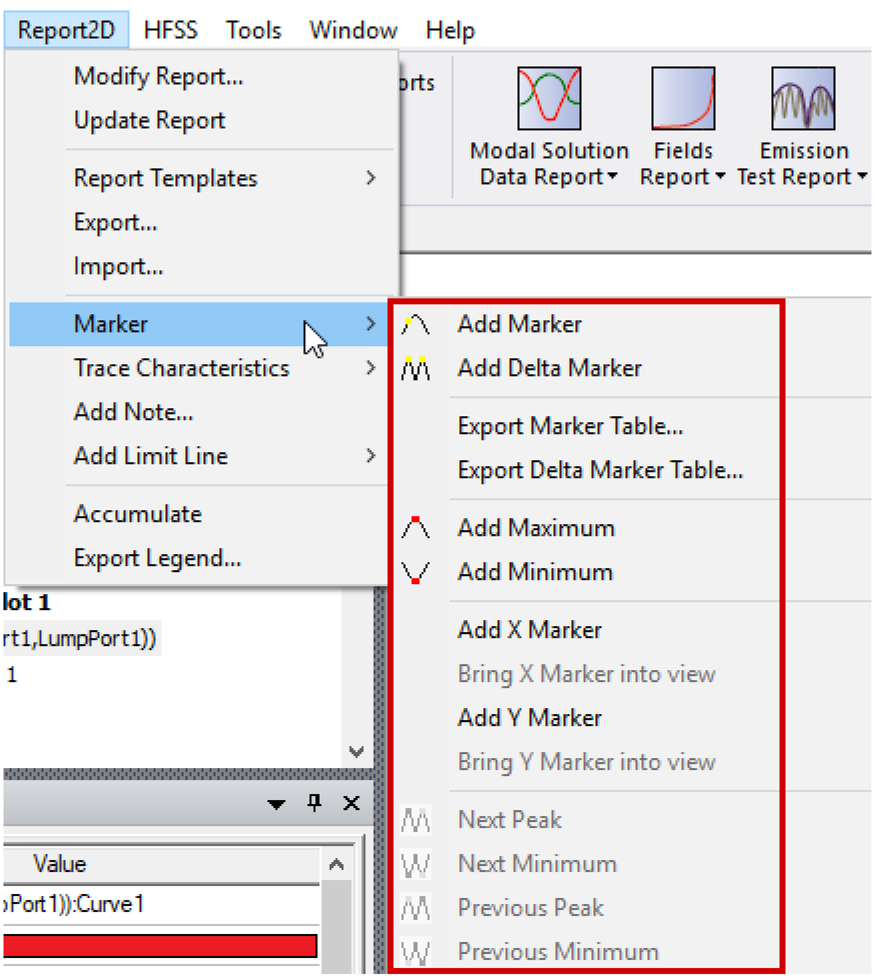
1. Click on the trace. This opens a Properties dialog for the plot with a tab named for the trace selected.
2. The editable properties include Point size, Point Style, whether to show points, whether to

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

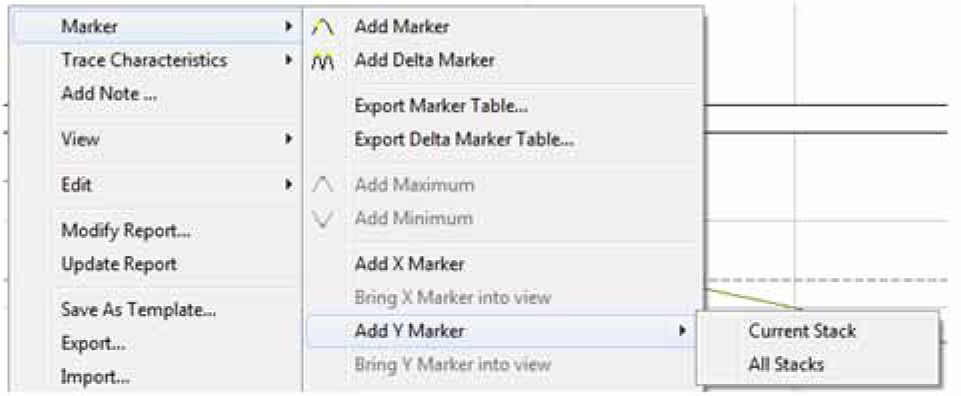
- **Next Peak**  – moves a selected marker on the next peak on a trace. You must exit marker mode and select a marker to enable this command.
- **Next Minimum**  – moves a selected marker to the next minimum on a selected trace. You must exit marker mode and select a marker to enable this command.
- **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

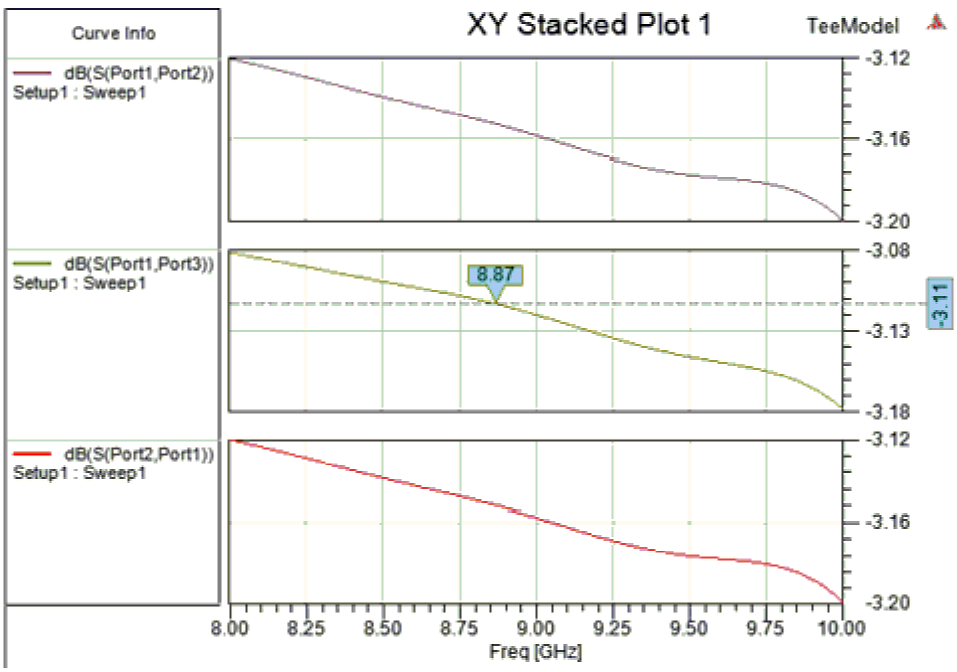
Y Markers allows for easy analysis and comparison of curves at a particular y-coordinate. Y Markers can be used to compare stacked curves.

Creating Y Markers in Stacked Plots

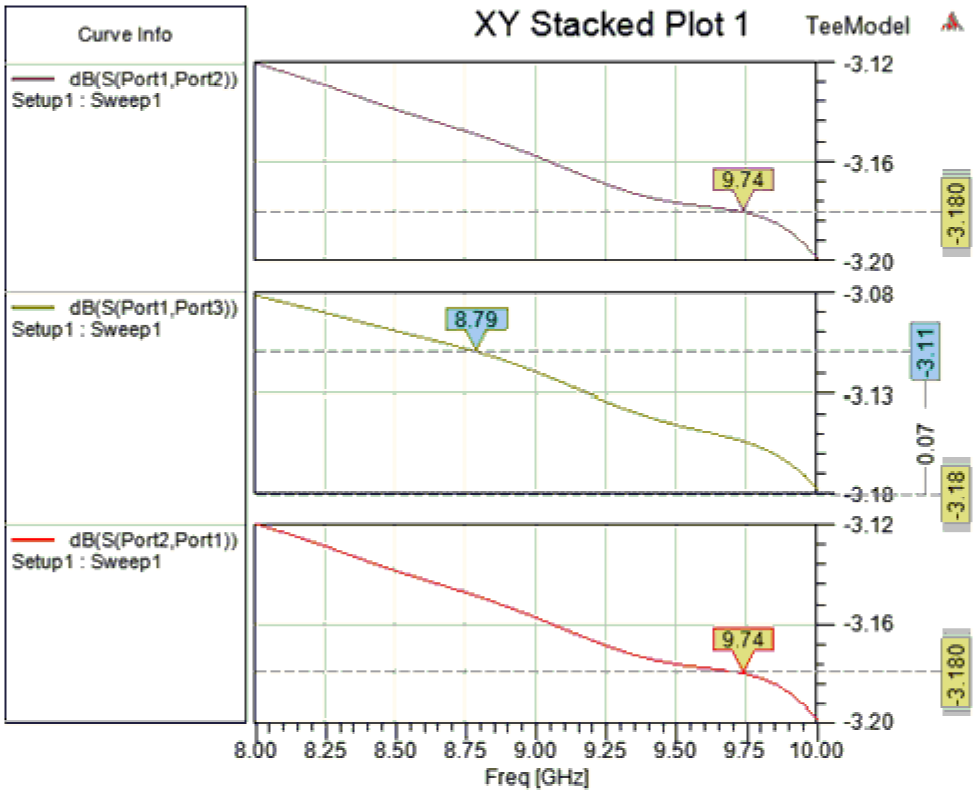
There are two ways to create Y Markers in Stacked Plots. You can create a Y Marker for a particular stack or for all stacks. Right-clicking on any stack shows the following shortcut menu:



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:



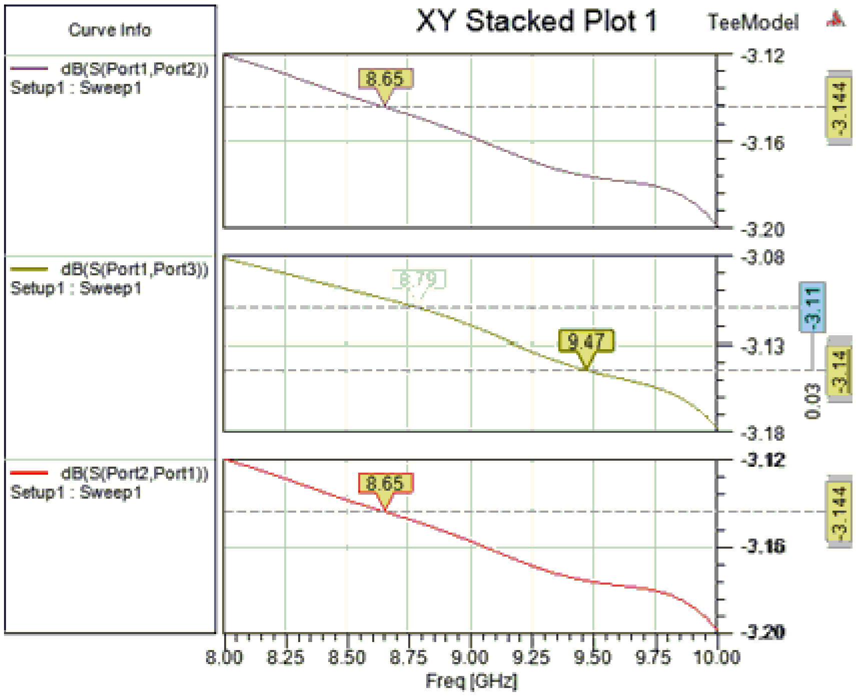
Add Y Marker >All Stacks creates one Y marker in each stack with same value. Initially this value is the minimum Y value of the Y ranges in all the stacks. This is shown is figure below:



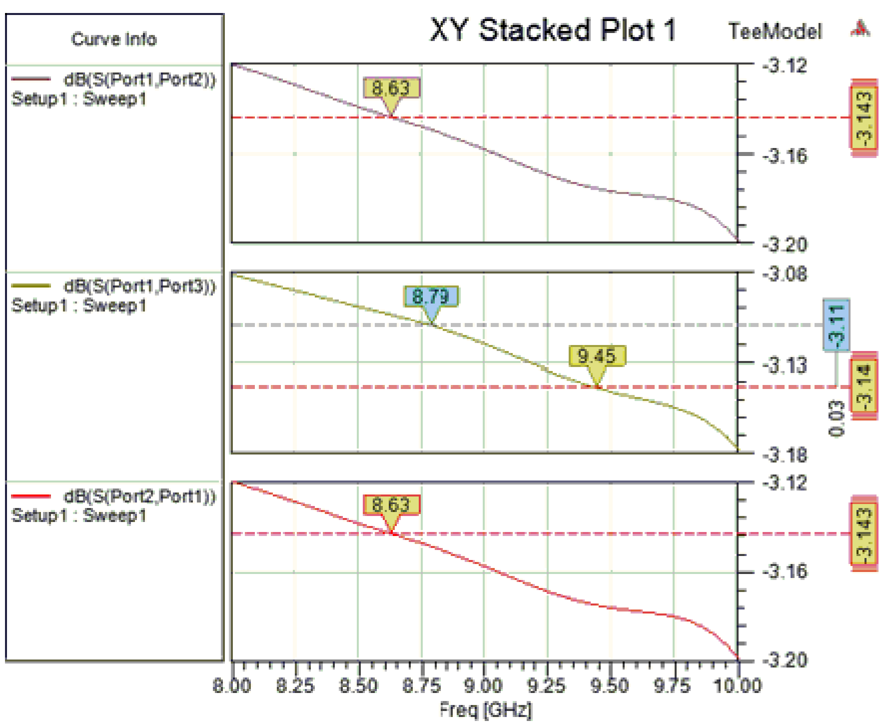
Notice that the Y Marker for All Stacks has a different appearance than the Y Marker for a particular stack, that is, it has double parallel lines above and below the Y Marker textbox.

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:

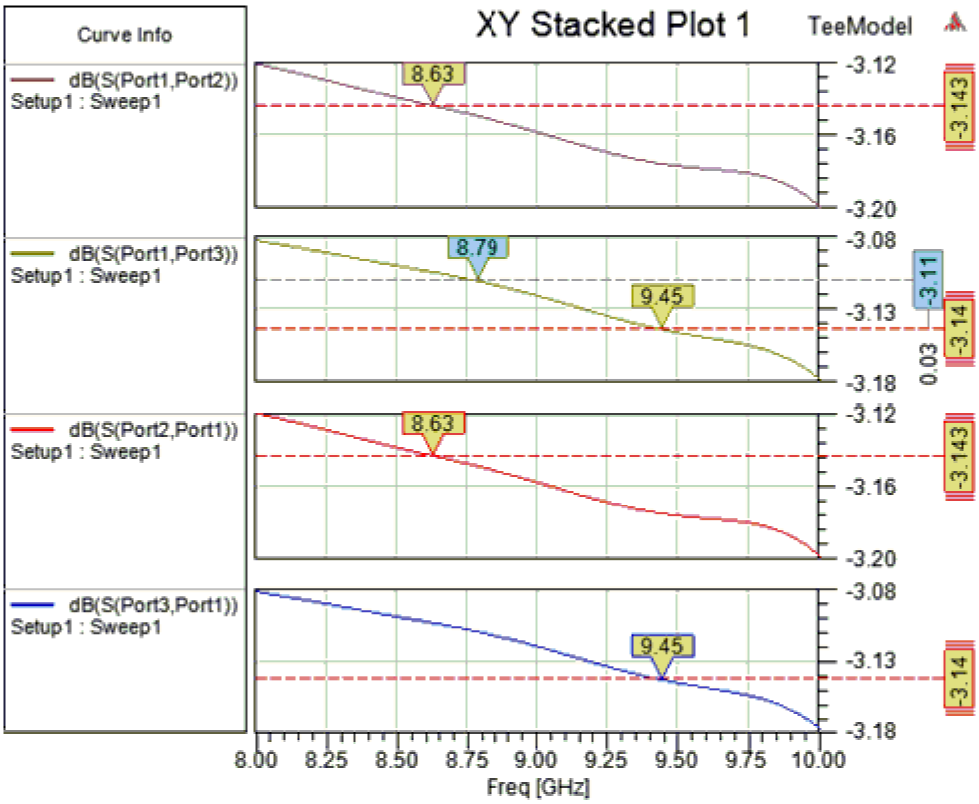


Also if a property of any one Y Marker is changed, all the "same" Y Markers show the change in property as well. For example the following figure shows that when the line color of a Y Marker in the top stack was changed to red color, a Y Marker in bottom stack show the same line color as well:



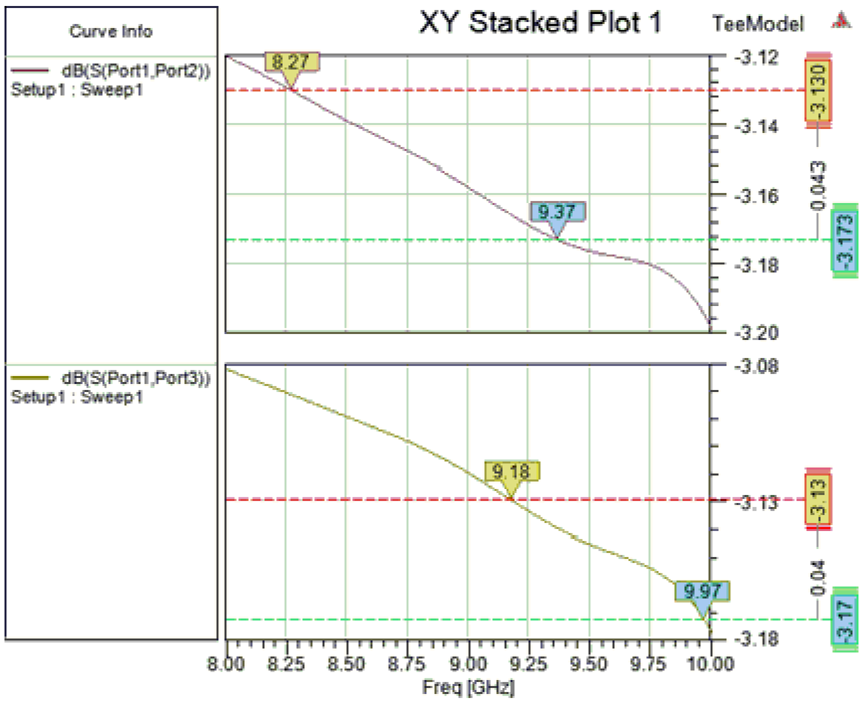
Automatic Y Markers for the new stack

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:



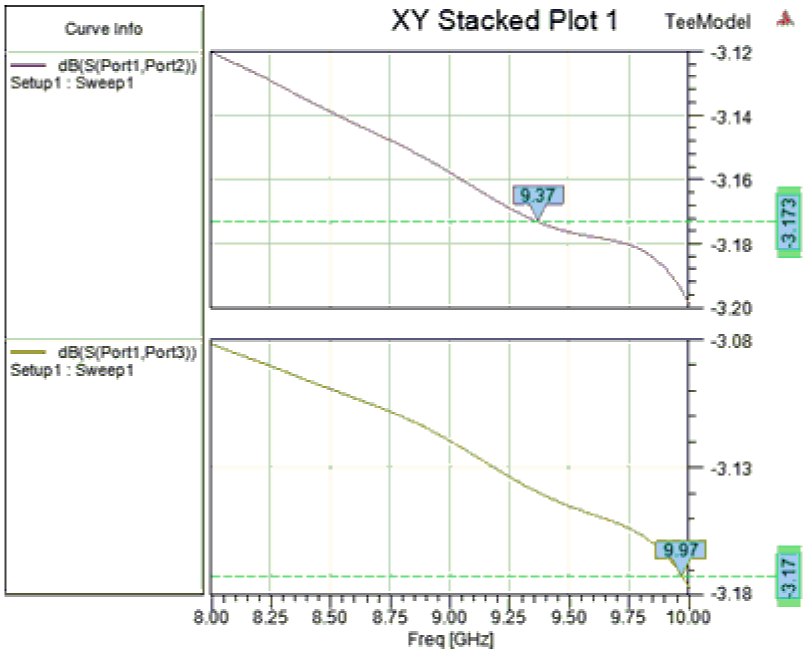
Y Marker Delta Annotations

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

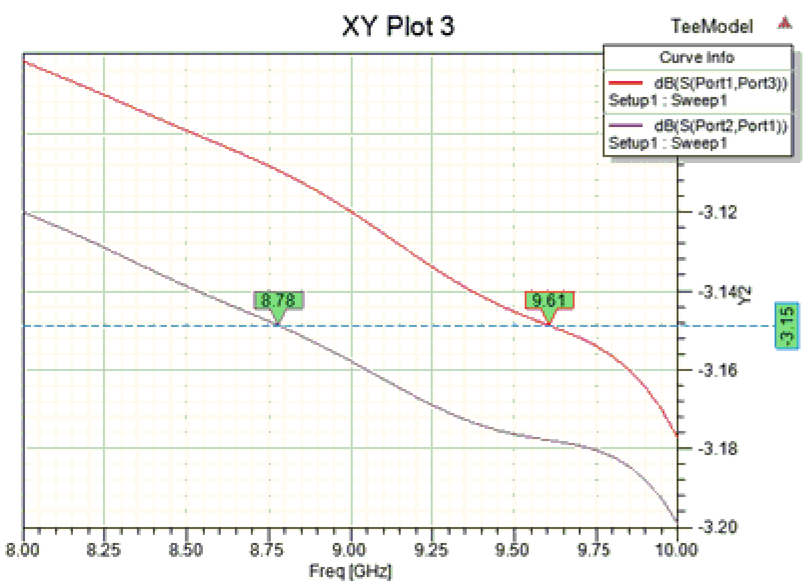
To delete a Y Marker, select a Y Marker in any stack and press the Delete key. This action will also delete all the corresponding Y Markers in all the stacks. For example, when the Y Marker with value -3.13 (red Y Marker) was deleted from the bottom most stack, all of the corresponding Y Markers were also deleted:



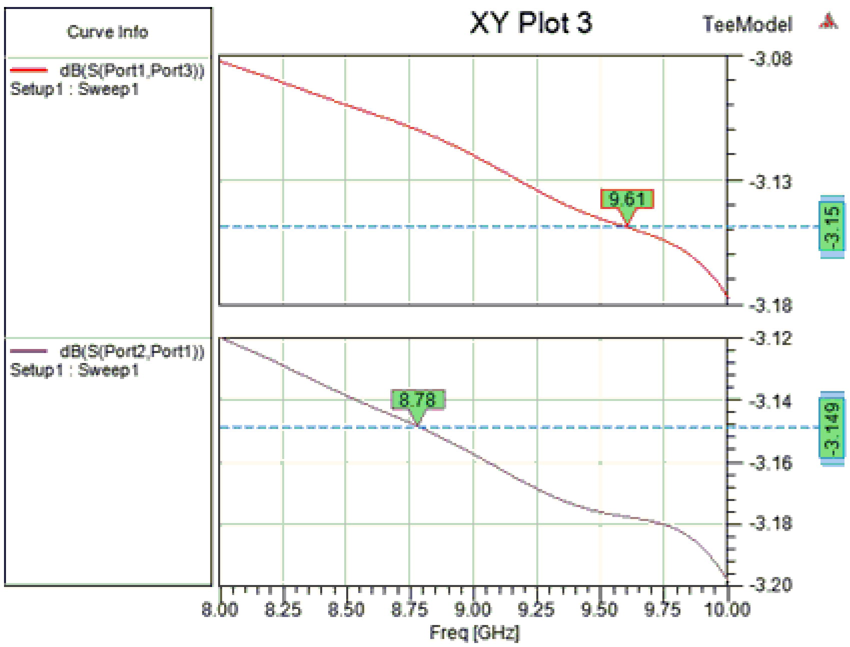
Note that on deleting a stack, Y Markers in other stacks are not affected.

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):

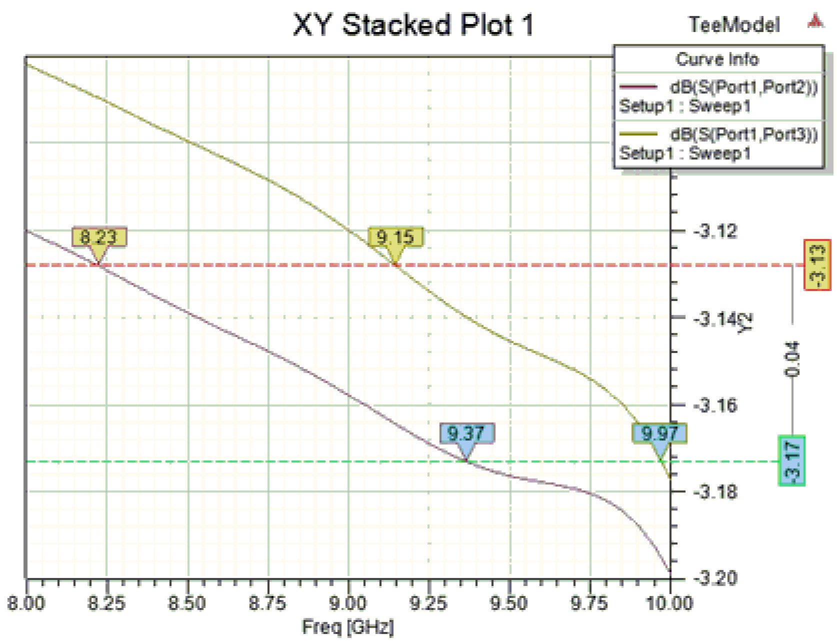


If you change the Display Type property of this plot to Rectangular Stacked Plot then a Rectangular Stacked XY Plot is created with each curve in its own stack and a Y Marker is shown in each stack with value -3.15 (blue Y Marker):



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).

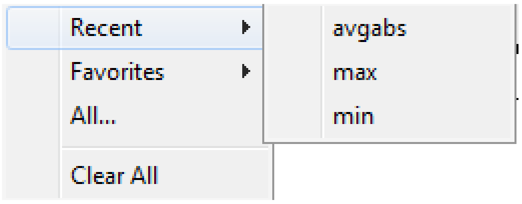
1. Double-click the X or Y axis of interest on an open plot display.

   This opens the **Properties** window for the Axis.

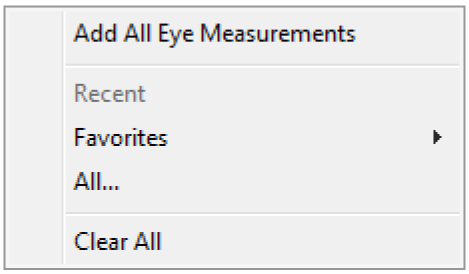
2. Under the **Axis** tab, use the scroll bar to find the **Specify Discard Values** property.
3. Click the check box to enable the property.
4. Enter a value in the **Discard Below** field. Units specified elsewhere in the Axis property are applied to this value. The Discard Below text box is inactive 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.

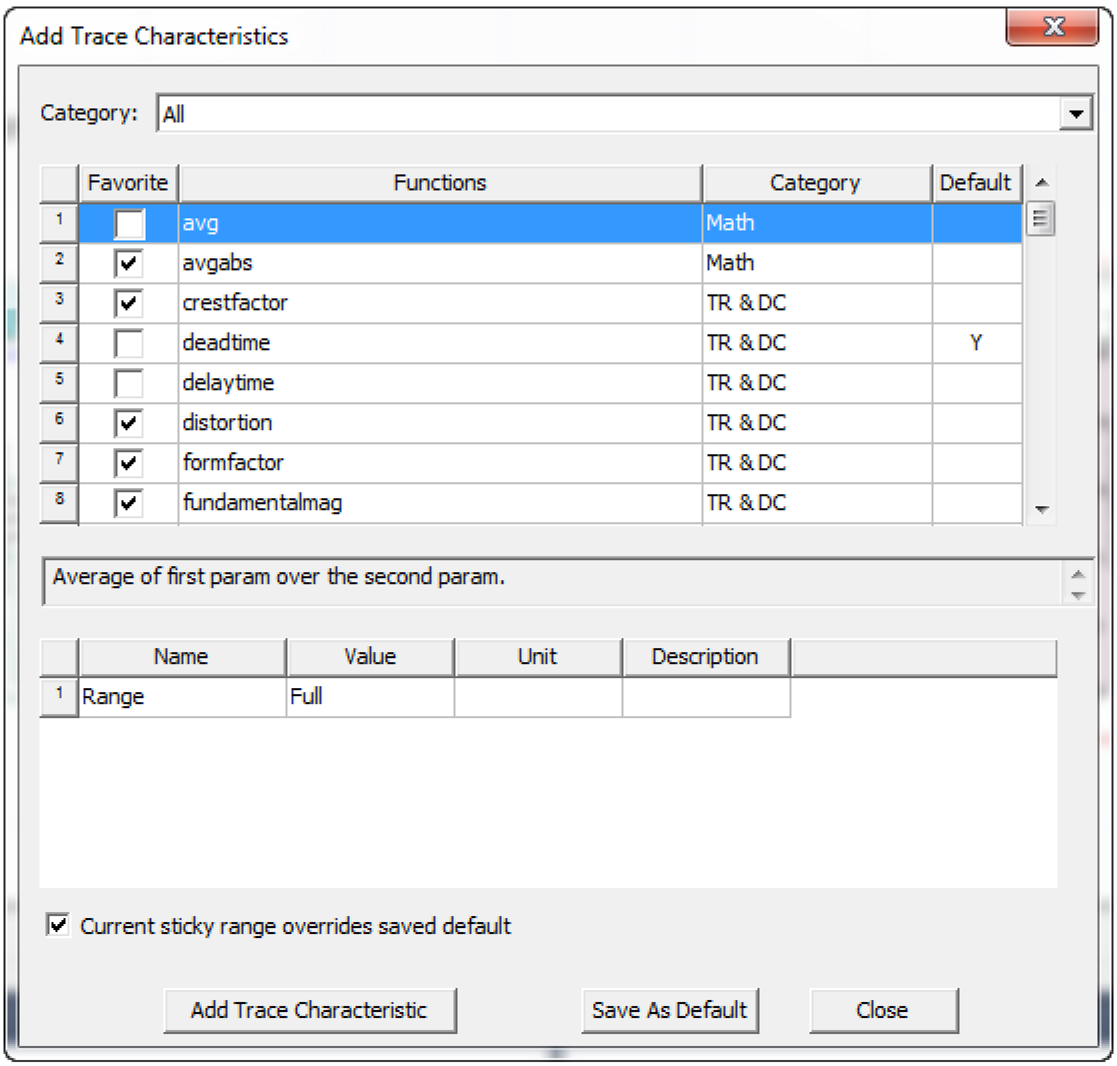
Adding Trace Characteristics to your Favorites

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 10 are displayed in the menu, and they are displayed in alphabetical order.
5. Click **Close**. You can view the current favorites by selecting Favorites in the Category drop-down menu.

> **Note:**
>
> You can remove favorites by clearing the **Favorite** check box for one or more functions, and clicking **Close**.

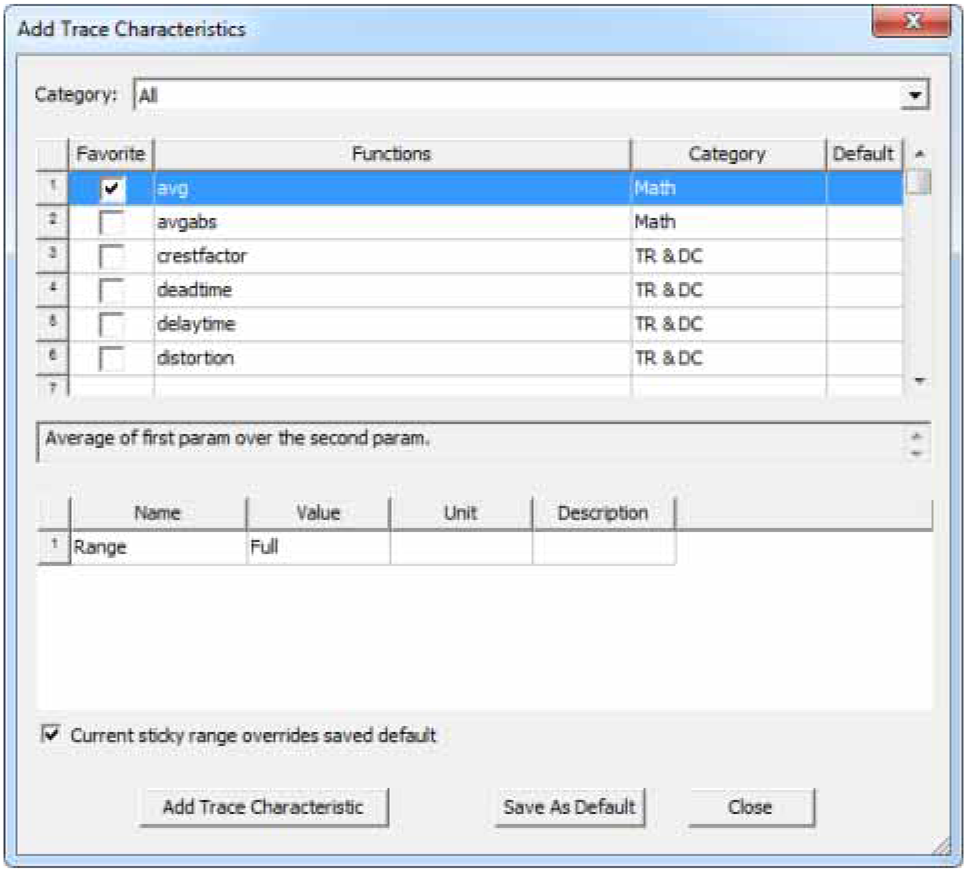
Adding Characteristics Using the Add Trace Characteristics Dialog Box

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,

Category	Functions
	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.

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.

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**.

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 which contains a definition with the same name, then an incremented number is appended to the pasted report or trace name.

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:

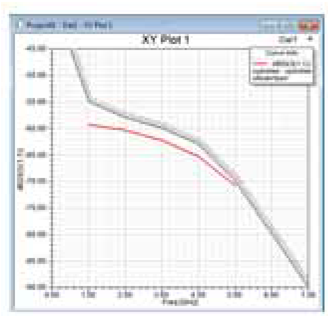
Right-click the report in the target design to which you would like to copy the trace data and select **Paste**.

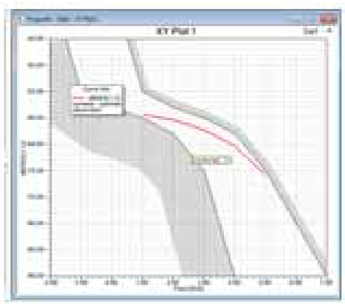
> **Note:**
>
> If you copy and paste report or trace data which contains the same name definition as a report or trace in the target design then an incremented number will be appended to the pasted name.

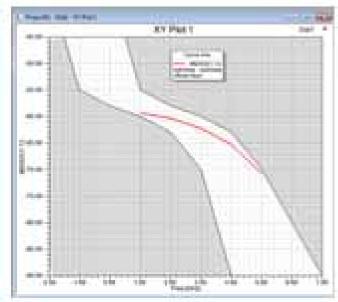
Limit Lines in Cartesian Plots

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)



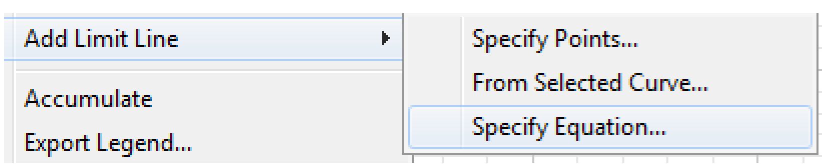




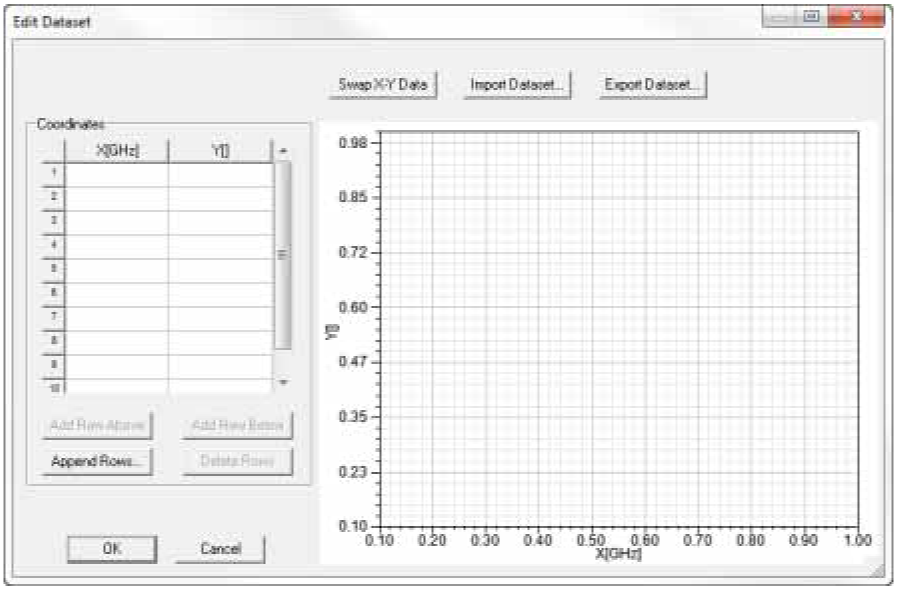
> **Note:**
>
> Limit lines are available only on Rectangular (XY) plots, not on XY-like plots such as Bode or Rectangular Stacked. On 2D Plots, the axes extents are based on the extents of the curves.

To create a limit line:

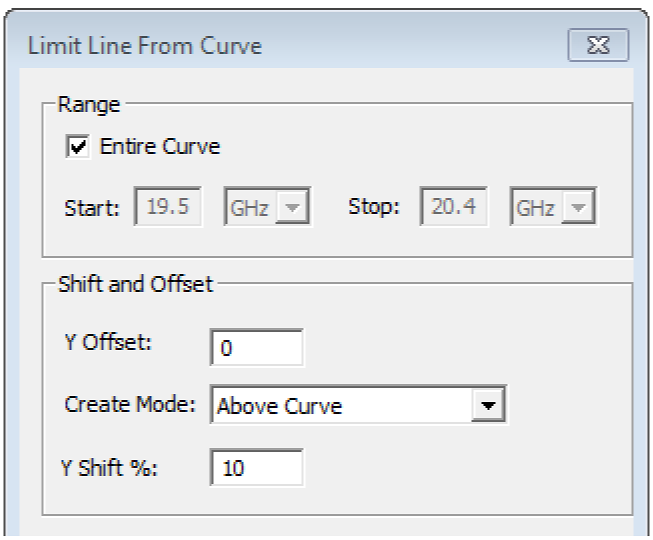
1. Click **Report2D> Add Limit Line** or right-click an XY plot and select **Add Limit Line...** from the Context menu. You then select whether to **Specify Points**, or **From a Selected Curve**, or **Specify Equation**.

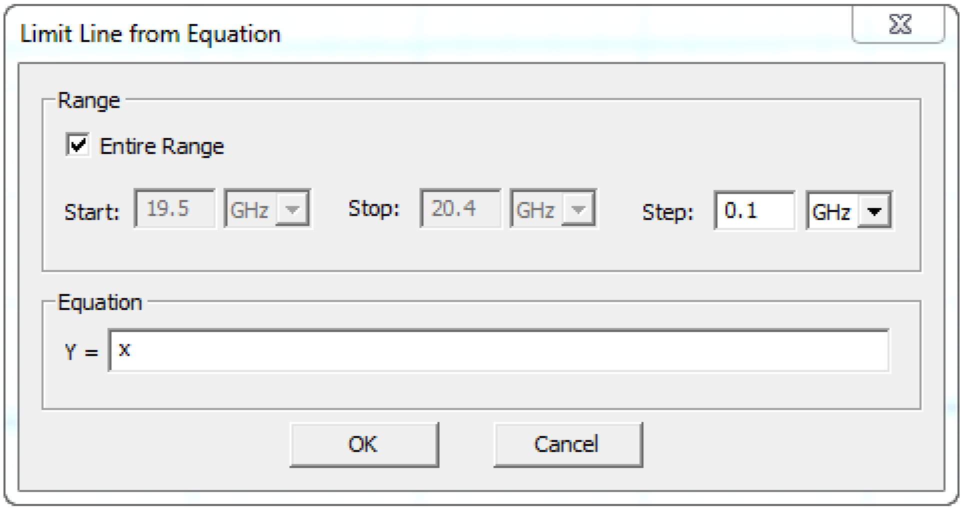
   Select **Specify Points** to open an *Edit Dataset* dialog box so that you can specify points.



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.



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:

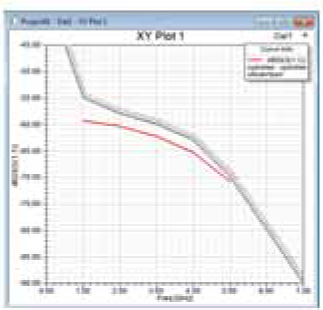
   Each limit line is associated with a particular Y axis (because it has to be scaled the same way as all the curves associated with the axis, follow its log/linear scale and so on). This Y axis association defaults to the first available Y axis when the limit line is created. However, if the plot contains 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.

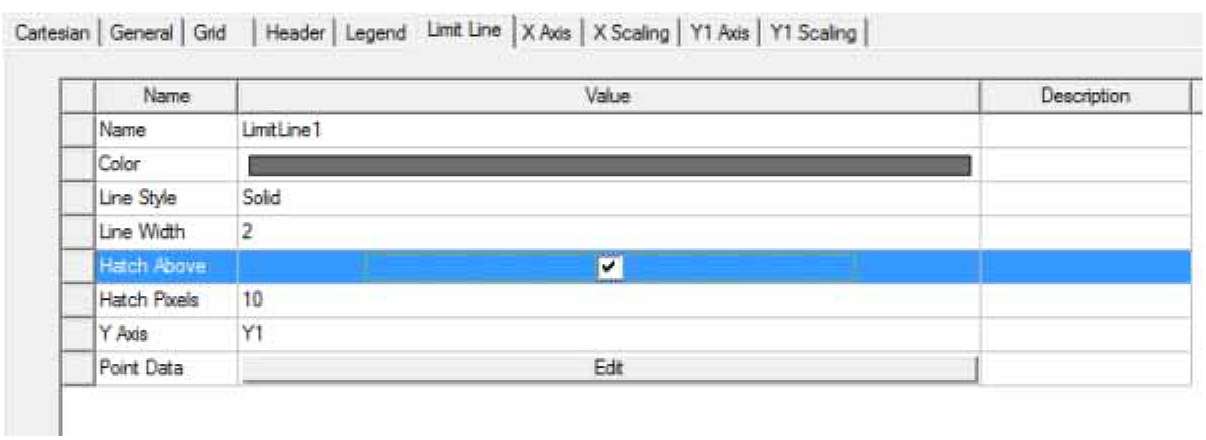
   If you uncheck “Entire Curve,” the Start and Stop fields are enabled and initialized based on the zoom level.

   - 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.



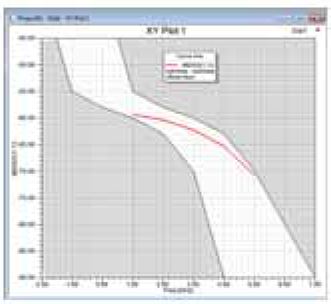
Name	Value	Description
Name	LimitLine1	
Color		
Line Style	Solid	
Line Width	2	
Hatch Above	✔	
Hatch Pixels	10	
Y Axis	Y1	
Point Data	Edit	

- 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.

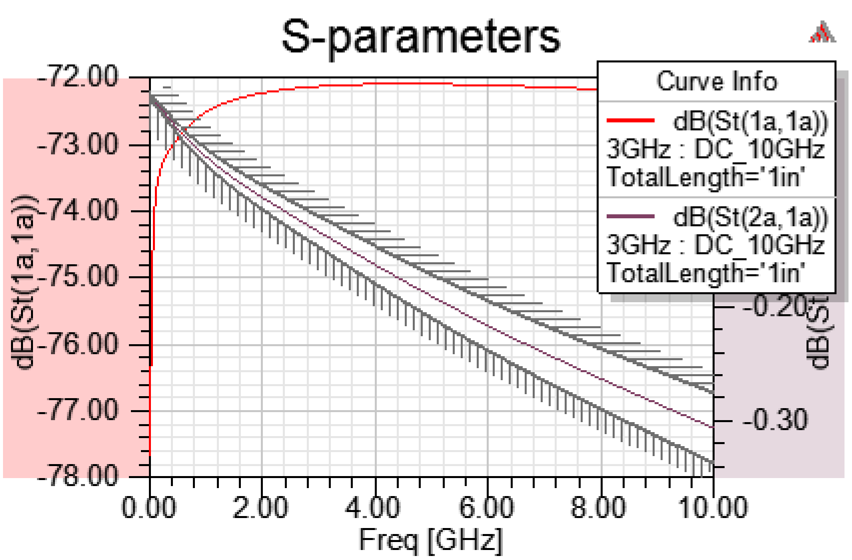
For a limit line specified as an Equation, the Properties also include:

- 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.

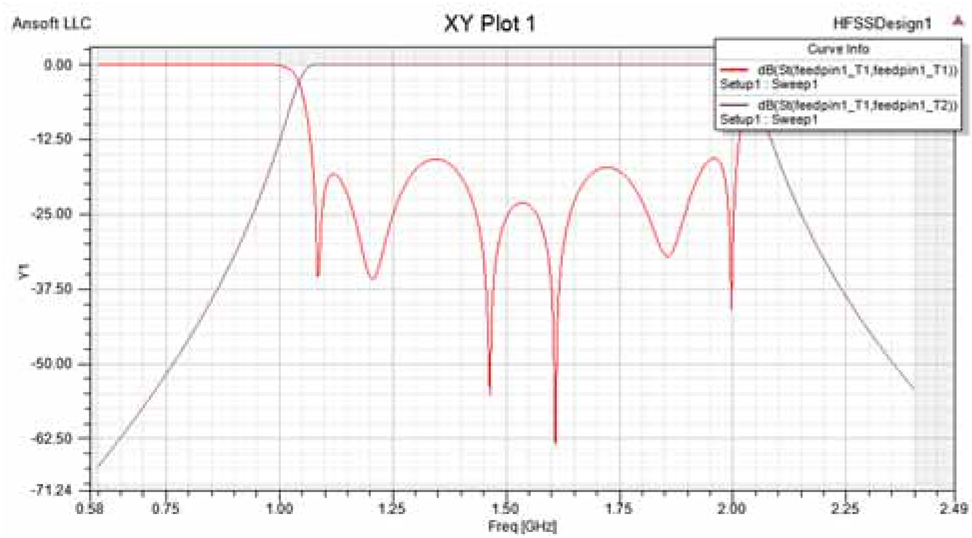


The following example shows the a limit line from curve plot, where the Hatch Above property for the lower limit line has been unchecked.

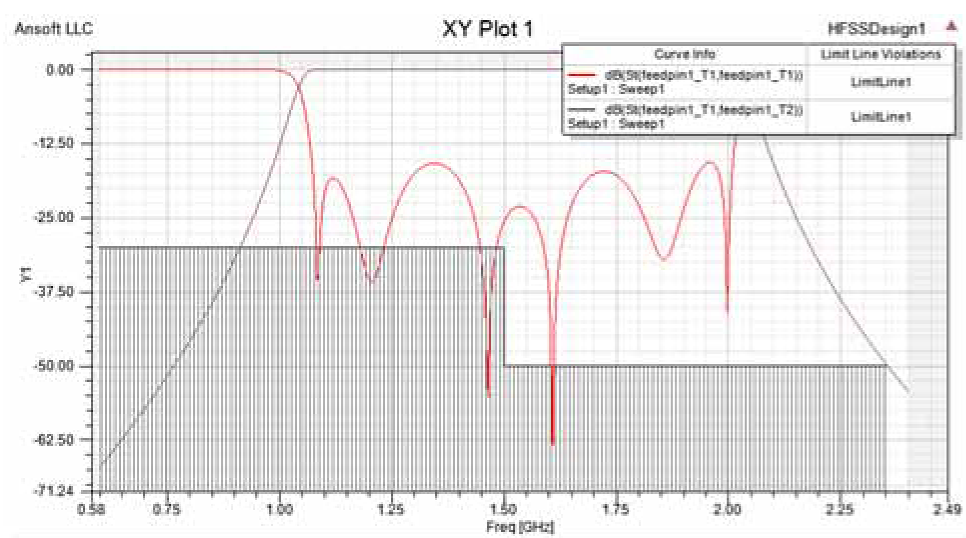


Limit Line Violations

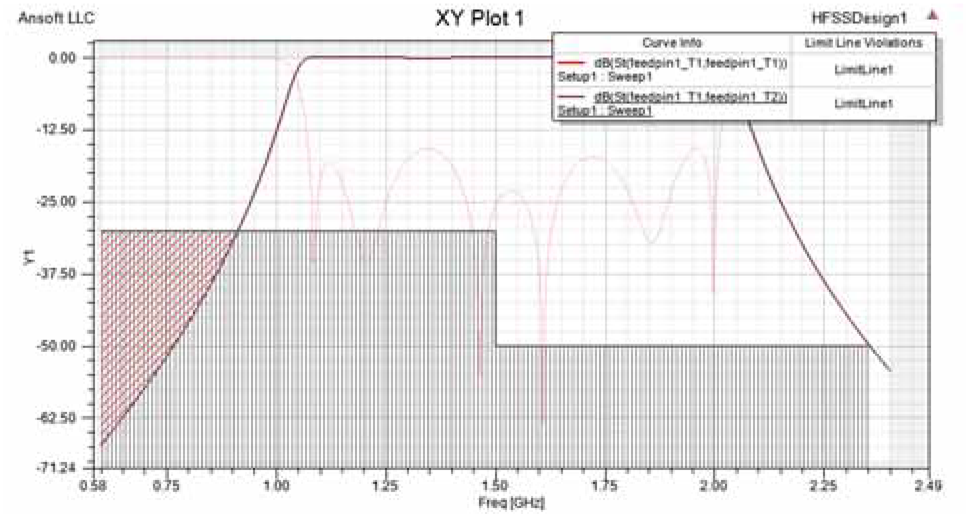
You can use a plotting feature to help you discern whether a curve violates a limit line or not. Consider following plot which shows two curves:



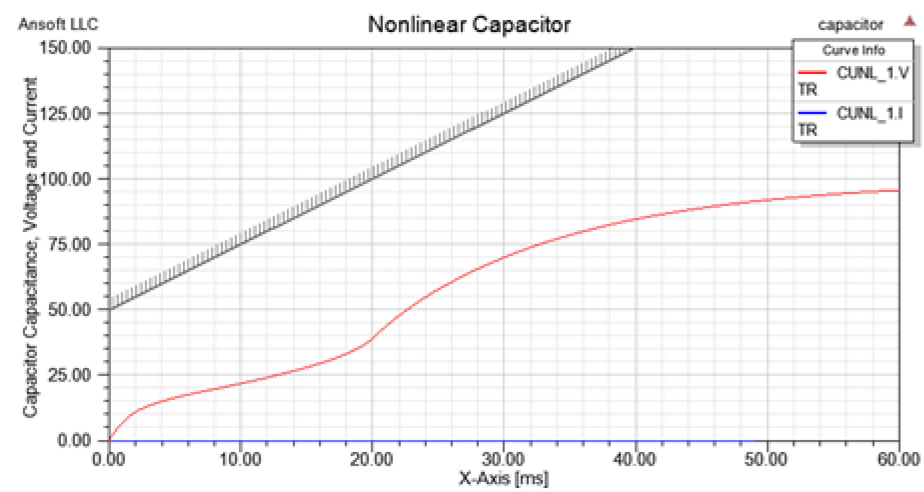
Suppose that the response cannot be below -30 dB until 1.5 GHz, and cannot be below -50 dB at 1.5 GHz or higher. You add a limit line for this requirement in the plot using 'Add a Limit Line' functionality. The plot automatically calculates whether a curve violates this requirement, that is, the limit 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 Circuit, 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.

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 quanitities) is computed as: 20.0*log10(x) + 60.0
dBu	(for electric field quanitities) is computed as: 20.0*log10(x) + 120.0
dBW(x)	10*log10(|x|).
dB10	10*log(|x|) to base 10.
dB10normalize	10*log [normalize(mag(x))].
dB20	20*log(x) to base 10.
dB20normalize	20*log [normalize(mag(x))].
deadtime	Obtains the latest time when the qtyl is within a tolerance of zero.
delaytime	Obtains the time from zero to 50% of the target point.
degel	Conversion from degrees electrical to seconds with respect to Hz.
deriv	Derivative of a given parameter.
distortion	Returns the total distortion for the selected simulation quantity and an additional argument frequency, which is the frequency in Hz at which to calculate the fundamental RMS of the simulation quantity.
even	Returns 1 if integer part of the number is even; returns 0 otherwise.
exp	Exponential function (the natural anti-logarithm) of the simulation quantity.
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	Returnns the root mean square value of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
root	nth root function.

rSidelobeX	Returns the X value of right side-lobe occurrence.
rSidelobeY	Returns the Y value of right side-lobe occurrence.
settlingtime	Returns the latest time at which the value of the selected simulation quantity fell outside its tolerance band. The target value of the quantity and the +/- bandwidth of the tolerance band are the additional args.
sgn	Sign extraction.
sin	Sine.
sinh	Hyperbolic sine.
slidingmean	Returns the moving average value of the selected simulation quantity (specified by the first argument). The average is calculated over a period (specified by the second argument).
slidingrms	Returns the moving RMS value of the selected simulation quantity (specified by the first argument). The RMS value is calculated over a period (specified by the second argument).
sqr	Square of the selected simulation quantity.
sqrt	Square root of the selected simulation quantity.
stddev	Returns the standard deviation of given values.
sum	Returns the sum of the given values.
tan	Tangent.
tanh	Hyperbolic tangent.
undershoot	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.
uppercutoff	Returns the upper 3dB frequency of the selected simulation channel in Hz.
variance	Calculates the variance of the given values.
XAtYMax	Threshold crossing time: report first time (x value) at which an output quantity crosses YMax.
XAtYMin	Threshold crossing time: report first time (x value) at which an output quantity crosses a user definable threshold.
XAtYVal	Returns the X value at the first occurrence of Y value.
XWidthAtYVal	Returns the X width between the first 2 occurrence of Y value.
xdb10beamdwidth	Width between left and right occurrences of values 'x' db10 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
xdb20beamwidth	Width between left and right occurrences of values 'x' db20 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.

YAtXMax	Returns the X value at maximum value of Y.
YAtXMin	Returns the Y value at minimum value of X.
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).

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:

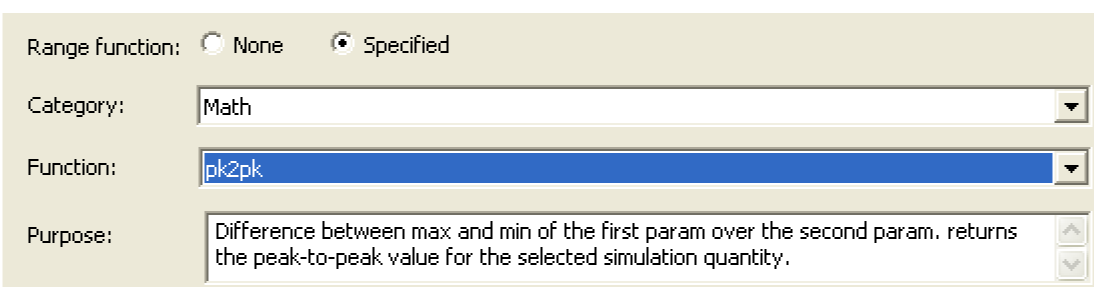
1. Click **Range Function** in the *Report* dialog box.

   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.
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.

Math Functions	
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.

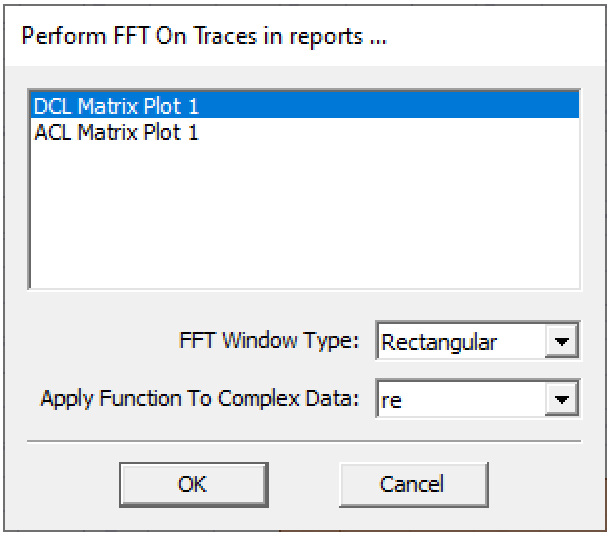
Eye Measurement Range Function Parameters

The Eye Measurement category of range functions provide the means to evaluate several characteristics of eye diagrams. Each of the Eye Measurement functions includes the following parameters. Specify the Value by editing the Value text field for the parameter name. Specify the unit for the parameter by selecting from the Unit drop-down menu.

Name	Default Value	Default Unit	Description
Unit Interval	0	ns	Unit interval of signal
Start Offset	0	ns	Offset at beginning of signal
End Offset	0	ns	Offset at end of signal
Autocrossing amplitude	1		Nonzero number means that crossing amplitude is calculated automatically.
Crossing amplitude	0	mV	Specify crossing amplitude used for eye measurement data computation.

Perform FFT on a Report

You can perform FFT on an existing 2D plot by using the **Results** > **Perform FFT** command. You can perform TDR on an existing 3D plot by using **Circuit** >**Results** > **Perform FFT**. This opens the **Perform FFT on Traces in Reports** dialog box.



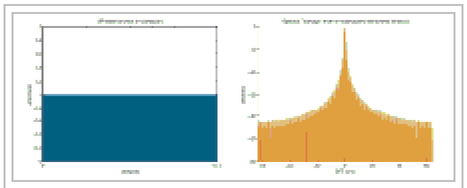
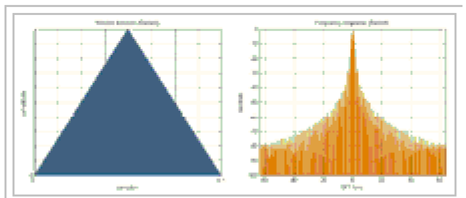
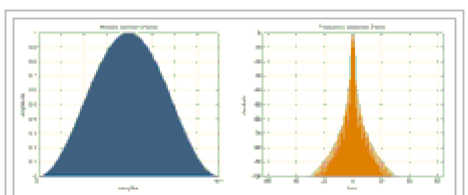
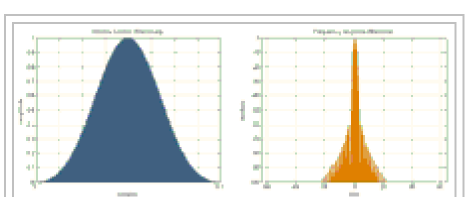
1. Select the report you want from the list.
2. Select the FFT window type to apply. Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies.
3. Select the function to apply to complex data.

   The new report 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

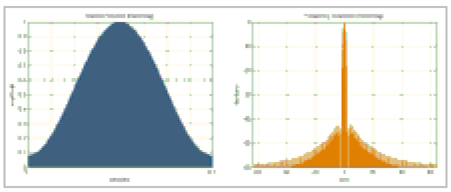
The window type list for **Perform FFT on Report** includes:

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. $$w(n) = 0.5\left(1 - \cos\frac{2\pi n}{N-1}\right)$$  Hann window; B = 1.50
Blackman	A high dynamic range function, with lower resolution, designed for wide band applications. $$w(n) = a_0 - a_1\cos\left(\frac{2\pi n}{N-1}\right) + a_2\cos\left(\frac{4\pi n}{N-1}\right)$$ where $a_0 = (1-\alpha)/2$; $\alpha_1 = 1/2$; $\alpha_2 = \alpha/2$  Blackman window; α = 0.16; B=1.73

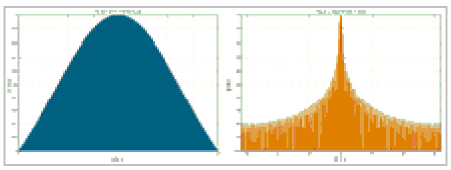
Window Function	Preferred Use
Hamming	A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe.

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$



Hamming window; B=1.37

Lanczos	The Lanczos window offers a windowed form of the infinite sinc filter, providing the central lobe of a horizontally stretched sinc, sinc(x/a) for -a £ x £ a.



Sinc or Lanczos window; B=1.31

Weber	
Welch	This approach applies a parabola-shaped window to the frequency domain data. It is based on the Bartlett method but splits the signal into overlapping segments, which are then windowed. The intent is to balance the influence of data in the center of the function.

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.

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

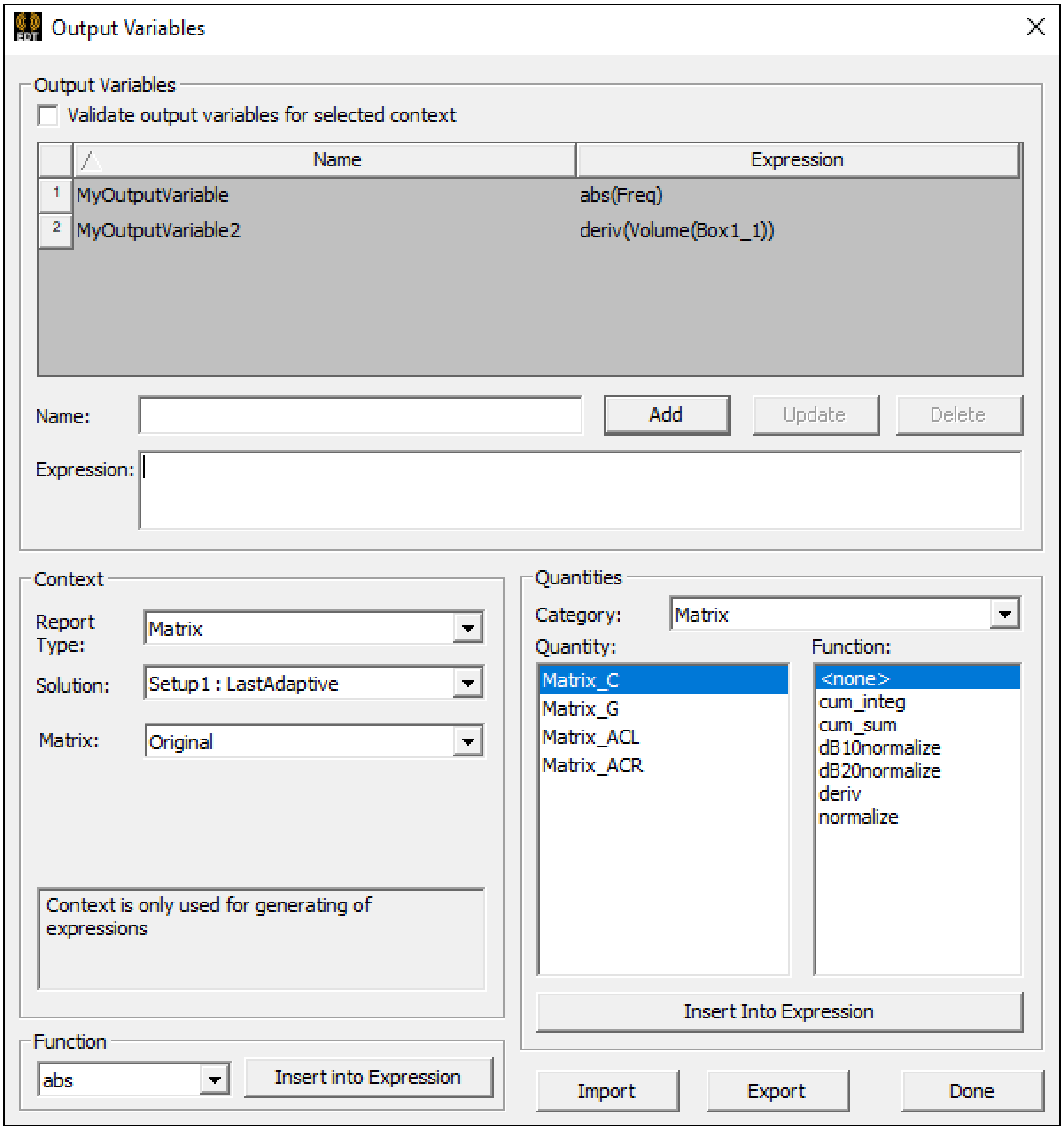
There are two modes of animation: frequency-based and phase-based. Frequency-based animation creates a series of frames by calculating the field overlay at each of the swept frequencies. Phase-based animation creates a series of frames by calculating the field overlay at a fixed set of phase deviations around a selected frequency. The animation controls can also change the simulation design point.

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 **Circuit** > **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.

Derivative Tuning for Reports

The Derivative Tuning feature available in the results menu is based on the derivatives that you can request for selected variables in the solution setup. It is limited to quantities like S-parameters and Far Fields, where far-field tuning supports the following:

- Incident Wave
- Linked Field
- Voltage Source
- Current Source
- Magnetic Bias : Note: Sensitivity analysis of a design variable associated with the magnetic bias region assumes that the magnetic bias field does not change. In some

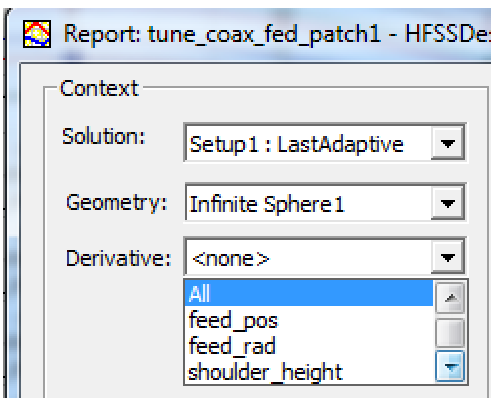
cases this assumption may not be appropriate and requires user discretion to ensure its applicability especially for non-uniform ferrite models.

Tuning far field quantities determines the derivatives of the response of electromagnetic devices with respect to variations in geometric or material properties. Designs can include port excitations or non-port excitations (that is, incident waves, linked field, voltage source and current source.

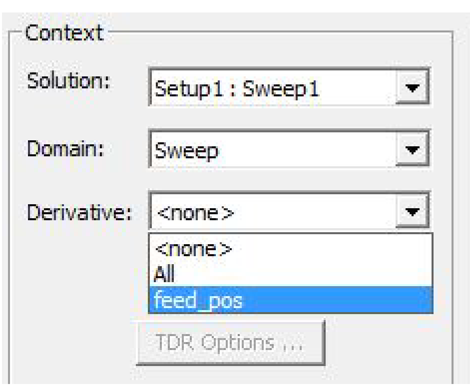
Note: Sensitivity analysis of a design variable associated with the magnetic bias region assumes that the magnetic bias field does not change. In some cases this assumption may not be appropriate and requires user discretion to ensure its applicability especially for non-uniform ferrite models.

Results obtained by the “Tune Reports” functionality is only an estimate, it is recommended that designs be re-analyzed with the variables values changed to include any tuning offsets to get an accurate result.

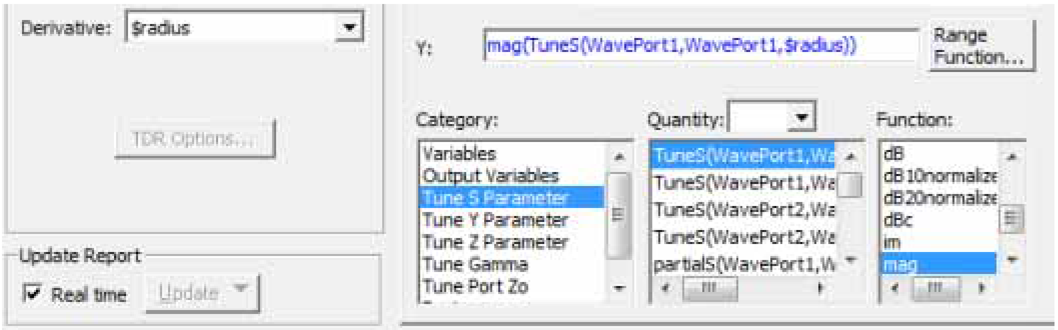
Frequencies, and local quantities cannot be tuned. If you have defined variables in the solution setup, the Context area of the Reporter displays the Derivative field, from which you can select the variable(s) for a tunable report.



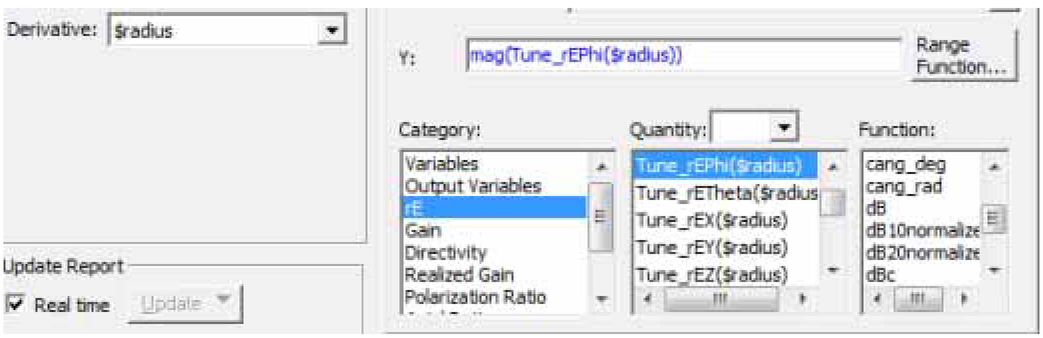
A common way to use this feature is to produce, for an output quantity of interest, two curves in one plot. One curve is produced with the **Report** dialog selection for the Context Derivative as <none>. The other has the All or specific variable selection.



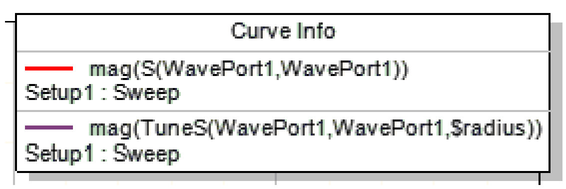
When you select All or a variable, notice that the Context field of the Report dialog shows the names of various parameters prefixed with Tune. Some of the Quantities associated with the category also show a Tune prefix. This labeling makes it easier to distinguish the Tuned traces from the reference traces.



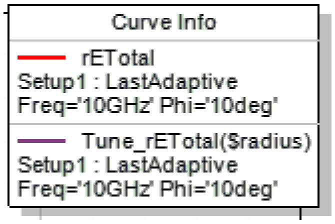
In the case of a Far Fields plot, the Tune prefix appears in only in the Quantity field.



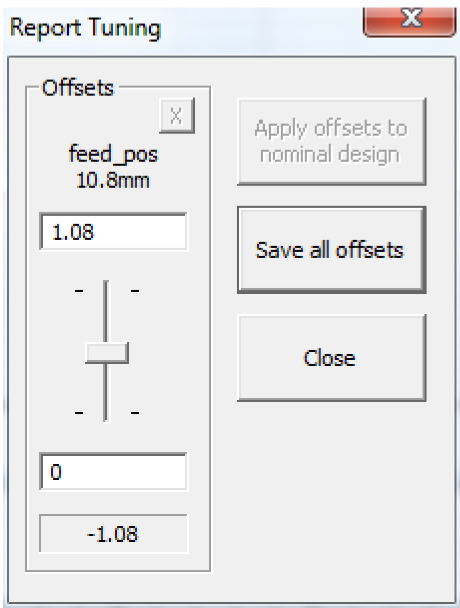
This gives you a plot with initially two identical curves, one on top of the other. The curve info table shows the trace names. This example shows the curve info for a plot with a normal S-parameter trace with a TuneS trace that can be tuned using the value of the $radius variable.



This example shows the curve info for a plot with a normal rETotal trace with a Tune_rETotal trace that can be tuned using the value of the $radius variable.



You can then right-click Results in the Project tree and select **Tune Reports**. The **Report Tuning** window appears. You can use the slider to tune the Tune curve interactively while the reference curve stays to provide a reference. This way you can see interactively how small changes in variables affect the result. You can then apply those offsets to the original variable values and re-solve the design.



The overall procedure for Report Tuning follows.

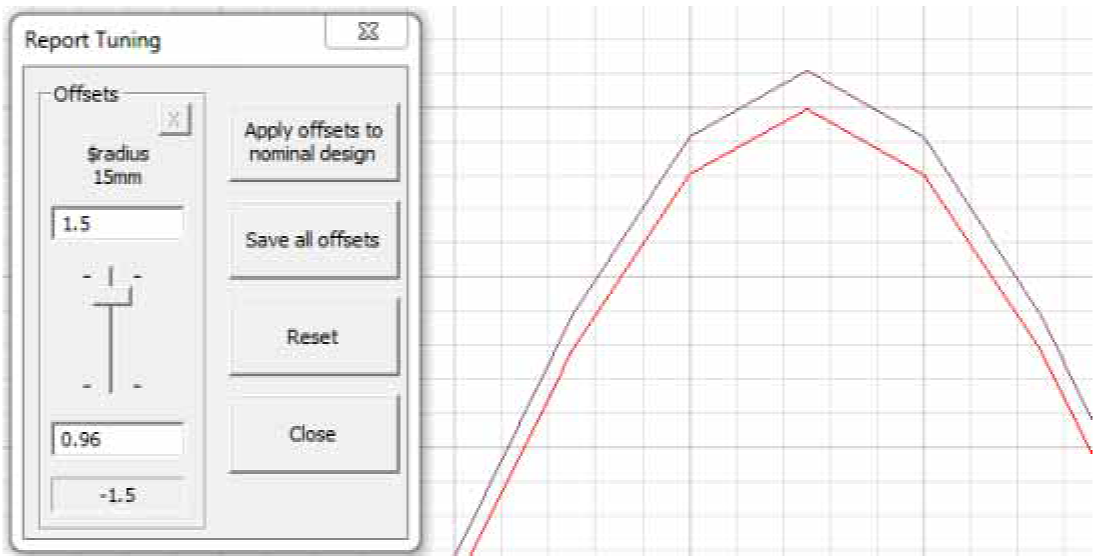
1. Generate a solution with one or more variables for which you select Use on the **Derivatives** tab of the solution setup.
2. Use the Reporter to create one curve with Derivative selection in the Context panel of the report dialog set to None.
3. Then create another curve, but in the Context pane. select for Derivatives, All or the variable of interest. Select to build your new trace from the Categories and Tune Quantities listed.

This gives you two identical curves, one top of the other. Notice that you are not limited in the number of traces that you define. You may choose to limit the number of traces for more ease in observing the results of tuning.

4. Click the **HFSS>Results>Tune Reports** or right-click **Results** in the Project tree and click **Tune Reports** from the short cut menu. The menu item is disabled if no variables have been selected in the Derivatives tab of the solution setup.

This displays a **Report Tuning** dialog which lists the variables available for tuning. The example above shows a **Report Tuning** dialog for a design with only one variable but it can show more.

5. You can use the slider to adjust the value of each available Tune variable. When you move a slider, the **Apply offsets to nominal design** button and the **Save all offsets** buttons are enabled. The Report shows the change to all Tune traces, relative to any reference traces you define.



The Tune dialog displays the change to the variable selected. For example, if the variable is $length with a value of 1mm, and the slider shows 0.1mm, then the effective value of $length for the purpose of derivatives is 1.1mm. If you exit the dialog by applying the offset or offsets (click **Apply offsets to nominal design**), $length is assigned a value of 1.1mm. You can then re-solve, and get results based on the derivative's prediction.

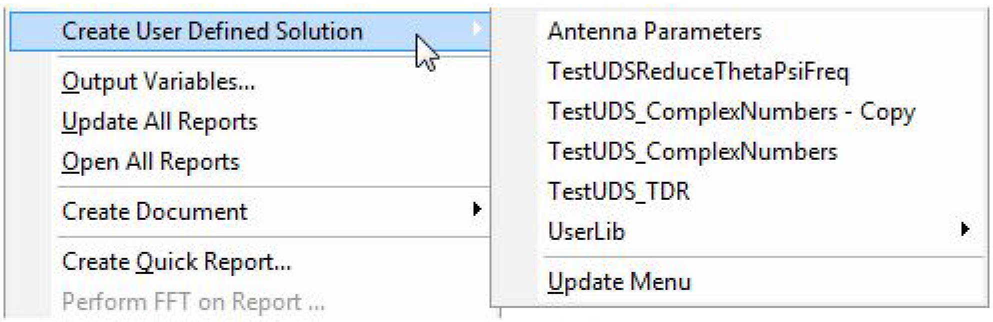
6. **Close** lets you close the dialog.

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.

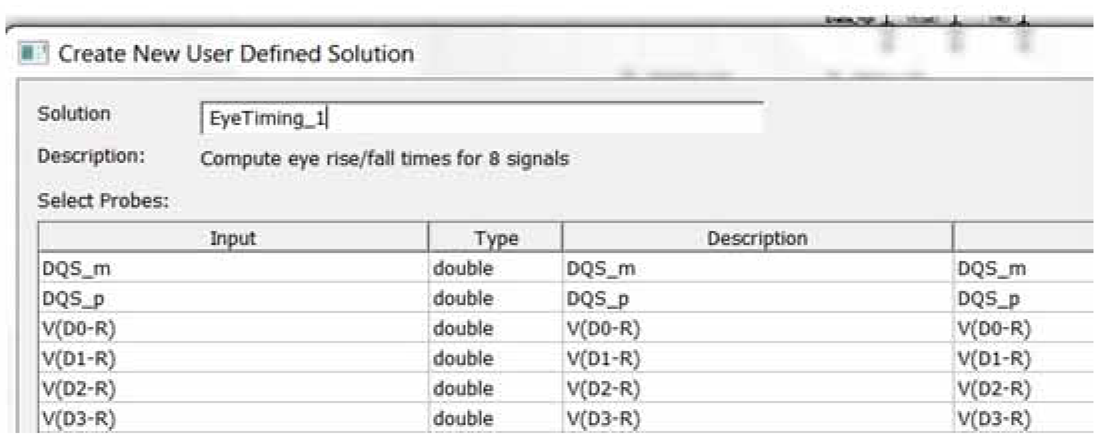
Use **Results**> **Create User Defined Solution**> **Update Menu** to refresh the menu to include the new UDO scripts that might have been copied to syslib, userlib, or Personallib; or exclude them if they have been deleted, after the launch of desktop. Once the user-defined-solution is created, the solution and the calculations defined by UDO become available in Reporter as any other quantities in a new "User Defined" report type.



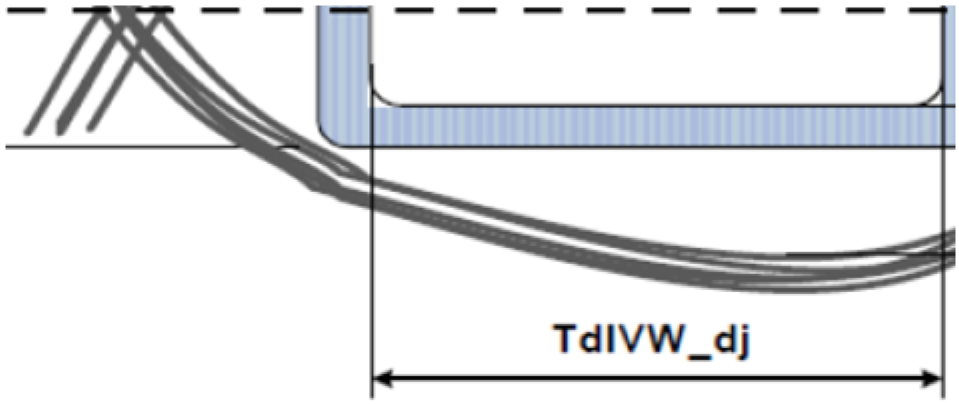
Named Probes and Properties in User Defined Outputs

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.



Computation of Traces Based UDO Calculations

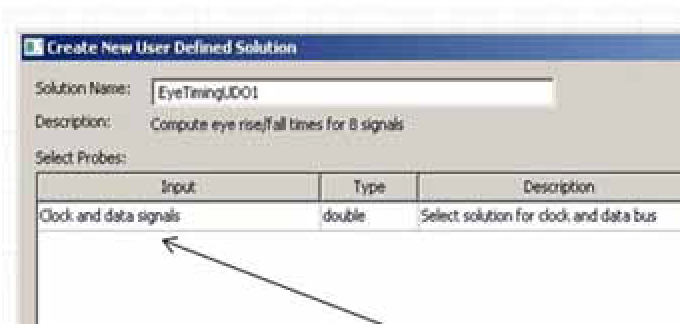
When traces that are based upon UDO outputs are computed, the data for probes is computed and passed to the UDO script for each design variation. Along with the probe data, the values of properties entered by user are also passed. The information about the UDO calculations that need to be computed is 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 enquired from UDOs at the time of trace computation and not at the time of creation of user defined solution.



This means that you select solution, context, and values of intrinsic variables just once, and the same information is used (in this case) for all clock and data signals. The expression for those signals comes from the UDO code.

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.		

<table>
<tr><td>Python Syntax</td><td>Validate (<<i>errorList</i>>, <<i>probeList</i>>, <<i>propertiesList</i>>, <<i>dynamicProbes</i>>)</td></tr>
<tr><td>Python Example</td><td><pre>
def Validate(self, errorStringList, probeList, propList,
dynamicProbes):
    if probeList == None or probeList.Count == 0:
        errorStringList.Add("Empty probe list")
        return False
    return True
</pre></td></tr>
</table>

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	see code below

```
# 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

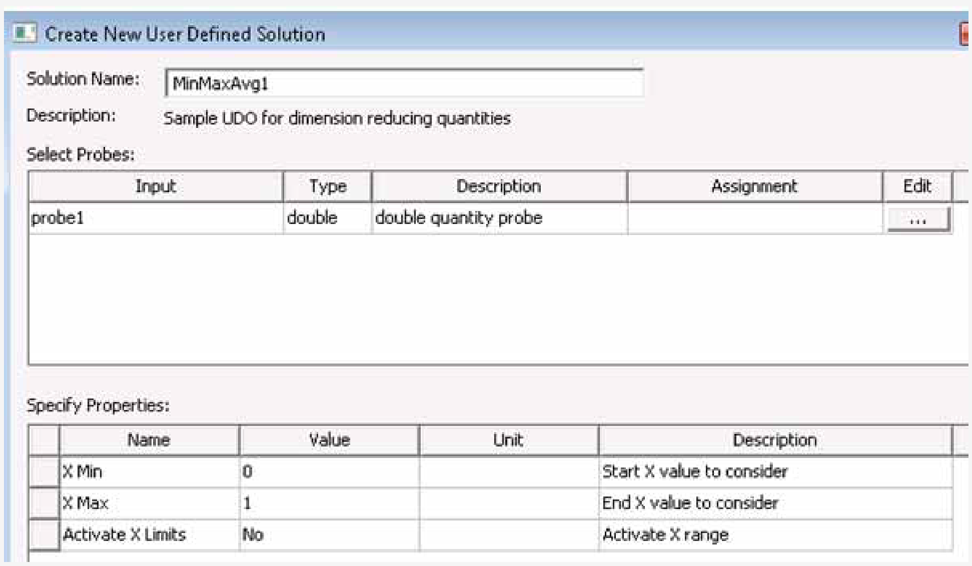
For each quantity that the UDO creates, it must also describe the quantity (unit and other details). This method is called for each quantity name (across all categories) as returned from an earlier call of the GetQuantityNames method.

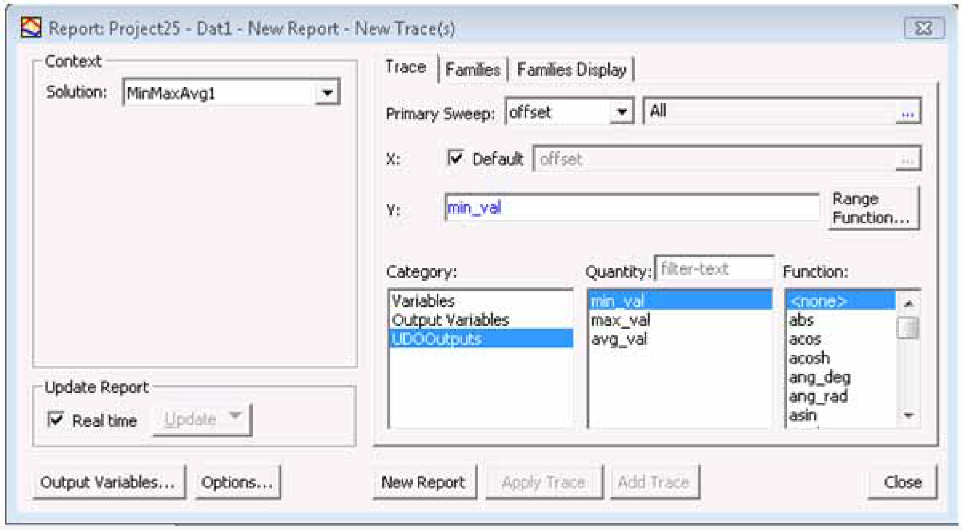
UI Access	NA
Parameters	Name <*quantityName*> — Type String — Description Name of quantity
Return Value	Object of type QuantityInfo.

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 / Type / Description <*dynamicProbes*> / List< UDSDynamicProbes > / [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	(see code below)

```
# 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 / Type / Description <*inData*> / IUDSInputData / Used to get the input probe data. <*outData*> / IUDSOutputData / Used to set the UDO solution quantity and sweep data. <*propList*> / IPropertyList / Used to get the user entered values for each of the properties defined during the GetInputUDSParams call. <*progressMonitor*> / IProgressMonitor / 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	(code below)

```
# 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 / Type / Description *<probeName>* / String / 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 / Type / Description *<probeName>* / String / 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. *<sweepName>* / String / 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 / Type / Description <*probeName*> / String / 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	(code below)

```
# 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	(see code below)

```
# 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	Type	Description
	<probeName>	String	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 / Type / Description *<qtyName>* / String / Name of the quantity. *<qtyData>* / List<Floats> / 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 / Type / Description *<qtyName >* / String / Name of the quantity. *<qtyData >* / List<floats> / 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 3 supported property types that could be used in the UDO script:

- INumberProperty to specify number properties (with unit support).
- IMenuProperty to allow the user to select from a list of options.
- ITextProperty to allow the user to enter text.

The IPropertyList type implements a collection for these properties:

- IPropertyList Abstract class
- IProperty Abstract class
- INumberProperty Abstract class
- ITextProperty Abstract class
- IMenuProperty Abstract class

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 Type Description <*TaskName*> String 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 Type Description <*TaskName*> String 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 Type Description <*Name*> String 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 Type Description <Percent> Integer 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.

<table>
<tr><td>UI Access</td><td colspan="3">NA</td></tr>
<tr><td rowspan="2">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PassFail></td><td>Boolean</td><td>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.</td></tr>
<tr><td>Return Value</td><td colspan="3">None</td></tr>
</table>

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 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

Additional .NET Assemblies

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.

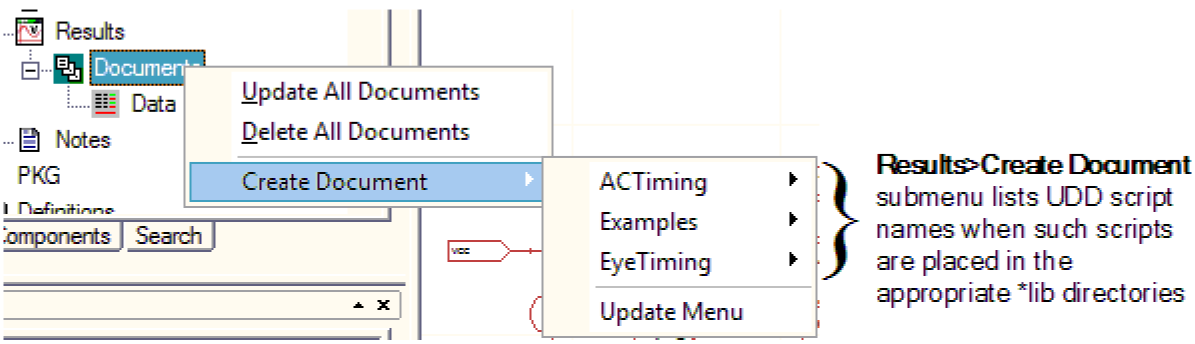
1. Build your .NET assembly for .NET 2.0 runtime.
2. Drop the built assembly in any **Lib** directory upstream of the UDO script location: that is, if you have your UDO script in *C:\Users\x\PersonalLib\UserDefinedOutputs\a\b\c\myudo.py* and have a .NET assembly called *com.Acme.UDOLib* You can keep 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.

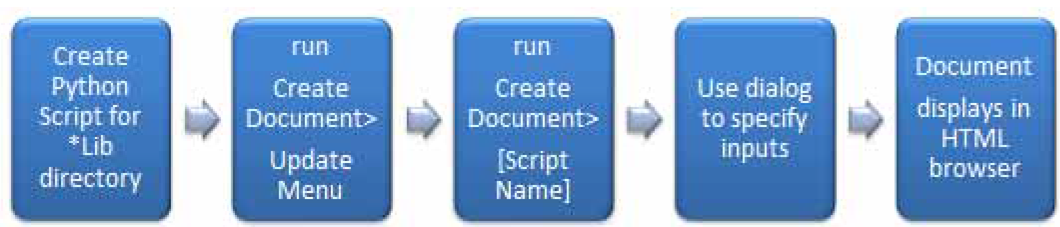
```
Import sys
sys.path.append("full path to your .NET assembly location")
```

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:



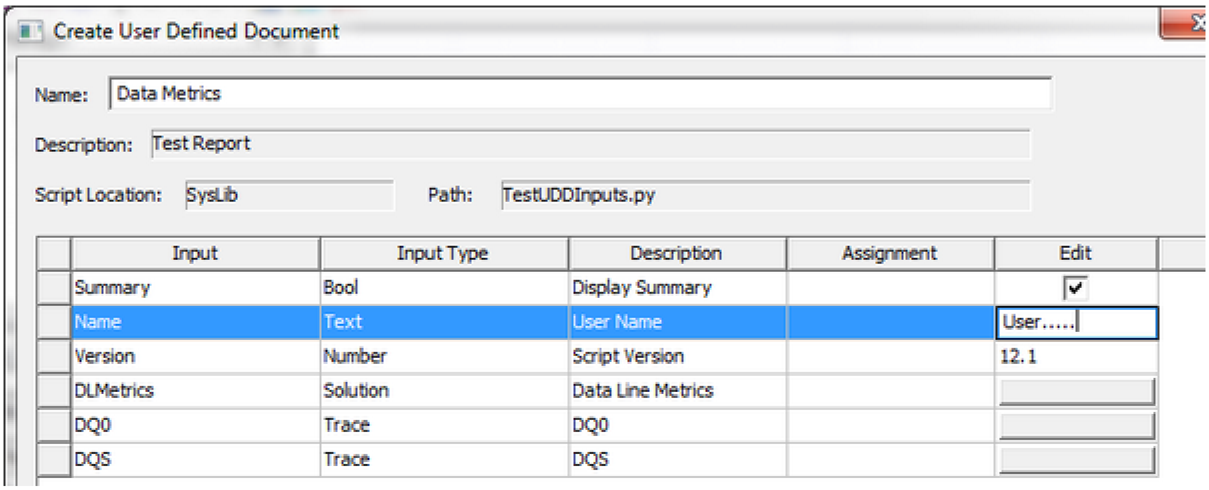
The UDD Python scripts must be placed in the **UserDefinedDocuments** directory under either of **syslib**, **userlib**, or **Personallib** with any subdirectory structure needed. The Lib directory can contain Python scripts that have common code that other scripts can use.

Use **Circuit**> **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 **Circuit**> **Results**> **Create Document** menu.

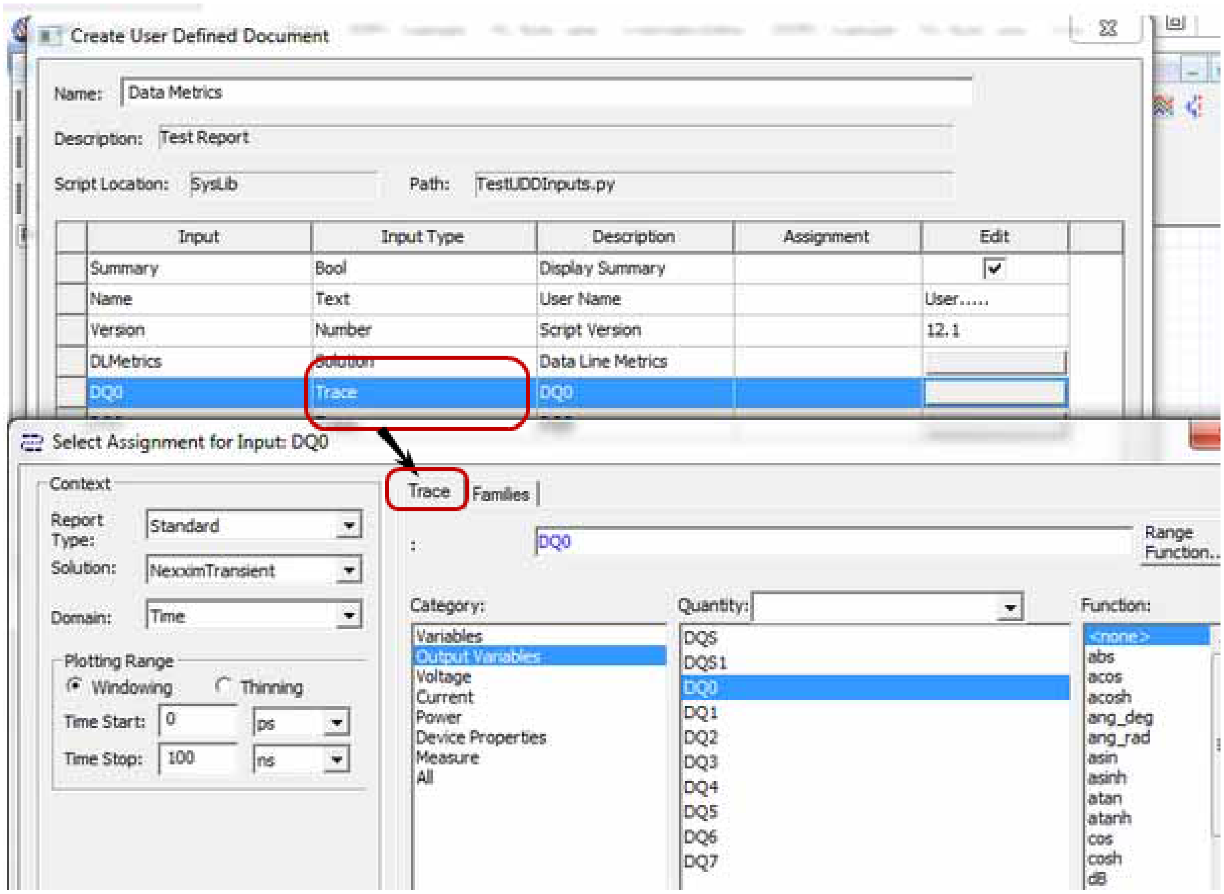
Create User Defined Document Dialog Inputs

User defined documents allow data from traces, solutions, and report types as inputs. A UDD can specify the named inputs, which you select or enter the values of in the *Create User Defined Document* dialog box that displays when you run **Circuit**> **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.

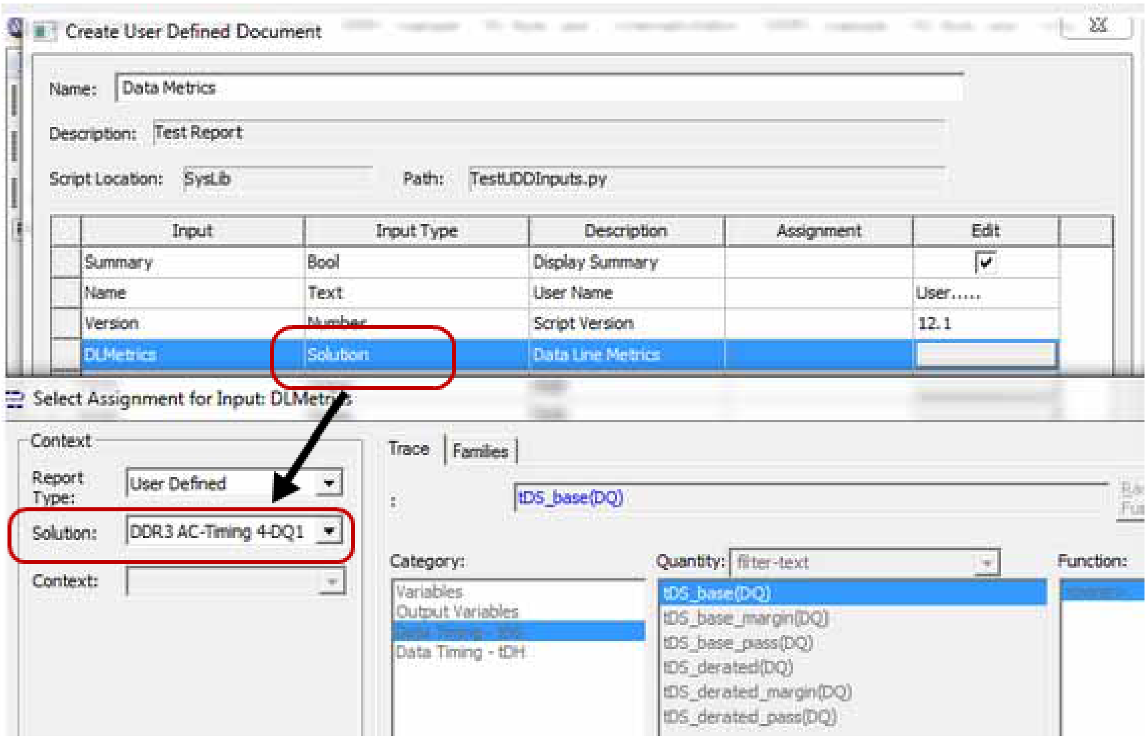


Input Type can also be Solution. You can select an entire solution when the document is created or modified. The solution data in its entirety is now available to the user and can be accessed from the Python script.

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:

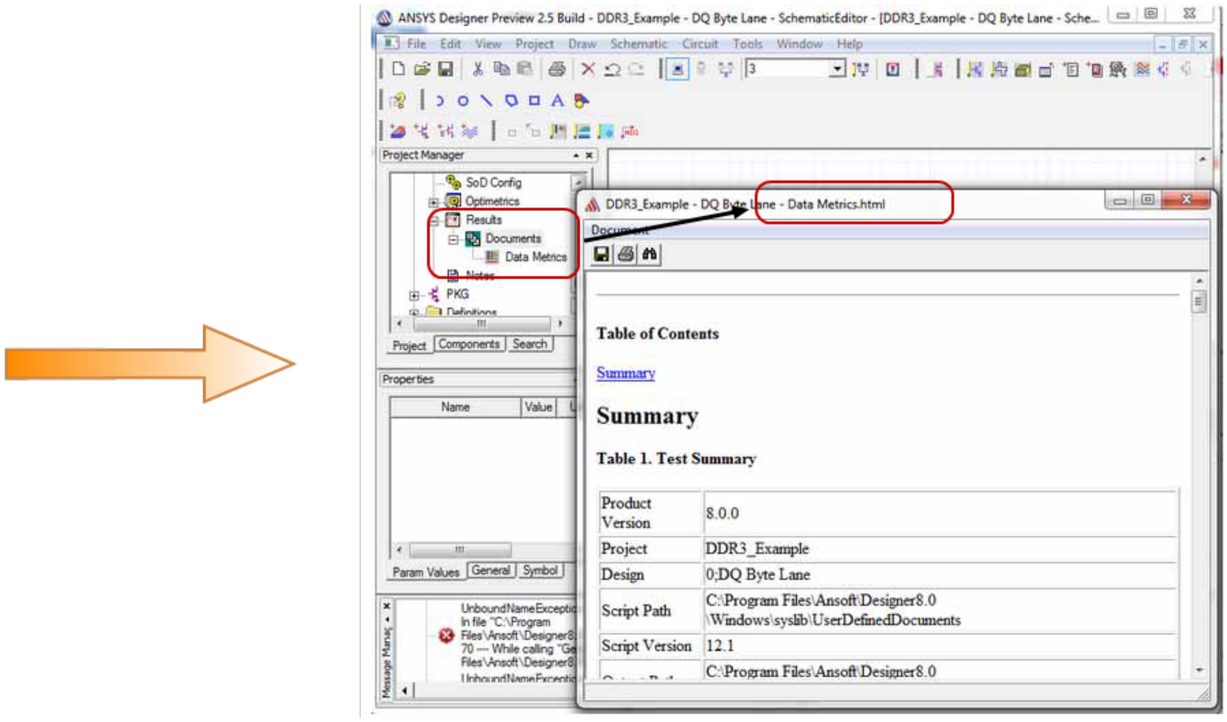
The Category/Quantity/Function portion of the dialog is disabled for user input.



UDD Document Creation and Display

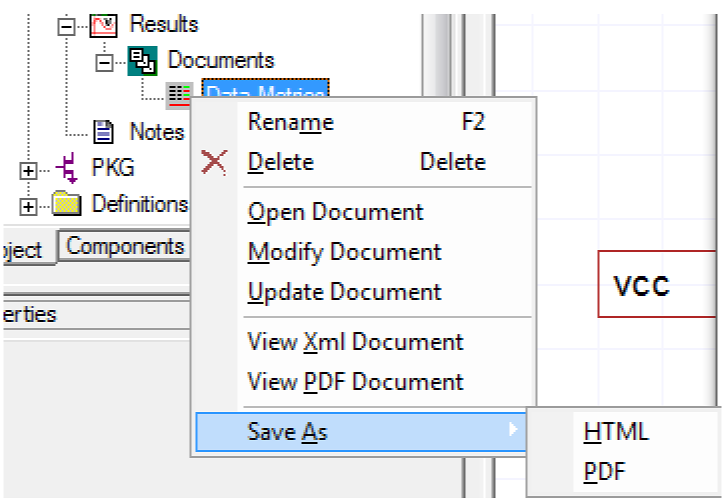
After all the input selections for a UDD are confirmed, based on the script, XML, HTML, and PDF documents are generated based on the inputs provided by the user. (The XML, HTML, and PDF generation is based on specific calls in the Python script, which are explain in a following section.) A web browser 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.



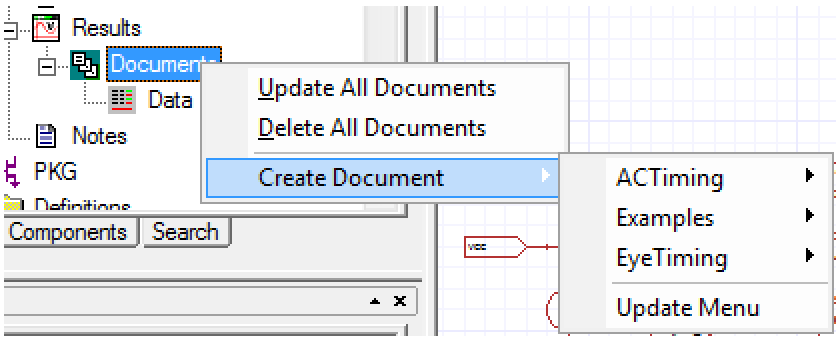
Managing Documents Listed in the Project Window Under Results

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

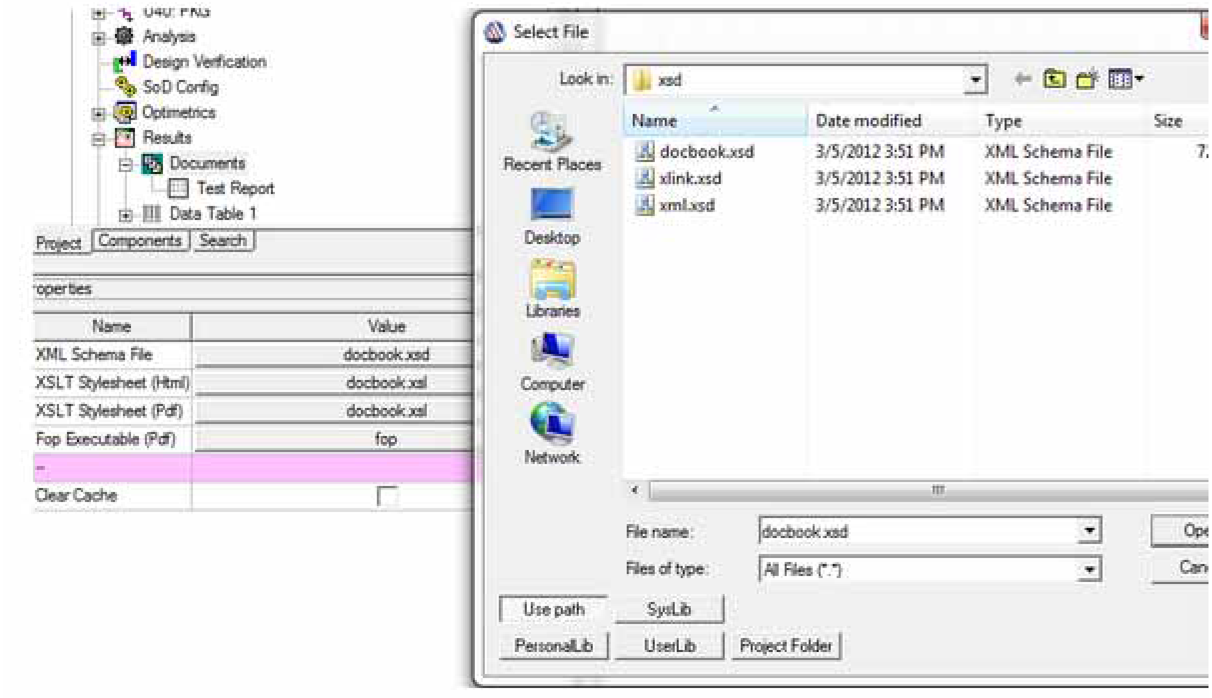
Right-click the documents folder has the menu options to **Update All Documents** or **Delete All Documents**. It also provides the option of creating a document from here.



Document Folder's Property Window

When the documents folder is selected, the *Property* window shows the following properties:

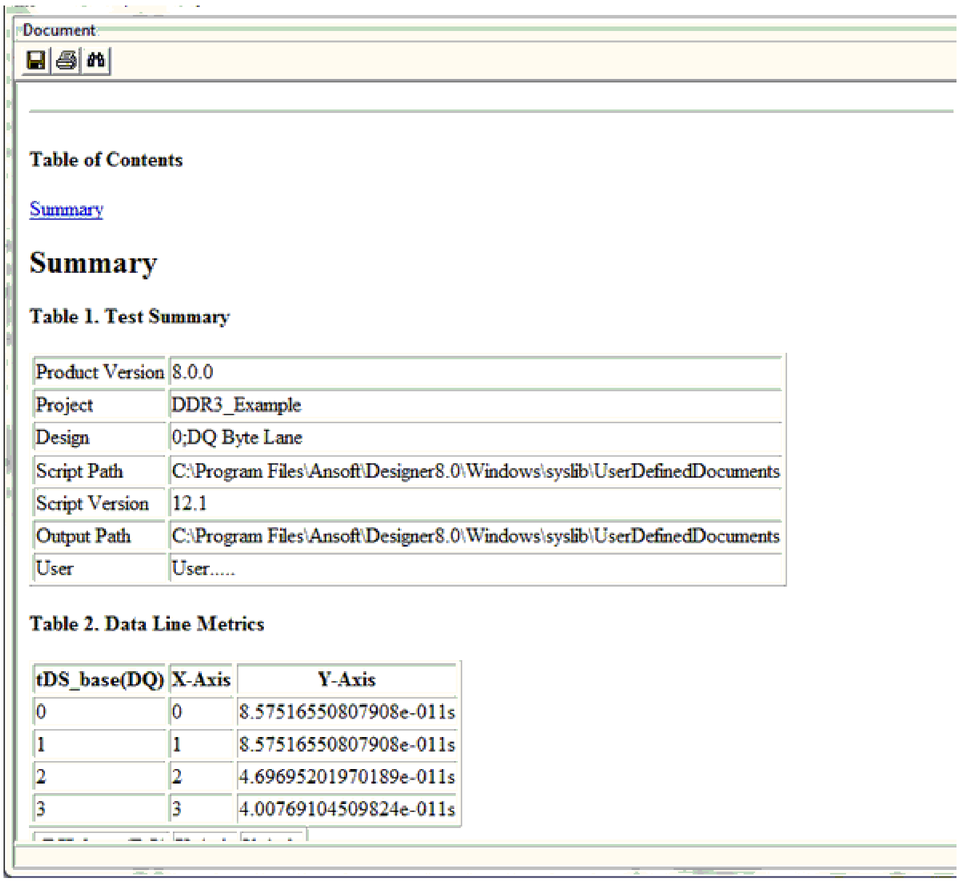
- **XML Schema File** - File path to the XML schema file.
- **XSLT StyleSheet (HTML)** - File path to the XSLT stylesheet file used for HTML generation.
- **XSLT StyleSheet (Fo)** - File path to the XSLT stylesheet file used for PDF generation.
- **Fop Executable (PDF)** - File path the Fop executable 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 / Type / Description *<TaskName>* / String / 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 / Type / Description *<TaskName>* / String / 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	Type	Description
	<PassFail>	Boolean	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.

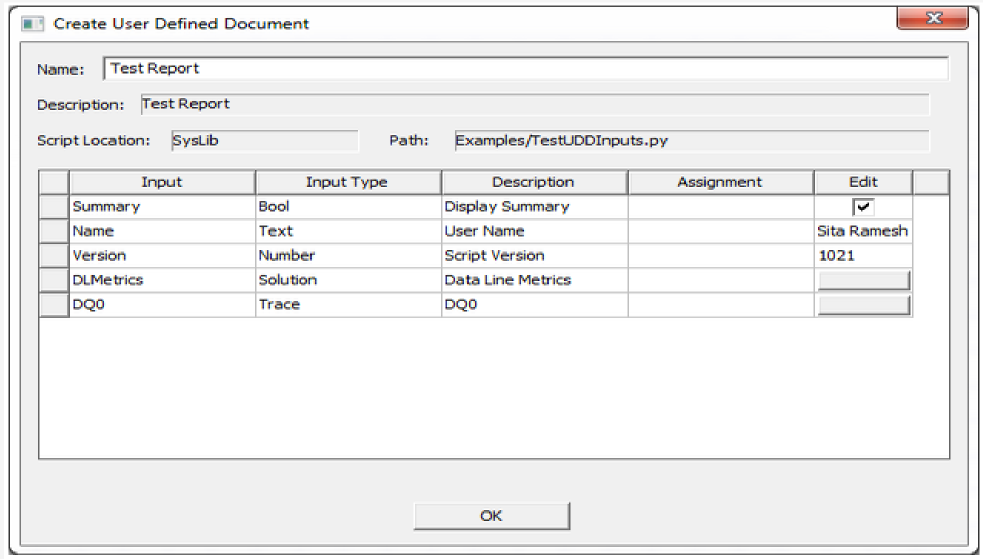
Python Syntax	ShowDefaultSetupDialog()
Python Example	`def ShowDefaultSetupDialog(self):` `    return True`

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 **Circuit**> **Results**> **Create Document**. The name and description of the UDD are also displayed in this dialog.



Parameters	Name	Type	Description
	<uddInputs>	List< UDDInputParams >	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 Syntax	Generate(*<uddInputs>*, *<generator>*, *<progressMonitor>*)

Python Example	<pre>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()</pre>

	`` 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	(see code below)

```
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	<pre>def HandleUDDEvents(self, uddLinks): if uddLinks[0] == "Open Report": # Get Design Name oDesign = self.GetUDDDesignContext() if oDesign != None: oDesign.OpenReport(uddLinks[1]) return True</pre>

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	<pre>def GetUDDSchema(self): return "C:\\Program Files\\AnsysEM\\v242\\Win64\\common\\docbook\\schema\\xsd\\docbook.xsd" schemaPath = pluginExt.GetUDDSchema()</pre>

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\\AnsysEM\\v242\\Win64\\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\\AnsysEM\\v242\\Win64\\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\\AnsysEM\\v242\\Win64\\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, PeronalLib, 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.

Post-processing and Generating Reports for 2D and Circuit

After HFSS 3D Layout, Planar EM, Nexxim, or Nexsys has generated a solution, all the results are available for analysis. The graphical plots and other reports described in this topic help you to visualize the relationship between design parameters and analysis results.

Creating Reports for 2D and Circuit Projects

A report is generated using the **New Report** window. The appearance of the **New Report** window differs depending on the type of design:

- EM Design (HFSS_3DLayout, Planar EM)
- Circuit Design (Nexxim, Nexsys)

the type of report:

- Standard Report
- Constellation Report
- Eye Diagram Report
- Statistical Eye Report

and the type of display:

- Polar Plot
- Data Table
- Smith Chart
- 3D Polar Plot
- Rectangular Plot
- Smith Contour Plot
- 3D Rectangular Plot
- Rectangular Stacked Plot
- Rectangular Contour Plot

> **Note:** Depending on the nature of your installation and the projects loaded into Electronics Desktop, at times only a subset of the types listed above may appear in the display-types submenu.

When you have selected the display type (e.g., by clicking **Product** > **Results** > **Create Standard Report** > **Rectangular Plot**) a **New Report** window opens that allows you to create the report. The following topics describe how to fill out the **New Report** window for various types of reports and displays.

2D and Circuit Report Types

When you click **Product > Results (** or right-click **Results** in the **Project Tree**), one or more of the following report types and **display types** are available, depending upon the nature of your installation and the projects loaded into Electronics Desktop.

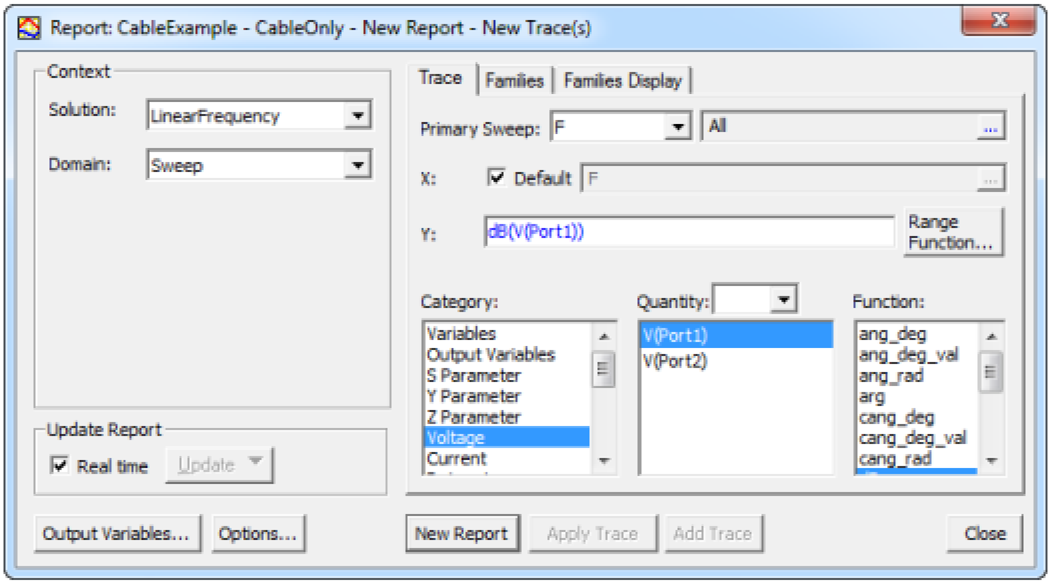
Report Type	Display Type
Create Standard Report	A “Standard” report can be depicted in any of the following display types: • Rectangular Plot • Data Table

	• Polar Plot • Smith Chart • Rectangular Stacked Plot • 3D Rectangular Plot • 3D Polar 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 • Stacked Eye Diagram 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.

Note: Depending on the nature of your installation and the projects loaded into Electronics Desktop, at times only a subset of the types listed above may appear in the display-types submenu.

2D and Circuit New Report Window

When you click **Product > Results (** or right-click **Results** in the **Project Tree**) and select a **report type** and a **display type** , a **New Report** window similar to the following opens:



> **Note:** Depending on the nature of your installation and the projects loaded into Electronics Desktop, at times only a subset of the following report options may appear in the **New Report** window.

In the **New Report** window, make selections in the following group boxes to create a report:

Context Group Box

Context group box

- **Solution** – lists the available sweeps.
- **Domain** – lists the available domains.
- **Show** — lists the quantity types to display. (Differential Pairs only)

Harmonics Group Box

- For Harmonic Balance simulation setups (HB1Tone, HBNtone, Oscillator, OscillatorNTone, PXF, TVNoise) this box displays the first five harmonic quantities of a Sweep or Network Function domain.
- **Edit** – You can select/filter and change the displayed harmonic quantities by clicking **Edit** to open the output quantity window; any changes that are made are reflected in the original setup window.

Plotting Range Group Box

- Long-time domain simulations may contain extremely large amounts of data.
- **Windowing** – shows data within a range of times:

  - Time Start — allows you to set the start time and units.
  - Time Stop — allows you to set the stop time and units.
  - Note that the maximum number of points displayed is 20,000,000. This may limit the range that can be shown. For example a Transient Analysis with 10ms Stop Time and 1ns Step Time creates 100,000,000 points; only a subset of these (2ms worth) can be displayed. To see the entire range, increase the Step Time.
- **Thinning** — removes unnecessary detail:
  - dy/dx tolerance — a delta voltage that should be set to be smaller than the overall voltage swing (min, max) and greater than the voltage fluctuations to be ignored. This number is used to calculate slope variations and eliminates points that have equal slopes (in these variations). For square or triangular waves, this can limit the number of points to two per group box of constant slope.
  - Number of Points gives the maximum points displayed and is applied after the dy/dx tolerance. It must be less than the maximum number of points.
- **Time Start** – allows you to set the start time and units.
- **Time Stop** — allows you to set the stop time and units.

Constellation Trace Tab

- **Primary Sweep** — drop-down menu allows you to select a primary sweep, or browse for the selection by clicking **[...]**.
- **Constellation** — displays the applicable function. Contains a drop-down menu for selecting the value(s) and a **[...]** browse button for selecting from a list of sweeps (if available). Clicking **Range Function** opens the **Set Range Function** window where a different function can be chosen and applied apply.
- **Category** window- these depend on the Solution type and the design. This lets you specify the category of information for the report.
- **Quantity** – allows you to specify a quantity for the **Constellation** report.
- **Function** – allows you to specify a function for the **Constellation** report.

Eye Diagram Trace Tab

- **Primary Sweep** — drop-down menu allows you to select a primary sweep, or browse for the selection by clicking **[...]**
- **Eye Diagram** — displays the applicable function. Contains a drop-down menu for selecting the value(s) and a **[...]** browse button for selecting from a list of sweeps (if available). Clicking **Range Function** opens the **Set Range Function** window and allows you to choose a different function to apply.
- **Category** window- these depend on the Solution type and the design. This lets you specify the category of information for the report.
- **Quantity** – allows you to specify a quantity for the **EyeDiagram** report.
- **Function** – allows you to specify a function for the **EyeDiagram** report.

Constellation Parameters Group Box

- Radio buttons allow you to choose between **Vector** and **Constellation**
- When **Constellation** is selected, you may specify a **Sample Period** and **Offset**.

Eye Parameters Group Box

- Allows you to specify a **Unit Interval** and **Offset**.

Update Report Group Box

- **Real Time** checked — enable real time updates for all reports.
- **Real Time** unchecked — enables the adjacent dropdown menu which allows you to choose **Update This Report** or **Update All Reports**. Both selections update traces to the latest data available.

Trace Tab

- **Primary Sweep** — drop-down menu allows you to select a primary sweep, or browse for the selection by clicking the **[...]**
- **X:**
  - Contains a drop-down menu for selecting the value(s) and a **[...]** button for selecting from a list of sweeps (if available).
- **Y**:
  - **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
  - **Category** window— these depend on the **Solution** type and the design. This lets you specify the category of information for the Y component.
  - **Quantity** – allows you to specify a quantity for the Y component.
  - **Function** – allows you to specify a function for the Y component.

Families Tab

- Lists the number of families available.
- Each member of a family defines one point on a curve.
- Radio buttons allow you to choose between **Sweeps** and **Available variations**. Allows you to edit sweeps; **Edit** lets you select variable **Values**.
- **Nominals** field (unavailable if none exist in the design) allows you to choose one of the following:
  - **Set All Variables to Nominal**
  - **Set All Unswept Variables to Nominal**
  - **Choose Nominals**

Families Display Tab

- Allows you to choose on the following types of families display: **AllFamilies**, **Statistics**, or **Histogram**.

Dialog Box Command Buttons

- **Output Variables** – opens the **Output Variables** window.
- **Options** – opens the Report Setup Options window. 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 contains a field for setting the maximum number of significant digits to display for numerical quantities.
- **New Report** – adds a report to the **Project** tree under the **Results** icon. The new report is displayed in the **Project** window.
- **Apply Trace** — applies the currently configured trace.
- **Add Trace** – this is enabled when you have created or selected a report. Add further traces. The new trace displays in the **Project** window under the report.
- **Close** – closes the **New Report** window.

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, S_{11} say, 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 so by any expression. If the complex S_{11} lies on the positive Y axis say, ang_deg_val(S_{11}) is 90 and "1 + ang_deg_val(S_{11})" is 91.

2D and Circuit Display Types

The information in a report can be displayed in several formats. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Results**. Navigate to the appropriate report type, then choose from one of the following display types:

Rectangular	A 2D rectangular (x-y) graph.

Plot	
Polar Plot	A 2D circular chart divided by spherical coordinates.
Data Table	A grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or another quantity.
Smith Chart	A 2D polar chart of S-parameters upon which a normalized impedance grid has been superimposed.
Rectangular Stacked Plot	A 2D rectangular (x-y) stacked graph.
3D Rectangular Plot	A 3D rectangular (x-y-z) graph.
3D Polar Plot	A 3D circular plot divided by spherical coordinates.
Rectangular Contour Plot	A 2D rectangular contour chart of the eye simulation results.
Smith Contour Plot	A 2D Smith contour chart of the load-pull results.
Eye Diagram Plot	A 2D rectangular-eye diagram with transient data available for post-processing
Stacked Eye Diagram Plot	One or more eye diagram plots in a horizontal stack.
PAM-4 Eye Plot	An eye diagram created from PAM-4 voltage traces with 4 symbol levels that create three distinct sub-eye openings between the three sequential pairs of symbol levels.
Statistical Eye Plot	A 2D rectangular-eye diagram with no retained data for post-processing

> **Note:** Depending on the nature of your installation and the projects loaded into Electronics Desktop, at times only a subset of the types listed above may appear in the display-types submenu.

2D and Circuit Rectangular Plot

A rectangular plot is a 2D, x-y graph of results.

1. Click **Product** > **Results** (or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Results**) and point to a appropriate **report type** . Then select **Rectangular Plot** to open the **New Report** window.

2. In the **Context** group box make selections on 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** and **display type** selected.
   c. Show field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.
3. Under the **Trace** tab **Primary Sweep** group box, make selections on the following categories:
   a. In the **Category** group box, click the type of information to plot.
   b. In the **Quantity** group box, 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. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
4. In the **X (Primary Sweep)** group box, make selections for the following:
   a. Select the Primary value(s) on the dropdown menu.
      To select an X component that is different on the Primary Sweep, uncheck the Default field to enable the X field and **[...]** button. Click **[...]** to open the **Select X Component** window. 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 window to assign the X component.
   b. If sweeps are available, select **[...]** to open a window that lets you select particular sweep or sweeps, or all sweeps.
   c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (e.g., 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 **[...]** button.
5. From the **Families** tab, confirm or modify the sweep variables that is 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** window.

   The function of the selected quantity is plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the **Project Manager** window 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**.

2D and Circuit Polar Plot

In Ansys Electronics Desktop, a polar plot is a 2D circular chart divided by the spherical coordinates R and theta, where R is the radius, or distance on the origin, and theta is the angle on the x-axis. Following is the general procedure for drawing a polar graph of results:

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** . Then select **Polar Plot** top open the **New Report** window.
2. In the **Context** group box make selections on 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** and **display type** selected.
   c. Show field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.
3. In the **Trace** tab **Polar Component** group box, specify the information to plot:
   a. From the **Category** dropdown menu, select the type of information to plot.
   b. From 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. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. 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** window.

The function of the selected quantity is plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the **Project Manager** window 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**.

2D and Circuit Data Table

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 **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Results**. Navigate to the appropriate **report type** . Then select **Data Table** to open the **New Report** window.
2. In the **Context** group box make selections on the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list.
   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** and **display type** selected.
   c. Show field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.
3. Under the **Trace** tab **Primary Sweep** group box, make selections on the following categories:
   a. From the **Category** drop-down menu, select the type of information to plot.
   b. From the **Quantity** group box, click the values to plot. Use **Ctrl**+click to make multiple selections.
   c. In the **Function** group box, click the mathematical function to apply to the quantity for the plot.
   d. The **Value** field displays the currently specified Quantity and Function. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
4. In the **X (Primary Sweep)** group box, make selections for the following:
   a. Select the Primary value(s) on the dropdown menu.
   To select an X component that is different on the Primary Sweep, uncheck the Default field to enable the X field and **[...]** button. Click **[...]** to open the **Select X Component** window. 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 window to assign the X component.
   b. If sweeps are available, select **[...]** to open a window that lets you select particular sweep or sweeps, or all sweeps.

c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (e.g., 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 **[...]** button.

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** window.

   The Y quantity is listed at each variable value or additional quantity value you specified. The data table is listed under **Results** in the **Project Manager** window. The plot is listed under **Results** in the **Project Manager** window 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**.

2D and Circuit Smith Chart

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. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** and select **Smith Chart** to open the **New Report** window.
2. In the **Trace** tab **Polar Component** group box, specify the information to plot:
   a. From the **Category** dropdown menu, select the type of information to plot.
   b. From the **Quantity** list, click the values to plot. Use **Ctrl**+click to make multiple selections.
   c. In the **Function** group box, click the mathematical function to apply to the quantity for the plot.
   d. The **Value** field displays the currently specified Quantity and Function. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. 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** window.

The function of the selected quantity is plotted against the values you specified on a polar plot. 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 Manager** window 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**.

2D and Circuit Rectangular Stacked Plot

A rectangular stacked plot is a 2D stacked x-y graph of results.

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** . Then select **Rectangular Stacked Plot** to open the **New Report** window.
2. In the **Context** group box make selections on the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list.
   b. Show field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.
3. Under the **Trace** tab **Z Component** group box, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
4. From 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 on the drop down list.
   - If sweeps are available, select the **[...]** button to display a window that lets you select particular values. The quantity is plotted against the primary sweep variable listed.
5. In the **X (Primary Sweep)** group box, make selections for the following:
   a. Select the Primary value(s) on the dropdown menu.
   To select an X component that is different on the Primary Sweep, uncheck the Default field to enable the X field and **[...]** button. Click **[...]** to open the **Select X Component** window. 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 window to assign the X component.

b. If sweeps are available, select **[...]** to display a window that lets you select particular sweep or sweeps, or all sweeps.

c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (e.g., 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 **[...]** button.

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** window.

The function of the selected quantity or quantities is plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the **Project Manager** window. 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**.

2D and Circuit 3D Rectangular Plot

A rectangular plot is a 3D, x-y-z graph of results.

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** . Then select **3D Rectangular Plot** to open the **New Report** window.
2. In the **Context** group box make selections on the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list.
   b. Show field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.
3. Under the **Trace** tab **Z** Component group box, specify the information to plot along the z-axis:
   a. In the **Category** group box, click the type of information to plot.
   b. In the **Quantity** group box, 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. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
4. From the **Trace** tab **Y** (Secondary sweep) lines, specify the information to plot along the y-axis in one of the following ways:

- From the drop-down menu, select the sweep variable to use.
- If sweeps are available, select the **[...]** button to display a window that lets you select particular values. The quantity is plotted against the primary sweep variable listed.

5. In the **X (Primary Sweep)** group box, make selections for the following:
   a. Select the Primary value(s) on the drop-down menu.
   To select an X component that is different on the Primary Sweep, uncheck the Default field to enable the X field and **[...]**. Click **[...]** to open the **Select X Component** window. This lets you specify the X component as you do the Y; that is, in terms of Categories that define the selectable Quantities, and Functions to apply. After making selections, **OK** the window to assign the X component.
   b. If sweeps are available, select **[...]** to display a window that lets you select particular sweep or sweeps, or all sweeps.
   c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (e.g., 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 **[...]** button.
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** window.

   The function of the selected quantity or quantities is plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the **Project Manager** window. 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**.

2D and Circuit 3D Polar Plot

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 on the origin, theta is the angle on the x-axis, and phi is the angle on the origin in the z direction. Following is the general procedure for drawing a 3D polar plot of results:

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** . Then select **3D Polar Plot** to open the **Report** window.
2. In the **Context** group box make selections on the following field or fields, depending on the design and solution type.

a. Solution field with a drop-down menu.
b. Show field with a drop-down menu. Whether this field appears depends on if differential pairs exist and are enabled.

3. In the **Trace** tab **Mag** group box, 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.
   b. From the **Quantity** group box, 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. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
4. From the **Trace** tab **Theta (Secondary Sweep)** line, make a selection on the sweep variable drop-down menu and specify all values or select values to plot along the theta-axis:
5. From 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 creates a new report in **Project Tree**, displays the report with the defined trace, and enables **Add Trace** on the **Report** window.

   The function of the selected quantity or quantities is plotted against the R-, phi-, and theta-axes on a 3D polar graph. The plot is listed under **Results** in the **Project Manager** window. 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**.

2D and Circuit Rectangular Contour Plot

This type of plot is a 2D rectangular contour chart of the eye simulation results. Its traces can have two characteristics for bit error rates: MinEyeHeight and MinEyeWidth.

To display a rectangular contour plot after a successful analysis:

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** . Then select **Rectangular Contour Plot** to open the **New Report** window.
2. Select the output quantities of interest and click **Add Trace**.
3. Click **Done**. A Report window opens to display a graph of the analysis results.

2D and Circuit Smith Contour Plot

To display a contour chart of the load-pull results after a successful analysis:

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** and select **Smith Contour Plot** to open the **New Report** window.
2. Select the output quantities of interest and click **Add Trace**.
3. Click **Done**. A Report window opens to display a graph of the analysis results.

Circuit Eye Diagram Plot

An eye diagram illustrates an overlay of rising and falling bit waveforms with a given unit interval or bit time.

1. Click **Circuit > Results (** or right-click **Results** in the **Project Tree**) , select **Create Eye Diagram Report** > **Rectangular Plot** to open the **New Report** window.
2. In the **Context** group box, make selections on the following 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** and **display type** selected.
   c. **Show** field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.
3. In the **Eye Parameters** group box, set the **Unit Interval** and **Offset** to match the specification for the source or driver of the signal.
4. Under the **Trace** tab **Primary Sweep** group box, make selections on the following categories:
   a. In the **Category** list, click the type of information to plot. For an Eye Diagram, **Voltage** is the usual Category.
   b. In the **Quantity** list, click the value to plot. The unit interval used to generate the quantity must be the same as the one specified for the report.
   c. In the **Function** list, click the mathematical function of the quantity to plot, typically <none>.
   d. The **Eye Diagram** field displays the currently specified Quantity and Function. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies to the currently specified Quantity and Function.
5. In the **X (Primary Sweep)** group box, make selections for the following:

a. Select the Primary value(s) on the dropdown menu.
   To select an X component other than the Primary Sweep, click the **[...]** button to open the **Select X Component** window. 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 window to assign the X component.
b. If sweeps are available, select **[...]** to display a window that lets you select particular sweep or sweeps, or all sweeps.
c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (e.g., 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 **[...]** button.

6. From the **Families** tab, confirm or modify the sweep variables that is 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** window.

The function of the selected quantity is plotted against the swept variable values or quantities you specified on an eye diagram. The plot is listed under **Results** in the **Project Manager** window and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**. The added trace is overlaid on the existing eye traces.

2D and Circuit Stacked Eye Diagram Plot

A stacked eye diagram creates a multiple display of two or more eye diagrams with the same unit interval bit time. A stacked eye diagram is a convenient way to compare a set of eye diagrams with the same unit interval.

1. Click **Circuit > Results (** or right-click **Results** in the **Project Tree**) , select **Create Eye Diagram Report**, then select **Stacked Eye Diagram Plot** to open the **New Report** window.
2. In the **Context** group box, make selections on 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** and **display type** selected.

c. **Show** field with a drop down selection list. Whether this field appears depends on if differential pairs exist and are enabled.

3. In the **Eye Parameters** group box, set the **Unit Interval** and **Offset** to match the specification for the source or driver of the signal. **The unit interval is required.**
4. Under the **Trace** tab **Primary Sweep** group box, make selections on the following categories:
   a. In the **Category** list, click the type of information to plot. For an Eye Diagram, **Voltage** is the usual Category.
   b. In the **Quantity** list, click the value to plot. The unit interval used to generate the quantity must be the same as the one specified for the report.
   c. In the **Function** list, click the mathematical function of the quantity to plot, typically <none>.
   d. The **Eye Diagram** field displays the currently specified Quantity and Function. Edit this field directly. Note that invalid expressions are displayed in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies to the currently specified Quantity and Function.
5. In the **X (Primary Sweep)** group box, make selections for the following:
   a. Select the Primary value(s) on the dropdown menu.
   To select an X component that is different on the Primary Sweep, click the **[...]** button to open the **Select X Component** window. 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 window to assign the X component.
   b. If sweeps are available, select **[...]** to display a window that lets you select particular sweep or sweeps, or all sweeps.
   c. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (e.g., 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 **[...]** button.
6. From the **Families** tab, confirm or modify the sweep variables that is 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** window.

The function of the selected quantity is plotted against the swept variable values or quantities you specified on an eye diagram. The plot is listed under **Results** in the **Project Manager** window and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**. The added trace becomes the lowest of the stacked eye diagrams.

> **Note:** The stacked eye diagram is very useful for comparing eye diagrams (e.g., for AMI analysis, stack the waveforms before and after the channel).
>
> - Sweep a variable and stack the resulting set of eye diagrams.
> - When Eye Measurements can be displayed, the Stacked Eye Diagram format displays the measurements in a convenient side column that does not block the chart.

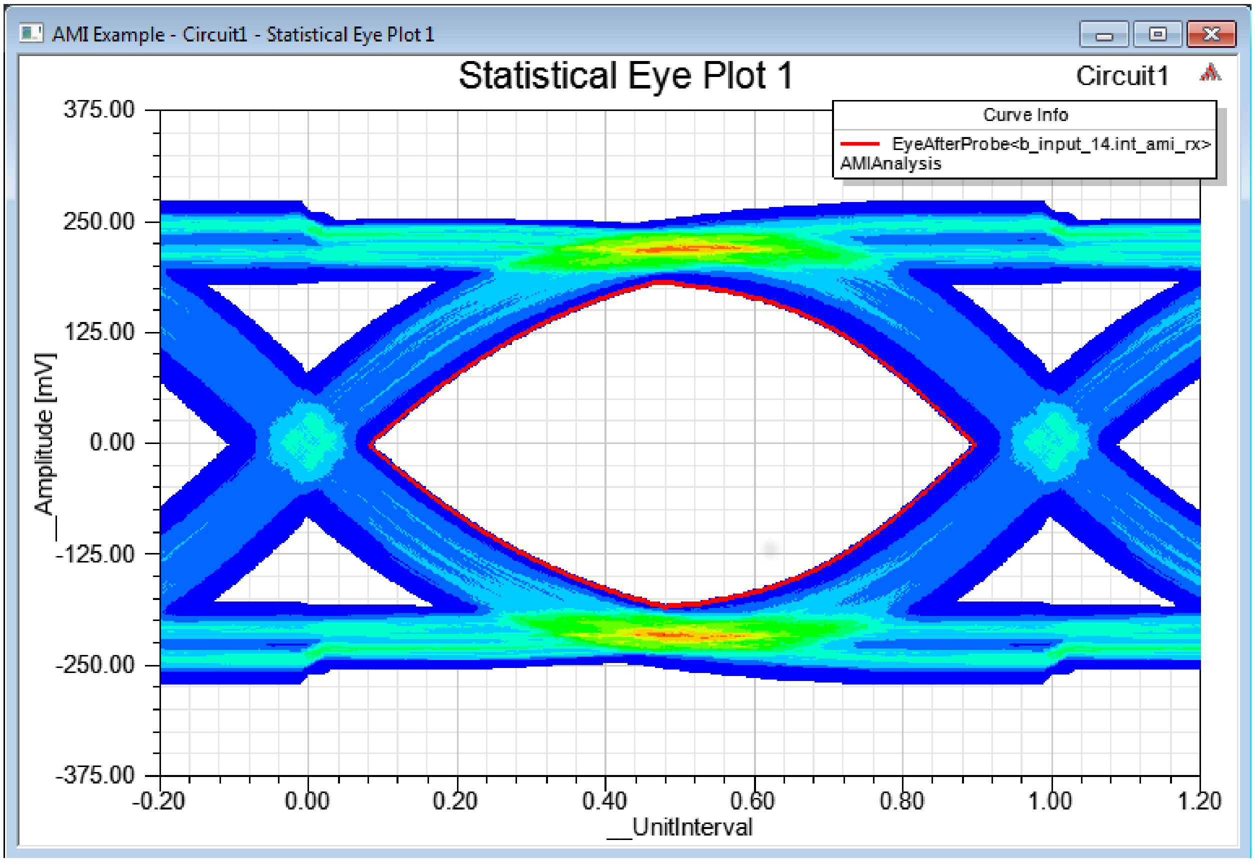
Circuit Statistical Eye Plot

> **Note:** Statistical Eye Plots are intensity plots of Probability Mass Function (PMF) data. For bit-by-bit analyses (transient, QuickEye, AMI) the PMF data is computed on the waveform generated for the given bit pattern, and the inner eye contour of the plot corresponds to the worst-case eye for that bit pattern. *Inner eye contours on Statistical Eye Plots are not BER contours.* Contour plots provide BER contours.

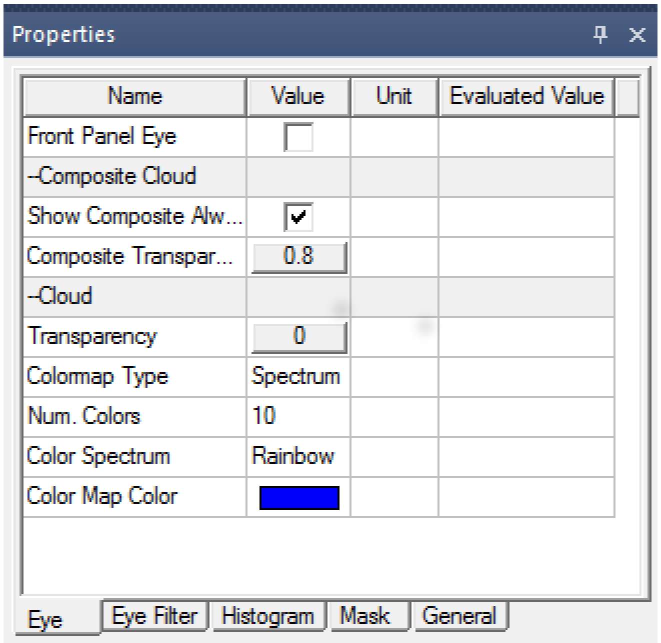
To display a statistical eye plot of the transient, QuickEye, or AMI results after a successful analysis:

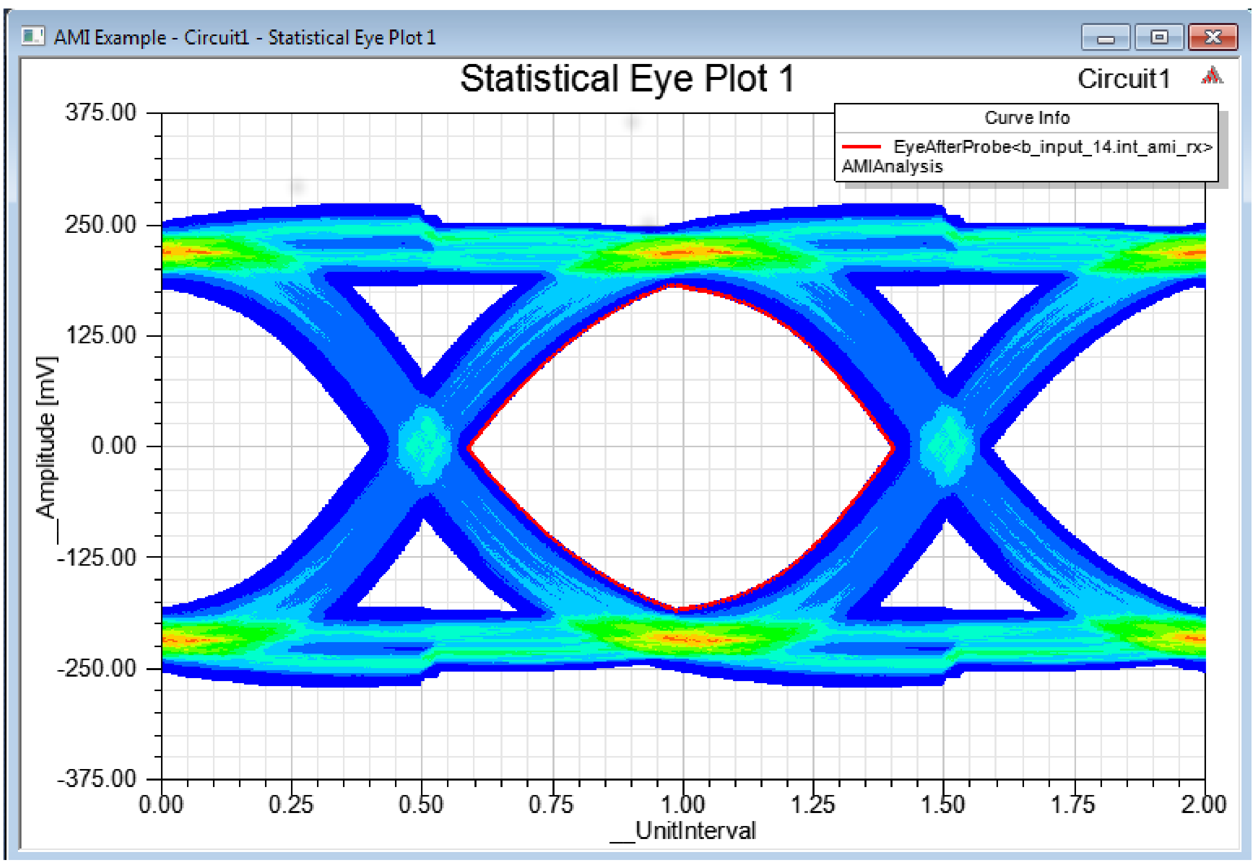
1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type**. Then select **Statistical Eye Plot** to open the **New Report** window.
2. Select the output quantities of interest and click **Add Trace**.
3. Click **Done**. A Report window opens to display a graph of the analysis results.

A Statistical Eye Plot resembles the following:



This eye diagram matches the UI extents displayed by the bathtub plot for the same trace and have a very similar eye opening position. The trace can be selected and the display changed to a "Front Panel" eye *(the default option for Statistical Eye Plots is a non-"Front Panel" eye)* via the "Eye" tab of the property window. This displays the [0, 2] UI extent front group box eye.


Properties
Name
Value
Unit
Evaluated Value
Front Panel Eye
-Composite Cloud
Show Composite Alw...
Composite Transpar...
0.8
-Cloud
Transparency
0
Colormap Type
Spectrum
Num. Colors
10
Color Spectrum
Rainbow
Color Map Color
Eye
Eye Filter
Histogram
Mask
General

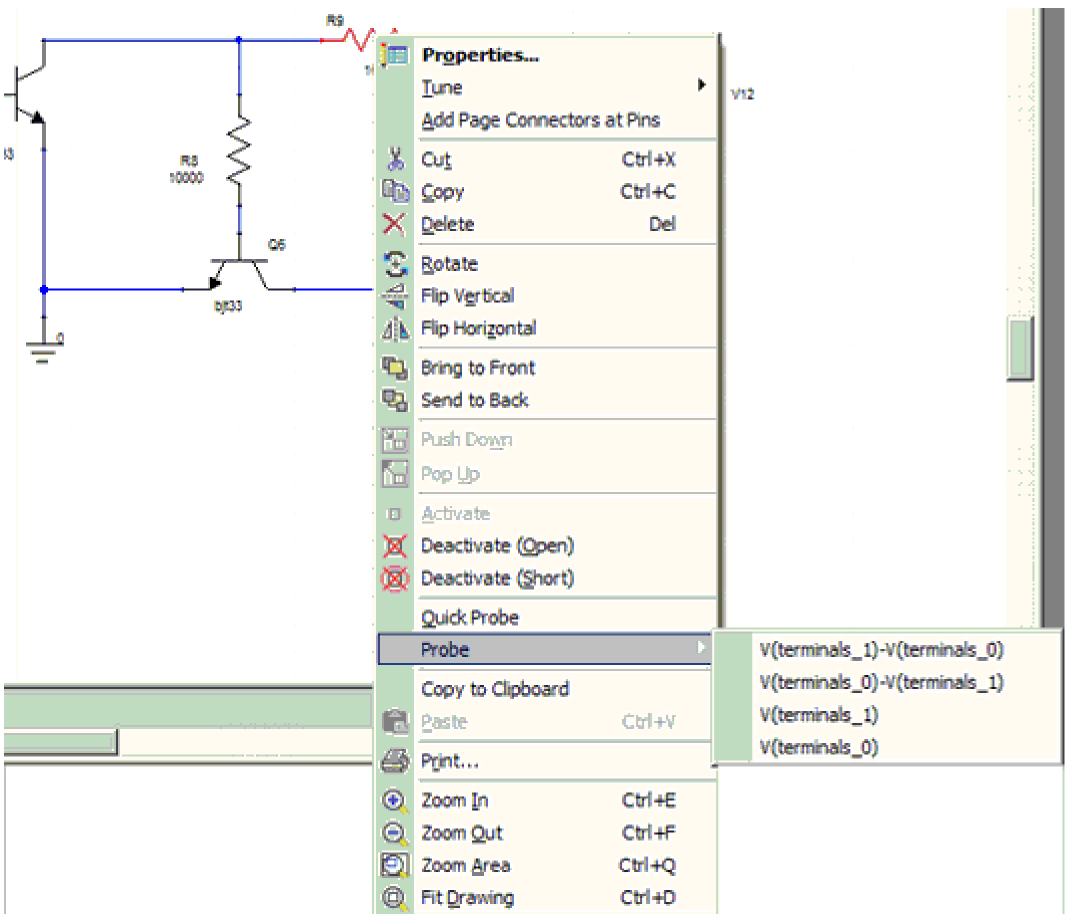

AMI Example - Circuit1 - Statistical Eye Plot 1
Statistical Eye Plot 1
Circuit1
Curve Info
EyeAfterProbe<b_input_14.int_ami_rx>
AMIAnalysis
375.00
250.00
125.00
0.00
-125.00
-250.00
-375.00
__Amplitude [mV]
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
__UnitInterval

2D and Circuit Plot-On-Schematic

One convenient way to generate a 2D report is to create a **Plot-On-Schematic**. A Plot-On-Schematic generates a standard 2D report but also associates it with the component being simulated. It then displays the plot alongside the component in the **Schematic** editor.

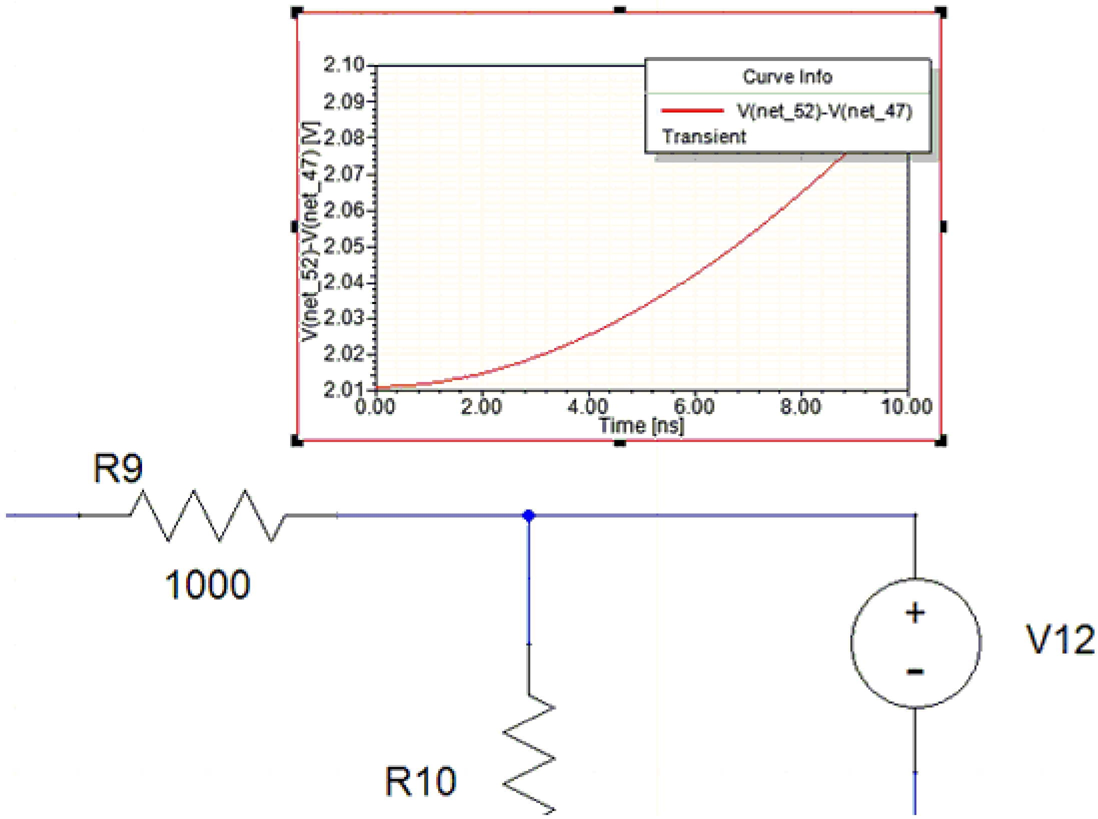
After simulating an analysis, select a component in the **Schematic Editor**, right-click over the component and highlight **Probe**. When you highlight the **Probe** menu item, a list of terminals that can be plotted is displayed (such as voltage across the component or various combinations of terminals). For larger pin components, more options appear in the **Probe** list, such as voltage across various terminals (including collector-to-base, base-to-collector, individual terminals, and any current or power options).

You can also generate a plot of a selected net, or a plot of multiple-selected nets. For instance, when you select two nets, a right-click option of **Quick Probe** allows you to create a **Plot-On-Schematic** of V(firstNet)-V(secondNet).

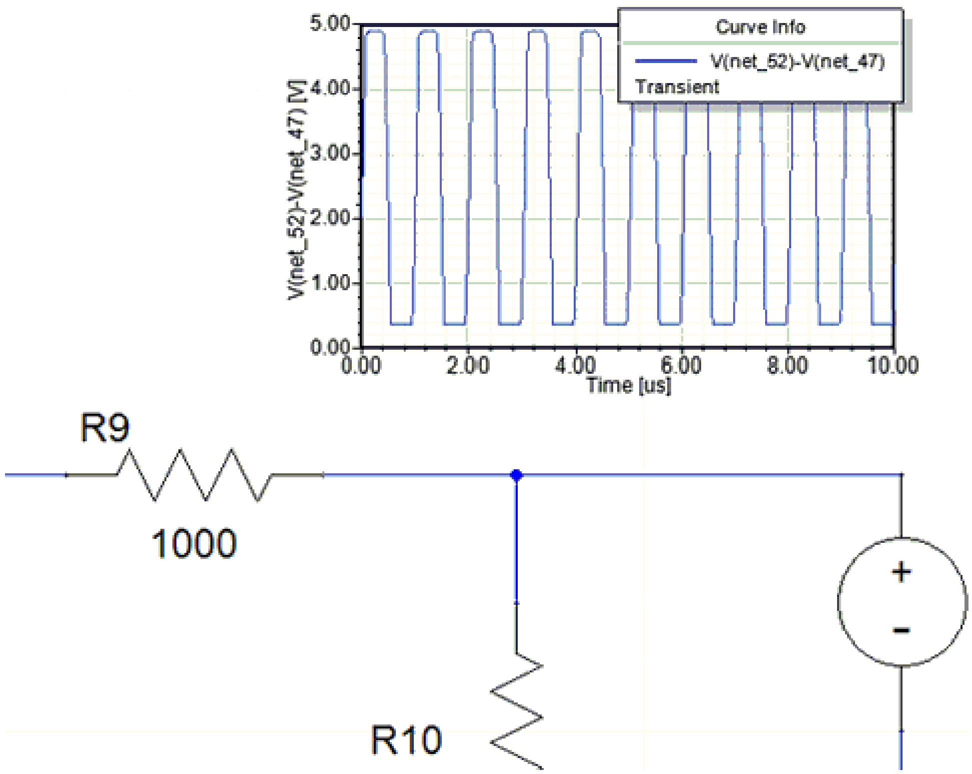


When you select an entry in the **Probe** list, the quantity is automatically plotted and placed on the schematic next to the selected component. The plot is associated with the component — as the component moves, the plots moves with it. (The plot can also be directly relocated by dragging and dropping.)

- The created plot also shows up in the **Results** folder in the **Project Tree**.
- Each component/net/port can have only one plot associated with it.
- You can have multiple plots on the same schematic, but only one such plot for each component.



If you right-click a plot in the schematic, choose the **Edit in Place** option (same as double-clicking the plot). The plot then becomes fully editable.

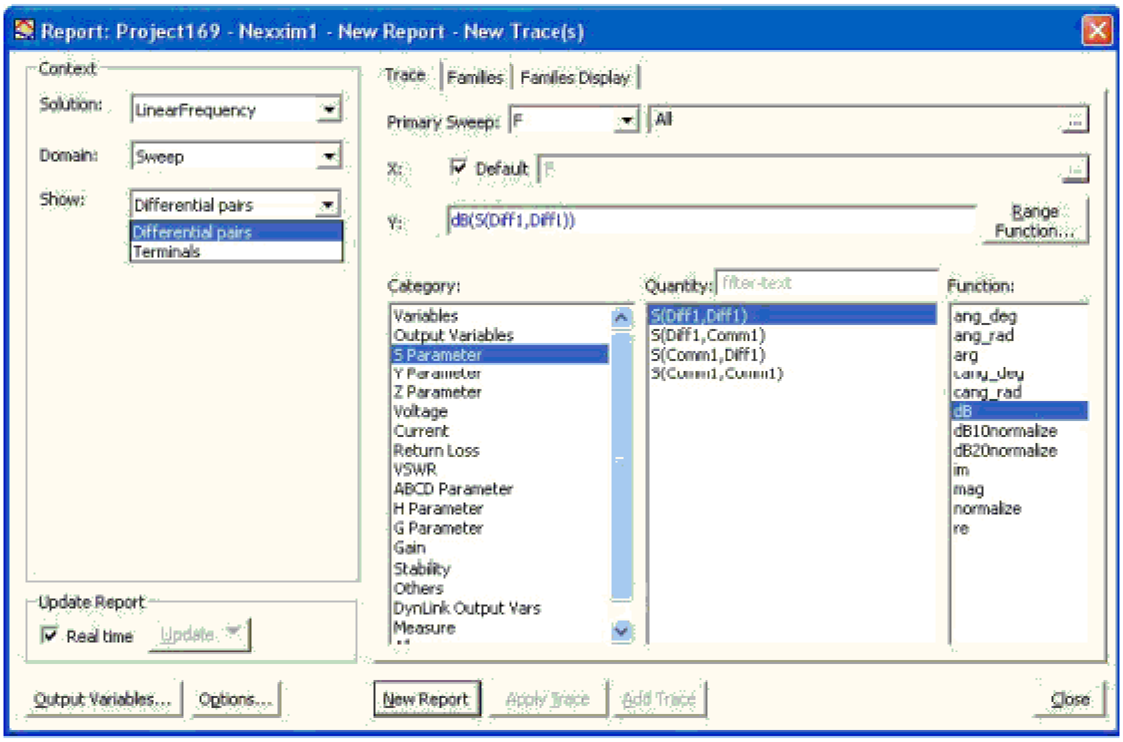


- You can move the legend bar around, change the color of traces, or change scaling for X and Y-axis — every operation that is normally performed in a regular plot window.
- You can also add plot markers.
- When you “click out” of the plot, it reverts back to being another schematic object.
- Plots are dynamically updated when you modify the analysis, so after you re-analyze, the **Plot-On-Schematic** is also updated.

For more information, see Modifying Reports.

2D and Circuit Reports with Differential Pairs

If differential pairs exist and are enabled, they change the reporter window – note the **Show** option in the **Context** group box. When differential pairs is selected as a **Show** option, the **Quantity** column changes on the original port names to the differential and common mode quantities.



2D and Circuit Creating a Report from a File

A report can be saved from one design and imported into another (or imported into the same design later). This allows you to take a static snapshot of a set of simulation data and view it at a later stage of the design process, or view the same dataset in one or more different designs.

To save a report to an Ansys Report Data File:

1. Create the report that you wish to save.
2. Right-click in the report window and select **Export** to open the **ExportReport** window.
3. Select the **Ansys ReportData files** option (.rdat) in the **Save as type** selection window.
4. Click the **Export Report** window to navigate to a appropriate location and either create a file name or select an existing file for the exported report. Then click **Save**.

To create a report from an existing Ansys Report Data File:

1. Right-click the **Results** folder in the **Project Manager** window and select **Create Report From File**.
2. Browse to select the appropriate **Ansys ReportData file** (.rdat). Then click **Open**.
3. The selected report is created in a new report window.

Note:

- Most report types supported by Electronics Desktop (including Rectangular Plots, Smith Charts, and Eye Diagrams) can be exported to **Ansys ReportDatafiles**. However, three-dimensional plots can not be exported or imported.
- Limited editing of reports created from files is supported. For example, change the color of a trace or an eye mask definition.

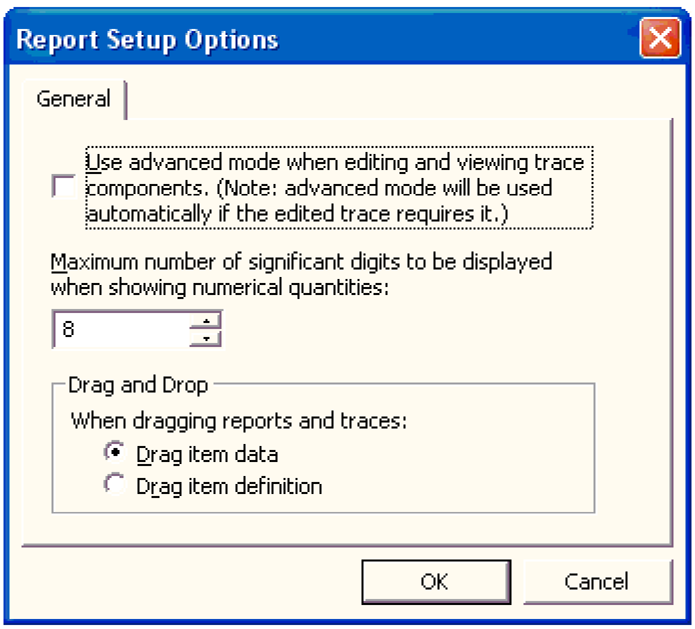
Setting 2D and Circuit Report Options

The following topics describe how to set up report options.

Setting Report Setup Options (2D and Circuit)

To set Report Setup Options in Electronics Desktop:

1. Click **Tools > Options > Report Setup Options** to open the **Report Setup Options** window.



2. Click the check box to specify whether to use **advancedmode** when editing and viewing trace components. (Advanced mode is used automatically if the trace requires it.)
3. Use the text-box/drop-down-menu to specify the number of **significantdigits** to use when displaying numeric values.
4. Specify the drag and drop behavior by clicking **Drag item data** or **Drag item definition**.

Setting Report2D Options

To set Report2D options in Electronics Desktop:

1. Click **Tools > Options > Report2D Options**. The **Report2D Options** window opens, displaying several available tabs.

   For properties controlled by check boxes, you can set values for all curves by clicking the column header cell that contains the property title. To change cell values, right-click on a cell field and select **Cut**, **Copy**, or **Paste**. To make changes to entire rows, right-click on a menu cell to select **Copy** or **Paste**.

   You can use a Restore Defaults button.

2. Click each tab, and make the appropriate selections.
3. Click **OK**.

Report2D Options: Curve Tab

These options are set on the **Curve** tab of the **Report2D Options** window.

1. Line style — select the options from the drop-down menu. The options are Solid, Dot, Dash, and Dot dash.
2. Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
3. Width — set the line width by editing the real value in the field.
4. Arrows — click the check box to use arrows on the curve ends.
5. Symbol — click the check box to have symbols mark the locations of data points on the curve.
6. Sym Freq — set the symbol frequency by editing the integer value in the field.
7. Sym Style — select the symbol to display for the designated data points. The sym style can be box, circle, vertical ellipse, horizontal ellipse, vertical up triangle, vertical down triangle, horizontal left triangle, horizontal right triangle.
8. Fill Sym — click the check box to set the symbol display as a solid or as hollow.
9. Sym Color — set the color for the symbol by double-clicking to open the Set color window. Select a default or custom color and click OK.

Report2D Options: Axis Tab

These options are set on the **Axis** tab of the **Report2D Options** window.

1. Axis Name — this describes the axis to which the following options refer.
2. Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.

3. Auto Scale — click the check box to toggle whether to auto scale the axis.
4. Min Scale — if Auto Scale is not selected, edit the real value to set the minimum value of the axis.
5. Max Scale — if Auto Scale is not selected, edit the real value to set the maximum value of the axis.
6. Auto Units — click the check box compute the correct units for the axis.
7. Units — click the cell to select from a menu of available units if you have not checked Auto Units.
8. Font color — set the font color of the axis by double-clicking to open the Set color window. Select a default or custom color and click OK.
9. Edit Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.

Report2D Options: Grid Tab

These options are set on the **Grid** tab of the **Report2D Options** window.

1. Grid Name — lists the name or letter of the grid. Not editable.
2. Line Style — select the options on the dropdown menu. The options are Solid, Dot, Dash, and Dot dash.
3. Line Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.

Report2D Options: Header Tab

These options are set on the **Header** tab of the **Report2D Options** window. For the Title and subtitle, independently specify the following:

1. Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
2. Font — click the cell to open the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.

Report2D Options: Note Tab

These options are set on the **Note** tab of the **Report2D Options** window.

1. Note Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.

2. Note Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.
3. Background Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
4. Background Visibility — click the check box to toggle the background for the note on or off.
5. Border Line Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
6. Border Visibility — click the check box to toggle the visibility of the note border.
7. Border Line Width — set the line width by editing the real value in the field.

Report2D Options: Legend Tab

These options are set on the **Legend** tab of the **Report2D Options** window.

1. Show Trace Name — click the check box to toggle the visibility of the trace name.
2. Show Solution Name — click the check box to toggle the visibility of the solution name.
3. Show Variation Key — click the check box to toggle the visibility of the variation key.
4. Text Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
5. Text Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.
6. Background Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
7. Border Line Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
8. Border Line Width — set the line width by editing the real value in the field.
9. Grid Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.

Report2D Options: Marker Tab

These options are set on the **Marker** tab of the **Report2D Options** window.

1. Marker Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
2. Marker Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the

window.

3. X Marker — use the following options to set the X Marker properties.
   a. Show Intersection — check box to show the intersection.
   b. XMarker Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
   c. XMarker Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.
   d. Box Background Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
   e. Line Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
   f. Line Style — select the options from the drop-down menu. The options are Solid, Dot, Dash, and Dot dash.
   g. Line Width — set the line width by editing the real value in the field.

Report2D Options: Marker Table Tab

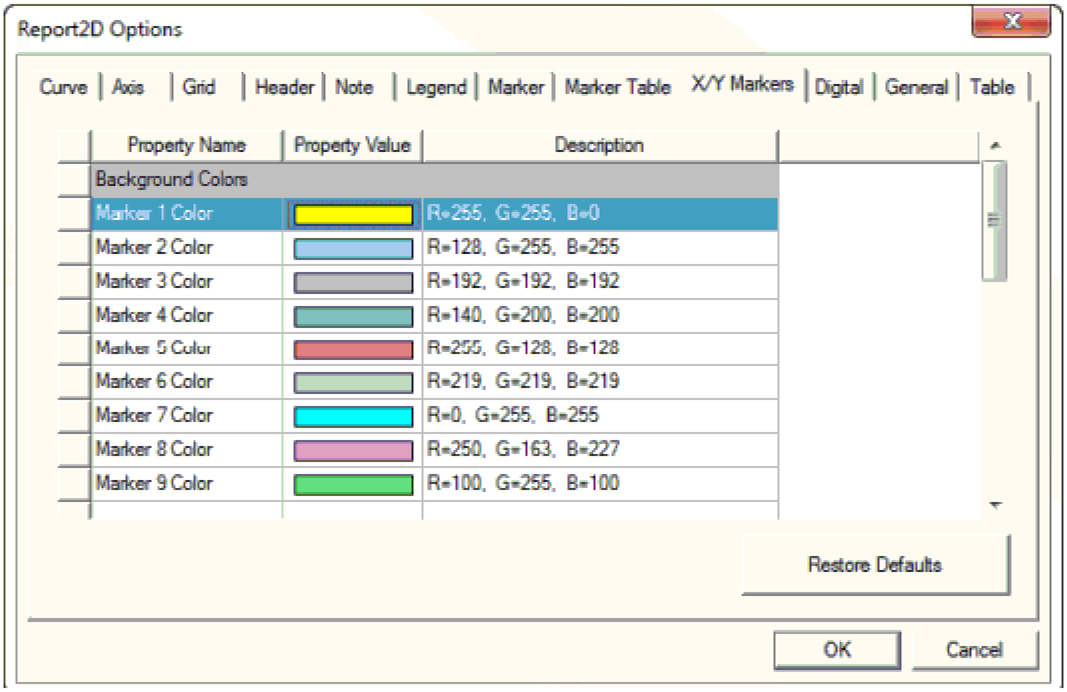
These options are set on the **Marker Table** tab of the **Report2D Options** window.

1. Precision — set the precision for marker placement by editing the real **Value** field.
2. Text Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
3. Text Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.
4. Background Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
5. Border Line Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
6. Border Line Width — set the line width by editing the real value in the field.
7. Grid Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
8. Grid Line Width — set the line width by editing the real value in the field.

Report2D Options: X/Y Markers Tab

Use this tab on the **Report2D Options** window to set the properties for the markers.

Background colors for Markers 1 through 10. You can set these by select the current color to open a color selection window, or by specifying RGB number values.



Properties, including

- On-screen intersection
- Marker Font
- Text color
- Line color
- Line style
- Line width
- Whether to Show Name
- Whether to Snap to Vertex

Inter marker deltas, including

- Whether to show delta
- Delta font
- Delta text color
- Line color
- Line style
- Line width

You can also **Restore Defaults**.

Report2D Options: Digital Tab

These options are set on the **Digital** tab of the **Report2D Options** window.

Digital Literal Foreground color.

Whether to Expand Arrays/Records

Digital Stack Hight in Pixels for the following:

- Analog
- Digital
- Enum
- Event
- Literal.

You can also **Restore Defaults**.

Report2D Options: General Tab

These options are set on the **General** tab of the **Report2D Options** window.

1. Background Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
2. Contrast Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
3. Highlight Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
4. Accumulate Depth — sets the maximum number of accumulated traces on a single report. When Accumulate Depth is turned on (i.e., set to a non-zero value),run an analysis to generate a plot, then change some parameter value(s) and run another analysis. The newly generated plot now contains both old and new results that can then be compared. The plot keeps adding new results with each subsequent analysis until the accumulate length is reached.
5. Curve Tooltip Option — click the check boxes to toggle the following properties:
   a. Show Trace Name
   b. Show Variation Key
   c. Show Solution Name
6. Clipboard Option - specify the following properties:
   a. Capture Aspect Size Ratio — this can be As Shown or Full Screen.
   b. Capture Background Color — this can be As Shown or White.

Report2D Options: Table Tab

These options are set on the **Table** tab of the **Report2D Options** window.

1. Text Font — click the cell to display the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.
2. Format — use the following properties to set the format:
   a. Field Width — set the table field width by editing the real value in the field.
   b. Precision — set the table precision by editing the real value in the field.
   c. Use Scientific Notation — click the check box to toggle scientific notation on or off.
3. Copy to Clipboard — use the following check boxes to toggle the following properties for table copy operations.:
   a. With Header
   b. With Tab Separator

Modifying 2D and Circuit Reports

After creating a report, See the following topics to modify the report's content, appearance and attributes.

Modifying 2D and Circuit Report Data

To modify the data that is plotted in a report:

1. From the **Project Manager** window, click the report you want to modify.
2. Right-click **Modify Report** to open the **Report**window.
3. The **Report** window command buttons permit you to create a new report with the settings you provide, or to modify an existing report.
   - **Output Variables** – opens the **Output Variables** window.
   - **Options** – opens the **Report Setup Options** window. 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.
   - **New Report**. Adds a report to the Project tree under the Results icon. The new Report is displayed in **Electronics Desktop** window.
   - **Apply Trace** – applies the selected traces in a report based on further processing or changes.
   - **Add Trace** – this is enabled when you have created or selected a report. Add one or more traces to include in the report.
   - **Close** – closes the **Report** window.

The updated report appears in the view window.

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*

Modifying the Background Properties of a 2D or Circuit Report

The standard View menu Zoom and Fit commands operate on reports. When zooming on a report, axis labels and ticks adjust automatically during the zoom operation and rescale to their final value after the zoom operation is complete.

To modify the appearance of a report, or the display properties an object in a report:

1. Open the report you want to modify.
2. You must select an editable object in the report to be able to edit its properties. Click an object to select it and view its Properties in the docked properties window. To open a floating Properties window, either double-click the selected object, or click **Edit > Properties** on the toolbar.

The selectable objects in reports are as follows:

- **Header** – this lets you edit the Properties for the text displayed at the top of the report, including the Title font, Company Name, Show Design Name, Subtitle Font. The plot title is tied to the report's name and is not a Header property. If you change the report name in the **Project Manager** window, plot title synchronizes. The Company Name and the Show Design Name check box are grouped in the Properties window as Subtitle. Edits to the Subtitle Font Property affects both of them.
- **General —** this window (or General tab for other Report properties windows) lets you edit the background color (the perimeter around the trace display) for the plot, the contrast color (the trace display background), the Field width, the Precision, 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.)
- **Legend —** this lets you edit the Properties for whether to Show Trace Name, Solution Name, and Variation Key. At least one of these three must be selected. You can also edit the Font, the background color of the Legend box, the Border Color, the Border Width, Grid Color (for the lines between Trace descriptions), and the Grid line width. Also see Modifying the Legend in a Report
- **Traces** – select traces either in the Legend or on the plot. The properties for traces include: Color, Line Style, Line Width, Trace Type, whether to Show a symbol, Symbol Frequency, Symbol style, whether to Fill symbol, symbol color, and whether to Show arrows. See Editing the Display Properties of Traces.
- X or Y **Axis Tab** — the defaults for most of these values are set in the **Report 2D Options** Axis tab.
- 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** window.
- Axis Color — set the color by double-clicking to open the Set color window. Select a default or custom color and click OK.
- Text Font — click the cell to open the Edit Text Font window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window.
- **Manual Format** (group box)
  - Number format — choose a format (i.e., Auto, Decimal, or Scientific notation).
  - Field Width — enter a real value.
  - Field Precision — enter a real value.
- X or Y **Scaling Tab** — These properties provide control over scaling.
  - Axis Scaling — choose scaling as Linear or Log. For the Y axis, all zero or negative values are discarded before log scaling is applied.
  - Specify Min — check box
  - Min — text entry in same units as axis units. Saved as SI internally.
  - Specify Max — check box
  - Max — text entry in same units as axis units. Saved as SI internally.
  - Specify Spacing — check box
  - Spacing — text entry in same units as axis units. Saved as SI internally
  - **Manual Units** (group box)
  - Auto Units — click the check box compute the correct units for the axis.
  - Units — click the cell to select from a menu of available units if you have not checked Auto Units.
  - **Infinity Visualization** (group box)
  - Map Infinity Mode — check box.

Each axis now can be set to treat infinity values in a user-defined way. When you check the Map Infinity Mode, any infinity values in the input data get the infinityMap value (negative infinity get the value*-1 and positive infinity the positive value specified). This can be useful if there are zeros, or very small values that Electronics Desktop treats as zero, in the data, for example, dB Gain.

  - Map Infinity To — enter a real value for the Map Infinity Mode.

3. Edit the properties, and **OK** the window to apply the changes.

Modifying the Legend in a 2D or Circuit 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** window.

3. Select the **Legends** tab.

This lists the legend (or legends) in the report.

4. Check the visibility check box, and OK the window to close it and apply the change.

To edit the display properties of a legend:

1. Select the legend in a report by clicking on the group box to display a docked properties window, or right-click on the legend and select **Edit > Properties** to open the floating properties window.

This lets you edit the Properties for whether to Show Trace Name, Solution Name, and Variation Key. At least one of these three must be selected.

You can also edit the Font by clicking the Font cell to open the **Edit Text Font** window. The window lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The window also contains a preview field. OK the selections to apply the font edits and close the window

You can also edit the background color of the Legend box, the Border Color, the Border Width, Grid Color (for the lines between Trace descriptions), and the Grid line width.

2. Click OK to close the Properties window and apply the selections. To **change the display name** for traces, see Editing Trace Properties.

To move a legend in a report:

1. Click and hold and 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.

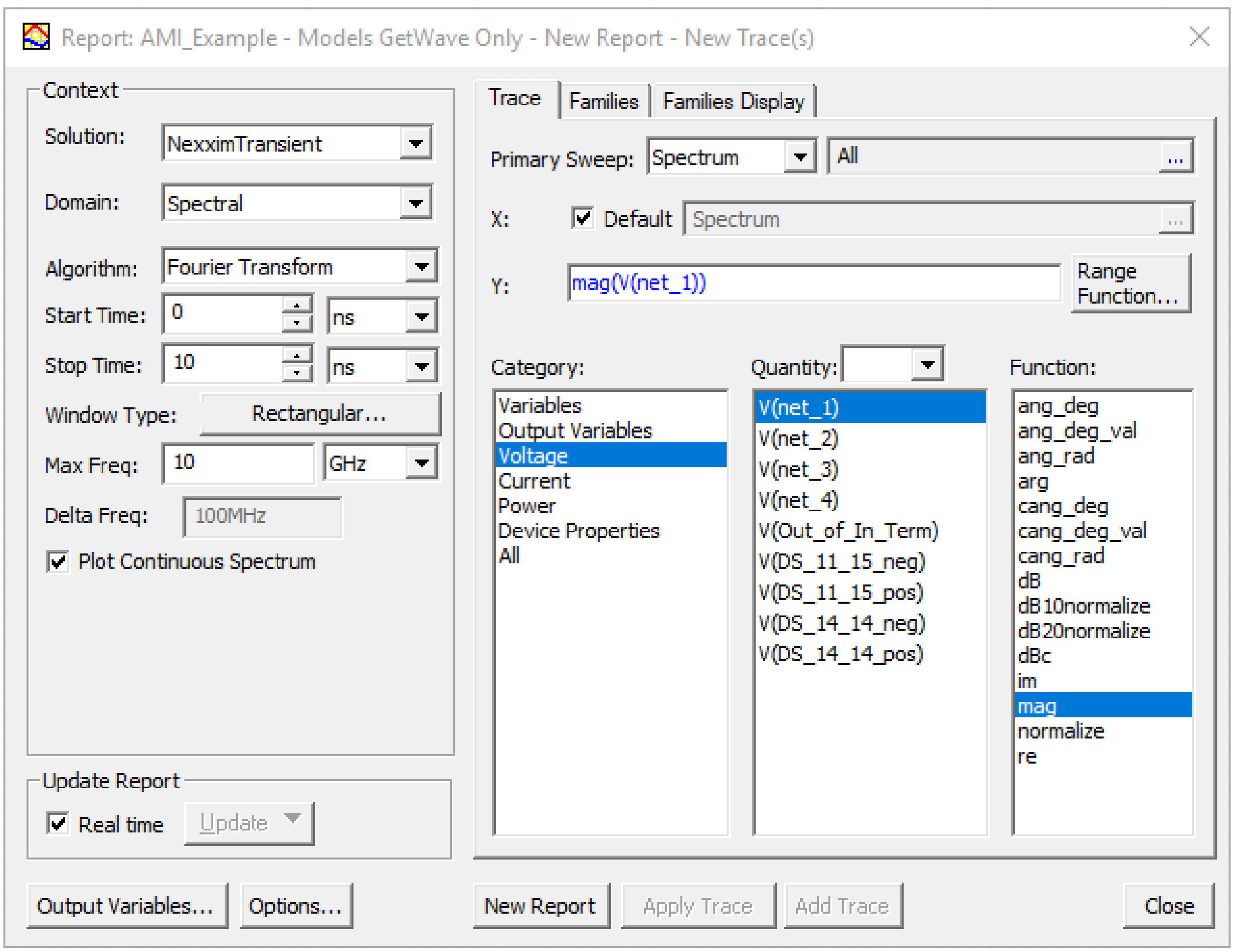
The mouse pointer changes to a horizontal or vertical line with arrow tips.

2. **Click+drag** the horizontal or vertical edge to the desire size.
3. Release.

Plotting Circuit Spectral Domain Data

After transient analysis, one can exam frequency components of the signal of interest by plotting spectral domain data on the Reporter.

1. Open a project in **Project Manager** window.
2. Right-click **Results**.
3. In **Solution**, choose **NexximTransient**.
4. In **Domain**, choose **Spectral**. Several fields appear in the **Report** window:



In the **Context** pane:

- **Start Time** and **Stop Time** define the signal duration T_0 = **Stop Time - Start Time**.

> **Note: Stop Time** and **Start Time** should lie inside the transient simulation time limits.

- **Max Freq** sets the maximum frequency for spectral analysis and plotting.
- **Delta Freq** is automatically supplied as $1/T_0$ Hz.
- The **Algorithm** drop-down menu contains 3 options: **Fourier Integration** (FI), **FFT**, and **Fourier Transform**.

  Fourier Integration and **FFT** both provide Fourier Series (or Fourier Coefficients) of the transient signal with DFT, where periodicity is implicitly assumed and the period is defined as T_0 = **Stop Time - Start Time**.

  Fourier Integration computes Fourier series by numerically integrating the input waveform from its transient simulation. The sampled waveform may not be on a set of evenly spaced time points. **Fourier Integration** is much slower compared with **FFT**, but it can be more accurate since it preserves the shape of underlying waveform without re-sampling.

  The **FFT** algorithm re-samples the input waveform by interpolating to evenly spaced time points. It uses the FFTW library that is publicly available and whose computational complexity is

  $$O(Nlog_2N)$$

  where N is the sample size after re-sampling and N = 2 *#Harmonics +1, where #Harmonics = **Max Freq** /**Delta Freq**.

> **Note:**
>
> - Based on the Nyquist-Shannon Theorem or Sampling Theorem, the sampling rate should be at least twice the bandwidth of the signal to avoid aliasing.
> - Spectral plots based on **Fourier Integration** and **FFT** are single-sided with magnitudes of non-zero harmonics doubled because the negative frequency components are folded to the positive side.

  Fourier Transform (FT) provides the Fourier transform of a signal defined in [**Start Time**, **Stop Time**] by leveraging the **FFT** algorithm. The Fourier transform of a time limited signal f(t) in [0, T_0] is

  $$\mathcal{F}(f) = \int_{-\infty}^{\infty} \mathrm{f}(t)e^{-i2\pi ft}dt \ = \int_{0}^{T_0} f(t)e^{-i2\pi ft}dt$$

The Fourier series of a periodic signal that is equal to f(t) in [0, T_0] and with period T_0 is

$$c_n = \frac{1}{T_0} \int_0^{T_0} f(t) e^{-i \frac{2\pi}{T_0} nt} dt$$

Hence,

$$c_n = \frac{1}{T_0} \mathrm{F} \left(\frac{n}{T_0} \right)$$

In other words, the Fourier series of f(t) based on **FFT** can be multiplied by T_0 in order to get the **Fourier Transform** of f(t) at sampled frequencies n/T_0.

> **Note:** Spectral plots based on **Fourier Transform** only show the positive frequency side.

- The **Plot Continuous Spectrum** check box provides a continuously plotted spectrum if selected and a discrete spectrum when cleared.
- **Window Type** determines which window type to apply.

  When applied to transient data, a window type can help resolve closely spaced harmonics and reduce spectral leakage due to the finite nature of the time signal. The following window types are available. (The “t” parameter is normalized time.)

  - **Rectangular**: 1.0
  - **Bartlett** (also known as the Parzen or triangular window): 1.0 - t
  - **Blackman**: 0.42 + 0.5*cos(π*t) + 0.08*cos(2*π*t)
  - **Hamming**: 0.54 + 0.46*cos(π*frac)
  - **Hanning**: 0.5 * (1.0 + cos(π*t))
  - **Kaiser**: BesselI0(KaiserParam * sqrt(1.0 - t*t)) / BesselI0(KaiserParam)
  - **Welch**: 1.0 - t^2
  - **Weber**: (0.828217*π^3*t^3 - 1.637363*π^2*t^2 + 0.041186*π*t + 0.99938)
  - **Lanzcos**: sin(π*t)

  Optionally, click the **Adjust Coherent Gain** check box to adjust signal levels (based on the window type) as if no window are used. When non-rectangular windows are used, the window automatically changes the power level of the signal. This is known as the “coherent gain” or “processing loss” of the window.

References:

[1] Fredric J. Harris, "From the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE* 66, no. 1 (Jan 1978): 51-83, http://web.mit.edu/xiphmont/Public/windows.pdf.

[2] Charan Langton and Victor Levin, "Fourier Transform of continuous and discrete signals," *The Intuitive Guide to Fourier Analysis and Spectral Estimation with MATLAB* (Mountcastle Academic, 2017), chap. 4, http://complextoreal.com/wp-content/uploads/2012/12/fft4.pdf.

[3] Paul Heckbert, "Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm" (Notes 3 class handout for Computer Graphics 2, 15-463, Carnegie Mellon University, Pittsburgh, PA, Jan 27,1998), http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf.

[4] Rolf Isermann and Marco Münchhof, " Spectral Analysis Methods for Periodic and Non-Periodic Signals," *Identification of Dynamic Systems* (Springer, Berlin, Heidelberg, 2011), chap. 3, https://link.springer.com/chapter/10.1007%2F978-3-540-78879-9_3.

[5] Julius O. Smith III, *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications*, 2nd ed. (W3K Publishing, 2007), https://www.dsprelated.com/freebooks/mdft/.

Delta Markers in 2D Reports (2D and Circuit)

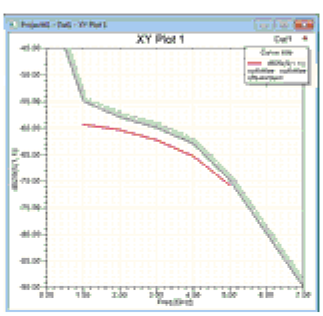
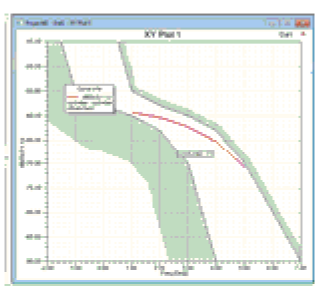
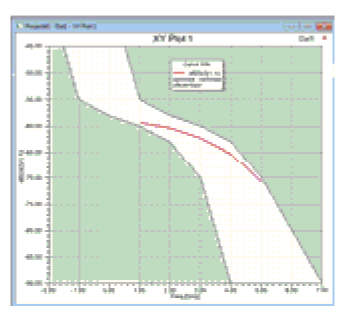
To view the difference between any two marker points in a report:

1. Set the first marker by clicking and holding the mouse button.
2. Move the mouse — without releasing the left — to another position, then release the left button to create the second marker.

The marker text window displays the difference between the two markers, instead of the absolute X and Y values.

Limit Lines in 2D and Circuit Cartesian Plots

Limit lines are simple graphical representations of constraints on XY plots. These are modeled as a sequence of XY point pairs. For example, designate a single limit line to delineate either an upper or lower limit, or two limit lines to delineate upper and lower limits. Add as many limit lines as you want. You can control the display properties of the line including color and hatch width.

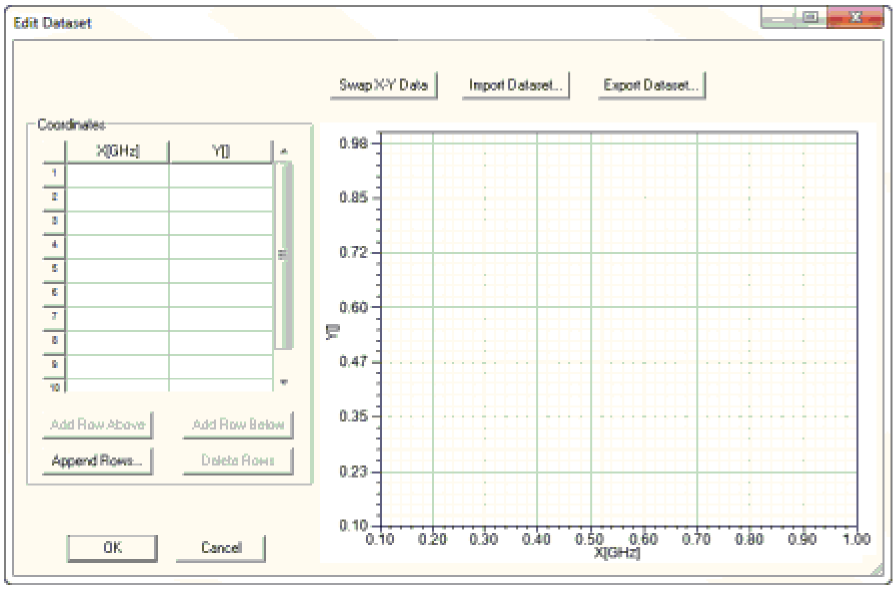




> **Note:** Limit lines are available only on Rectangular (XY) plots, not on XY-like plots such as Bode or Rectangular Stacked.

To create a limit line:

1. Click **Report2D>Add Limit Line**, or right-click an XY plot and select **Add Limit Line**.

   This opens an **Edit Dataset** window.



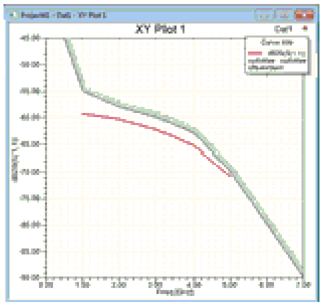
1. You can use this window to:
   - Enter the XY coordinate values directly.
   - Import XY values from a tab-delimited **.tab** file.
   - Export dataset to a file.

   To add additional data points, click **Append Rows ...**, or modify the table by clicking **Add Row Above**, **Add Row Below**, and **Delete Rows**, as necessary.

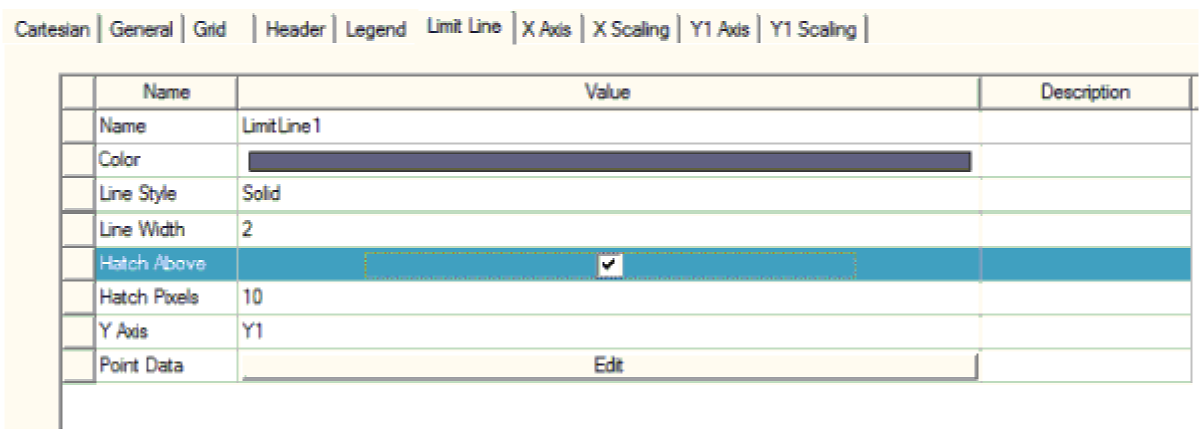
   You can press **Shift** +click to select multiple adjacent rows, or **Ctrl**+click to select any combination of rows for deletion.

> **Note:** Each limit line is associated with a particular Y axis (because it has to be scaled the same way as all the curves associated with the axis, follow its log/linear scale et cetera). This Y axis association defaults to the first available Y axis when the limit line is created. However, if the plot contains multiple Y axes, it can be associated with a different Y axis later via its properties tab.

2. Click **OK** to add the limit line you defined to the plot. The line divides the plot into two regions in the context of its length. By default, the upper region is hatched to designate constraint violation.



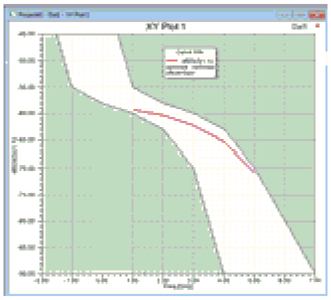
1. You can select the limit line in the plot to edit its properties on the **Limit Line** tab of the plot properties.



Name	Value	Description
Name	LimitLine1	
Color		
Line Style	Solid	
Line Width	2	
Hatch Above	✔	
Hatch Pixels	10	
Y Axis	Y1	
Point Data	Edit	

- Line properties: color, style and width
- Hatch direction (hatch above or below the limit line)
- Hatch width in pixels
- Y axis association
- The point data that defines the limit line itself

2. If you add a second limit line, designate it as hatch below by unchecking **Hatch Above** to produce a “tunnel” marking the upper and lower constraints.



2D and Circuit Spinning a 3D Report

You can set a 3D report spinning around any axis, to enable viewing from multiple angles without having to rotate repeatedly.

1. From the **View** menu, select **Spin**. Alternatively, right-click the 3D graph, select **View** on the dropdown menu, and select **Spin** on the subordinate menu. The cursor changes to the Spin mode cursor (curved arrows).
2. **Click**+**drag** the mouse in the direction and at the speed you want to spin the view. When you release the mouse button, the image begins spinning continually.
3. To stop spinning, click the mouse again in the report window. Start another spin in the same or another direction and speed by repeating step 2.
4. To end Spin mode, press **Esc** or deselect **Spin** on the **View** menu.

Reviewing 2D Polar Plots

For a polar plot of S-parameters, Electronics Desktop 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

VSWR

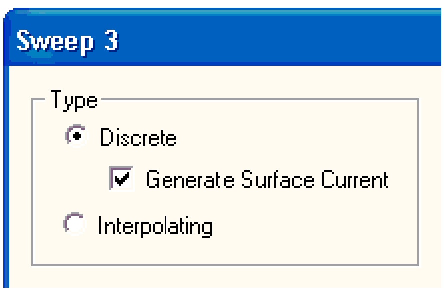
The voltage standing wave ratio, calculated on the equation $\frac{1+|S_{ij}|}{1-|S_{ij}|}$.

A scale under the plot displays the scale of points along the R-axis.

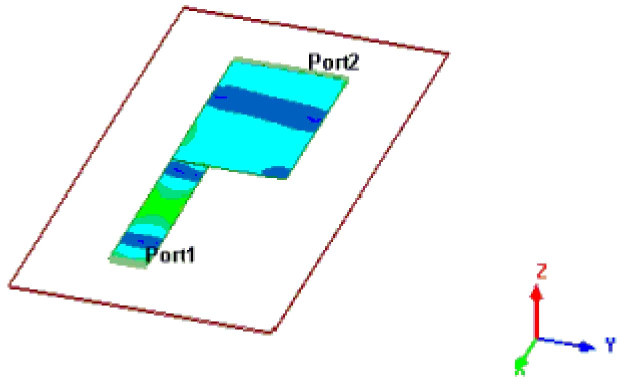
2D Overlaying Surface Currents on a 3D View

Surface currents calculated as the results of a Planar EM simulation can be displayed as overlays on the 3D viewer.

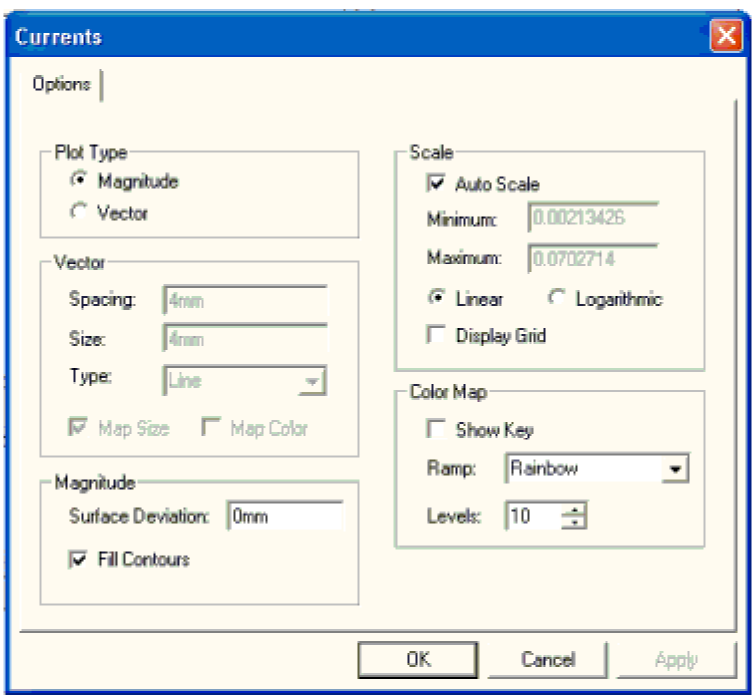
1. To ensure that the surface current information is generated, the sweep setup must specify a **Discrete** frequency sweep, and the **Generate Surface Current** option must be enabled (checked):



2. Run the Planar EM analysis with the sweep.
3. to open the surface current as an overlay, expand the **Analysis** icon in the **Project Manager** window, and select **Setup***m* **> Sweep***n* **> Results > Display Currents** (*m* and *n* identify the particular solution setup and sweep setup, respectively). 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 opens 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** window opens:



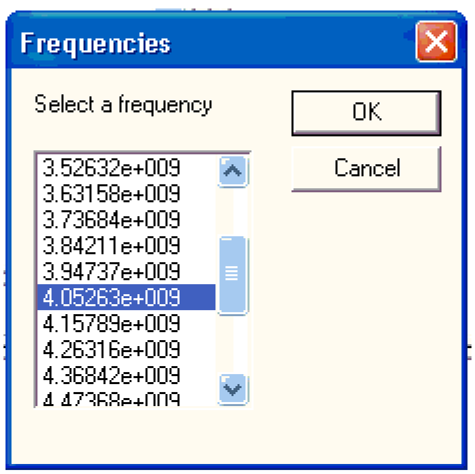
In the **Plot Type** group box, select **Magnitude** to enable the **Magnitude** group box options or select **Vector** to enable the **Vector** group box options.

In the **Scale** group box, 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** group box, 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 (default is off).

Click **Apply** to apply any changes to the display without closing the window. Click **OK** to apply any changes and close the window. Click **Cancel** to close the window without changing any options.

5. To select the frequency for the current overlay, expand the **Results** icon in the **Project Manager** 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** window opens:



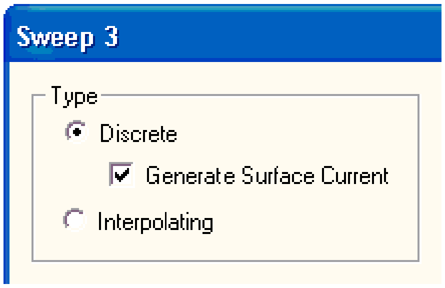
The list displays the frequencies that are swept in the analysis. When you select a frequency on the list, the overlay displays the surface current values calculated at that frequency. Click **OK** to leave the overlay at the selected frequency, or click **Cancel** to close the window without applying any frequency changes to the overlay.

6. To dismiss the overlay, expand the **Results** icon in the **Project Manager** window, right-click **Setup***m***:Sweep***n***:Currents***k*, and select **Delete s** (*m*, *n*, and *k* identify the particular solution setup, sweep setup, and surface current setup, respectively).

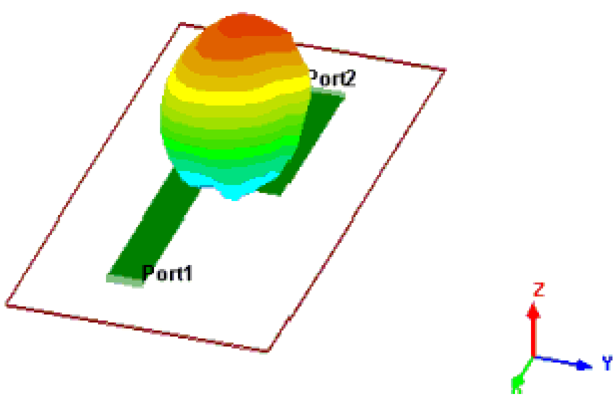
2D Overlaying Far Fields on a 3D View

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):



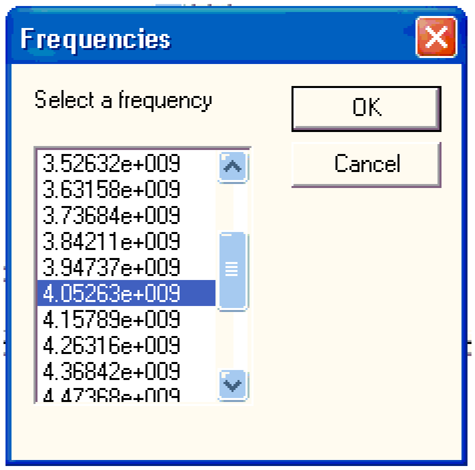
2. Run the Planar EM analysis with the sweep.
3. to open the far field as an overlay, expand the **Analysis** icon in the **Project Manager** window, and select **Setup***m* **> Sweep***n* **> Results > Far Field** (*m* and *n* identify the particular solution setup and sweep setup, respectively). Also 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 opens 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 Manager** window, and select **Setup***m***:Sweep***n***:Far Field***k* **> Frequency**. The

Frequencies window opens:



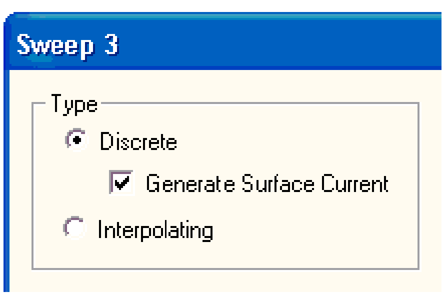
The list displays the frequencies that are swept in the analysis. When you select a frequency on the list, the overlay displays the far field values calculated at that frequency. Click **OK** to leave the overlay at the selected frequency, or click **Cancel** to close the window without applying any frequency changes to the overlay.

5. To dismiss the overlay, expand the **Results** icon in the **Project Manager** window, right-click **Setup***m***:Sweep***n***:Far Field***k*, and select **Delete** (*m*, *n*, and *k* identify the particular solution setup, sweep setup, and far field setup, respectively).

2D 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):

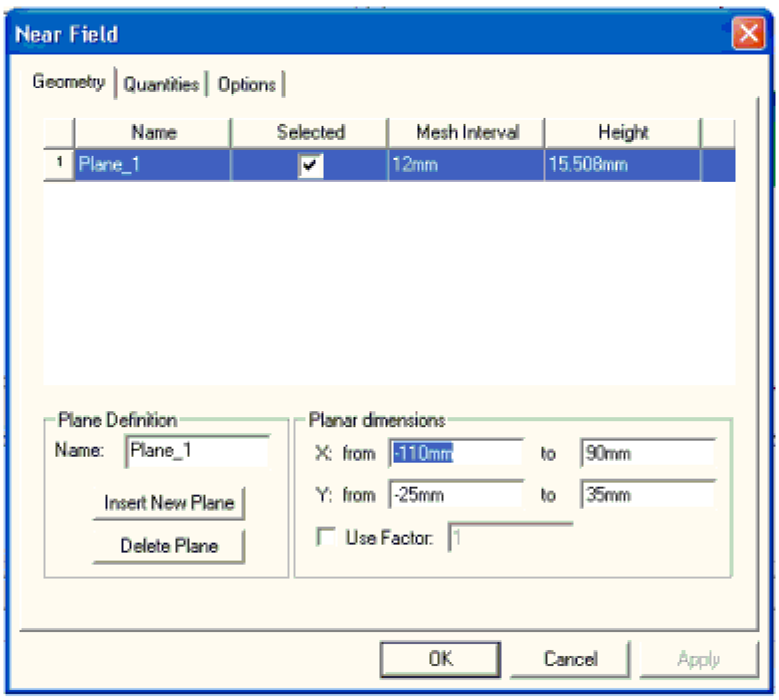


2. Run the Planar EM analysis with the sweep.

3. to open the near field overlay, expand the **Analysis** icon in the **Project Manager** window, and select **Setup***m* **> Sweep***n* **> Results > Near Field** (*m* and *n* identify the particular solution setup and sweep setup, respectively). Also select from a list of corresponding Setup/Sweep overlay choices which are displayed when you right-click **Field Overlays** in the **Project Tree** to open the **Near Field** window. The window has three tabs and at the bottom of each tab are three buttons:

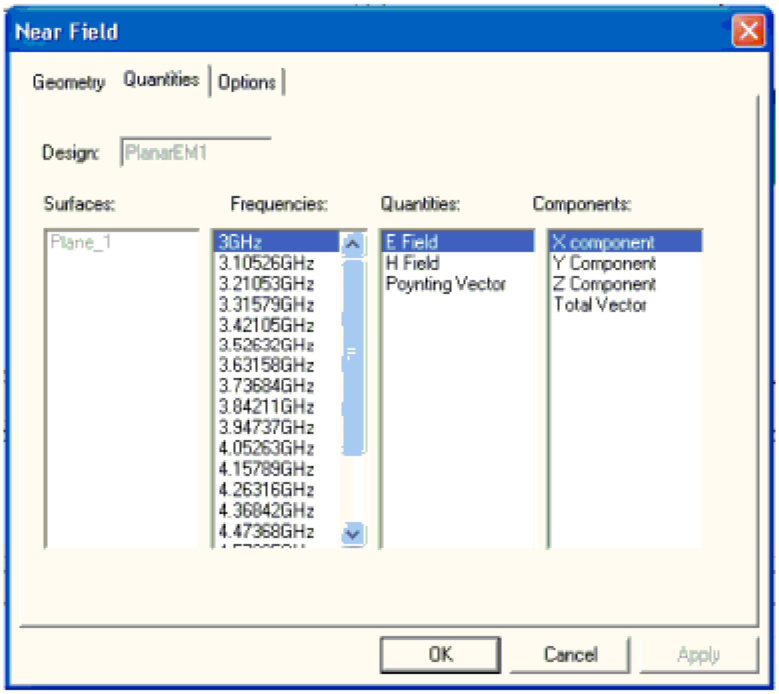
- **Apply** is activated whenever you change a value. Click **Apply** to start the display, and see the effect of each change. The window stays open.
- When no values are changed on any tab, the **OK** button starts the display. When one or more values have been changed, **OK** applies the changes. In either case, **OK** closes the window 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 window 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.

4. The **Near Field** window opens with the **Geometry** tab displayed:



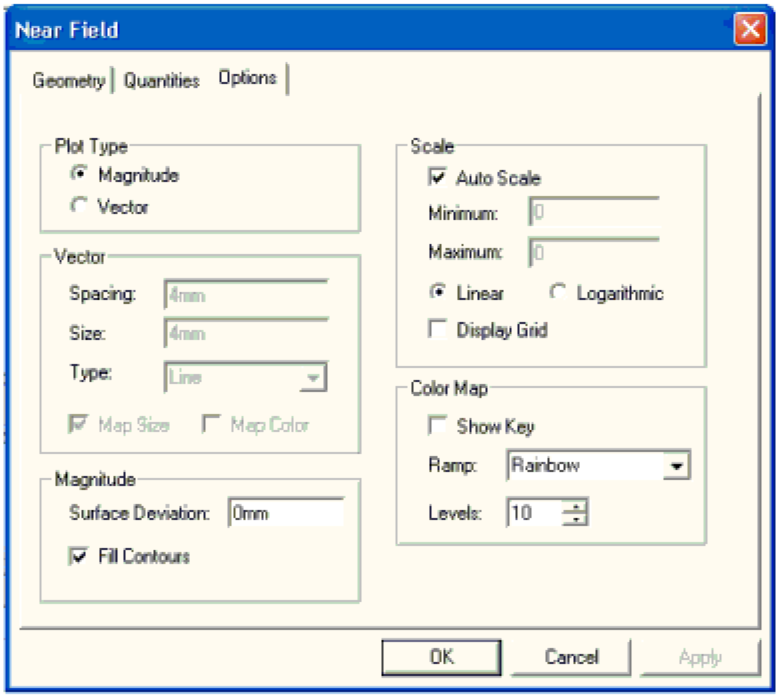
5. Use the options in the **Geometry** tab to select (or define) one or more planes for the calculation, including the dimensions to be used, and a scale factor if appropriate.

6. Click the **Quantities** tab to select the near field quantity to be calculated:



7. If more than one surface is involved, select a surface on the **Surfaces** group box.
8. Select a frequency on the **Frequencies** list. The frequencies are the ones swept in the analysis.
9. Select a field type on 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.
10. Select a vector component on the **Components** field.

11. Click the **Options** tab to specify the display options for the near field overlay:

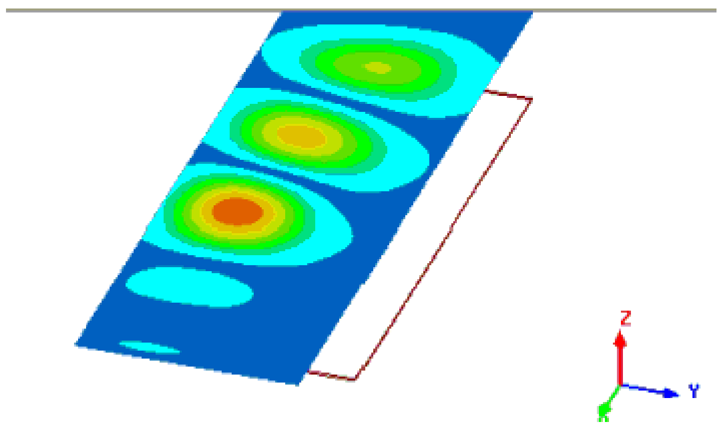


12. In the **Plot Type** group box, select **Magnitude** to enable the **Magnitude** group box options or select **Vector** to enable the **Vector** group box options.

13. In the **Scale** group box, 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).

14. In the **Color Map** group box, 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).

15. When you click **Apply** or **OK** in any of the **Near Field** window tabs, the 3D viewer window opens with the near field values overlaid on the geometry:

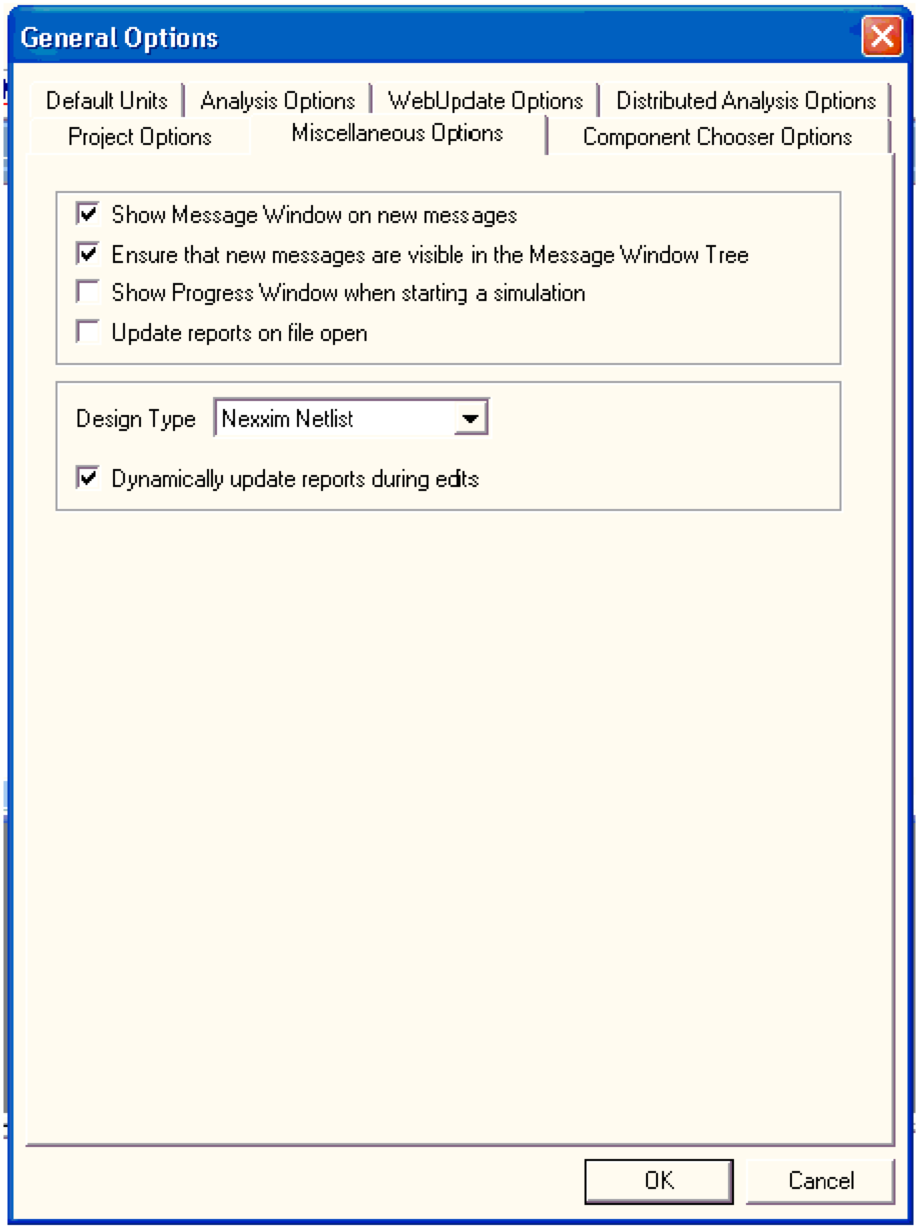


16. To dismiss the overlay, expand the **Results** icon in the **Project Manager** window, right-click **Setup** > **Sweep** > **Near Field**, and select **Delete**.

2D and Circuit Updating Post-processing Data

To manually update the report that is active in the design area using the latest simulation results, right-click in the report and select **Update Report** on the menu. Alternatively, click the **Report** entry in the menu and select **Update Report**(the text of the **Report** entry depends on the type of report being displayed).

In a schematic design, an option is available to control all schematic-based analyses. Select **Nexxim Circuit Options** from the **Tools > Options > General Options** drop-down. Here is the **Nexxim Options** window:



On all three options windows, the option **Dynamically update reports during edits** is available beneath the **Miscellaneous Options** tab and controls the version of solution data to be reported.

Electronics Desktop solution data is versioned. You might have solution data for a particular state of the Design, then edit the Design and re-simulate. Solution data is now available for both states of the Design. Suppose the following sequence occurs:

- You simulate and create a report. The report displays the current solution data (state 1).
- You edit the design. The report is marked 'Invalid' (a large "X" appears in the upper-right corner of the report window and on the report icon in the Results folder in the **Project Manager** window.
- You rerun the simulation. The report is updated with the latest solution data (state 2), and the report is marked as valid (the "X" disappears).
- Now you click **Undo**. Electronics Desktop still has the 'old' solution data (state 1). If the option **Dynamically update post-process data on edit** is off, the State 2 report data becomes invalid again. If **Dynamically update post-process data on edit** is on, Electronics Desktop reloads the now valid state 1 solution data into the plot and marks it valid.
- Upon **Redo**, the result again depends on the setting of the **Dynamically update post-process data on edit** option. If the option is off, the report becomes valid again (the state 1 data was not reloaded by the **Undo** operation, so the report is still displaying state 2). If the option is on, the state 2 data is reloaded dynamically and plot is also valid.

Deleting 2D and Circuit Reports

To Delete a Single Report

To delete a single report, right-click the report in the **Results** folder in the **Project Manager** window for the design, and click **Delete** on the menu. Or, select a report icon in the **Project Tree**, and right-click to open the shortcut menu, then click **Delete** on the shortcut menu or the "**X**" icon on the toolbar to delete the selected report.

To Delete All Reports

To delete all reports, on the **Product** menu, select **Results** > **Delete All Reports**, or, on the **Project Manager** window, expand the **Project Tree**. Then right-click the [*active design folder*] and select **Delete All Reports**. You can also right-click **Results** and select **Delete All Reports**.

Working with Traces in 2D and Circuit Reports

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 one or more variables has been solved, at several values, "sweep" over some or all of those values, resulting in a curve in 2D or 3D space.

A report can include any number of traces and, for rectangular graphs, up to four independent y-axes. Traces appear in the **Project Manager** window 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 a selection can be made:

- For Windows 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.

Adding a Trace to a 2D or Circuit Report

In general, to add a trace to a report:

1. Select a report in the **Project Manager** window and right-click and select **Modify Report.**
2. In the **Report** window specify the Y component information.
   a. From the drop-down menu, select the Category of information you want to plot.

      The Category drop-down menu lists the available categories for the Solution type and the active 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 on 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. 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 fields. Invalid functions appear in red.
   e. **Range Function** — opens the **Set Range Function** window. This applies currently specified Quantity and Function.
3. In the **Report** window specify the X axis information (e.g., Primary Sweep).
4. Click **Add Trace**.

A trace is added to the traces list under its report icon in the **Project Manager** window. The trace represents the function of the quantity you selected and is plotted against other quantities or swept variable values. Selecting a Trace in the **Project Manager** window 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 on the **Project Tree**.

By the default, the Trace name is the definition (the category, quantity and function). The trace is 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** window. 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 2D and Circuit Trace Properties

To edit trace properties such as the name, the component definition, or 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** window.

   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** window, 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 a **trace component definition**:

1. Select the trace in the **Project Manager** window.
2. In the docked Properties window for the trace, select the component field of interest and choose **Edit** on the drop-down menu.

   This displays the an edit Component field window.form which Edit the category, quantity and function.

3. Click OK to apply the changes and close the window.

To edit a **trace Context**:

1. Select the trace in the **Project Manager** window to open 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 on the drop-down menu.

To edit a **variable** for a trace:

1. Select the trace in the **Project Manager** window to open the docked properties window.
2. Under the -Variables category, on the Families line, click **Edit** to open the Edit families window.

   From this window, select **Sweeps** or **Variations**. Each selection changes the available editable values.

   If other nominal values are available, click the **[...]** button to select from a list.

Editing the Display Properties of Traces (2D and Circuit)

To edit the display properties of a trace:

1. Select a trace in an open **Report** window.
2. Clickce on the trace to view a **Docked Properties** window, or double-click to open **Properties** window.

The display properties window for a trace includes a **General** tab and an **Attributes** tab.

The General tab properties apply to the general appearance of the plot. They include the Background color, Contrast color, Field width, and Whether to use Scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.)

The Attributes Tab properties apply specifically to the Trace. The defaults are set in the Report2D options. They include:

- Name — not editable by selecting the trace on the Report. It shows the characteristics of the trace as defined in the **Report** window. To edit a trace name, see Editing Trace Properties
- Color — shows the Trace color. Double-click to open a Color window. You can select from Basic colors, or custom colors. You can define up to 16 custom colors by selecting or by editing the Hue, Saturation, Luminescence, and the Red, Green, and Blue values.
- Line style — a drop-down menu lets you select Solid, Dot, Dash, or Dot-dash.
- Line width — a 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.
- Show Symbol — whether to show a symbol at the data points on the line.
- Symbol Frequency — how often to show symbols on the trace.
- 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 — click the check box to set the symbol display as a solid or as hollow.
- Symbol Arrows — click the check box to use arrows on the curve ends

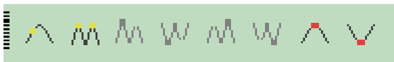
Note:

So curves with single points always appear, Box is the default symbol. None cannot be selected as the symbol.

3. Edit the properties of interest and OK the Properties window to apply the changes and close the window.

Adding Data Markers to 2D and Circuit Traces

The Reporter includes **Report 2D > Marker** menu commands and toolbar icons



that let you add markers to traces. A marker appears as “m*N*” at the marked point, where *N* increments from 1 as you place additional markers. Each marker can be selected and has editable properties including name, font, background and color. As you place markers, one or more marker legends may be displayed, depending on the **View > Active View Visibility** settings for the legends. The main marker legend appears in the upper-left of the plot, and lists the marker names and their X and Y values in a table. You can control the number format for the table values via the properties window, general tab. Under Marker/Other Number format, 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 open 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 a movable marker at the origin of the plot with a vertical line rising on the X axis. To move an X marker, click the X label and drag it to the appropriate 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. This marker is not cleared by the **Clear All** command, and must be deleted by selecting it and using the **Edit Delete** command.
- **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:

- **Marker** – this command lets you place a marker at an arbitrary point on a selected trace.
- **X Marker** – this command adds a movable marker at the origin of the plot with a vertical line rising on the X axis. To move an X marker, click the X label and drag it to the appropriate 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. This marker is not cleared by the **Clear All** command, and must be deleted by selecting it and using the **Edit Delete** command.
- **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.
- **Next Data Point** (Right) — moves a selected X marker to the next data point.
- **Previous Data Point** (Left) — moves a selected X marker to the previous 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 except X Markers.

Discarding 2D and Circuit Report Values below a Specified Threshold

To prevent real small numbers from skewing a plot, you can discard small values (under 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, drag 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 field is inactive if the Specify Discard Values check box is not enabled.
5. Click **OK** to apply the Discard Values to the report.

Add 2D or Circuit Trace Characteristics

Add or clear additional characteristics to a selected trace. To add additional characteristics to a selected trace:

1. Select a trace in a report plot or legend.
2. Click **Report 2D > Trace Characteristics**, or right-click the selected trace to open the shortcut menu.
3. Select **Trace Characteristics > Add**

This displays the **Add Trace Characteristics** window.

4. Select the **Category** and an associated Function to apply. The available categories depend on the plot, and Category enables the display of associated functions.

Category	**Functions for the Category**
Math	max, min, pk2pk, rms, avg, integ, integabs, avgabs, rmsAC, ripple, pkavg, XatYMin, XatYMax, XatYVal, cmplx()
PulseWidth	pulsefall9010, pulsefront9010, pulsefront3090, pulsemax, pulsemaxtime, pulsemin, pulsemintime, pulsetail50, pulsewidth5050, pw_plus, pw_plus_max, pw_plus_min, pw_plus_avg, pw_plus_rms, pw_minus_max, pw_minus_min, pw_minus_avg, pw_minus_rms
Overshoot, Undershoot	overshoot, undershoot.
TR & DC	crestfactor, formfactor, distortion, fundamentalmag, delaytime, risetime, deadtime, settlingtime,
Error	iae, ise, itae, itse
Period	per, pmax, pmin, prms
Radiation	xdb10bandwidth, xdb20bandwidth, lSidelobeX, lSidelobeY, rSidelobeX, rSidelobeY

Given a selected Function, and Category, the **Add Trace** window displays a field that explains the Purpose of the function. For a full list of functions and their definitions, see Selecting a Function.

5. Some categories and functions call for you to specify one or two additional values in a table. Save these values by clicking **Default**.
6. Click **Add** to add the specified characteristics to the Trace.

To remove existing trace characteristics:

1. Select a trace in a report plot or legend.
2. Click **Report 2D > Trace Characteristics**, or right-click the selected trace to open the shortcut menu.
3. Select **Trace Characteristics > Clear All**

Trace characteristics are clear on the selected trace.

Removing Traces from 2D and Circuit Reports

You can remove traces on the traces list in the following ways:

To *remove one trace* on the report:

- Select the trace you want to remove on the Project tree, then click **Delete**.

To *remove all traces* on the report:

- Select all the traces and click **Delete**.

Define Traces Using Range Functions (2D and Circuit)

Range Functions are special functions that use a 2D dataset as input, along with 0 or more additional parameters. Range functions can be used to produce a user-created 2D report display that is a collection of traces and their attributes. The trace collection and attributes are used to generate a portion of a report-definition that generates a family of curves in the report window.

Range Functions can be applied to a range (subset) of points on an X-Y plot, then calculates a single-number representation of the specified range, and display that number directly over a plotted wave-form. Range functions can be used to extract trace characteristics (such as max, min, overshoot) from a plot and use those values for additional plotting, or use the values for exporting to a file, where the data table formats are supported. Range Functions are listed in the **Reporter** window and **Output Variable** window for users to employ. Extracted trace characteristics may also be used in optimetrics.

For example, with a wave-form in a transient plot that contains a square pulse, a range function can calculate and display a single value that represents the High-to-Low/Low-to-High transition states. Or, with a plot that shows a pulse, the PulseWidth and RiseTime Range Functions can use the entire plotted curve to calculate a single number that represents the width or rise-time, and display that single-number representation directly over the specified range of plotted points.

- Range functions trace characteristics are displayed on the report window as a column in legend window
- Range functions are available from reporter and optimetrics
- Numerous range functions are available in the following categories:

— Pulse width functions

— Overshoot/Undershoot functions

— TR & DC functions

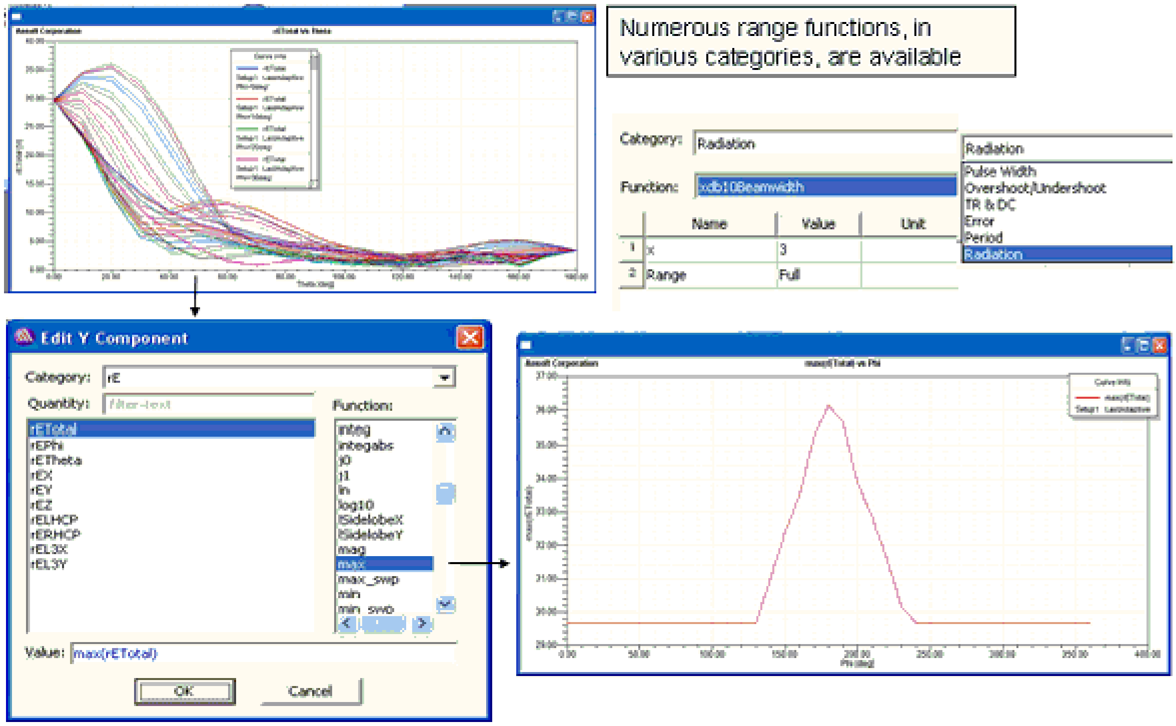
— Error functions

— Period functions

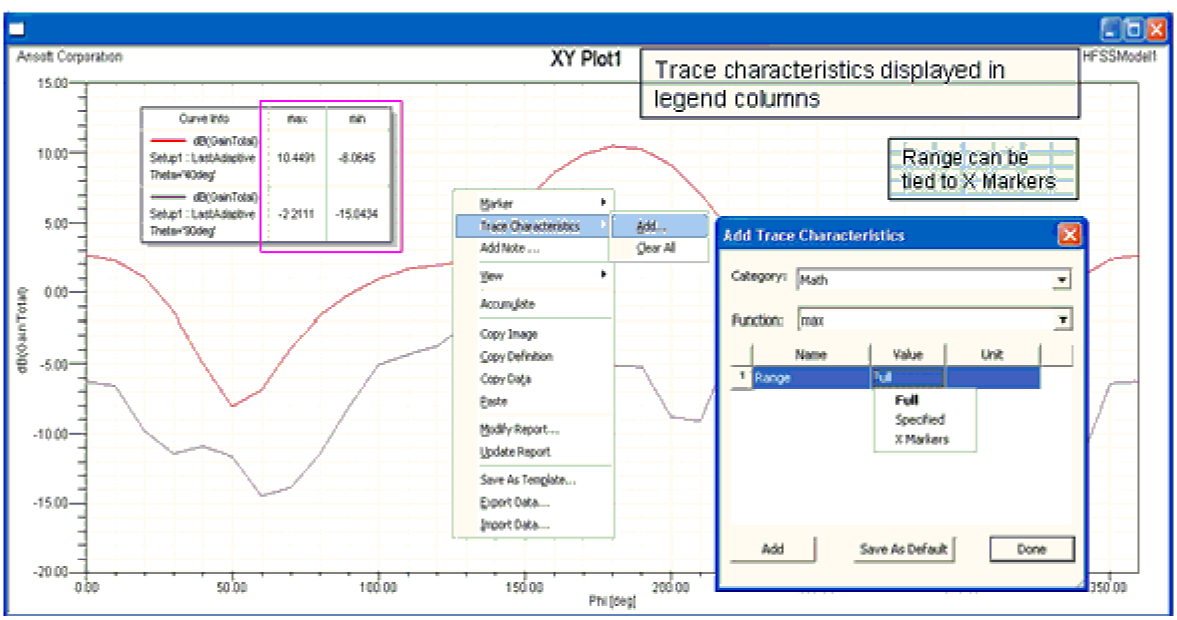
— Radiation functions

> **Note:**
>
> For all range functions, the second parameter is assumed to be a sweep variable, and if it is not, the following error message results: "**Second parameter to functions that apply to a range of values must be a sweep variable** ". You receives this message if you create a range function where the second parameter is a non-variable, such as deriv(dB(S11),dB(S12)). Whereas deriv(dB(S11),F) is acceptable.

Define Traces Using Range Functions



Trace Characteristics



The following table lists the available Range Functions:

Range Function	Category	Description	Parameter(s)
avg	Math	Average of first param over the second param.	N/A
avgabs	Math	Returns the mean of the absolute value of the selected quantity.	N/A
bandwidth	AC	Returns the bandwidth of the selected simulation quantity in Hz.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
cang_deg	TR & DC	Cumulative angle of the first param in degrees, along the second param (typically sweep variable). Returns double.	N/A
cang_rad	TR & DC	Cumulative angle of the first param in radians, along the second param (typically sweep variable). Returns double.	N/A
crestfactor	TR & DC	Returns the crest factor (peak/RMS) for the selected quantity.	N/A
deadtime	TR & DC	Obtains the latest time when the qtyl is within a tolerance of zero.	Tolerance:The +/- bandwidth around 0
delaytime	TR & DC	Obtains the time from zero to 50% of the target point.	Target: The target value for input
deriv	TR & DC	Derivative of a given param	N/A

distortion	TR & DC	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.	Frequency: Freq in Hz at which to calculate the RMS of the qty
duty	TR & DC	Duty cycle measurement.	N/A
eyeheight	TR & DC	Eye-diagram characteristics.	N/A
eyejitter	TR & DC	Eye-diagram characteristics.	N/A
eyewidth	TR & DC	Eye-diagram characteristics.	N/A
FFT	Math	Fast Fourier Transform. Returns output signal.	Incoming signal.
formfactor	TR & DC	Returns the form factor (RMS/Mean Absolute Value) for the selected quantity.	N/A
fundamentalmag	TR & DC	Returns the RMS value of the fundamental frequency for the selected quantity and an additional argument frequency, which specifies the fundamental frequency.	Frequency: Freq in Hz at which to calculate the RMS of the qty
gaincrossover	AC	Returns the gain crossover	Base Gain: Ref AC magnitude; Base

		frequency (where the gain is 0 dB) of the selected simulation quantity in Hz.	phase: Ref phase angle
gainmargin	AC	Returns the gain margin in dB at the phase crossover frequency of the selected simulation quantity.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
iae	Error	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument.	Target: Target value
integ	Math	Integral of the selected quantity. Uses trapezoidal area.	N/A
integabs	Math	Returns the integral of the absolute value of the selected qty.	N/A
ise	Error	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument.	Target: Target value
itae	Error	Returns the time-weighted absolute deviation of the selected quantity from a target value that is entered via an additional argument.	Target: Target value

itse	Error	Returns the time-weighted squared deviation of the selected qty from a target value that is entered via an additional argument.	Target: Target value
lowercutoff	AC	Returns the lower 3dB frequency of the selected simulation channel in Hz.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
max	Math	Maximum (of magnitudes).	N/A
min	Math	Minimum (of magnitudes).	N/A
overshoot	Overshoot/Undershoot (Overridable)	Calculates peak overshoot given a threshold value and number of evenly spaced points over entire time range.	Threshold: The reference value from where the overshoot/undershoot is calculated; Number of Points: Number of evenly spaced time points
peakgain	AC	Returns the peak gain of the selected simulation quantity in dB.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
peakgainfreq	AC	Returns the frequency in Hz at which the peak gain of the selected simulation quantity occurs.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
per	Period (Overridable)	Calculates period.	Threshold:Y transition value which determines the period; Number of Points: Number of evenly

			spaced time points
phasecrossover	AC	Returns the phase crossover freq, at which the phase is -180 degrees, in Hz for the selected simulation quantity.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
phasemargin	AC	Returns the phase angle in degrees at the gain crossover frequency of the selected simulation quantity.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
pk2pk	Math	Difference between max and min of the first param over the second param. Returns the peak-to-peak value for the selected simulation quantity.	N/A
pkavg	Math	Returns the ratio of the peak to peak-to-average for the selected quantity.	N/A
pmax	Period (Overridable)	Maximum period of input stream.	Threshold:Y transition value which determines the period; Number of Points: Number of evenly spaced time points
pmin	Period (Overridable)	Minimum period of input stream.	Threshold:Y transition value which determines the period; Number of Points: Number of evenly spaced time points
prms	Period (Overridable)	Rms of period of input stream.	Threshold:Y transition value which determines the period; Number of Points:

			Number of evenly spaced time points
pulsefall9010	Pulse Width	Returns the pulse fall time of the selected quantity derived from estimates of the signal at 90% and 10% of its peak value.	N/A
pulsefront1090	Pulse Width	Returns the pulse front time of the selected quantity derived from estimates of the signal at 10% and 90% of its peak value.	N/A
pulsefront3090	Pulse Width	Returns the pulse front time of the selected quantity derived from estimates of the signal at 30% and 90% of its peak value.	N/A
pulsemax	Pulse Width	Returns the pulse maximum on the front and tail estimates for the selected quantity.	N/A
pulsemaxtime	Pulse Width	Returns the time at which the maximum pulse value of the selected quantity is reached.	N/A
pulsemin	Pulse Width	Returns the pulse minimum on the front and tail estimates for the selected quantity.	N/A

pulsemintime	Pulse Width	Returns the time at which the minimum pulse value of the selected quantity is reached.	N/A
pulsetail50	Pulse Width	Returns the pulse tail time of the selected quantity on the virtual peak to 50%.	N/A
pulsewidth5050	Pulse Width	Returns the pulse width of the selected quantity as measured on the 50% points on the pulse front and pulse tail.	N/A
pw_minus	Pulse Width (Overridable)	Pulse width of the first negative pulse.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_minus_avg	Pulse Width (Overridable)	Average of the negative pulse width input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_minus_max	Pulse Width (Overridable)	Max. Pulse width of the negative pulse of input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_minus_min	Pulse Width (Overridable)	Min. Pulse width of the negative pulse of input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of

			evenly spaced time points
pw_minus_rms	Pulse Width (Overridable)	RMS of the negative pulse width input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_plus	Pulse Width (Overridable)	Pulse width of first positive pulse.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_plus_avg	Pulse Width (Overridable)	Average of the positive pulse width input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_plus_max	Pulse Width (Overridable)	Max. Pulse width of the positive pulse of input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_plus_min	Pulse Width (Overridable)	Min. Pulse width of the positive pulse of input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time points
pw_plus_rms	Pulse Width (Overridable)	RMS of the positive pulse width input stream.	Threshold:Y transition value which determines the pulse width ; Number of Points: Number of evenly spaced time

			points
ripple	Math	Returns the ripple factor (AC RMS/Mean) for the selected quantity.	N/A
risetime	TR & DC	Obtains the time taken to go from 10% to 90% of target point.	Target: The target value for input
rms	Math	Returns total RMS of the selected quantity.	N/A
rmsAC	Math	Returns the AC RMS for the selected quantity	N/A
settlingtime	TR & DC	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.	Target; Tolerance
undershoot	Overshoot/Undershoot (Overridable)	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.	Threshold: The reference value from where the overshoot/undershoot is calculated; Number of Points: Number of evenly spaced time points
uppercutoff	AC	Returns the upper 3dB frequency of the selected simulation channel in Hz.	Base Gain: Ref AC magnitude; Base phase: Ref phase angle
XAtYMax	Math	Returns the X value at maximum Y.	N/A
XAtYMin	Math	Returns the X value	N/A

		at minimum Y.	
XAtYVal	Math	Returns the X value at the first occurrence of Y value.	Y Value: Y value at which to find X

Copy and Paste of 2D and Circuit 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 is evaluated in 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 (e.g., a port name) an error is posted that indicates a solution does not exist for this trace.

> **Note:** 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:

From the **Project Manager** window, expand the **Results** folder. Then right-click the report and select **Copy Definition**.

To Paste a Report Definition:

From the **Project Manager** window, right-click **Results** and select **Paste**. A new report is created and it contains the copied definitions.

To Copy an individual Trace Definition(s):

From the **Project Manager** window, expand the **Results** folder and the report. Then right-click the trace or traces and select **Copy Definition**.

To paste the Trace Definition(s):

Right-click the report in the target design to copy the trace or traces and select **Paste**.

A new trace(s) is added to the report and it contains the copied trace definition(s).

> **Note:**
>
> If you copy and paste a report or trace definition to a design which contains a definition with the same name, then an incremented number is appended to the pasted report or trace name.

Copy and Paste 2D and Circuit 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 in 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 Manager** window and select **Copy Data**, or use the menu bar **Edit > Copy Data**, or right-click within a plot to display and select **Copy Data**.

To paste copied report data:

Right-click **Results** in the **Project Manager** window 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 Manager** window and select **Copy Data**.

To paste copied trace data:

Right-click the report in the target design to copy the trace data to 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 is appended to the pasted name.

Variables, Quantities, and Functions in 2D and Circuit Reports

The follow topics describes how to sweep variables, and select various quantities and functions for use in specifying values to be displayed in plots and reports.

Sweeping a Variable in a 2D or Circuit Report

In Ansys Electronics Desktop, 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.

To specify the swept variable values to plot a selected quantity against:

1. In the **Report** window, select the variable on the X (Primary Sweep) drop-down menu.
2. To modify the values that is plotted for a variable:
   a. Click **[...]** on the **X (Primary Sweep)** line of the **Report** window to display a list of the possible values.
   b. Select **All Values** or click **Edited** to display a window that lets you specify the sweeps to use.

   All of the selected variable's values is plotted.

2D and Circuit Selecting a Function

The value of a quantity being plotted depends upon its mathematical function, which you select on the**Trace** tab **Function** group box in the **Report**window. 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").

The following functions listed can be applied as **Range Functions** to previously specified Quantities and Functions using the **Set Range Function** window. Some of these functions can operate along an entire curve (min, max, integ, avg, rms, pk2pk, cang_deg, cang_rad). Their syntax is as follows:

- cang_deg(*quantity*) implicitly implies derivative over the primary sweep
- cang_deg(*quantity*, *SweepVariable*) explicitly means derivative over the sweep variable specified in the second argument (such as "Freq")

> **Note:** For all range functions, the second parameter is assumed to be a sweep variable, and if it is not, the following error message results: "**Second parameter to functions that apply to a range of values must be a sweep variable** ". You receive this message if you create a range function where the second parameter is a non-variable, such as deriv(dB(S_{11}),dB(S_{12})). Whereas deriv(dB(S_{11}),F) is acceptable.

You can select on the following functions in the **Trace** tab **Function** group box:

abs	Absolute value
acos	Arc cosine
acosh	Hyperbolic arc cosine
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
asin	Arc sine
asinh	Hyperbolic arc sine
atan	Arc tangent
atanh	Hyperbolic arc tangent
avg	Average of first parameter over the second parameter
avgabs	Absolute value of average.
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180.
cang_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.
cmplx	Complex number with the first parameter as the real part and the second parameter as the imaginary part.
conjg	Conjugate of the complex number.
cos	Cosine
cosh	Hyperbolic cosine
crestfactor	Peak/RMS (root mean square) for the selected simulation quantity
dB(x)	20*log10(|x|)
dBm(x)	10*log10(|x|) +30
dBW(x)	10*log10(|x|)
db10normalize	10*log [normalize(mag(x))]
db20normalize	20*log [normalize(mag(x))]
deriv	Derivative of a given parameter.
even	Returns 1 if integer part of the number is even; returns 0 otherwise
exp	Exponential function (the natural anti-logarithm)
formfactor	Returns root mean square RMS/Mean Absolute Value for the selected simulation quantity.
iae	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument. To use this function, you need to open the Add Trace Characteristics window and select the Error category.
im	Imaginary part of the complex number
int	Truncated integer function

integ	Integral of the selected quantity. Uses trapezoidal area.
integabs	Absolute value of integral.
ise	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics window and select the Error category.
itae	Returns the time-weighted absolute 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 window and select the Error category.
itse	Returns the time-weighted squared deviation of the selected qty from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics window 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)
ln	Natural logarithm
log10	Logarithm base 10
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	Maximum of magnitudes.
max_swp	Maximum value of a sweep.
min	Minimum magnitudes.
min_swp	Minimum value of a sweep.
nint	Nearest integer
normalize	Divides each value within a trace by the maximum value of the trace. ex. normalize(mag(x))
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise
overshoot	Obtains the peak overshoot over a point (double argument)
per	Calculates period.
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	Period max.
pmin	Period minimum
prms	Period Root Mean Square.
pulsefall9010	Pulse fall time of the selected simulation quantity according to the 90%-10% estimate.
pulsefront9010	Pulse front time of the selected simulation quantity according to the 10%-90% estimate.
pulsefront3090	Pulse front time of the selected simulation quantity according to the 30%-90% estimate.
pulsemax	Pulse maximum on the front and tail estimates for the selected simulation quantity.
pulsemaxtime	Time at which the maximum pulse value of the selected simulation quantity is reached.
pulsemin	Pulse minimum on the front and tail estimates for the selected simulation quantity.
pulsemintime	Time at which the minimum pulse value of the selected simulation quantity is reached.
pulsetail50	Pulse tail time of the selected simulation quantity on the virtual peak to 50%.
pulsewidth5050	Pulse width of the selected simulation quantity as measured on the 50% points on the pulse front and pulse tail.
PulseWidth Functions	
pw_plus	Pulse width of first positive pulse
pw_plus_max	Max. Pulse width of input stream
pw_plus_min	Min. Pulse width of input stream
pw_plus_avg	Average of the positive pulse width input stream
pw_plus_rms	RMS of the positive pulse width input stream
pw_minus_max	Max. Pulse width of input stream
pw_minus_min	Min. Pulse width of input stream
pw_minus_avg	Average of the negative pulse width input stream
pw_minus_rms	RMS of the negative pulse width input stream
polar	Converts the complex number in rectangular to polar
re	Real part of the complex number
rect	Converts the complex number in polar to rectangular
rem	Fractional part
ripple	Returns the ripple factor (AC RMS/Mean) for the selected quantity.

rms	Returns total root mean square of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
rsidelobex	The ‘x’ value for the right side lobe: the next highest value to the right of the max value.
rsidelobey	The ‘y’ value for the right side lobe: the next highest value to the right of the max value.
sgn	Sign extraction
sin	Sine
sinh	Hyperbolic sine
sqrt	Square root
tan	Tangent
tanh	Hyperbolic tangent
Undershoot	Obtains the peak undershoot over a point (double argument).
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
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 window 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 window and select the Radiation category.
y0	Bessel function of the second kind (0^{th} order)
y1	Bessel function of the second kind (1^{st} order)

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, S_{11} say, 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 so by any expression. If the complex S_{11} lies on the positive Y axis say, ang_deg_val(S_{11}) is 90 and "1 + ang_deg_val(S_{11})" is 91.

Selecting Solution Quantities to Plot (2D and Circuit)

When you create a report of Modal or Terminal solution data, 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 **Electronics Desktop** such as S_{11}, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or theta. The valid categories available depend on the type of quantity (real or complex) that is being plotted, the setup, the solution type, and the plot domain.

To select an S-parameter quantity to plot:

1. In the **Report** window, **Trace** tab, select one of the following categories:

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 on the original field solution.
S-parameter	S-parameters on the S-matrix. For designs which include a Frequency Selective Surface (FSS)-referenced radiation boundary, S11 and S21 represent the extracted reflection and transmission coefficients, respectively.
Y-parameter	Admittance matrix parameters computed on the S-parameters and port impedances.
Z-parameter	Impedance matrix parameters computed on the S-parameters and port impedances.
Gamma	Propagation constants for the S-parameters.
Port Z_o	Characteristic port impedances.
VSWR	Voltage standing wave ratio, calculated on the equation $\frac{1+\lvert S_{ij}\rvert}{1-\lvert S_{ij}\rvert}$.

2. Select a quantity to plot on the **Quantity** group box. The available quantities depends upon the selected category and the setup of the design.

Selecting a Field Quantity to Plot (2D and Circuit)

When plotting field quantities, the quantity can be a value that was calculated by **Electronics Desktop** 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."
2. In the **Report** window, select Geometry for the Context, unless you are plotting scalar (e.g., integration).
3. In the **Report** window, select one of the following categories:

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 on the original field solution.
Calculator	Includes scalar and vector field quantities automatically calculated by

Expressions	Ansys Electronics Desktop, 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 on the **Quantity** group box. The available quantities depends upon the selected category and the setup of the design.

2D and Circuit Selecting a Far-Field Quantity to Plot

When plotting far-field quantities, the quantity can be a value that was calculated by **Electronics Desktop** such as antenna gain, a value from a calculated expression, or an intrinsic (inherent) variable value such as frequency or theta.

To select a far-field quantity to plot:

1. When you create the report, specify the Report Type as "Far Fields."
2. In the **Report** window, select one of the following categories for the field setup:

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 on the original field solution.
Gain	Gain is 4π times the ratio of an antenna's radiation intensity in a given direction to the total power accepted by the antenna.
Axial Ratio	Axial ratio of the electric field.
Antenna Params	Ansys Electronics Desktop-calculated quantities that include peak directivity, radiated power, accepted power, radiation efficiency, max U, and array factors. For far-field setups, the decay factor for lossy materials is calculated as a constant for all far fields.
Radar Cross-Section (Bistatic RCS)	The radar cross-section (RCS) or echo area (σ) is measured in meters squared and represented for a bistatic arrangement (i.e., when the transmitter and receiver are in different locations). This is represented by: $$\sigma = \frac{4\pi r^2 \lvert E_{\mathrm{scat}} \rvert^2}{\lvert E_{\mathrm{inc}} \rvert^2}$$ where: • E_{scat} is the scattered E-field. • E_{inc} is the incident E-field. *RCS is supported for designs with Plane Incident Waves. RCS is not supported for other types of incident waves.*

2D and Circuit Directive Gain

Directive Gain is the ratio of the radiation intensity at a specified observation angle to the average radiation intensity. For the observation angle (q,f), the directivity is:

$$D(\theta,\phi) = \frac{U(\theta,\phi)}{U_{ave}} = \frac{U(\theta,\phi)}{\frac{1}{4\pi}\oiint U(\theta,\phi)d\Omega}$$

where U is the radiation intensity, which is defined as:

$$U(\theta,\phi) = \frac{1}{2}\mathrm{Re}(\vec{E}\times\vec{H})\cdot r^2\hat{r}$$

The average radiation intensity is related to the radiated power by:

$$U_{ave} = \frac{P_r}{4\pi}$$

Directivity

Directivity is the maximum value of Directive Gain, or:

$$D = \frac{U_{\max}}{U_{ave}}$$

Reference

Warren Stutzman and Gary Thiele, *Antenna Theory and Design*, Wiley, 1981.

2D and Circuit Radiation Efficiency

The radiation efficiency is the ratio of the radiated power to the accepted power given by:

$$e = \frac{P_{\mathrm{rad}}}{P_{\mathrm{acc}}}$$

where:

- P_{rad} is the radiated power in watts
- P_{acc} is the accepted power in watts

> **Note:** Because the radiation efficiency is calculated on the accepted power, a port must be defined for radiation efficiency to be displayed.

Setting a Range Function (2D and Circuit)

To apply a range function to the Y, Z, or Mag component of a trace:

1. Click **Range Function** in the **Reports** window.

   This opens the **Set Range Function** window. The functions available are the same as described in Selecting a Function.

2. Click the Specified radio button on the Range function line.

   This enables the Range Function fields.

3. Select the **Category**, then an associated Function to apply. The available categories depend on the plot, and Category enables the display of associated functions.

Category	Functions for the Category
Math	max, min, pk2pk, rms, avg, integ, integabs, avgabs, rmsAC, ripple, pkavg, XatYMin, XatYMax, XatYVal
PulseWidth	pulsefall9010, pulsefront9010, pulsefront3090, pulsemax, pulsemaxtime, pulsemin, pulsemintime, pulsetail50, pulsewidth5050, pw_plus, pw_plus_max, pw_plus_min, pw_plus_avg, pw_plus_rms, pw_minus_max, pw_minus_min, pw_minus_avg, pw_minus_rms
Overshoot, Undershoot	overshoot, undershoot.
TR & DC	crestfactor, formfactor, distortion, fundamentalmag, delaytime, risetime, deadtime, settlingtime,
Error	iae, ise, itae, itse
Period	per, pmax, pmin, prms
Radiation	xdb10bandwidth, xdb20bandwidth, lSidelobeX, lSidelobeY, rSidelobeX, rSidelobeY
Eye Measurements	EyeLevelZero, EyeLevelOne, EyeAmplitude, EyeHeight, EyeSignalToNoise, EyeOpeningFactor, EyeWidth, EyeJitterP2P, EyeJitterRMS, EyeRiseTime, EyeFallTime

Given a selected Function, and Category, the **Set Range Function** window displays a field that explains the Purpose of the function. For a full list of functions and their definitions, see Selecting a Function.

Selecting a function causes the display of a description in the **Purpose** field. If the function requires a value (such as the XatYVal Math function or the pw_minus_max Pulse Width function), the following table the function field displays the name, editable value field, unit, and description.

The Eye Measurement functions have five editable parameters.

4. Click the **Over Sweep** drop-down menu to select from available sweeps.
5. To select from available Sweeps, or to edit them, click **[...]** and uncheck **Use All Sweeps**.

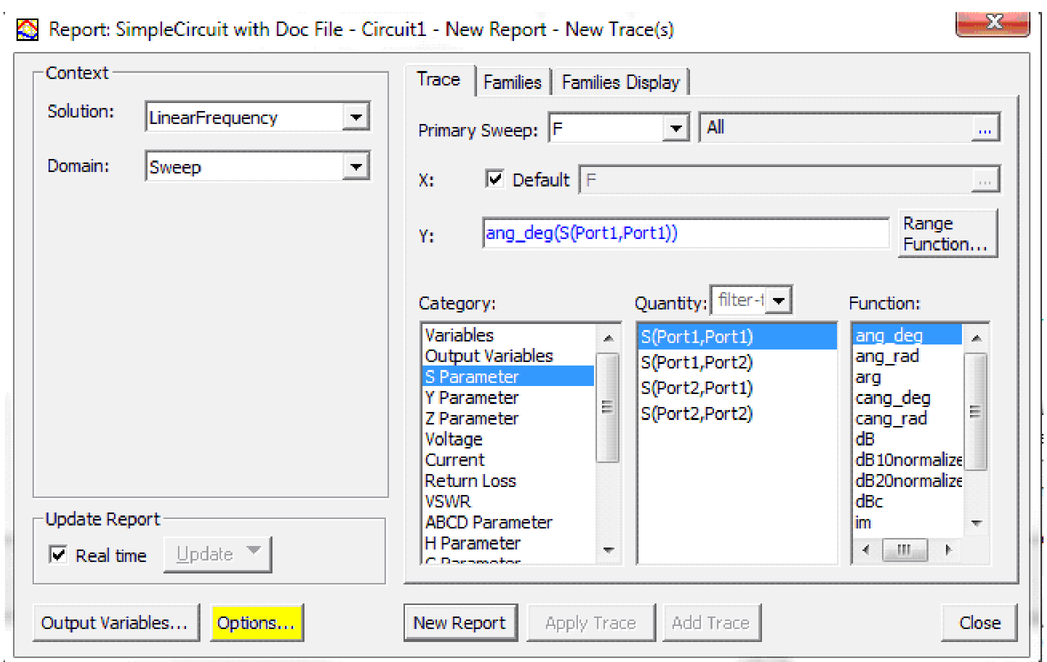
This enables a list of the sweeps. The sweep(s) you select is displayed on the Over Sweep line. You can use the buttons to **Clear All Selections** or **Select All** sweeps.

6. Select the Sweeps **Default** or **Edited** buttons to specify whether to accept the default or edited sweeps.
7. To edit the sweeps further, select **[...]** to open the **Edit Sweep** window.

For frequency variables, this lets you specify a single value, linear step, linear count, decade count, or octave 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.

Handling a Large Number of 2D or Circuit Ports



The tree view is only displayed when the following environment variables are configured.

- ANSYS_ENABLE_LARGE_PORTS = 1

  Required for Matrix parameters to display in tree view when the number of ports > 10.
- ANSYS_DISABLE_TREE_LIMIT = 1

  Required for Matrix parameters to display in tree view when the number of ports <= 10.

Working with Eye Diagrams

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. For more information about setting up eye diagram report types, see the following topics:

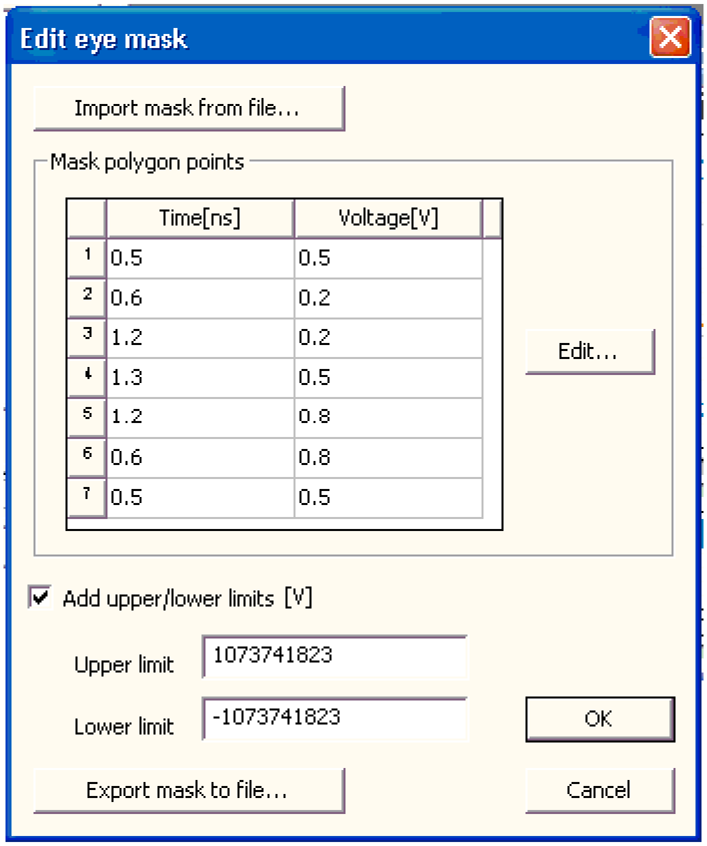
- Eye Diagram Report
- Stacked Eye Diagram Report
- Statistical Eye Diagram Report

Modifying Circuit Eye Diagram Reports

Double-click the Report to open a Property window with several tabs.

- Click the **General** tab to set the background color, plot area, and other general parameters.
- Click the **Mask** tab to enable and define a mask or keep-out region that is overlaid on the eye diagram. Click **Show Mask** to enable the edit fields. You can specify the fill color, boundary color, and transparency.

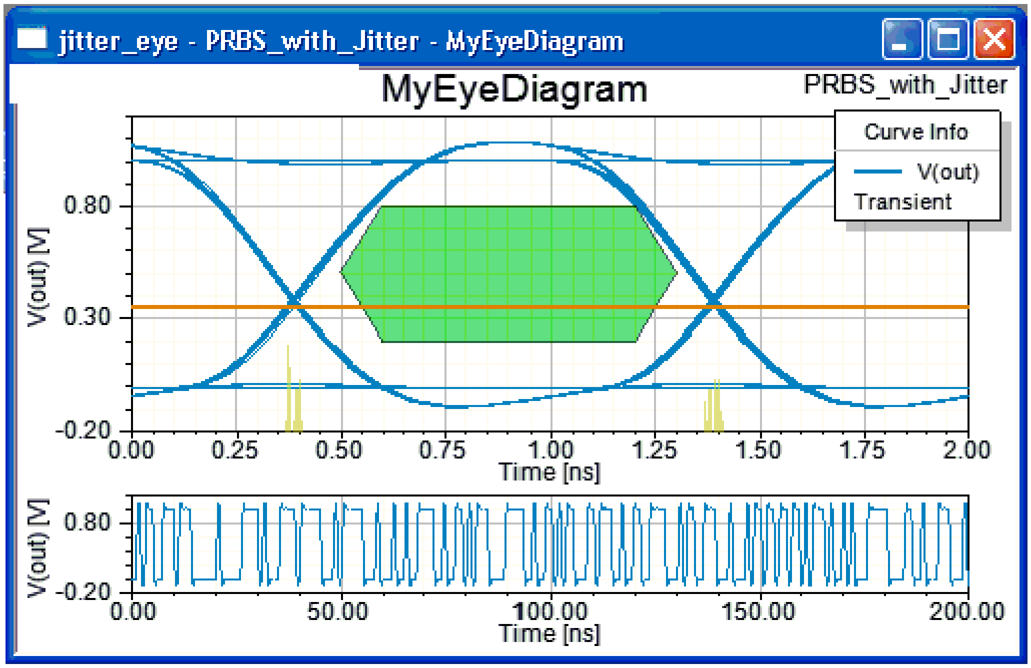
Click **Edit** on the **Mask** tab to open the Edit Eye Mask window:



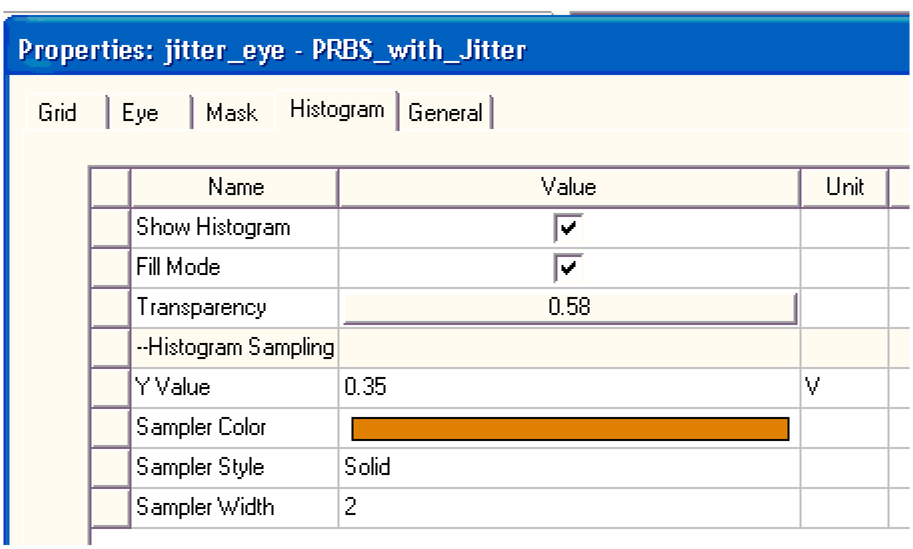
Use the window to specify edit points that define the mask polygon and upper and lower limits as applicable. If appropriate, save the mask setting to a file.

Click **OK** to open the mask on the eye diagram.

- Here is how the mask appears on the graph:



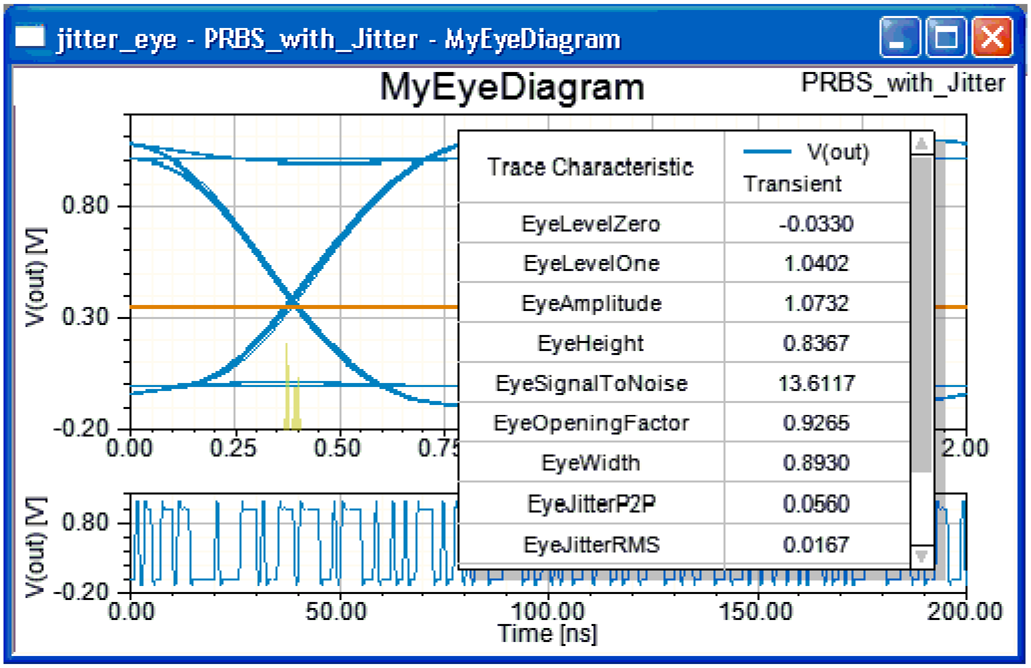
- Click the **Histogram** tab to control the display of a histogram that counts the number of crossings of a specified Y-value. In the illustration above, the histogram is in yellow and the Y-value is the orange line. Here is the **Properties** page showing the histogram setup fields:



Adding Trace Data to the Eye Diagram Report

Right-click the Report and select **Trace Characteristics > Add** to open the **Trace Characteristics** window. Select a **Category** of data, then select a **Function** within that category. Click **Add**. The value of the trace characteristic is added to the report display.

Right-click the Report and select **Trace Characteristics > AddAll Eye Measurements** on the menu to open the trace measurements in tabular format. Here is the result:



Eye Diagram References

A tutorial on interpreting eye diagrams may be found at:

`http://www.complextoreal.com/chapters/eye.pdf`

Other references on eye diagrams may be found at:

`http://www.scientificarts.com/logo/logos.html`

`http://www.findarticles.com/p/articles/mi_m0HPJ/is_n4_v45/ai_16226058`

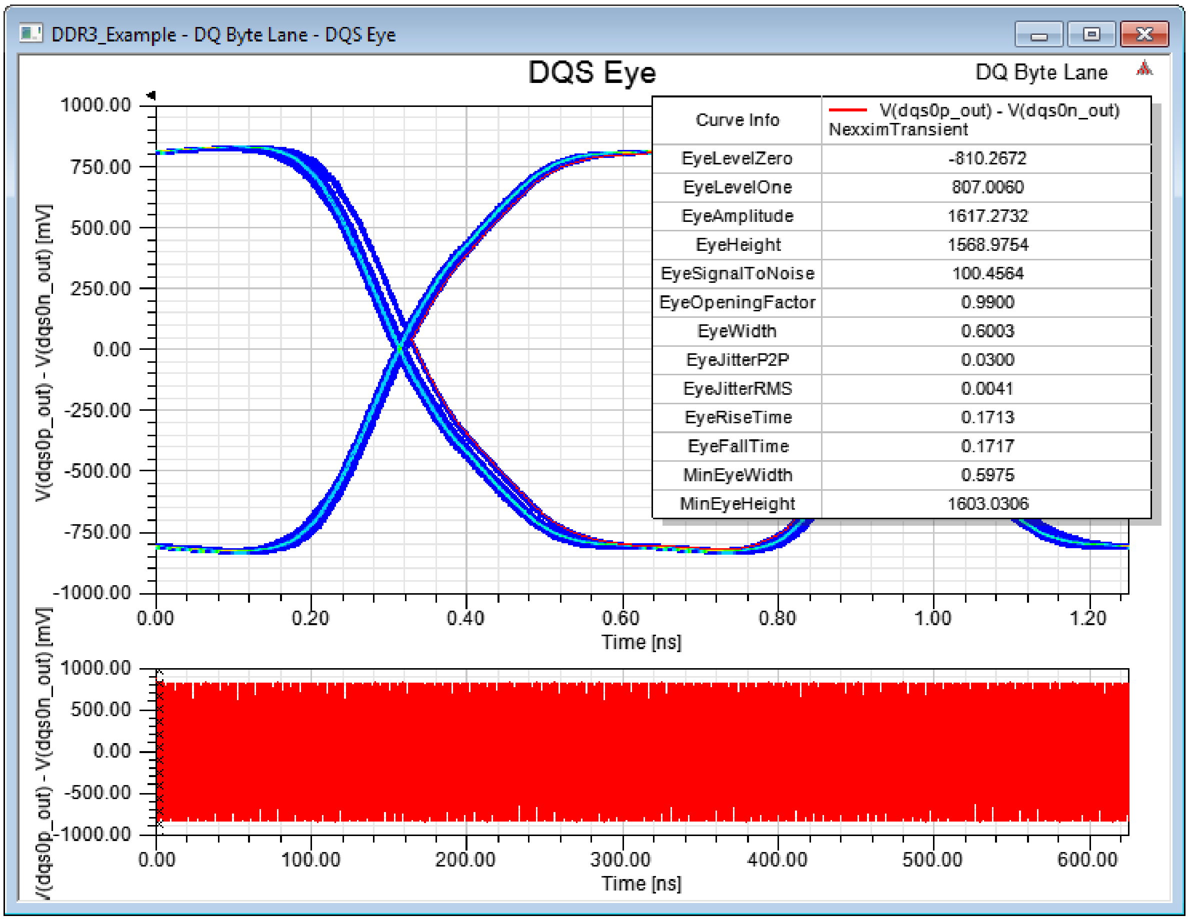
Eye Measurements (2D and Circuit)

Several standard eye-diagram measurements are provided in **Electronics Desktop**. The following eye measurements are supported:

- EyeLevelZero
- EyeLevelOne
- EyeAmplitude

- EyeHeight
- EyeSignalToNoise
- EyeOpeningFactor
- EyeWidth
- EyeJitterP2P
- EyeJitterRMS
- EyeRiseTime
- EyeFallTime
- MinEyeWidth
- MinEyeHeight

You can right-click the report and select **Trace Characteristics > AddAll Eye Measurements** to display all the eye measurements in tabular format.

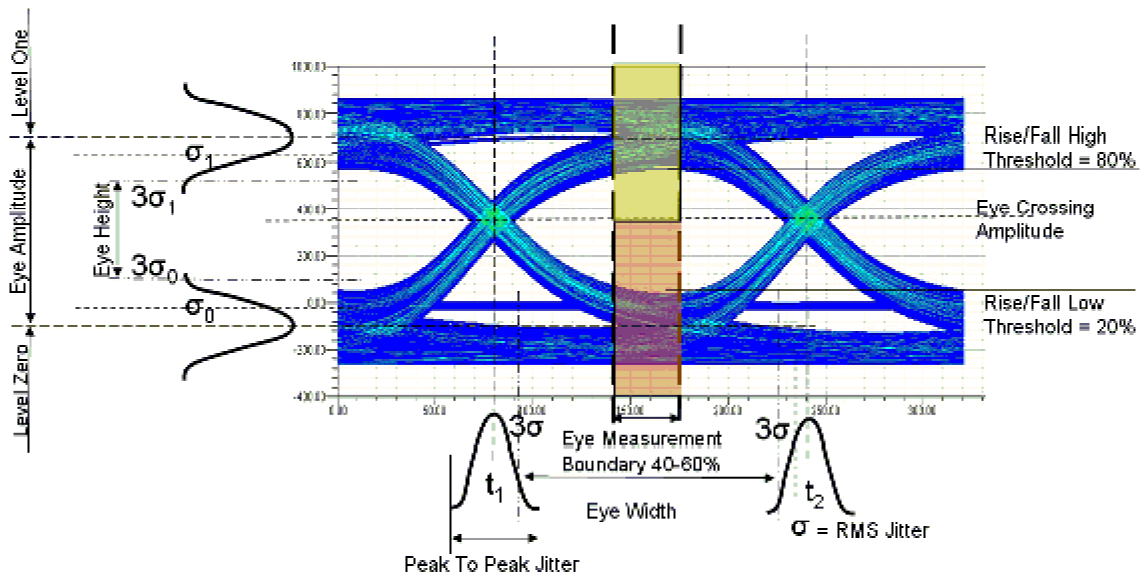


To add an eye measurement, right-click the eye diagram and select one of the following:

- Select **Trace Characteristics/Add** to add eye measurements individually
- Select **Trace Characteristics/Add All Eye Measurements** to add all available eye measurements

To remove an eye measurement, right-click the eye diagram:

- Select **Trace Characteristics/Clear All**



Values in the Eye Measurements Table are Calculated as Follows:

- **EyeLevelZero** and **EyeLevelOne** are the mean values of the lower/upper vertical histograms in the 40-60% center region of the eye, where "center" is defined by the eye-crossing points t1 and t2. Vertical histograms are created above or below the eye-crossing point (represented as yellow/orange shaded regions in the preceding figure). The mean value of these histograms yields EyeLevelZero and EyeLevelOne.
- **EyeAmplitude** = (EyeAmplitude = EyeLevelOne - EyeLevelZero). This is the voltage value for the eye crossing point, which is computed by creating a vertical histogram at the eye crossing points. Eye Crossing Amplitude is computed as the mean of the histogram. A front group box eye diagram has two eye crossings. Eye Crossing Amplitude is calculated as being the mean of these crossing amplitudes, and is then used to compute other horizontal eye measurements, such as eye width, RMS jitter, and peak-to-peak jitter.
- **EyeHeight** = [(EyeLevelOne - $3\sigma_1$) - (EyeLevelZero + $3\sigma_0$)] where σ_1 and σ_0 are the standard deviations of the vertical histograms used to determine EyeLevelZero and EyeLevelOne.
- **EyeSignalToNoise** = [EyeLevelOne - EyeLevelZero] / ($\sigma_1 + \sigma_0$)
- **EyeOpeningFactor** = [EyeAmplitude - $\sigma_1 - \sigma_0$] / EyeAmplitude

- **EyeWidth** = [(t_2 - 3σ) - (t_1 + 3σ)] where σ is the standard deviation of the horizontal histograms used to determine eye-crossing points.
- **EyeJitterP2P** is the width of the horizontal histogram across the eye-crossing point.
- **EyeJitterRMS** is the standard deviation of the horizontal histogram across the eye-crossing point.
- **EyeRiseTime** and **EyeFallTime** are the differences in mean values of the horizontal histograms centered on the 20% and 80% threshold values. Horizontal histograms are created at the 20% and 80% amplitude thresholds for both rising and falling edges. The difference in the mean values of the histograms for rising edges yields EyeRiseTime, while the difference in mean values for falling edges yields EyeFallTime.
- **MinEyeWidth** is the minimum horizontal opening at the eye-crossing amplitude, typically at the center of the eye. Unlike statistical eye width, if different traces exist for different unit intervals, then MinEyeWidth represents the minimum width of all the traces.
- **MinEyeHeight** is the minimum vertical opening at the eye-measurement point, typically at the center of the eye. Unlike statistical eye height, if different traces exist for different unit intervals, then MinEyeHeight represents the minimum height of all the traces.

Additional Values Not Listed in the Eye Measurements Table:

- **AutoDelay** is the amount of time required to shift the eye diagram so it is centered. You can control this delay either by specifying Manual Delay or allow the system to automatically compute the delay via Auto Delay.
- **AutoComputeCrossAmplitude**

  With Auto Mode “on”, the eye width is calculated as the average of the two peaks of the vertical histogram. This is shown as the "Eye Crossing Amplitude" in the preceding figure. Both statistical and minimum eye widths is calculated at this amplitude.

  With Auto Mode "off", the user can manually set the crossing point voltage to raise/lower where the eye width calculations are performed. Both statistical and minimum eye widths is calculated at this amplitude. The user should enter the voltage value where he thinks the actual crossing point lies--that is, where eye width should be computed (both statistical and MinEyeWidth).
- **AutoComputeEyeMeasurementPoint**

  With Auto Mode “on”, the eye height is calculated as the average of the two eye-crossing points, t1 and t2, as shown in the preceding figure. This point is the Eye Measurement Point. Both statistical and minimum eye heights is calculated at this time point.

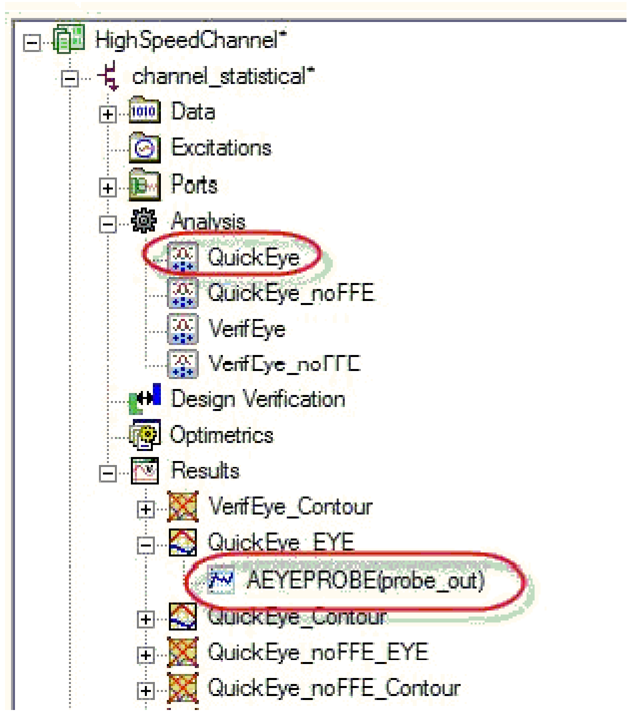
  With Auto Mode "off", the user can manually set the crossing point time in order to shift where the eye height calculations are performed. Both statistical and minimum eye heights is calculated at that time point.

Auto Delay does not affect eye height if the Eye Measurement Point is selected according to the eye diagram that is currently displayed. For example, if the UI is 1ns, the front group box eye is drawn over 2ns. With Auto Delay “off”, assuming that the first crossing is at 0.3ns and the second is at 1.3ns, the user should enter 0.8ns as the eye measurement point. With Auto Delay “on”, the first crossing is shown at 0.5ns and second is shown at 1.5ns. In this case, the corresponding eye measurement approximation point is 1ns.

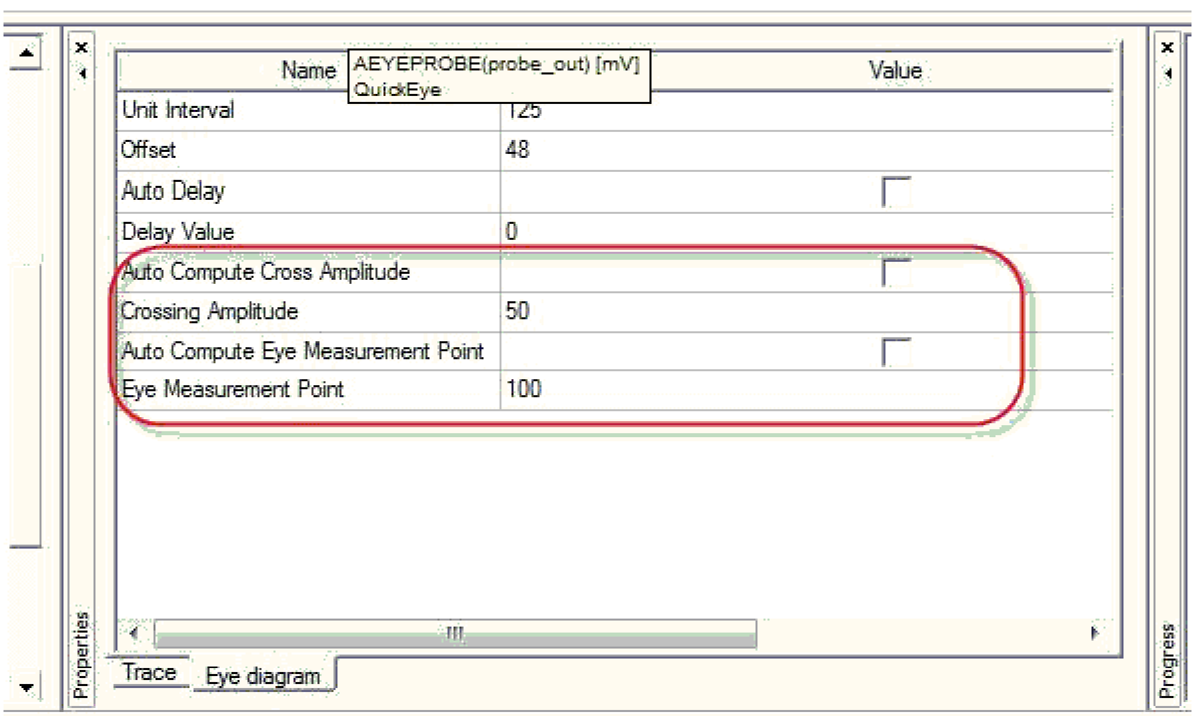
- **EyeCrossingTime** is calculated by creating a horizontal histogram across a small, narrow strip at the middle of the eye (eye-crossing points, t1 and t2). Eye Crossing Time is the mean of the histogram. The middle of the eye is computed as the middle of the vertical extremities of the eye--that is, the midpoint of max & min voltage values across the complete eye diagram. Later, an internal algorithm refines the crossing times by re-computing the crossing points based on a more accurate amplitude. The refined crossing time calculation is then used as the new middle of the eye. A front group box eye diagram has two eye crossings and two Eye Crossing Times.
- **EyeMeasurementPoint** (time) is calculated as the mean of the two eye crossing times in a front group box eye diagram. This value is then used to compute several vertical eye measurements, such as eye height, eye amplitude, level one, and level zero, etc. If Auto Mode is “off”, the user should enter the time value where he thinks the actual vertical eye measurements should be conducted (at present, only minimum eye height is measured).

  Auto Delay does not affect this value if the user follows the eye diagram that is currently displayed. Consequently, the user should enter the diagram time value where he thinks the eye measurement point actually lies. For example, let 1ns be the unit interval (UI) and assume the front group box eye is drawn over 2ns. With Auto Delay “off”, if the first crossing is at 0.3ns and the second is at 1.3ns, the user should enter 0.8ns as the approximate eye measurement point. With Auto Delay “on”, the first crossing is shown at 0.5ns and the second is shown at 1.5ns, so the approximate eye measurement point is 1ns.

The eye measurements described above can be accessed by analyzing **QuickEye** simulation in HighSpeedChannel.adsn (Nexxim standard examples) and putting your cursor on **AEyeProbe**.



The two check boxes are in the **Eye diagram** tab of the trace properties:



Eye Measurement Range Function Parameters (2D and Circuit)

The Eye Measurement category of range functions provide the means to evaluate several characteristics of eye diagrams. Each of the Eye Measurement functions includes the following parameters. Specify the Value by editing the Value field for the parameter name. Specify the unit for the parameter by selecting from the Unit drop-down menu.

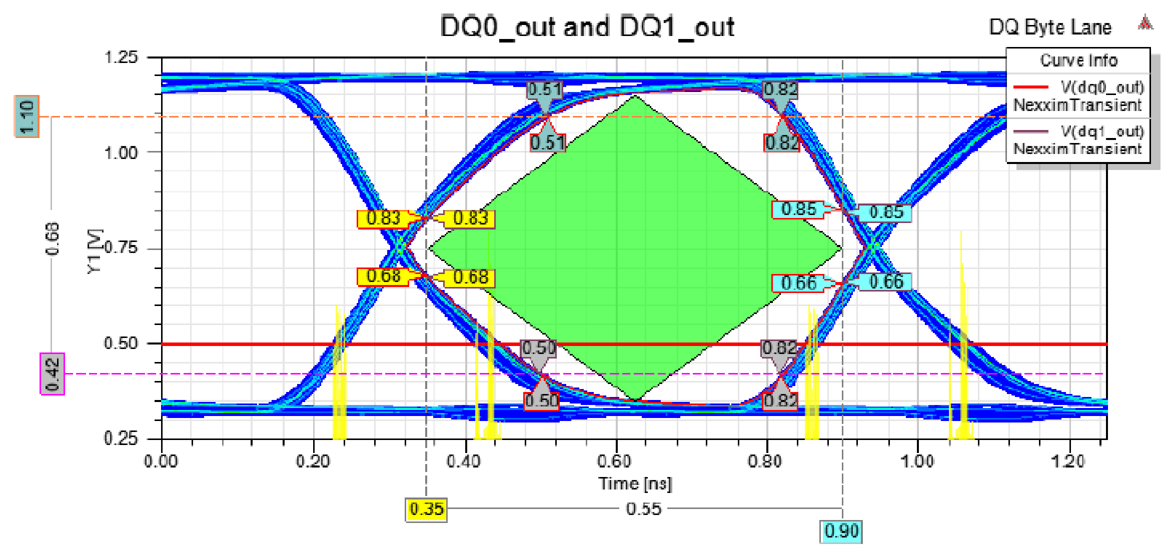
Name	DefaultValue	Default Unit	Description
Unit Interval	0	ns	Unit interval of signal
Start Offset	0	ns	Offset at beginning of signal
End Offset	0	ns	Offset at end of signal
Autocrossing amplitude	1		Nonzero number means that crossing amplitude is calculated automatically.
Crossing amplitude	0	mV	Specify crossing amplitude used for eye measurement data computation.

Using Stacked Eye Diagrams

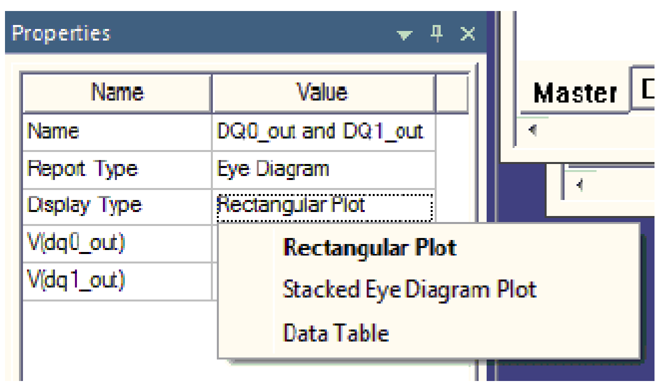
The Rectangular Eye Diagram is optimized for viewing a single eye. But when multiple eyes are overlaid in the same Rectangular Eye Diagram, it is not very easy to compare the eyes. It is also very tedious to compare Y-marker and X-marker intersections. The Stacked Eye Diagram feature allows multiple eye plots to be stacked vertically for easy comparison. For details on the setup, see Stacked Eye Diagram.

Converting a Rectangular Eye Diagram to Stacked Eye Diagram

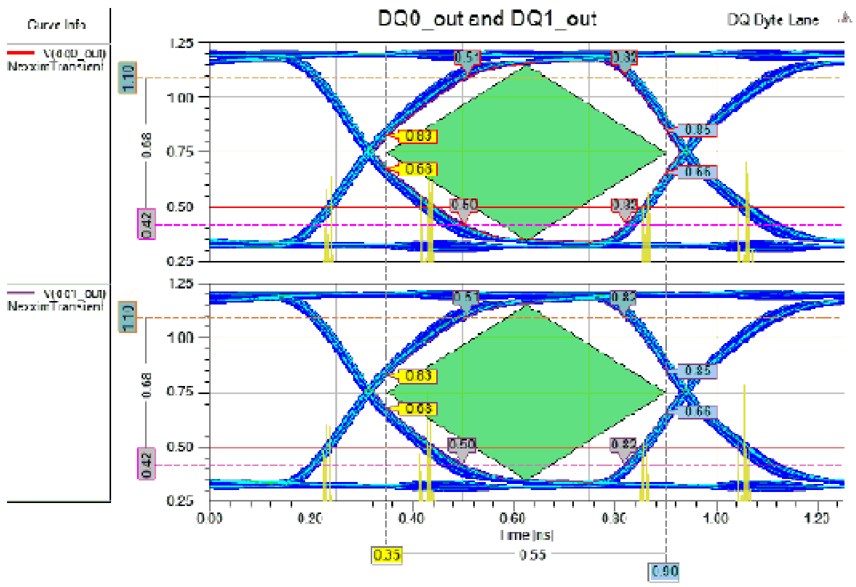
The following Rectangular Eye Diagram has two eyes. This is not clearly evident from this plot, except on the legend, and hence the two eyes cannot be compared easily.



To convert this plot to a Stacked Eye Diagram, select the plot in the Results folder in the **Project Manager** window and change its **Display Type** property to **Stacked Eye Diagram Plot** as shown in the following screenshot.



After conversion, the Eye diagrams are stacked as in the following view:



Only Rectangular Eye Diagram display types can be converted to Stacked Eye Diagrams. Statistical Eye Diagram Plots cannot be converted to Stacked Eye Diagrams.

In a Stacked Eye Diagram all the eyes are stacked over each other and are easily comparable in terms of X and Y data ranges, eye mask violations, histograms, X- and Y-marker intersections.

Histograms

Histograms can be enabled in a Stacked Eye Diagram by selecting any eye , then enabling the Show Histogram property under the Histogram tab in the Properties window, just as in Rectangular Eye Diagrams. The histograms are enabled for all eyes in the Stacked Eye Diagram, and are aligned as shown in the illustration above.

X and Y Markers

X-markers and Y-markers can be added to Stacked Eye diagrams on the **Report 2D > Marker** menu. X-markers align across all the stacked eyes.

Y-markers are specific to each eye. When you set up a Y-marker, select Current Stack to put the Y-marker only on the selected diagram, or All Stacks to add Y-markers to all diagrams. The Y-markers move up and down in tandem.

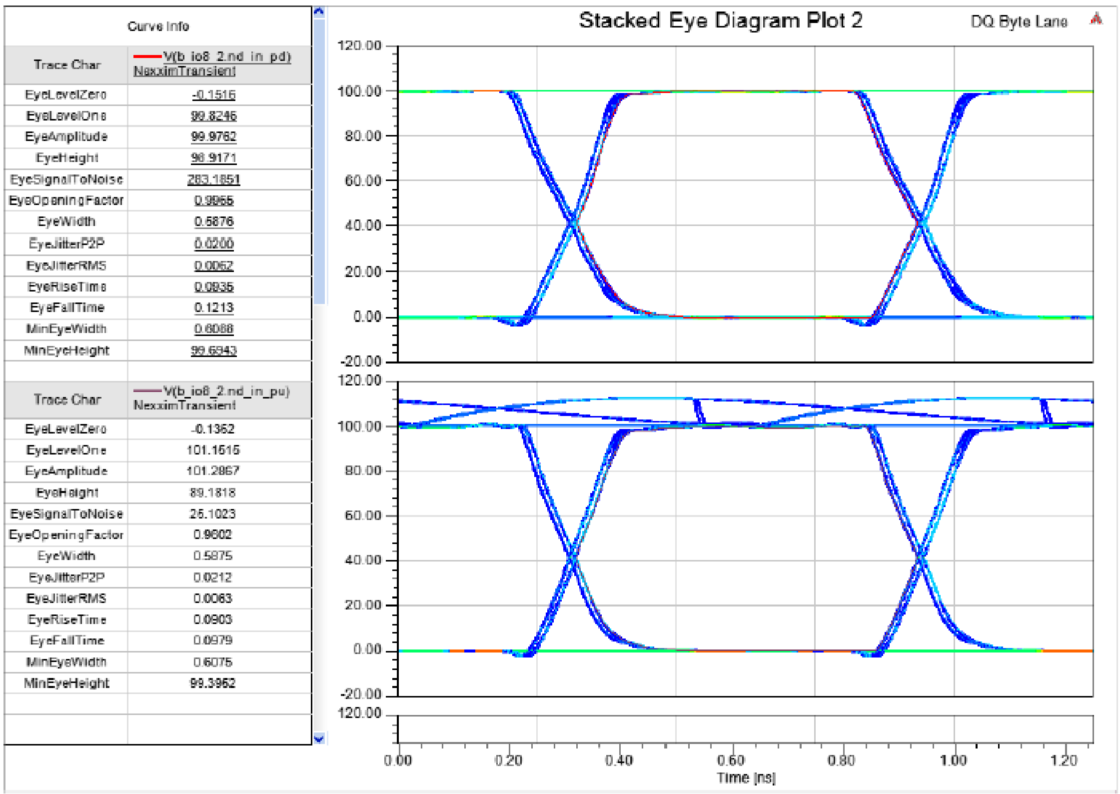
Eye Masks

An Eye Mask can be enabled and specified just as in a Rectangular Eye Diagram. If enabled and specified, the same Eye Mask is shown for each eye in Stacked Eye Diagram.

Eye Measurements

Eye measurements can be enabled through the shortcut menu selection **Trace Characteristics > Add All Eye Measurements**.

Eye Measurements are shown in a Legend window. To maximize plot area, Eye Measurements and other trace characteristics are shown in rows instead of columns for Stacked Plots, alongside each stack, as show in the following:



2D and Circuit PAM-4 Eye Plot

Eye diagrams created from PAM-4 voltage traces differ from regular (NRZ) eye diagrams because they have 4 symbol levels that create three distinct sub-eye openings between the three sequential pairs of symbol levels.

A PAM-4 modulation can be set up using any of the following:

- Transient Analyses
- Eye Source for QuickEye
- AMI Transmitter for AMI Analysis

For more information, see PAM-4 Transient Analysis.

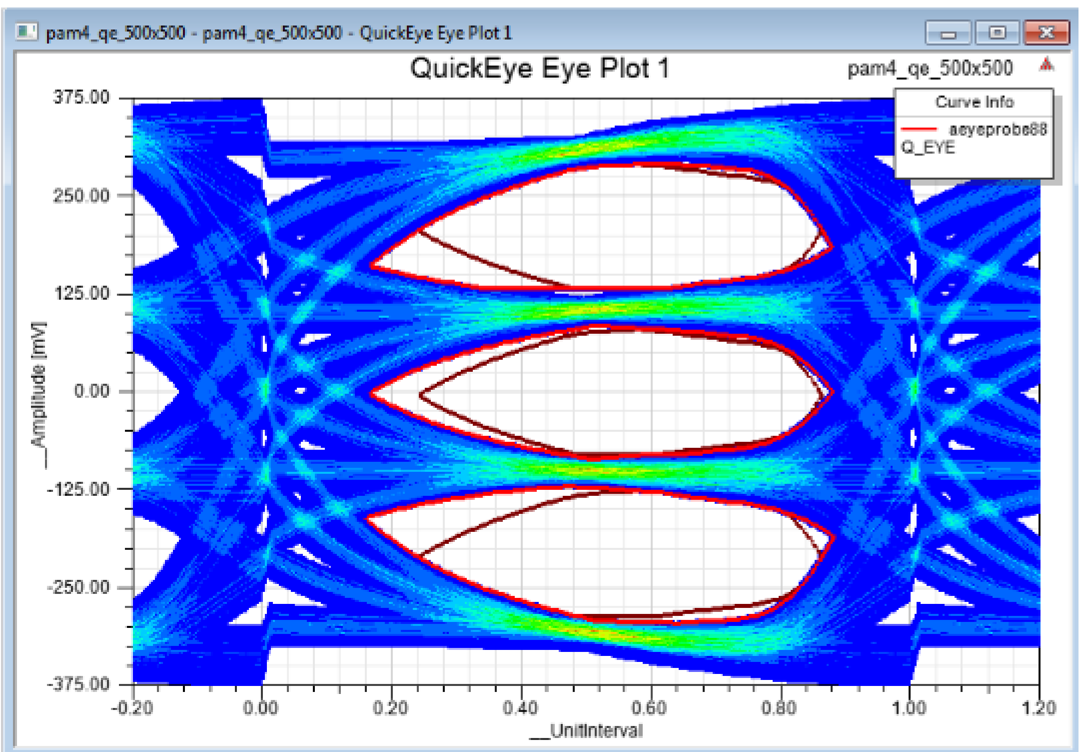
After a successful analysis using PAM-4 modulation, to display an eye plot of the Transient, QuickEye, or AMI results:

1. Click **Product** > **Results** or, on the **Project Manager** window, expand the **Project Tree** and [*active design folder*], then right-click **Results**. Navigate to the appropriate **report type** . Then select **Statistical Eye Plot** or **Eye Diagram Plot** to open the **New Report** window.
2. Selections in the report window are context-sensitive. Whatever Eye Probe/Scope settings are chosen determine the relevant output quantities from which you may select. For example, setting the Eye Scope context to an existing Eye Probe result in the display of quantities that are relevant to the Eye Probe. Whereas choosing Others in the Eye Scope list box displays all other voltage, current and power quantities.

   Select the output quantities of interest and click **Add Trace**.
3. Click **Done**. A Report window opens to display a graph of the analysis results.

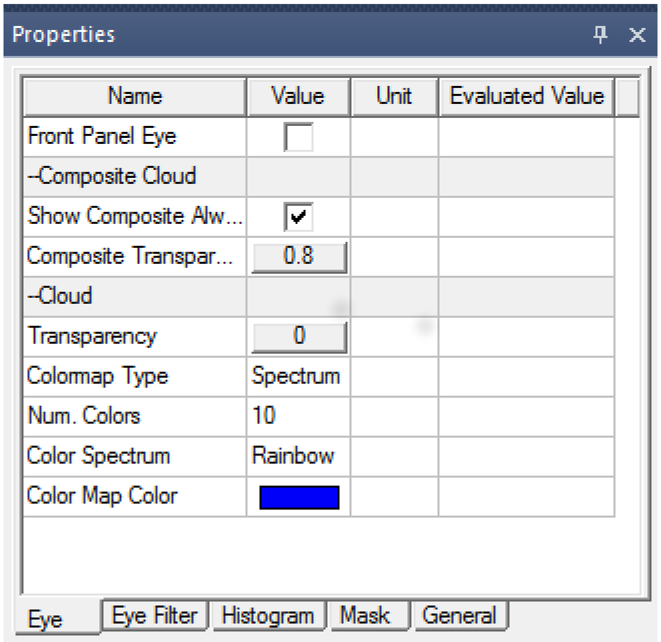
Statistical Eye Plots of PAM-4 Results

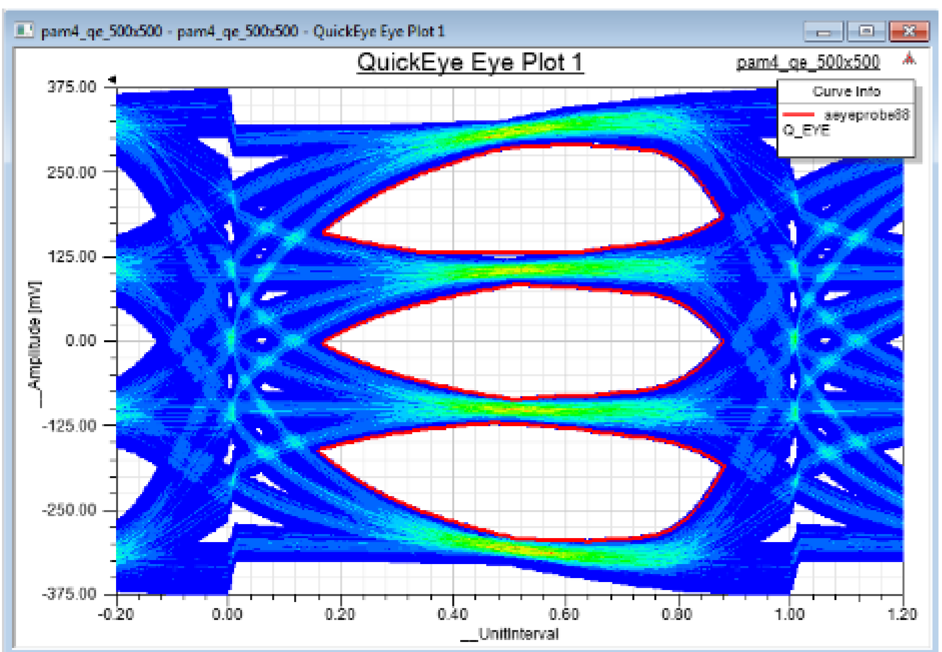
A Statistical Eye Plot of a PAM-4 transmission resembles the following:



An additional “Consolidated” eye, is also displayed. The consolidated eye is created by superimposition of the three sub-eyes aligned along their voltage reference levels. The consolidated eye shows the worst-case eye opening and serves as a simplified measure of the goodness of the eye. The user chooses to use the visual and measurements on the consolidated or individual sub-eyes.

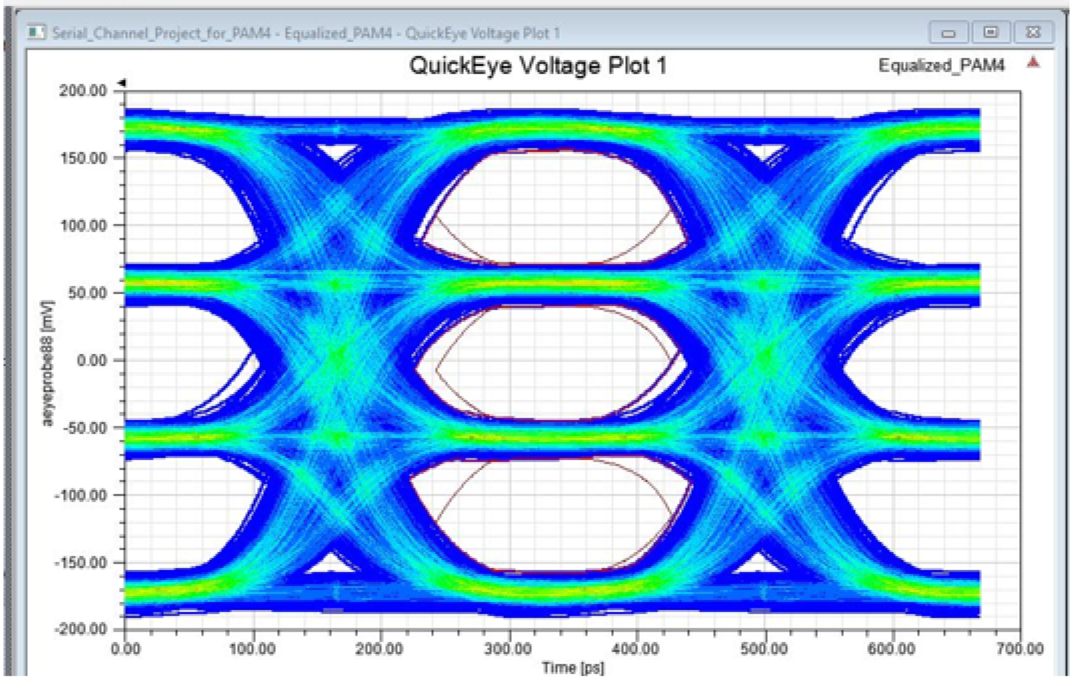
The consolidated eye display is controlled by a trace property:

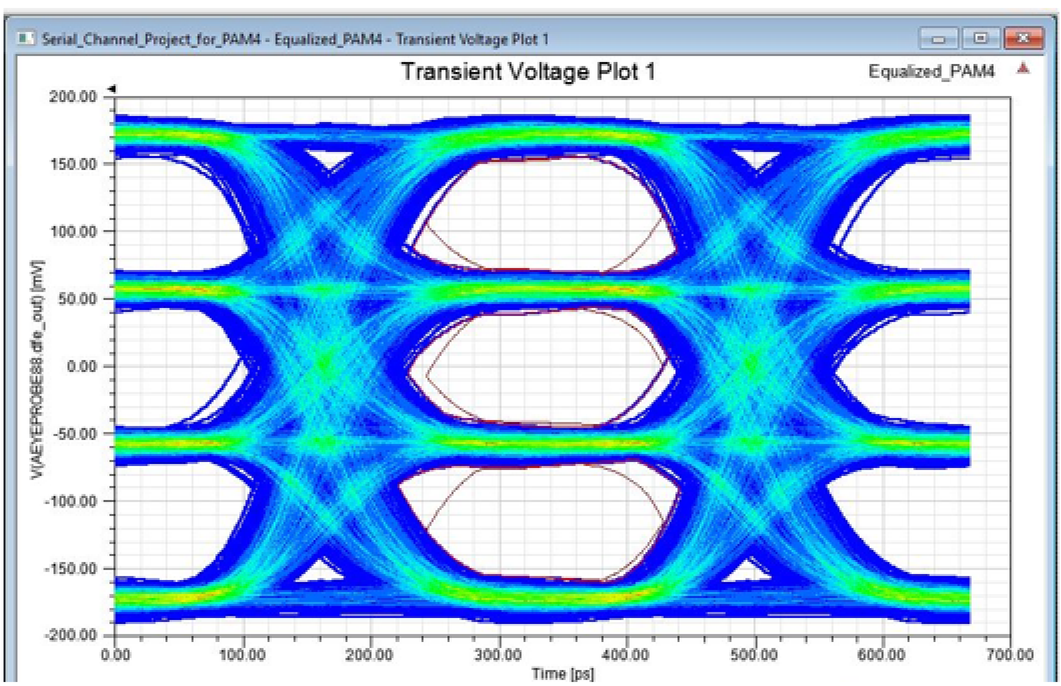




Voltage Plots of PAM-4 Results

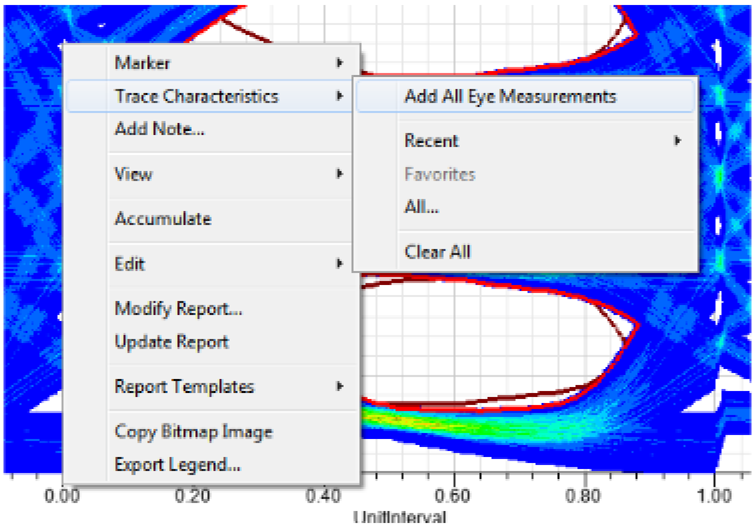
When you generate a Rectangular Eye Diagram from PAM-4 data, the waveform sub-view can be zoomed in to show the boundaries of the unit intervals (black lines) and the inner-eye contour overlay:


Serial_Channel_Project_for_PAM4 - Equalized_PAM4 - QuickEye Voltage Plot 1
QuickEye Voltage Plot 1
Equalized_PAM4
aeyeprobe88 [mV]
Time [ps]
200.00
150.00
100.00
50.00
0.00
-50.00
-100.00
-150.00
-200.00
0.00
100.00
200.00
300.00
400.00
500.00
600.00
700.00

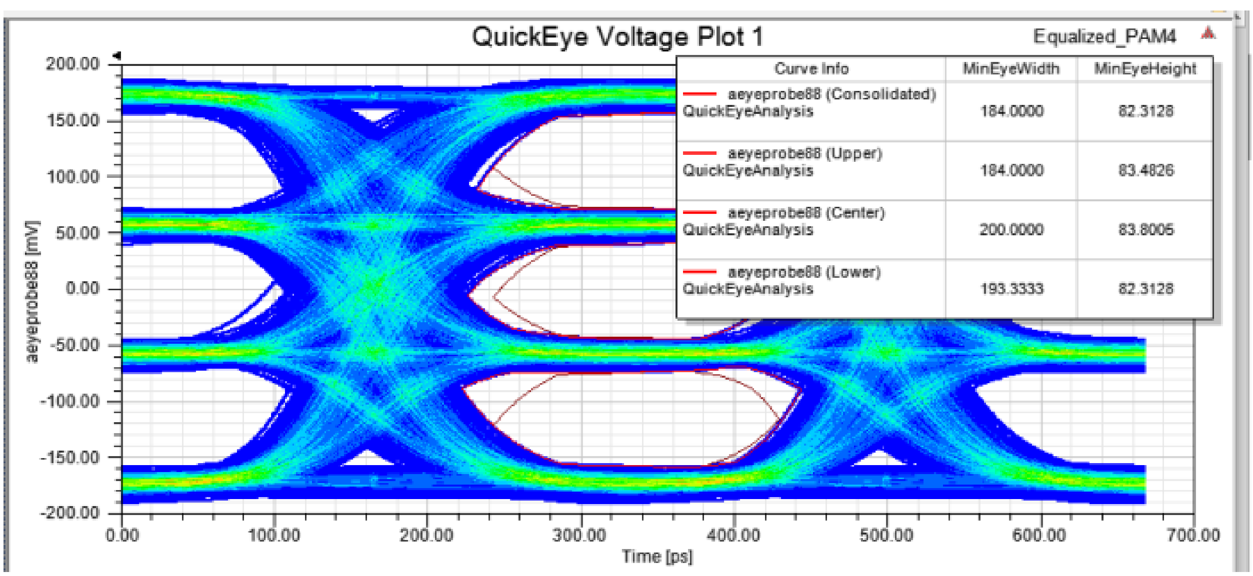

Serial_Channel_Project_for_PAM4 - Equalized_PAM4 - Transient Voltage Plot 1
Transient Voltage Plot 1
Equalized_PAM4
V(AEYEPROBE88.dfe_out) [mV]
Time [ps]
200.00
150.00
100.00
50.00
0.00
-50.00
-100.00
-150.00
-200.00
0.00
100.00
200.00
300.00
400.00
500.00
600.00
700.00

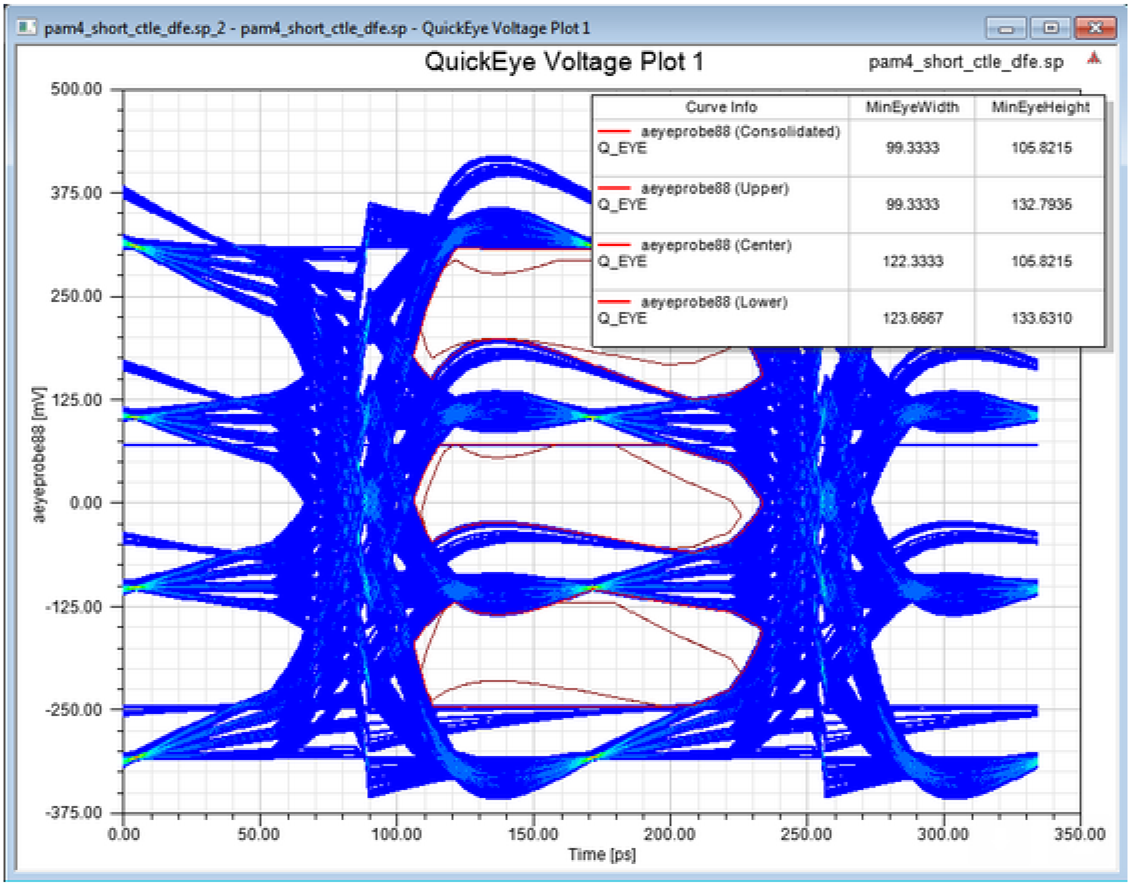
Eye Measurements for PAM-4 Plots

PAM-4 eye diagrams support only the MinEyeWidth and MinEyeHeight measurements. To add trace characteristics, right-click the plot and select on the following options:



When **Add All Eye Measurements** is clicked, only the supported MinEyeHeight and MinEyeWidth are added to the plot. Any other measurements display NaN if added.





PAM-4 trace characteristics are displayed for each sub-eye and include the consolidated measurement. The sub-eye names and top-to-bottom sorting allow easy mapping of the numbers to the visual. When the Legend is exported (via the shortcut menu), the entire table along with the sub-eye name annotations is also exported.

> **Note:** Mask violations for transient analysis are not supported.

Related Topics

2D and Circuit Report Types

2D and Circuit New Report Window

2D and Circuit Report Display Types

Sweeping a Variable in a Report

Working with Traces

Delta Markers in 2D Plots

Working with Eye Diagrams

Modifying the Background Properties of a Report

Specifying Output Variables in 2D and Circuit Reports

The **Output Variables** window contains four group boxes:

- **Context —** Specify the Report type, the Solution, and for appropriate report types, the Domain and Show fields. Changing the Report type affects whether the Domain menu appears, and may affect the functions listed in the **Function** group box.
- **Output Variables —** Specify the name and expression for a new output variable.
- **Calculation —** Insert quantities into the **Expression** area of the **Output Variables** group box.
- **Function —** Insert completed expressions into the **Expression** field of the **Output Variables** group box.

2D and Circuit Adding a New Output Variable

To add an output variable:

1. Click **Product Menu** > **Results** > **Output Variables** or, in the **Project Manager** window, right-click **Results** and select **Output Variables** to open the **Output Variables** window. Variables defined using the **Product Menu** > **Results** > **Output Variables** command appear in the list at the top of the window.
2. In the **Output Variables** group box, enter a name for the new variable in the *Name* field.
3. To enter an expression, do one or both of the following:
   a. Type part or all the expression directly in the **Expression** field. Valid functions appear in blue. Invalid functions appear in red.
   b. Insert part or all the expression using the options in the Function group box.
4. Click **Add** to add the new variable to the list.
5. Repeat steps 2 through 5 to add additional variables.
6. When you are finished adding output variables, click **Done** to close the **Output Variables** window.

2D and Circuit Building an Expression Using Existing Quantities

When entering an expression for a new output variable, insert part or all the expression using the options in the **Function** group box in the **Output Variables** window.

To add an input variable by inserting part or all of the expression:

From **Product Menu** > **Results** > **Output Variables** or, in the **Project Manager** window, right-click **Results** and select **Output Variables** to open the **Output Variables** window. In the **Output Variables** window, enter a name for the new variable in the *Name* field.

To insert a quantity:

1. From the **Report Type** drop-down list, select the type of report from which the user wants to select the quantity.
2. From the **Solution** drop-down menu, select the solution from which the user wants to select the quantity.
3. From the **Category** group box, select the appropriate type of quantity to enter.
4. From the **Quantity** group box, select the appropriate quantity or the geometry.
5. From the **Function** group box, select a ready-made function (this option is the same as inserting the function on the **Function** group box).
6. If applicable, on the **Domain** list, select the solution domain.
7. Click **Insert Into Expression**.

   The selected quantity is entered into the **Expression** area of the **Output Variables** window.

To insert a function:

1. From the **Function** group box, select a ready-made function on the drop-down menu.
2. Click **Insert Function into Expression.**

   The function appears in the **Expression** area of the **Output Variables** group box.

1. When the user is finished defining the variable in the **Expression** field, click **Add** to add the new variable to the list.
2. Repeat steps 2 through 6 to add additional variables.
3. When the user is finished adding output variables, click **Done** to close the **Output Variables** window.

> **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, the user can plot values in units other than SI. If the user wants to plot the polar angle of a complex simulation result, S_{11} say, 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 the user treats "1" as degrees, make it explicit and use "1deg + ang_deg(S_{11})" instead.
>
> If the user 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 so by any expression. If the complex S_{11} lies on the positive Y axis say, ang_deg_val(S_{11}) is 90 and "1 + ang_deg_val(S_{11})" is 91.

Deleting Output Variables (2D and Circuit)

To delete output variables:

1. Remove all references to the output variable in the project.
2. Save the project to erase the command history.
3. Click **Product Menu > Results > Output Variables** or, in the **Project Manager** window, right-click **Results** and select **Output Variables**.

   This opens the **Output Variables** window.

4. Select the variable and click **Delete**.
5. Click **OK** to close the window.

Viewing Matrix Data in 2D and Circuit Reports

To view matrices computed for the S-parameters, impedances, and propagation constants during each adaptive, non-adaptive, or sweep solution:

1. From the **Project Manager** window, right-click the solution setup of interest, then click **Matrix Data** to open the **Solution Data** window

   The **Matrix Data**tab is selected.

2. In the **Design Variation** field, specify the design with the matrices you want to view.

Optionally, choose a design variation solved during an Optimetrics analysis on the **Set Design Variation** window. This lists all the solved variations in the design.

3. In the **Simulation** dropdown menu, select the solution setup and solved pass - adaptive, single frequency solution, or frequency sweep - for which you want to view matrices.
4. Select the type of matrix you want to view: **S-matrix**, **Y-matrix**, **Z-matrix**, **Gamma**, or **Z** $_{\mathbf{o}}$ (characteristic impedance.) The available types depend on the solution type.
5. Select the format in which to display the matrix information. The available formats depend on the matrix type being displayed.
6. Select the solved frequencies to display:
   - to open the matrix entries for all solved frequencies, select **All Freqs**. It is selected by default.
   - To show the matrix entries for one solved frequency, clear **All Frequencies**, then select the solved frequency for which you want to view matrix entries. For adaptive passes, only the solution frequency specified in the **Solution Setup** window is available. For frequency sweeps, the entire frequency range is available.
   - To insert or delete one or more displayed frequencies, click **Edit Freqs**. This command is only available if the sweep type is Fast or Interpolating. If choose to export the matrix data for the Fast or Interpolating sweep after modifying the frequencies in the **Edit Frequencies** window, only those frequencies displayed under the **Matrix Data** tab is exported.

The data is displayed in the table. By default, wave ports are listed in alphabetical, then numerical order, just as they appear in the excitation tree. To change the port order, change setting for Default Matrix sort order in **Electronics Desktop** General options.

2D and Circuit Selecting the Matrix Display Format

Display matrix data in the following formats. The available formats depend on the type of matrix being displayed.

Magnitude, Phase (deg)	Displays the magnitude and phase (in degrees) of the matrix type.
Real, Imaginary	Displays the real and imaginary parts of the matrix type.
dB, Phase (deg)	Displays the magnitude in decibels and phase in degrees of the matrix type.
Magnitude	Displays the magnitude of the matrix type.
Phase (deg)	Displays the phase in degrees of the matrix type.
Real	Displays the real parts of the matrix type.
Imaginary	Displays the imaginary parts of the matrix type.
dB	Displays the magnitude in decibels of the matrix type.

Renaming 2D and Circuit Matrix Data

From the **Project Manager** window, right-click a port excitation to rename it. When you rename a port excitation, the associated data is reordered so it can be presented in the same manner. The reordering is done to match the tree-sort order presented for the ports (renamed matrix data is reordered so alphabetic values appear before numeric values).

Exports of the matrix data is ordered in the same manner. This reordering is conducted as part of post-processing and does not force a re-solve.

Animation in 2D and Circuit Reports

The following topics describe how to post-process field overlay displays to create and view various animated reports. Any of the field overlay displays can be animated by cycling the overlays as a series of frames. See 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.

Related Topics

2D Frequency Animation

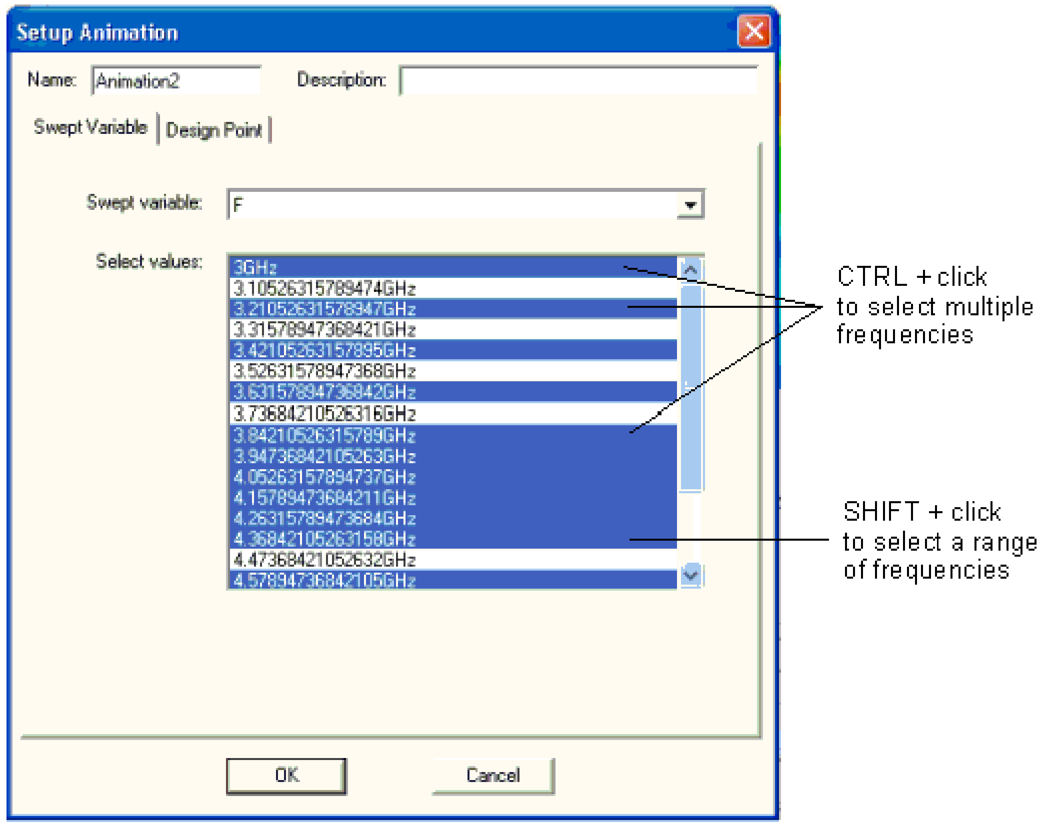
2D Phase Animation

Changing the Design Point in 2D Animation

2D Frequency Animation

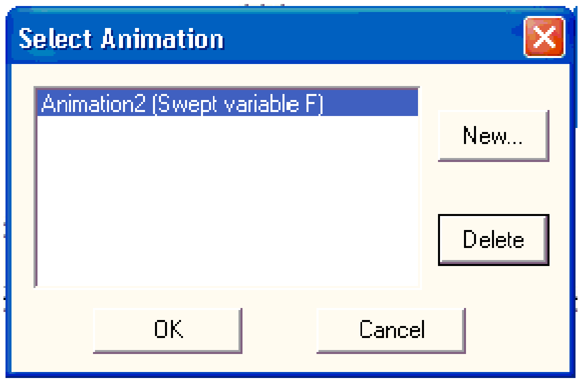
1. To initiate the animation of the overlay that is currently displayed, do one of the following:
   - Select **Animate** on the **View** menu.
   - Expand the **Results** icon in the Project window, right-click the overlay entry, and select **Animate** from the menu.

2. If no animations have been defined previously, the **Setup Animation** window 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 **Ctrl** to select multiple individual frequencies, or hold down **Shift** to select a contiguous range of frequencies. [These selection modes are illustrated in the window example above].
- Click **OK**.

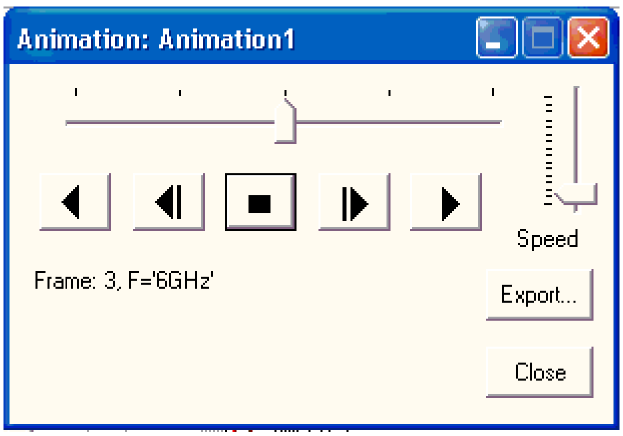
If one or more animations have been defined, selecting **Animate** from one of the menus opens the **Select Animation** window:



The **Select Animation** window provides the following operations:

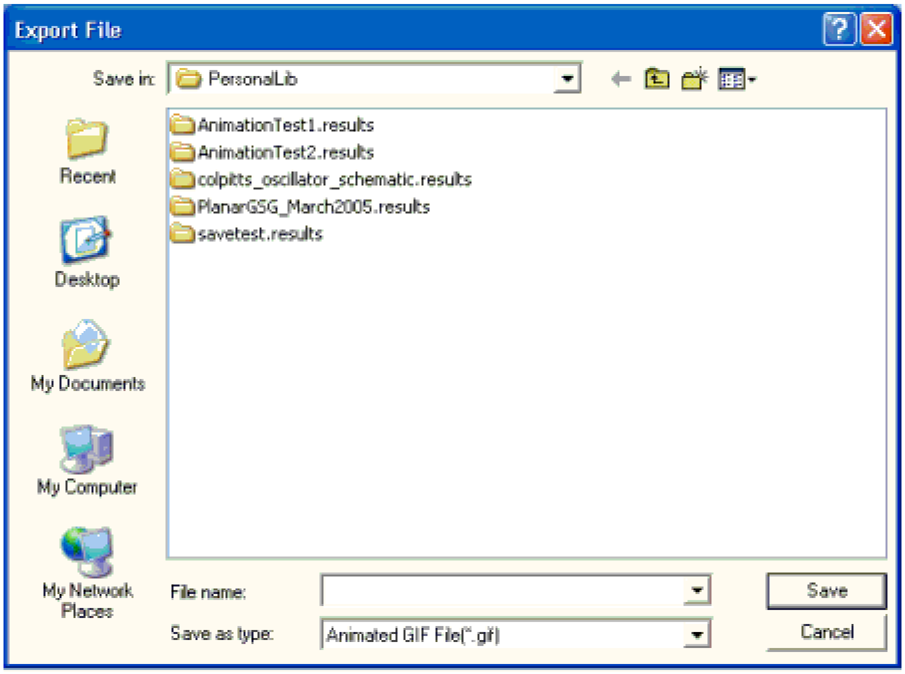
- Click to select one of the animations and click **OK** to start that animation.
- Click **New** to open the **Setup Animation** window described above, and close the **Select Animation** window.
- Select an animation and click **Delete** to delete the definition.

3. Electronics Desktop calculates the frame data. If the Progress window is displayed, monitor the progress of the calculation. When the frames have been calculated, the animation begins and the **Animation** control window opens:



- Click the playback 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 window shows the progress of the animation. Click the **Speed** slider to control the speed of the animation.

- To export the frame data to a file, click **Export**. The **Export File** window opens:



Specify the directory and file name. Click the **Save as type** menu to select the file format (Animated GIF or AVI). Click **Save** to save the data and close the window.

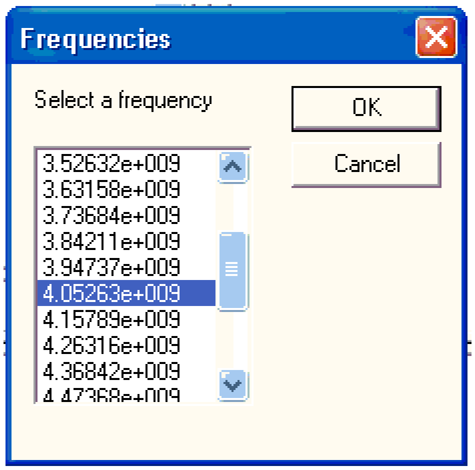
- Click **Close** on the **Animation** control window to stop the animation and close the window.

2D Phase Animation

1. To prepare for phase animation, you must select the base frequency.

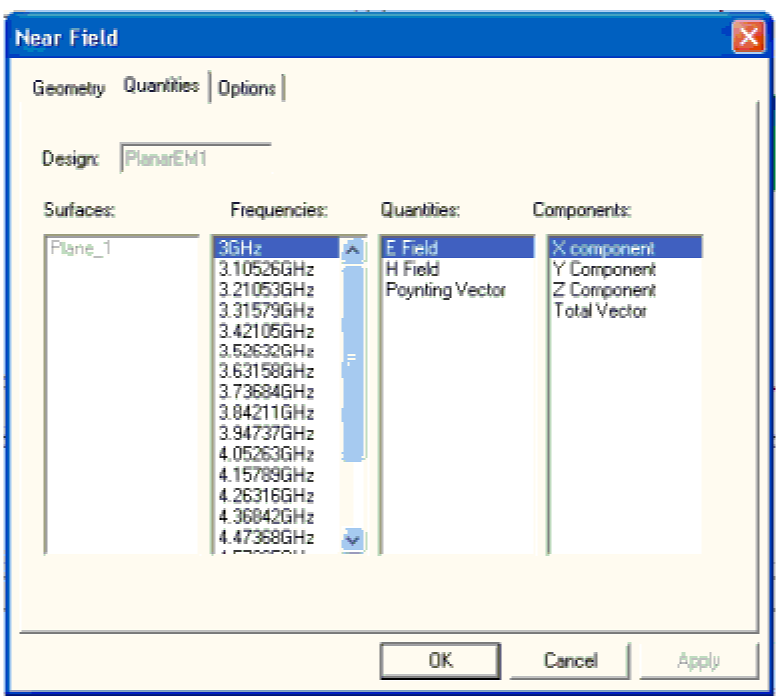
- For surface current and far field overlays, expand the **Results** icon in the **Project Manager** window, right-click the overlay, and select **Frequency** on the menu. The

Frequencies window opens:



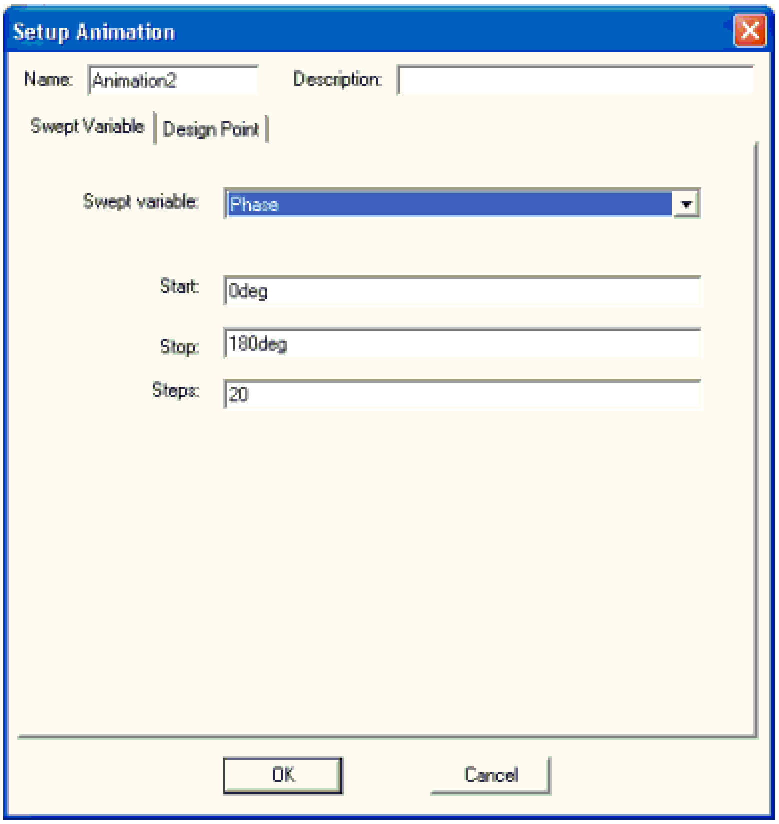
The list displays the frequencies that are swept in the analysis. Select a frequency on the list; the overlay displays the field values calculated at that frequency. Click **OK** to leave the overlay at the selected frequency.

- For near field overlays, expand the **Results** icon in the **Project Manager** window, right-click the overlay, and select **Properties** on the menu. Click the **Quantities** tab to select a reference frequency for the phase animation:



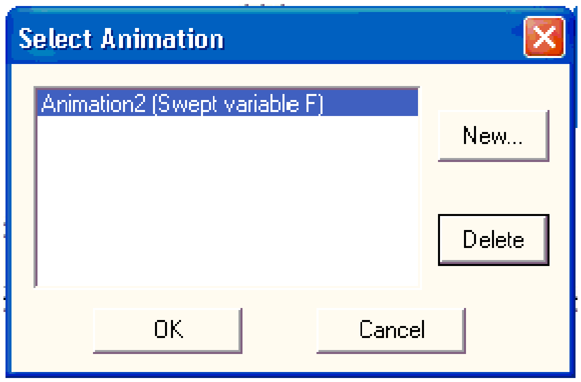
Click **OK** to apply the frequency and close the window.

2. To initiate the animation of the overlay that is currently displayed, do one of the following:
   - Select **Animate** on 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** window 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**.

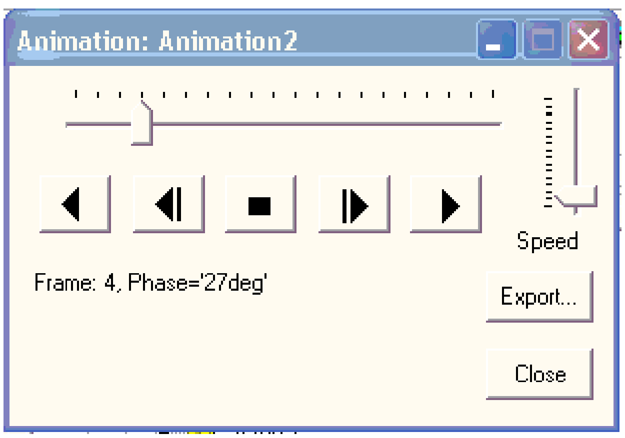
If one or more animations have been defined, selecting **Animate** from one of the menus opens the **Select Animation** window:



The **Select Animation** window provides the following operations:

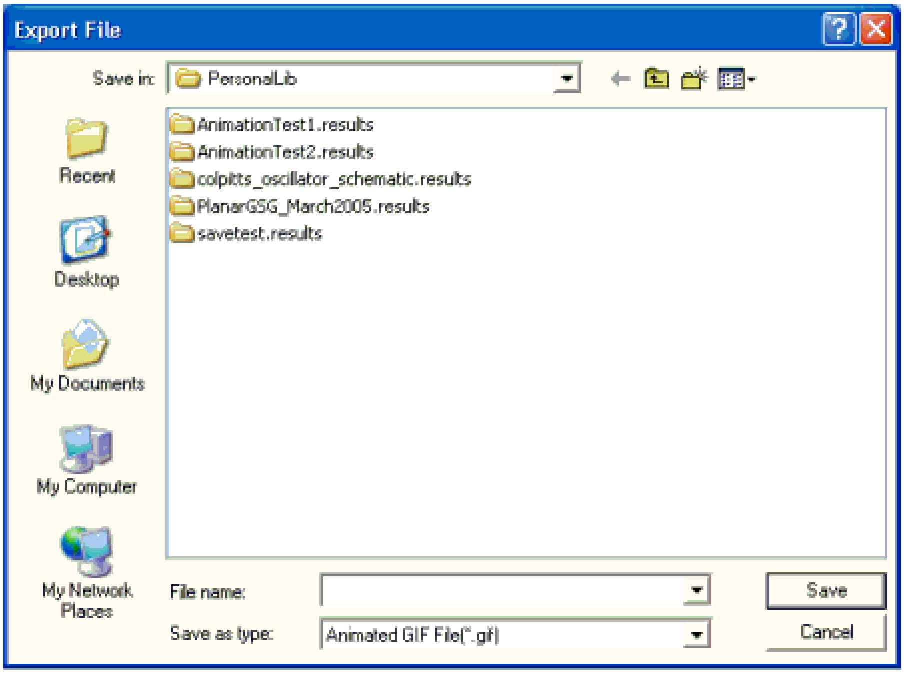
- Click to select one of the animations and click **OK** to start that animation.
- Click **New** to open the **Setup Animation** window described above, and close the **Select Animation** window.
- Select an animation and click **Delete** to delete the definition.

4. Electronics Desktop calculates the frame data. If the Progress window is displayed, monitor the progress of the calculation. When the frames have been calculated, the animation begins and the **Animation** control window opens:



- Click the playback 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 window shows the progress of the animation. Click the **Speed** slider to control the speed of the animation.

- To export the frame data to a file, click **Export**. The **Export File** window opens:

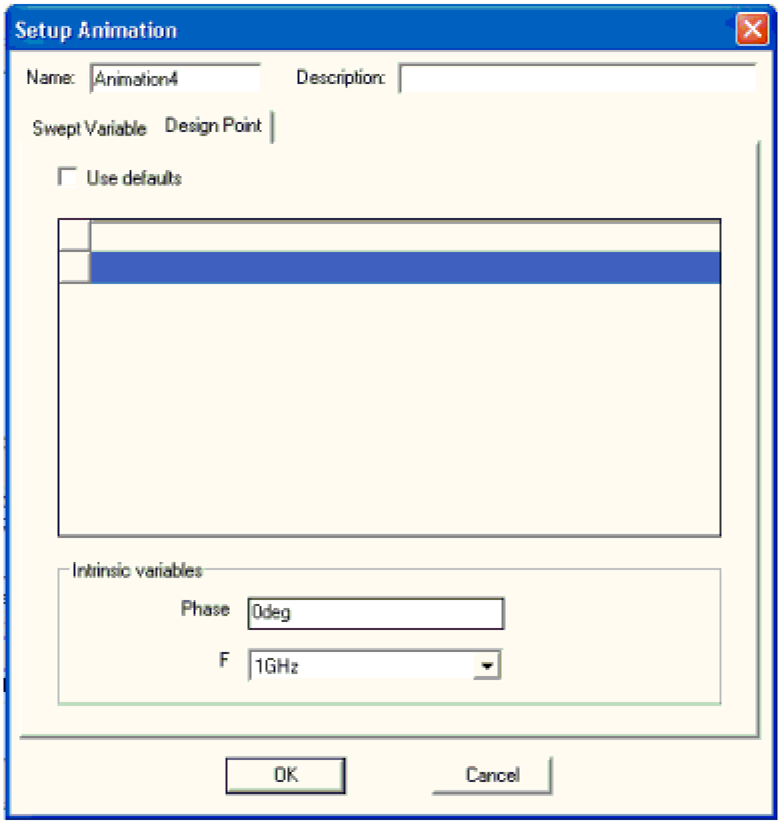


Specify the directory and file name. Click the **Save as type** menu to select the file format (Animated GIF or AVI). Click **Save** to save the data and close the window.

- Click **Close** on the **Animation** control window to stop the animation and close the window.

Changing the 2D 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 on the ones used in the original simulation. From the **Animation Setup** window, select the **Design Point** tab and deselect the **Use defaults** option. The following fields are displayed:



Make any appropriate changes, then click **OK** to apply the changes and close the window. Clicking **Cancel** closes the window without making any changes.

User-Defined Outputs in 2D and Circuit Reports: Introduction

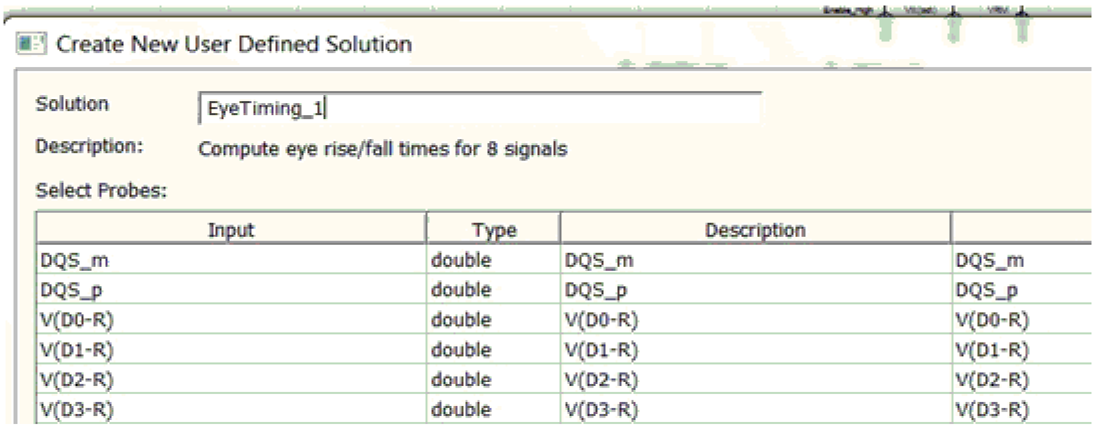
User-defined outputs (UDOs) allow users to define calculations through IronPython scripts or any .NET language (and used by the IronPython script). The UDO scripts need to be in the **UserDefinedOutputs** directory under either of **syslib**, **userlib** or **Personallib** with any directory structure needed for organization (the **Lib** directory name is special and its purpose is explained later on in the document)

The UDO scripts that are in syslib/UserDefinedOutputs, userlib/UserDefinedOutputs or Personallib/UserDefinedOutputs become available to the user to create "user-defined Solutions" through the **Results>Create user-defined Solution** menu. Use **Results>Create user-defined Solution>Update Menu** to refresh the menu to include the new UDO scripts that might have been copied to syslib, userlib or Personallib, or exclude them if they have been deleted, after the launch of desktop. Once the user-defined-solution is created, the solution and the calculations defined by UDO become available in Reporter as any other quantities in a new "user-defined" report type.

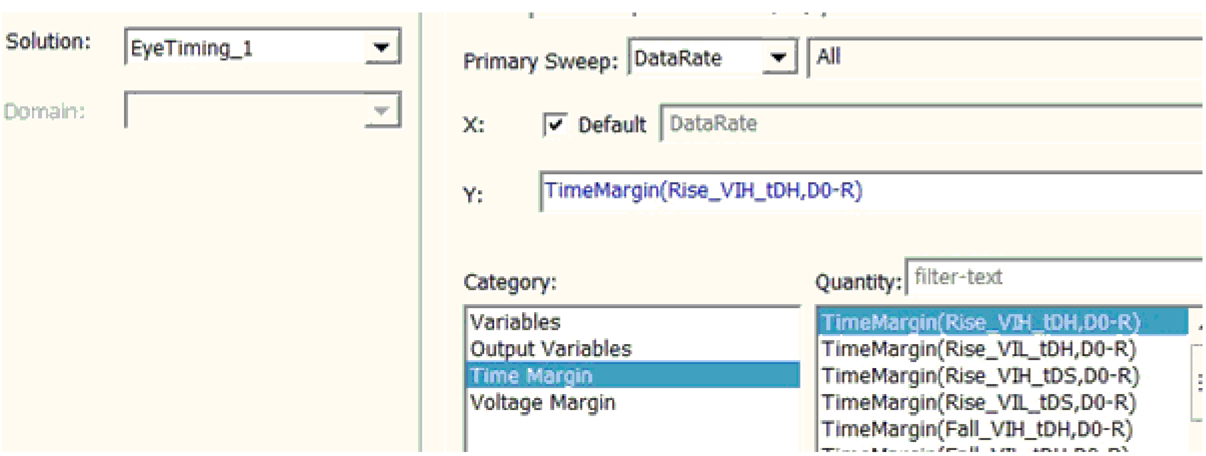
Named Probes and Properties in user-defined Outputs (2D and Circuit)

UDOs allow processing data across traces, solutions and report types. A UDO specifies the named probes and properties for which user selects/enters the values at the time of creation of user-defined solution. Probes are very similar to traces except that the user selects the values of only intrinsic variables for probes. The values of design/project variables are selected when a trace is created based upon the user-defined solution in reporter.

For example, you could create a user-defined solution called EyeTiming_1.



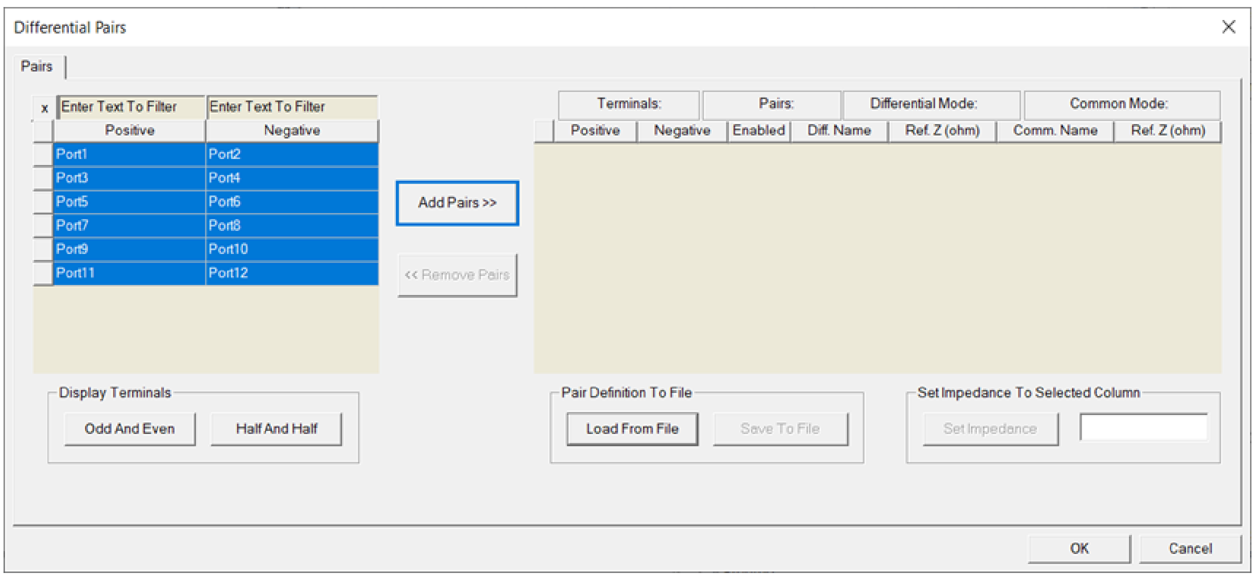
You can then access this solution in the Reporter.



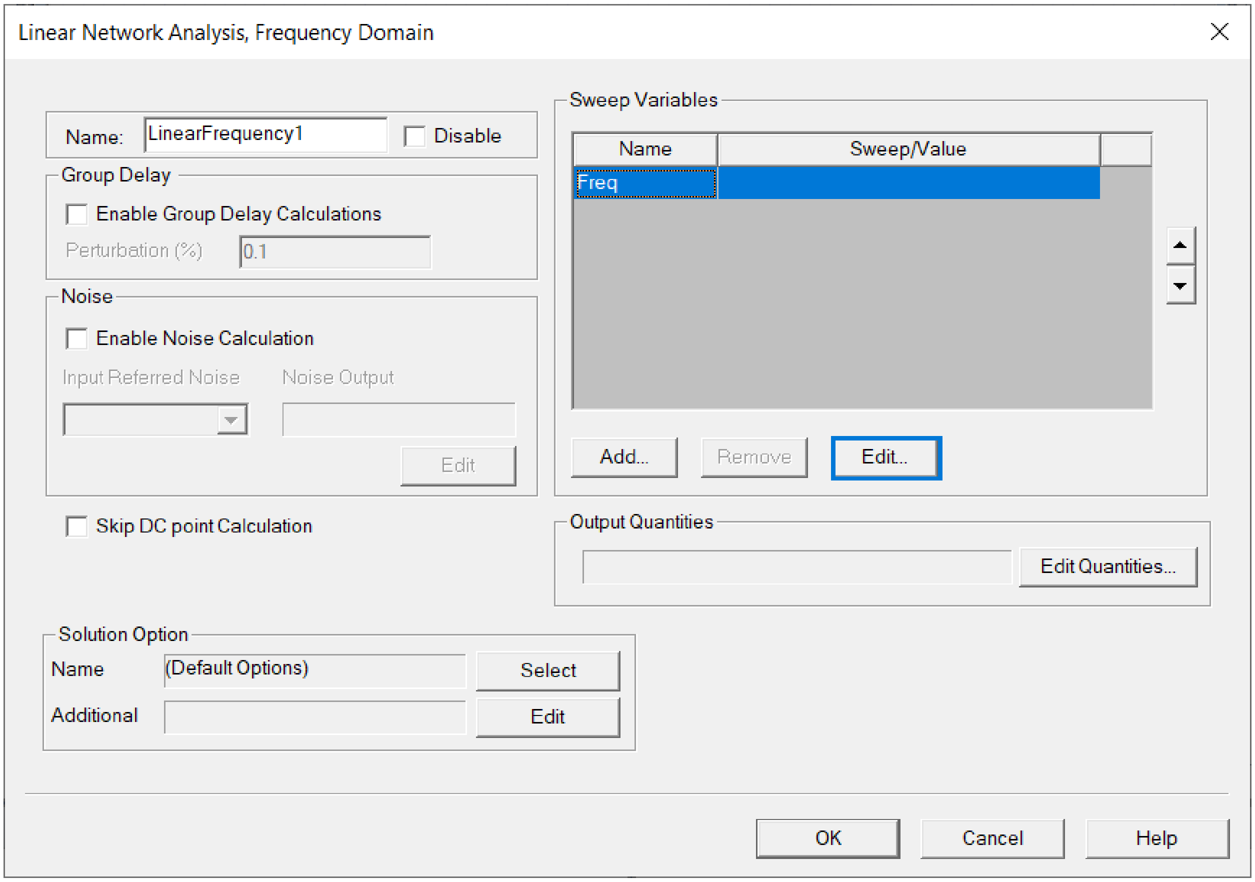
Generate COM, or ERL, IDL, and ICN Calculations With A User Defined Solution.

To generate Channel Operating Margin (COM), or Effective Return Loss (ERL), Insertion Loss Deviation (ILD), and Integrated Cross Talk (ICN) calculations individually within Circuit, have a circuit design ready to be analyzed with an n-port model. Additionally, a relevant config file(.cfg) is assumed in the following instructions.

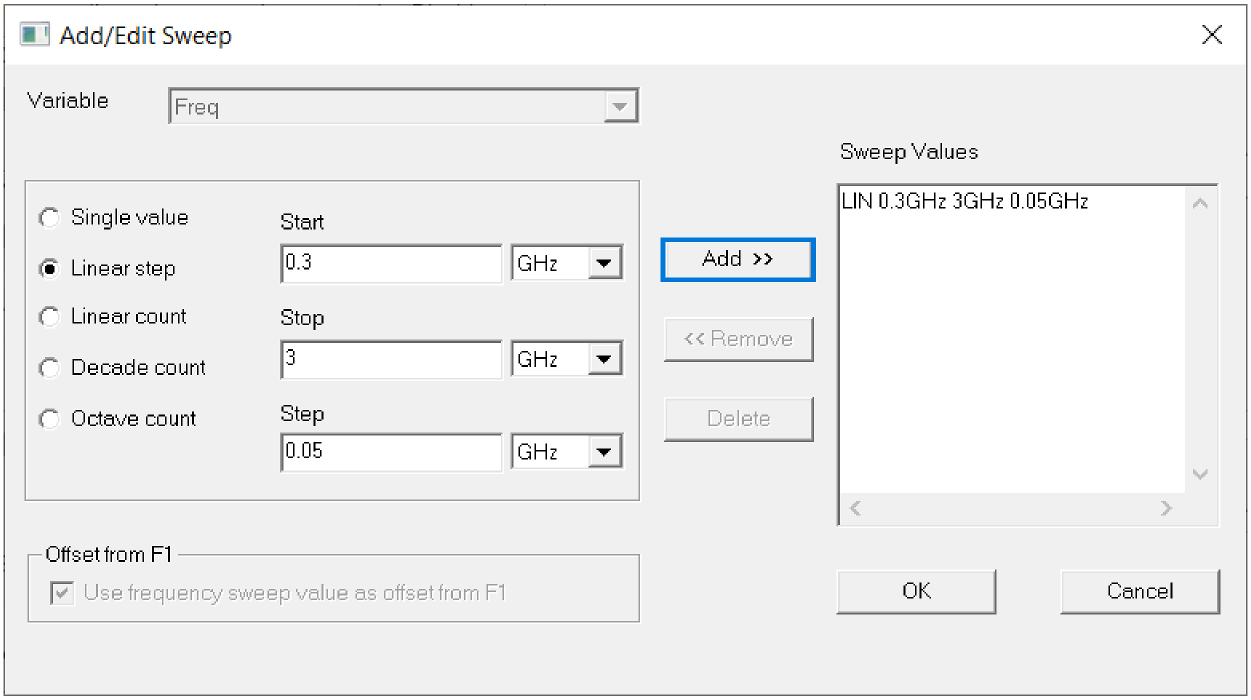
1. In the **Project Tree**, expand the active project.
2. Right-click **Ports** and select **Differential Pairs**. The **Differential Pairs** dialog opens.
   i. From the **Pairs** tab, select the pairs, then click **Add Pairs**.
   ii. Click **OK** to close the dialog.



3. If an analysis solution is not yet set up, add an analysis solution. For example, a linear network analysis:
   i. From the **Project Tree**, right-click **Analysis > Add Nexxim Solution Setup... > Linear Network Analysis**.
   ii. In the **Name** field, enter a suitable name, such as LinearFrequency.
   iii. In the Sweep groupbox, select the **Freq** row, then click **Edit...** to open the **Add/Edit Sweep** dialog.



iv. Enter the desired sweep values using the variable step/count options, then click **Add>>**. For example Start: 0.3 GHz ; Stop: 3 GHz ; Step: 0.05 GHz.



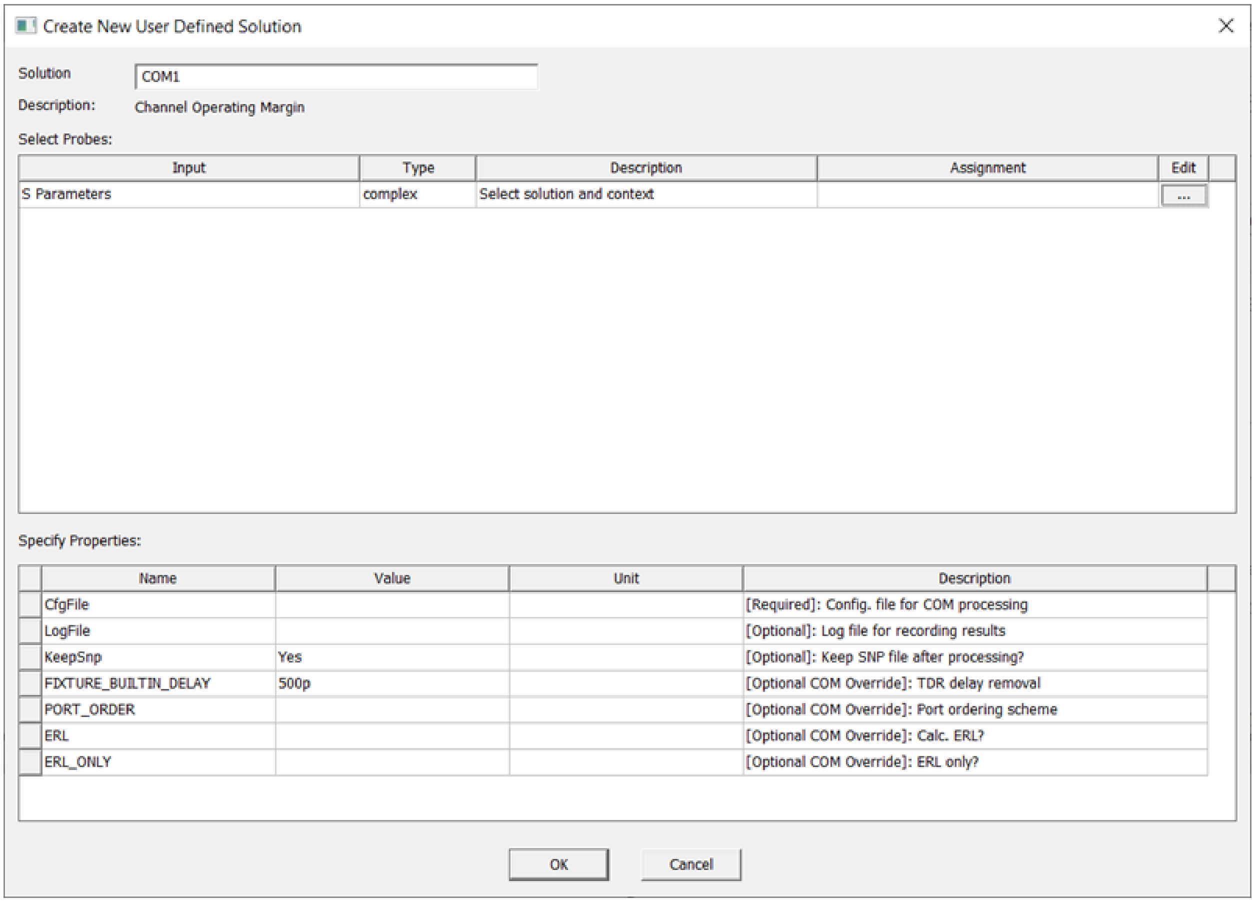
v. Click **OK** to close the **Add/Edit Sweep** dialog, then click **OK** to close the **Linear Network Analysis** dialog.

4. From the **Project Tree**, right-click **Analysis** then select **Analyze**. Wait for the design to be analyzed.

5. Create the new User Defined Solution. Select the calculation: COM, or ERL, IDL, ICN individually.

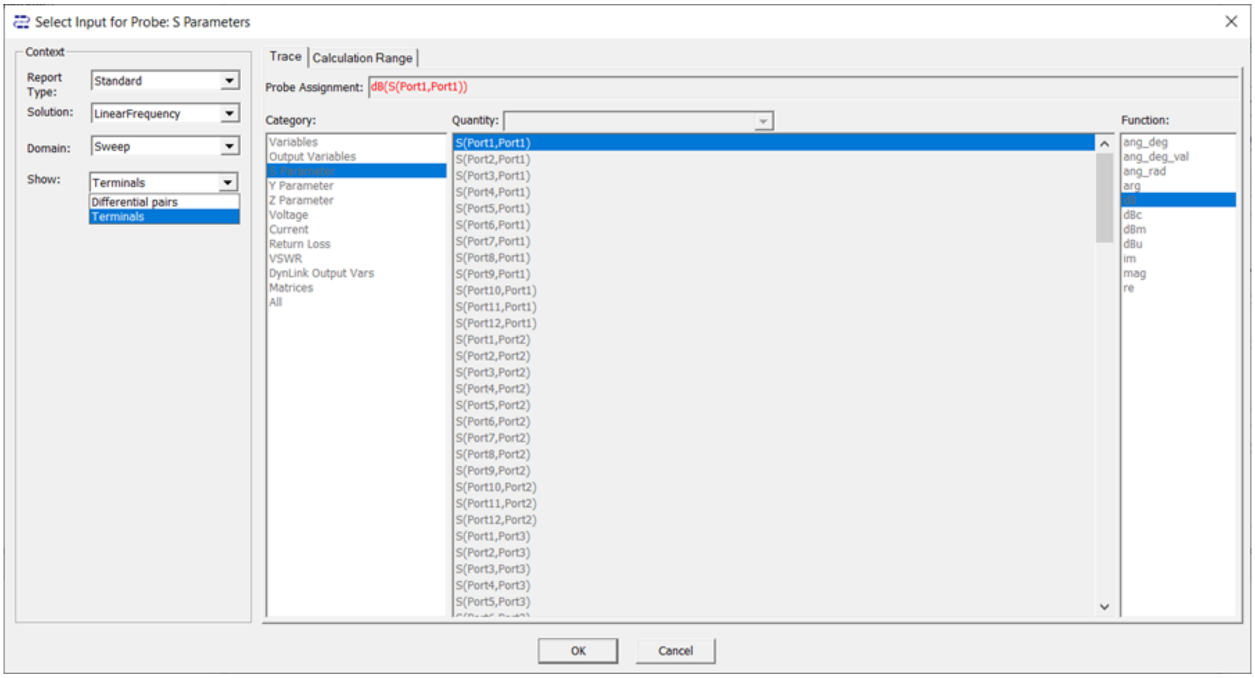
   For a COM calculation report:

   From the **Project Tree**, right-click **Results** and select **Create User Defined Solution> CircuitDesigns> COM** to open the **Create New User Defined Solution** dialog.



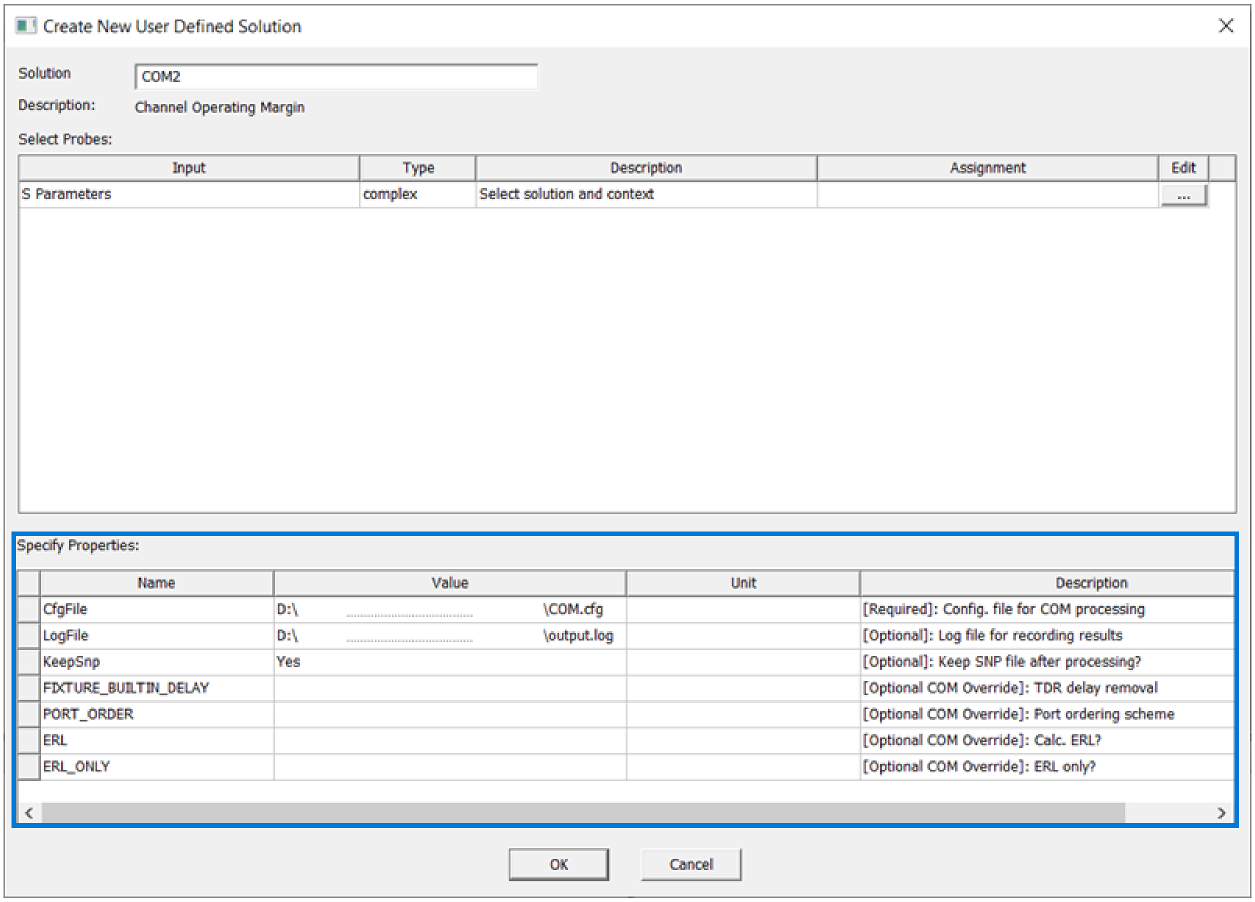
i. In the **Name** field, enter a suitable name, such as **COM1**.

ii. From the **Select Probes** table, in the **S Parameters** row, click **...** under the **Edit** column to launch the **Select Input for Probe: S Parameters** dialog.



1. From the **Context** group box, expand the **Show:** drop-down menu and select **Terminals**.
2. Click **OK** to close and return to the **Create New User Defined Solution** dialog.

iii. In the **Specify Properties** table, enter the file path for the relevant Config file(.cfg) in the **CfgFile Value** field. Additionally, in the **Logfile Value** field, specify a log file(.log) path if desired.



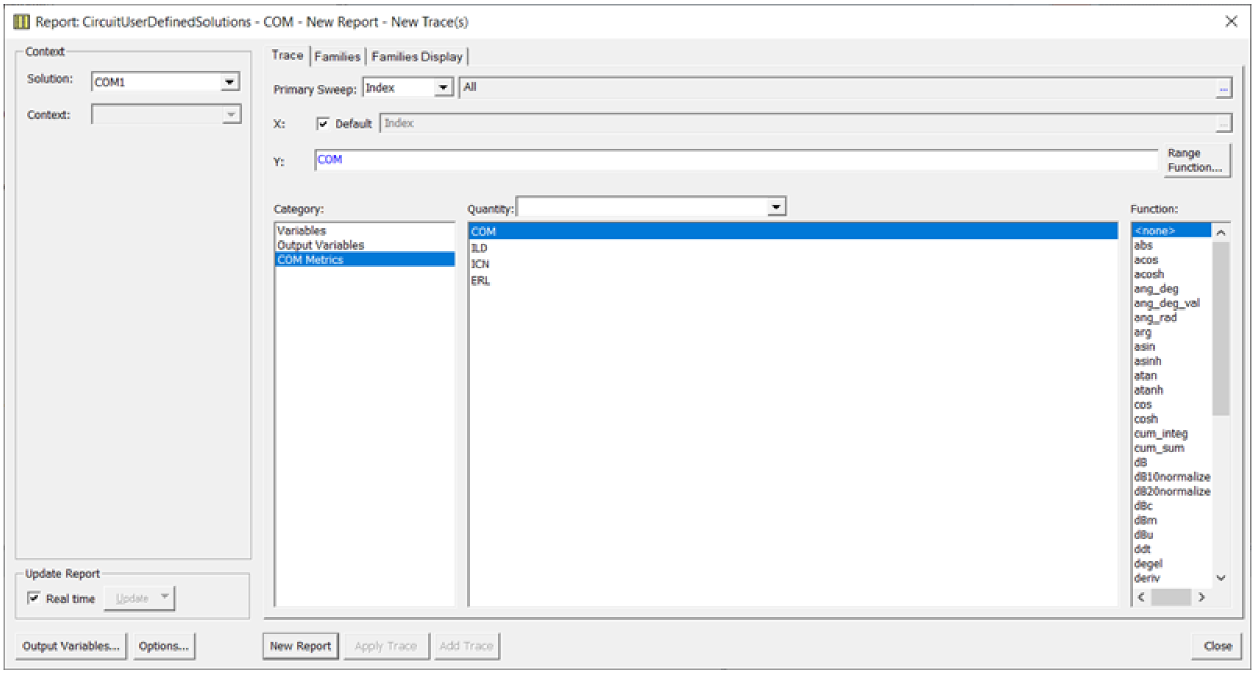
> **Note:** If other property values are updated in the **Specify Properties** table, such as **FIXTURE_BUILTIN_DELAY**, the table entered properties will override any corresponding properties specified in the config file.

iv. Click **OK** to close the dialog.

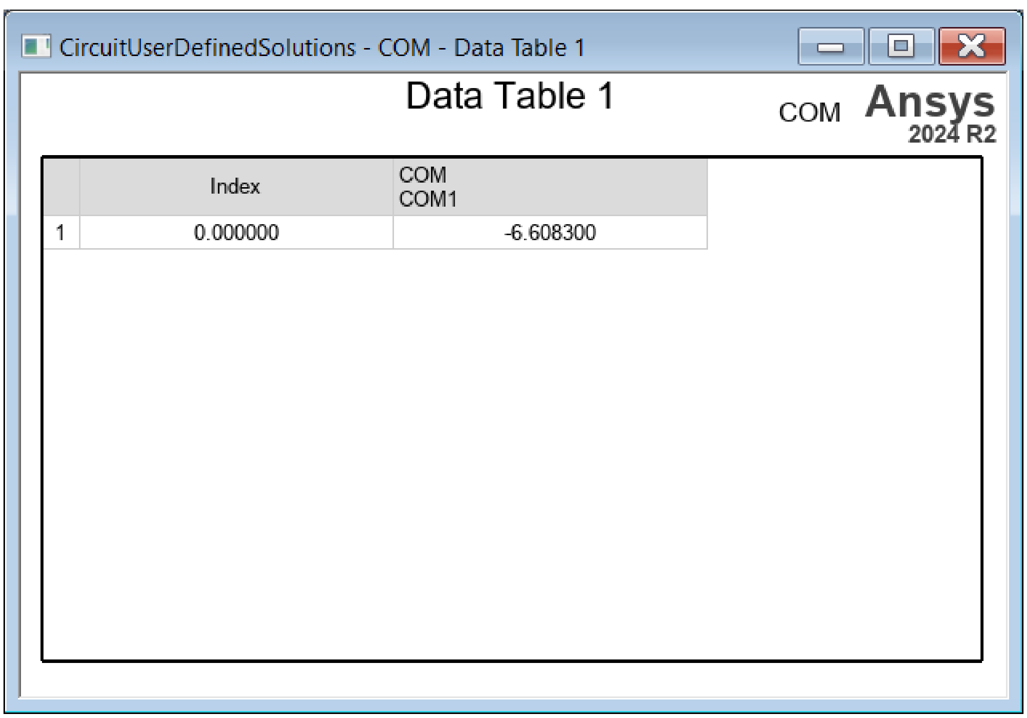
6. From the **Project Tree**, right-click **Results**, then select **Create User Defined Report > Data Table to open the Report: CircuitDesign Solutions** dialog.

   i. In the **Context** group box, expand the **Solution:** drop-down menu and select the User Defined Solution, in this case COM1. Ensure only COM Metrics is selected in the **Catagory** list, and only COM is selected in the **Quantities** list to create a Data table with only the COM calculation. Alternatively, selecting any combination of COM, ILD, ICN, and ERL in the quantities list will generate a data table containing those calculation values.

ii. Click **New Report**, then **Close**.



The new report has been generated!



> **Note:** The UI process to create reports for ERL, ICN, and ILD is similar. Select ERL or ICN or ILD in step 5 and enter the relevant config file path in step 5iii. Additionally, note that to create a User Defined Solution for ICN calculations, the snp file must contain more than 4 ports.

Creating a Transient EMI Receiver Probe Report

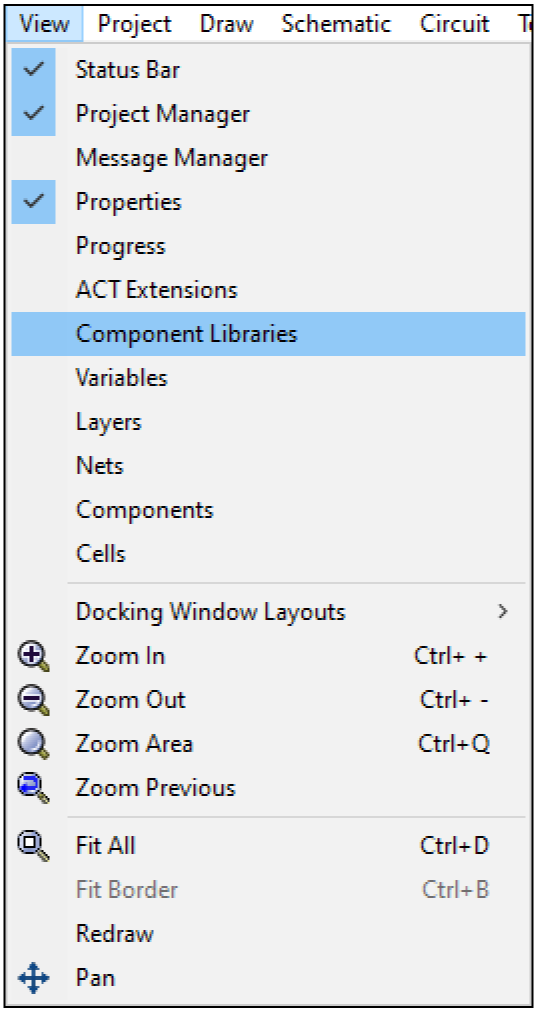
The FFT-based (i.e., Fast Fourier Transform) virtual EMI receiver (i.e., Time Domain Scan) probe reduces EMI/EMC compliance test time.

> **Note:** The following instructions assume the user has created a new project or opened an existing project, and the project has a design ready to be analyzed. Refer to the "Circuit Getting Started Guides" on page 1-8 for more information.

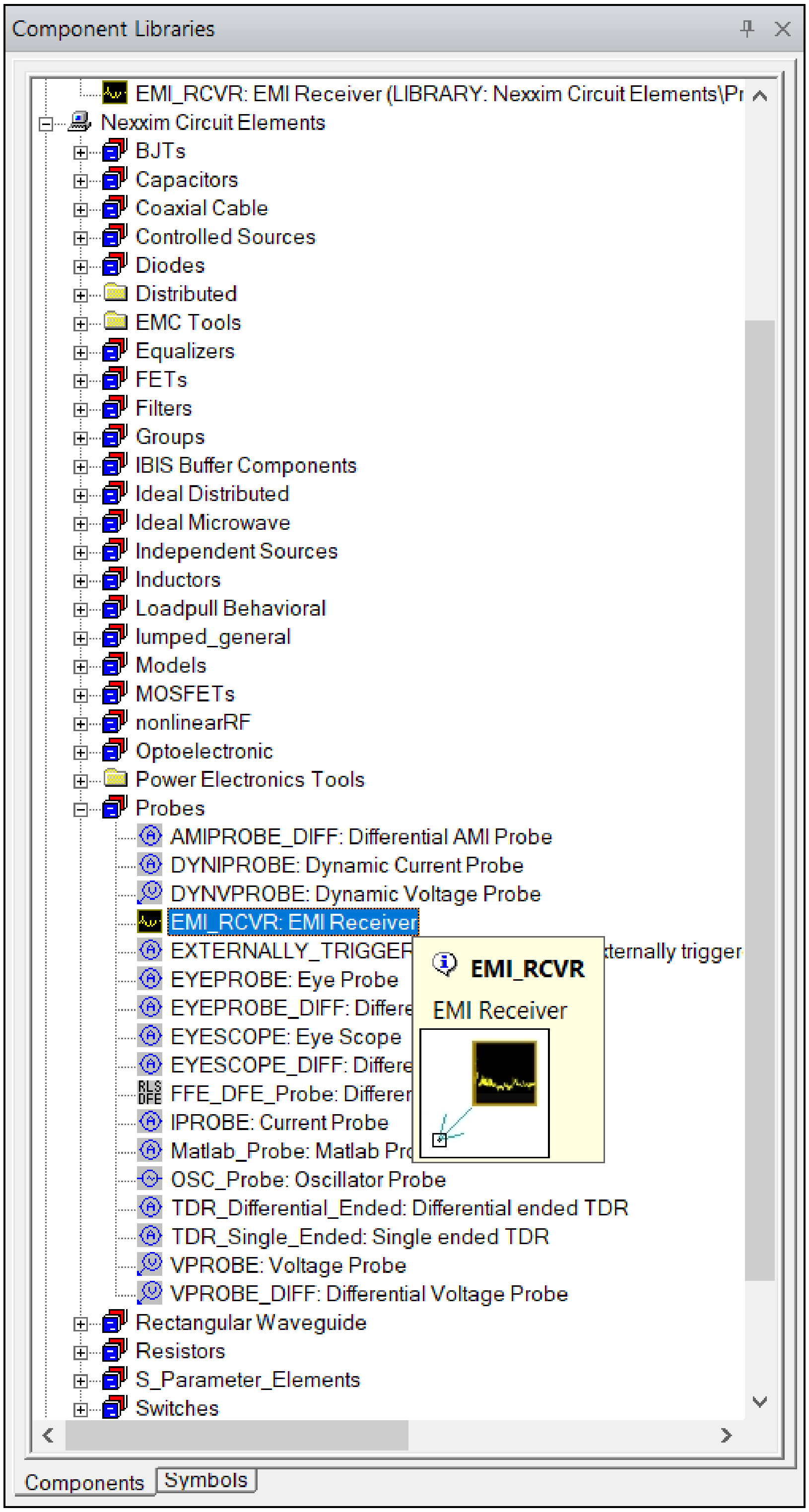
Adding EMI Receiver Probes to the Design

Complete the following steps to add one or more EMI receiver probes to a circuit design.

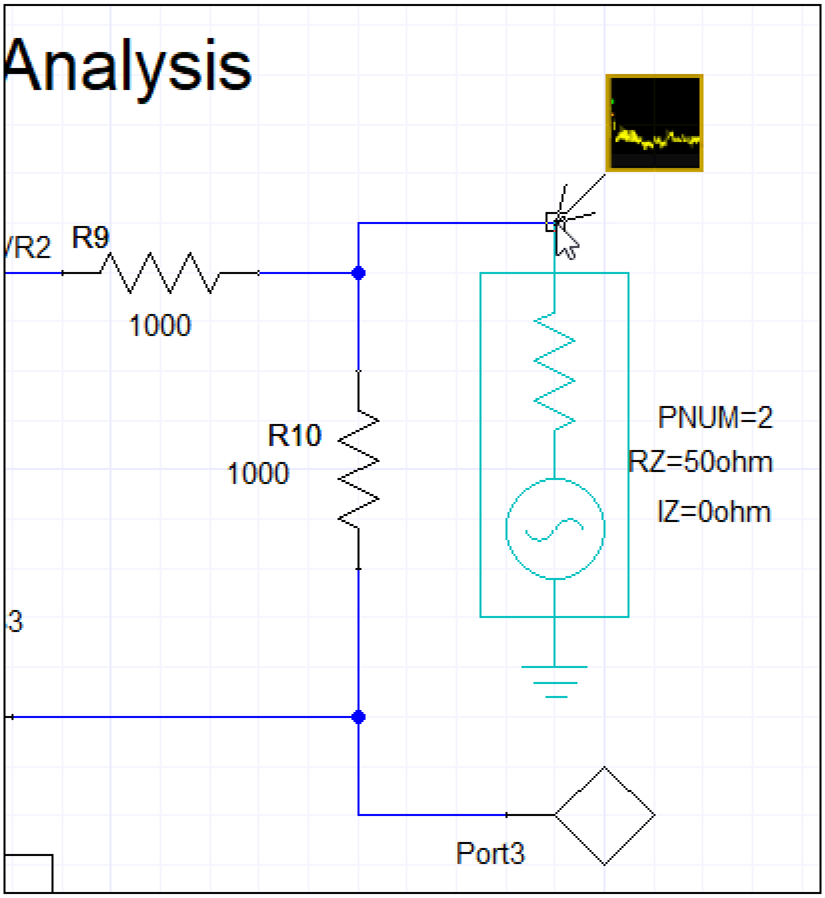
1. If necessary, open the **Component Libraries** window by navigating to View. A check appears adjacent to **Component Libraries** if the window is open.



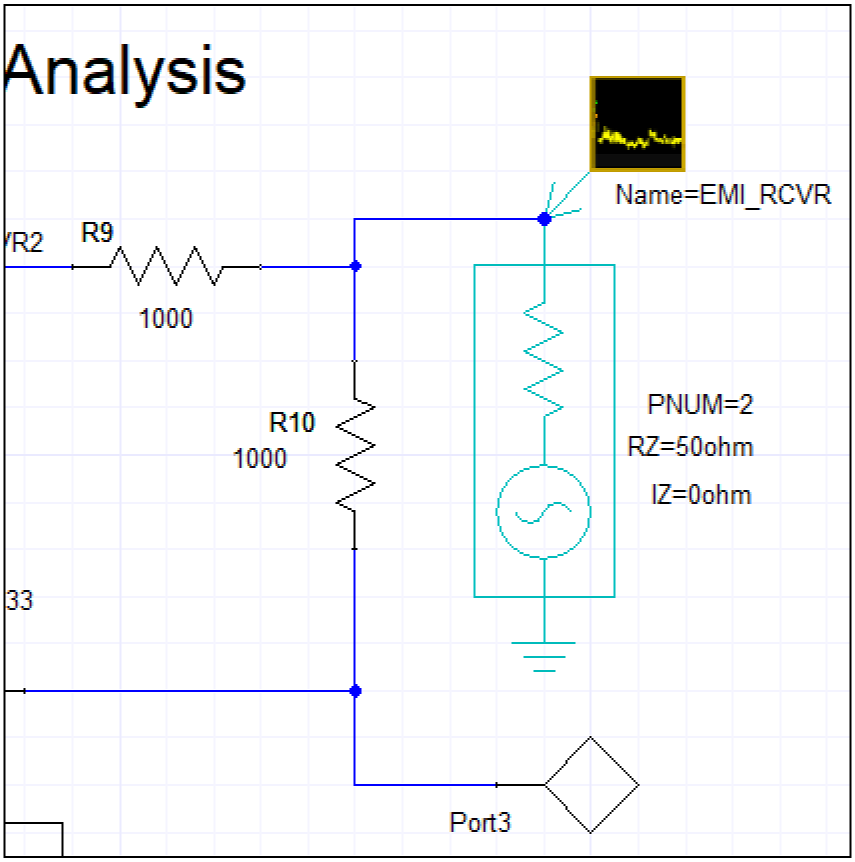
2. From the **Component Libraries** window, scroll to and expand **Nexxim Circuit Elements** > **Probes**.


Component Libraries
EMI_RCVR: EMI Receiver (LIBRARY: Nexxim Circuit Elements\Pr
Nexxim Circuit Elements
BJTs
Capacitors
Coaxial Cable
Controlled Sources
Diodes
Distributed
EMC Tools
Equalizers
FETs
Filters
Groups
IBIS Buffer Components
Ideal Distributed
Ideal Microwave
Independent Sources
Inductors
Loadpull Behavioral
lumped_general
Models
MOSFETs
nonlinearRF
Optoelectronic
Power Electronics Tools
Probes
AMIPROBE_DIFF: Differential AMI Probe
DYNIPROBE: Dynamic Current Probe
DYNVPROBE: Dynamic Voltage Probe
EMI_RCVR: EMI Receiver
EXTERNALLY_TRIGGER
EYEPROBE: Eye Probe
EYEPROBE_DIFF: Differe
EYESCOPE: Eye Scope
EYESCOPE_DIFF: Differe
FFE_DFE_Probe: Differer
IPROBE: Current Probe
Matlab_Probe: Matlab Pro
OSC_Probe: Oscillator Probe
TDR_Differential_Ended: Differential ended TDR
TDR_Single_Ended: Single ended TDR
VPROBE: Voltage Probe
VPROBE_DIFF: Differential Voltage Probe
Rectangular Waveguide
Resistors
S_Parameter_Elements
Switches
EMI_RCVR
EMI Receiver
Components
Symbols

3. **Hold+click EMI_RCVR: EMI Receiver** to select the probe and attach it to the cursor. Then drag the probe to an appropriate place in the **Schematic Editor**.



4. **Click** to drop the component at the chosen location.



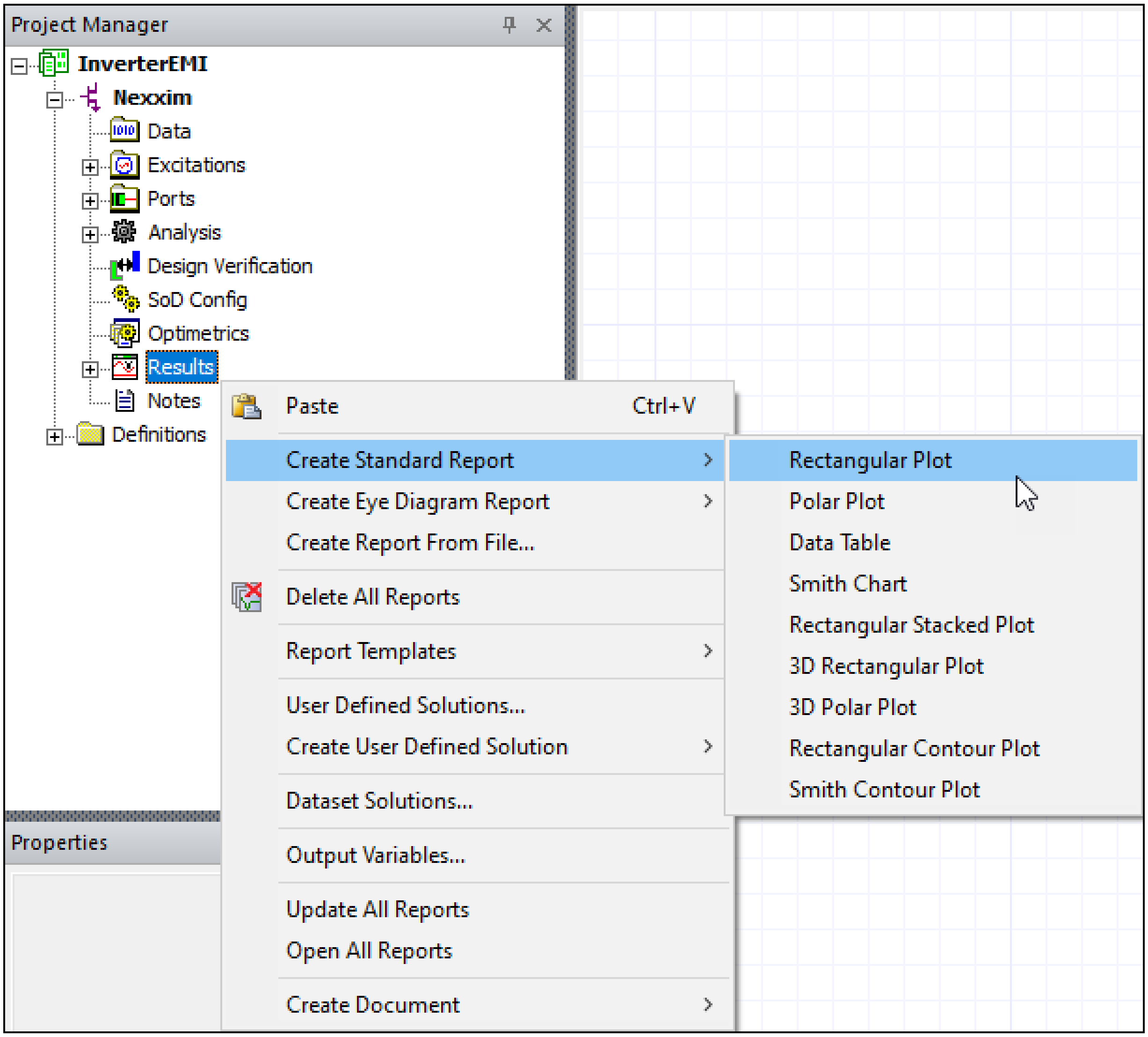
5. To add multiple EMI receiver probes, repeat steps **4-5**, as necessary.
6. If the user has not already done so, perform a **Transient Analysis Simulation** before continuing. (Refer to "Circuit Transient Analysis " on page 11-19).

Creating a Transient EMI Receiver Plot

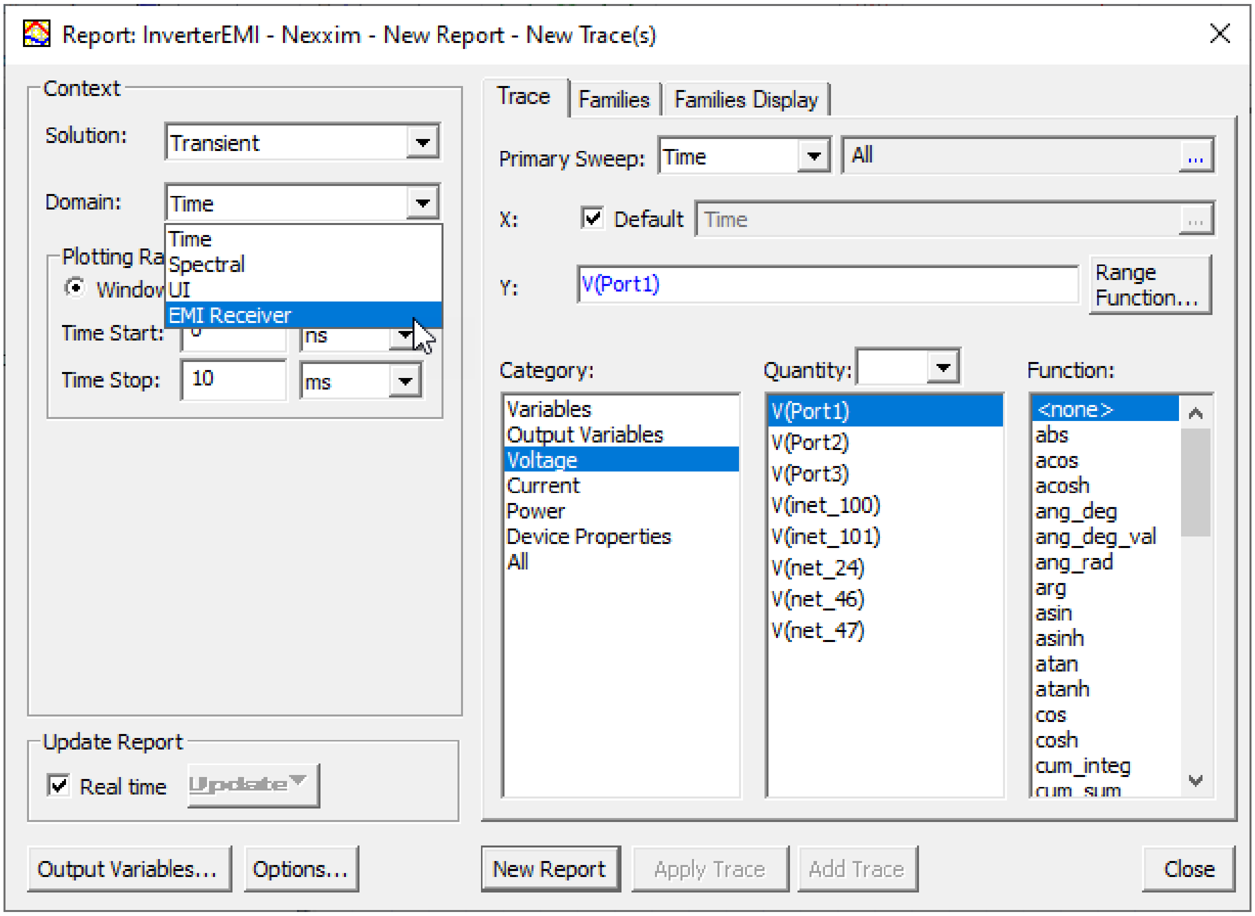
Complete the following steps to create a new transient EMI receiver plot report.

1. From the **Project Manager** window, expand **Project Tree** > **[*active design folder*]**. Then right-click **Results** and select **Create Standard Report** > **Rectangular Plot** to

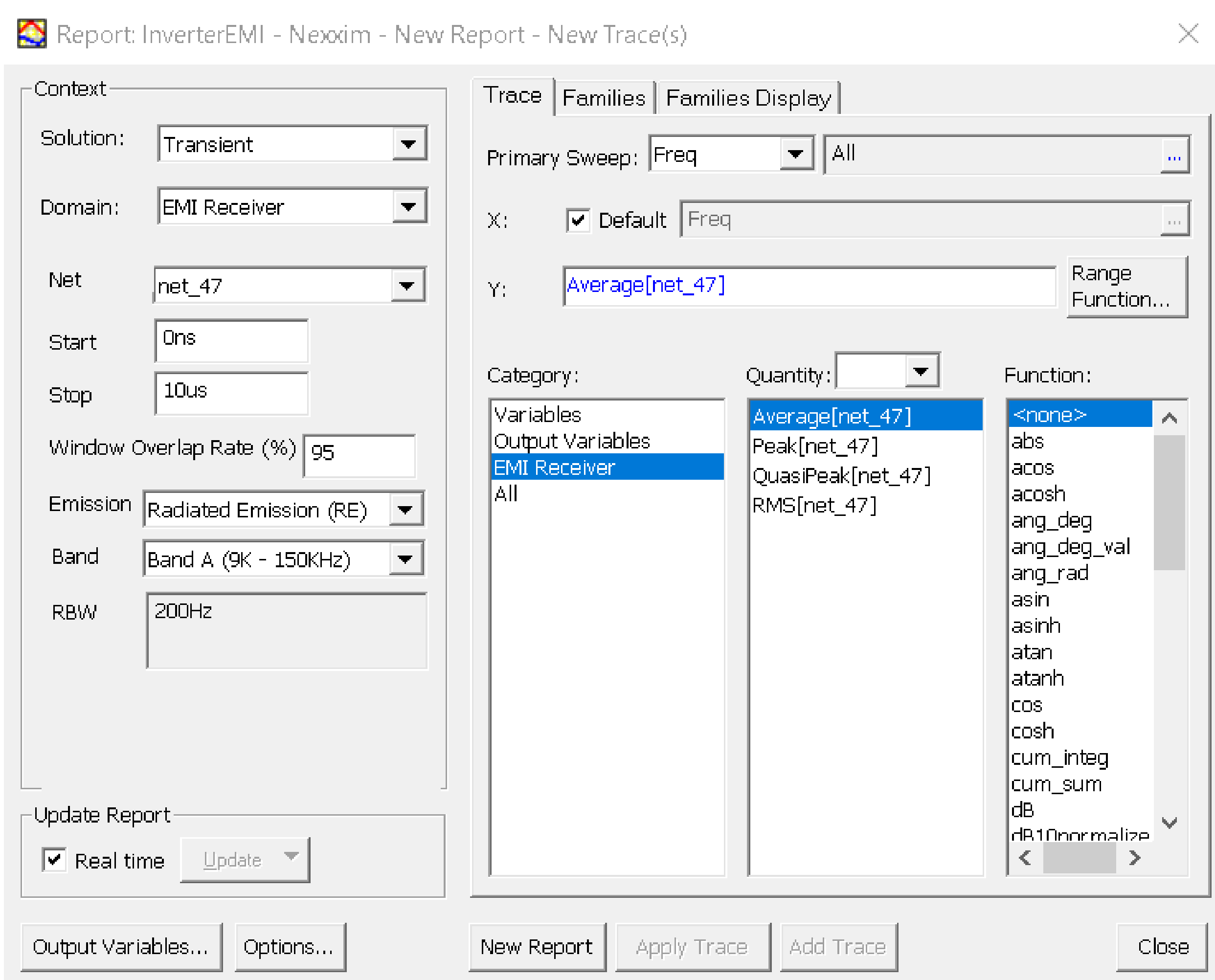
open the **Report** window.



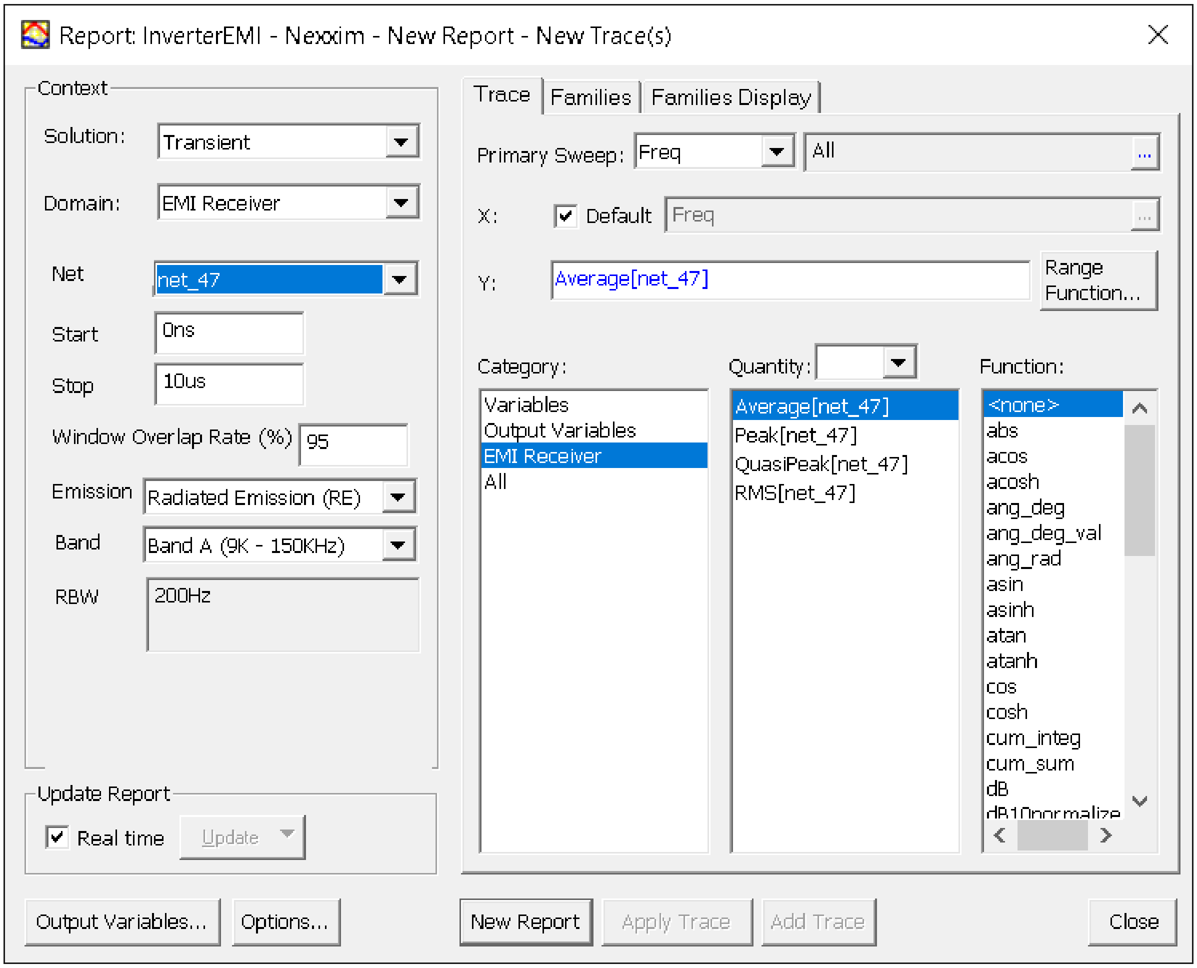
2. From the **Context** group box, select **EMI Receiver** from the **Domain** drop-down menu.



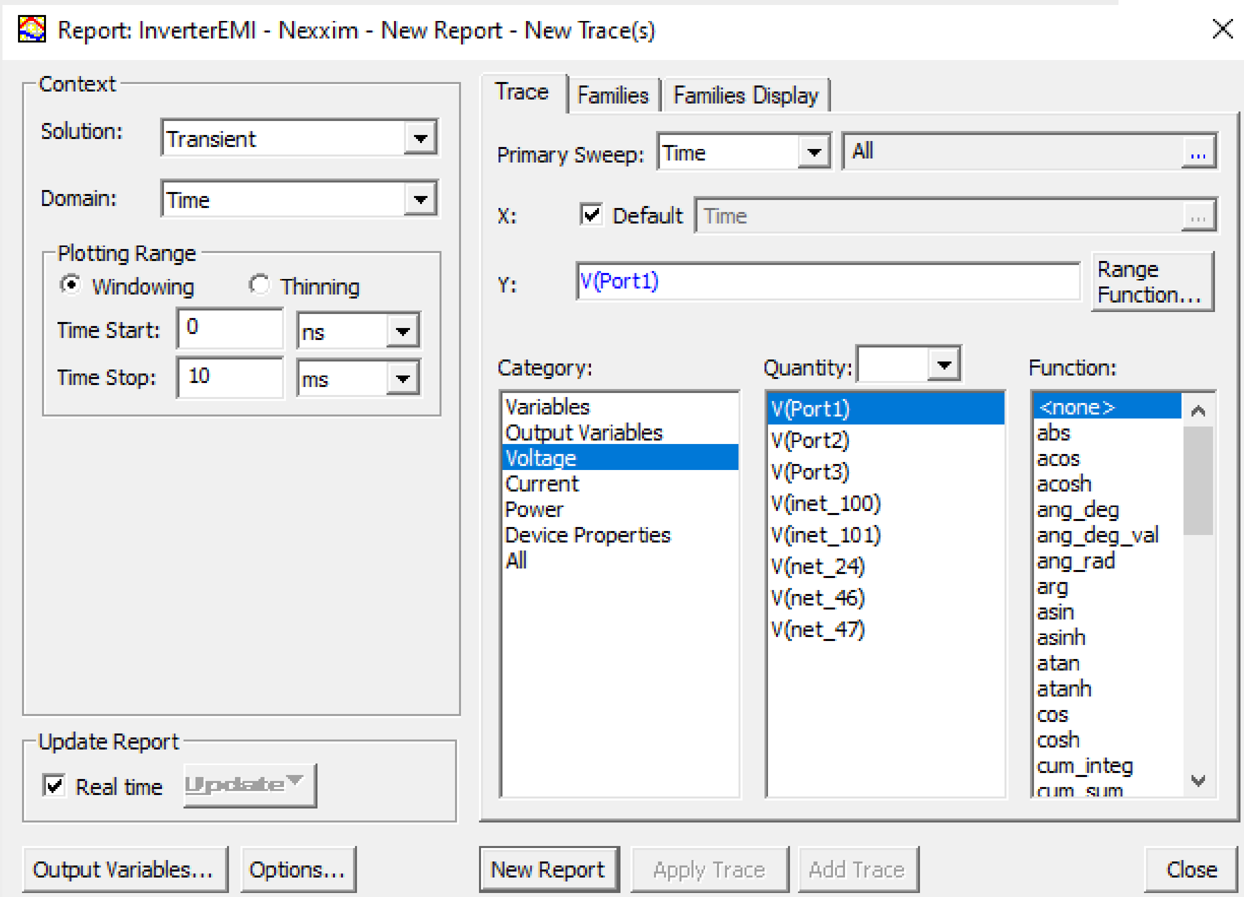
Choosing **EMI Receiver** reveals several new options beneath the **Domain** drop-down menu.



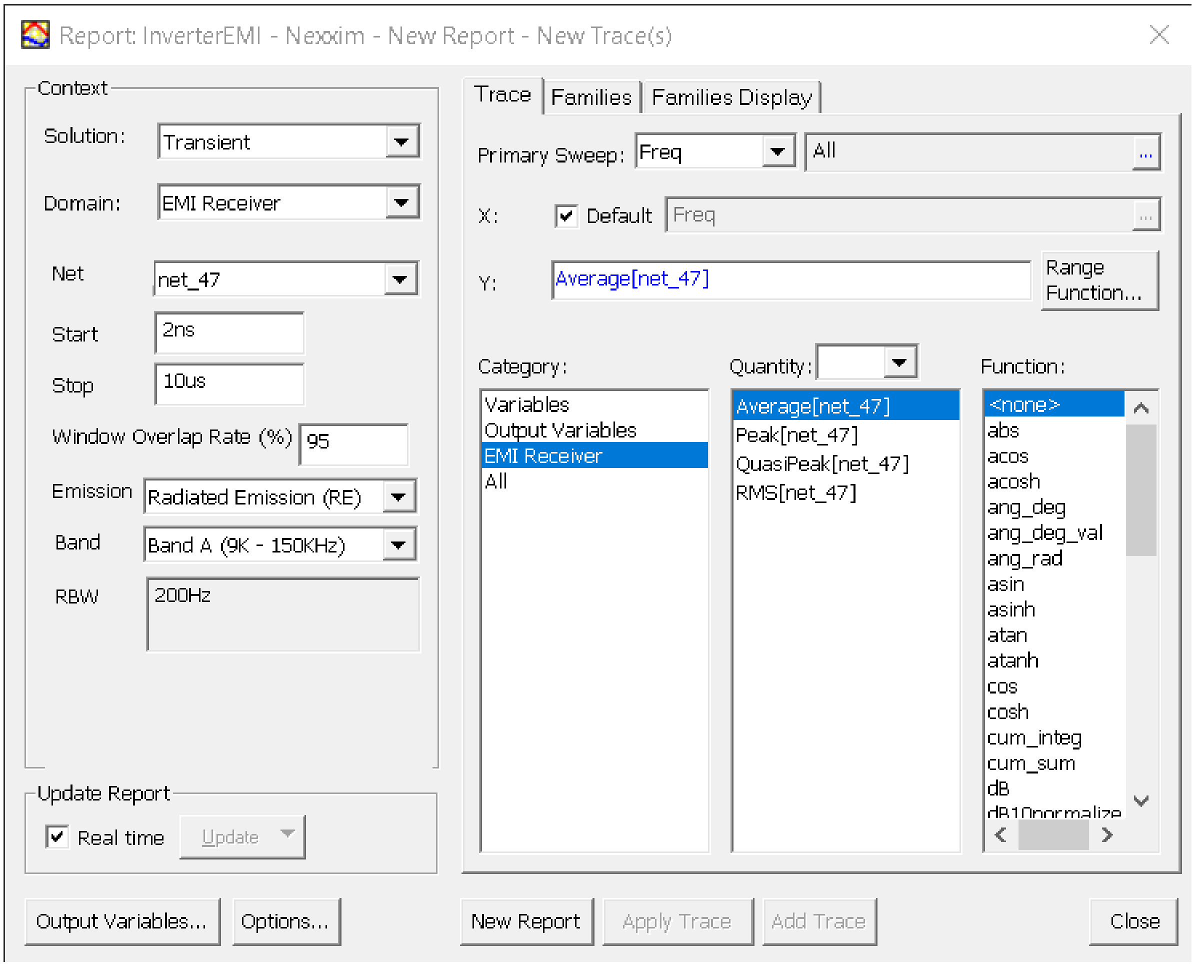
3. From the **Net** drop-down menu, select the net or port to which the appropriate EMI receiver component is connected (e.g., **net_47**). Selecting the net or port simultaneously changes the selections in the **Trace** tab > **Quantity** list.


Report: InverterEMI - Nexxim - New Report - New Trace(s)
Context
Solution:
Transient
Domain:
EMI Receiver
Net
net_47
Start
0ns
Stop
10us
Window Overlap Rate (%)
95
Emission
Radiated Emission (RE)
Band
Band A (9K - 150KHz)
RBW
200Hz
Update Report
Real time
Update
Trace
Families
Families Display
Primary Sweep:
Freq
All
X:
Default
Freq
Y:
Average[net_47]
Range Function...
Category:
Variables
Output Variables
EMI Receiver
All
Quantity:
Average[net_47]
Peak[net_47]
QuasiPeak[net_47]
RMS[net_47]
Function:
<none>
abs
acos
acosh
ang_deg
ang_deg_val
ang_rad
asin
asinh
atan
atanh
cos
cosh
cum_integ
cum_sum
dB
Output Variables...
Options...
New Report
Apply Trace
Add Trace
Close

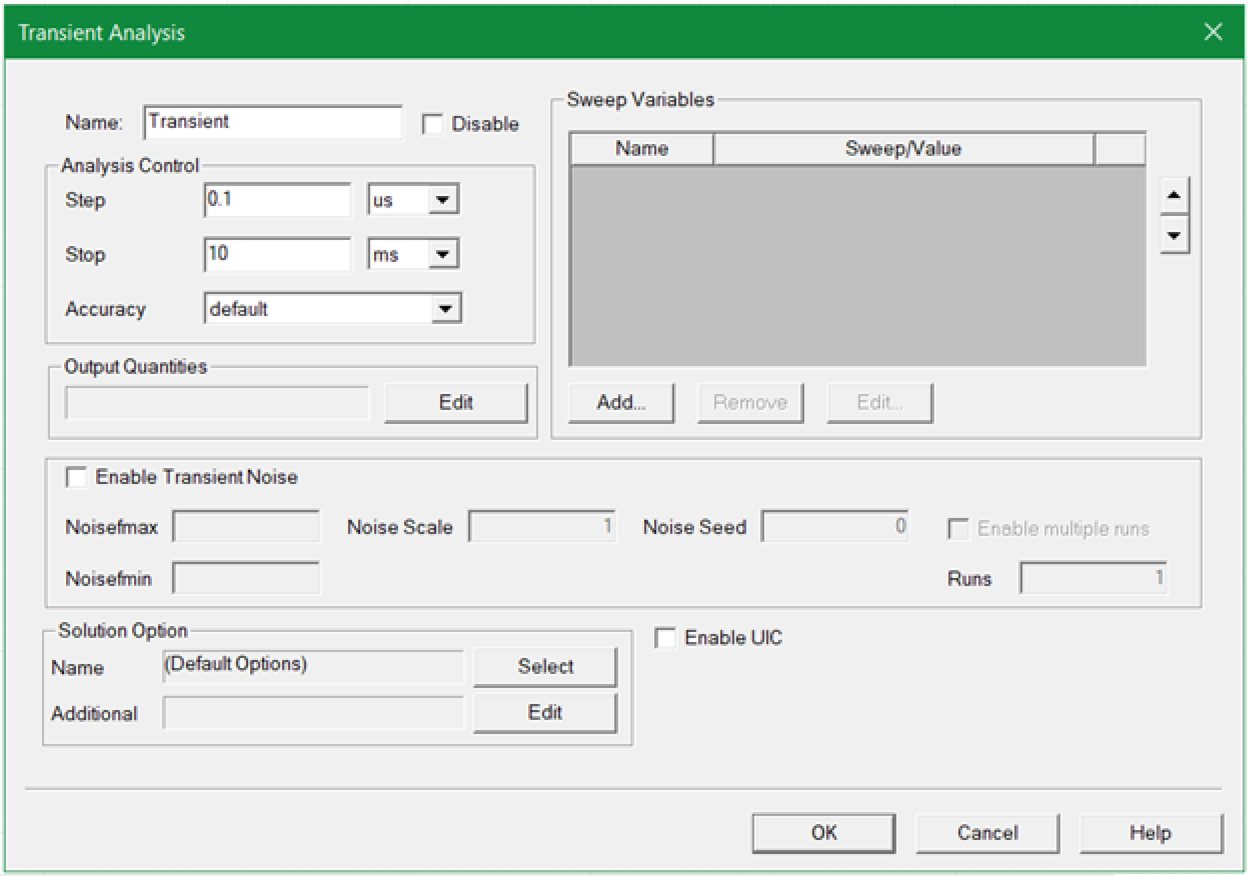
Note: EMI Receiver Probe reports are generated from the available transient signals recorded during transient analysis. To ensure that an EMI receiver probe is connected to a signal recorded for post-processing, repeat **step 1** to open the **Report** window. Ensure **Transient** is selected in the **Solution** drop-down menu, and **Time** in the **Domain** drop-down menu (i.e., the default selections). The **Quantity** list will populate with all of the signals available for post-processing reports.



4. The **Start** and **Stop** fields allow the user to select specific time frames within the transient analysis for EMI receiver analysis. Enter the desired values in the **Start** and **Stop** fields.


Report: InverterEMI - Nexxim - New Report - New Trace(s)
Context
Solution:
Transient
Domain:
EMI Receiver
Net
net_47
Start
2ns
Stop
10us
Window Overlap Rate (%)
95
Emission
Radiated Emission (RE)
Band
Band A (9K - 150KHz)
RBW
200Hz
Update Report
Real time
Update
Output Variables...
Options...
Trace
Families
Families Display
Primary Sweep:
Freq
All
X:
Default
Freq
Y:
Average[net_47]
Range Function...
Category:
Variables
Output Variables
EMI Receiver
All
Quantity:
Average[net_47]
Peak[net_47]
QuasiPeak[net_47]
RMS[net_47]
Function:
<none>
abs
acos
acosh
ang_deg
ang_deg_val
ang_rad
asin
asinh
atan
atanh
cos
cosh
cum_integ
cum_sum
dB
New Report
Apply Trace
Add Trace
Close

Note: The **Start**/**Stop** values should fall within the range entered in the **Step**/**Stop** fields during **Transient Analysis** setup.

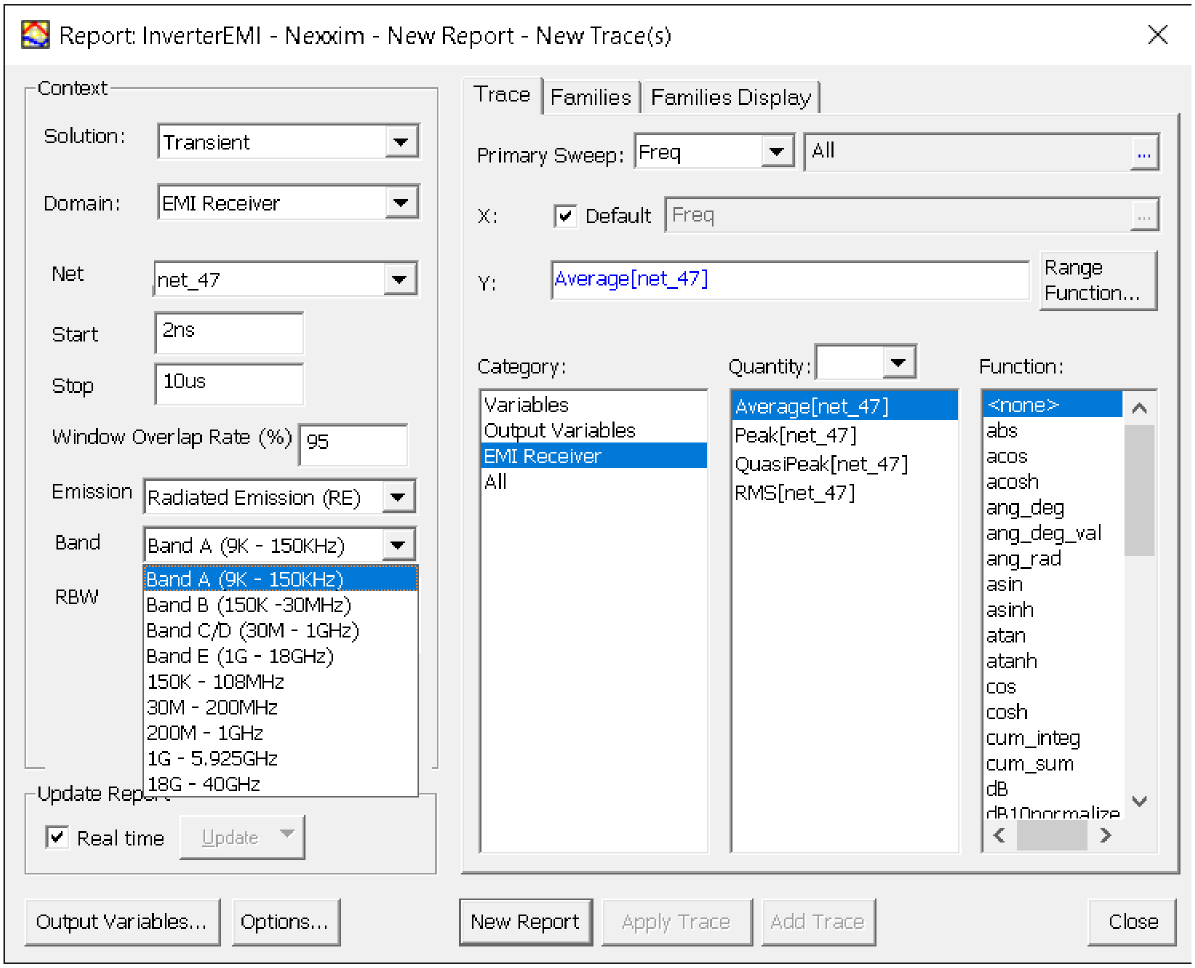


5. Gaussian window is the default window type as required in CISPR16-1-1 standard. If necessary, enter a different value between **50 - 97** in the **Overlap Rate (%)** field (default is **95**). The window overlap rate determines the time step of EMI analysis. A higher percentage will result in a smaller time step, at the cost of more computation and increased simulation time.

6. The field **Emission** has options **Radiated Emission (RE)** and **Conducted Emission (CE)** to choose from. The **Emission** information doesn't alter simulation settings or results, but shows up in plot name and title as part of a plot ID.

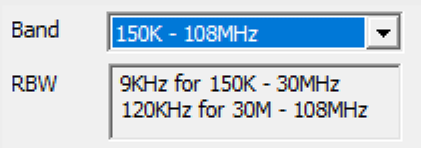
7. Select the appropriate band from the Band drop-down menu. Other than Band A, B, C/D and E as defined in CISPR16-1-1, CE or RE bands required for CISPR25 EMI/EMC analysis are included in the menu. Selecting a band simultaneously populates the **RBW**

field and the selections in the **Trace** tab>**Quantity** list, based on those defined in CISPR16-1-1.



Note:
1. For a sub-band of a CISPR16 Band, the RBW of the CISPR16 Band is applied to the sub-band. For example, Band 30M-200MHz is a sub-band of CISPR16 Band C/D(30M-1GHz), RBW=120KHz of Band C/D is applied to the sub-band 30M-1GHz.

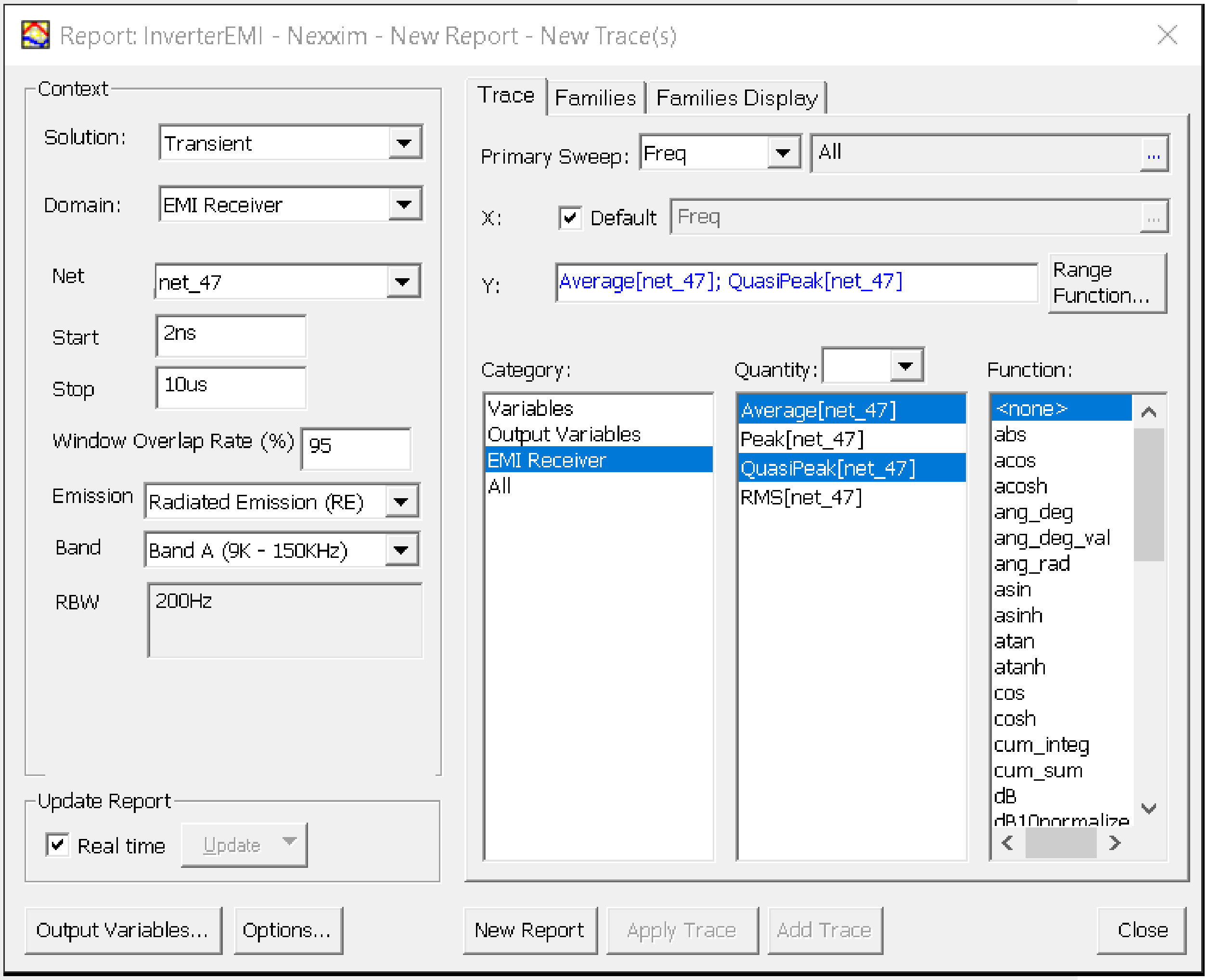
2. Considering Band 150k-108MHz crosses two CISPR16 Bands, RBW=9KHz for Band B(150K-30MHz) and RBW=120KHz for Band C/D (30M-1GHz) are applied in analysis.



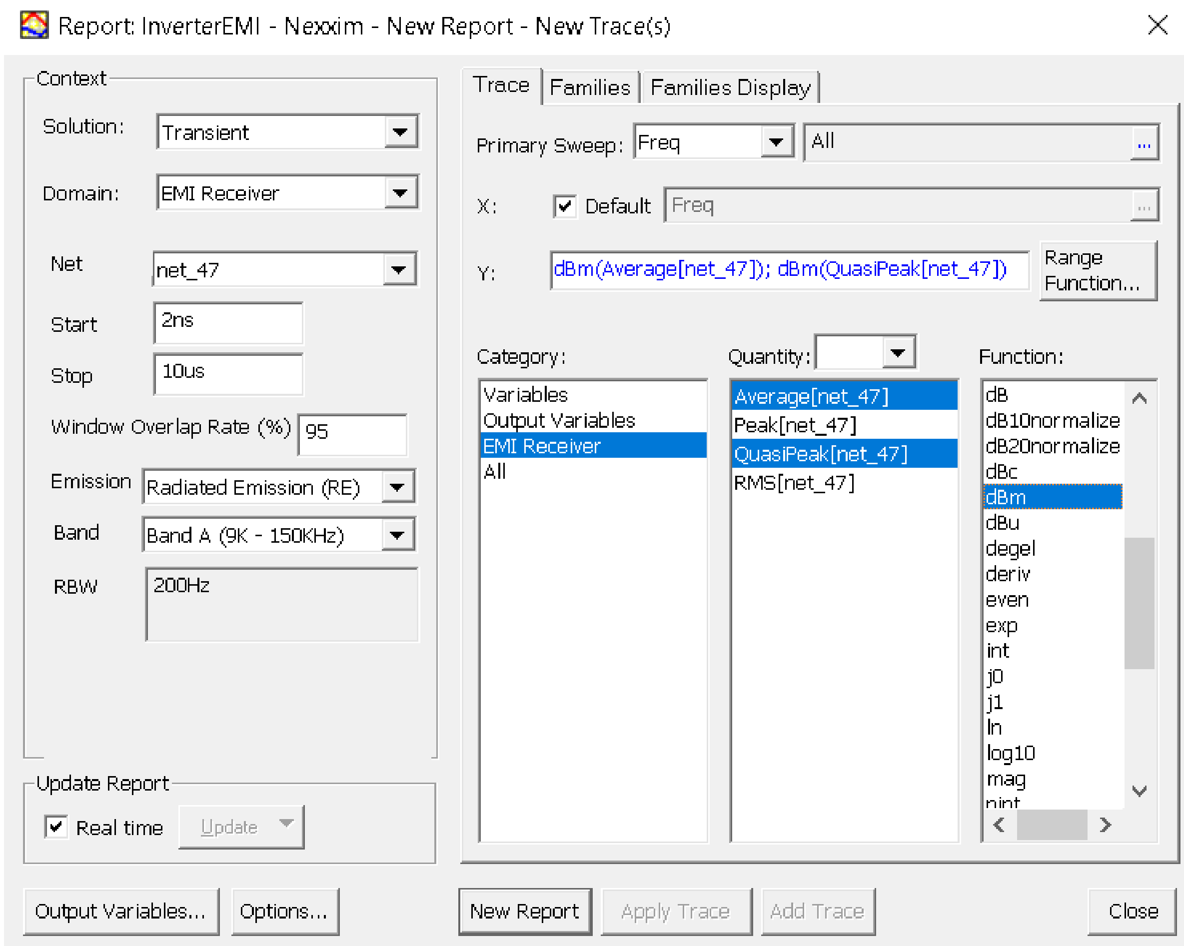
3. For bands above 1GHz, RBW =1MHz is applied following standards CISPR16-1-1 and MIL-STD-461.

8. Ensure **EMI Receiver** is selected from the **Category** list (selected by Default).
9. From the **Quantity** list, select one or more detector options (i.e., **Average**, **Peak**, **QuasiPeak**, and/or **RMS**). It should be noted that for Bands above 1GHz, QuasiPeak detection is not supported.

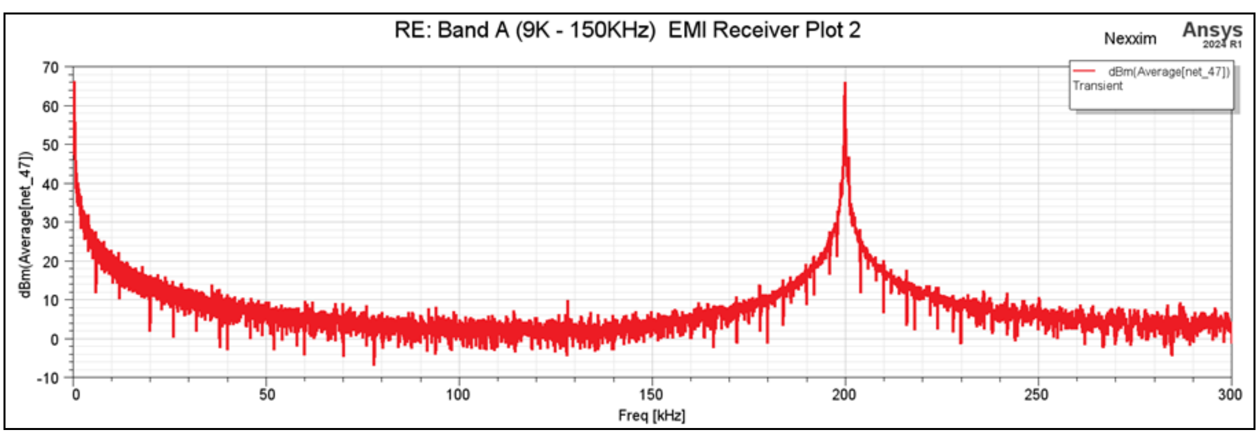
Note: Select multiple options from the **Quantity** list by **Ctrl+clicking** them.



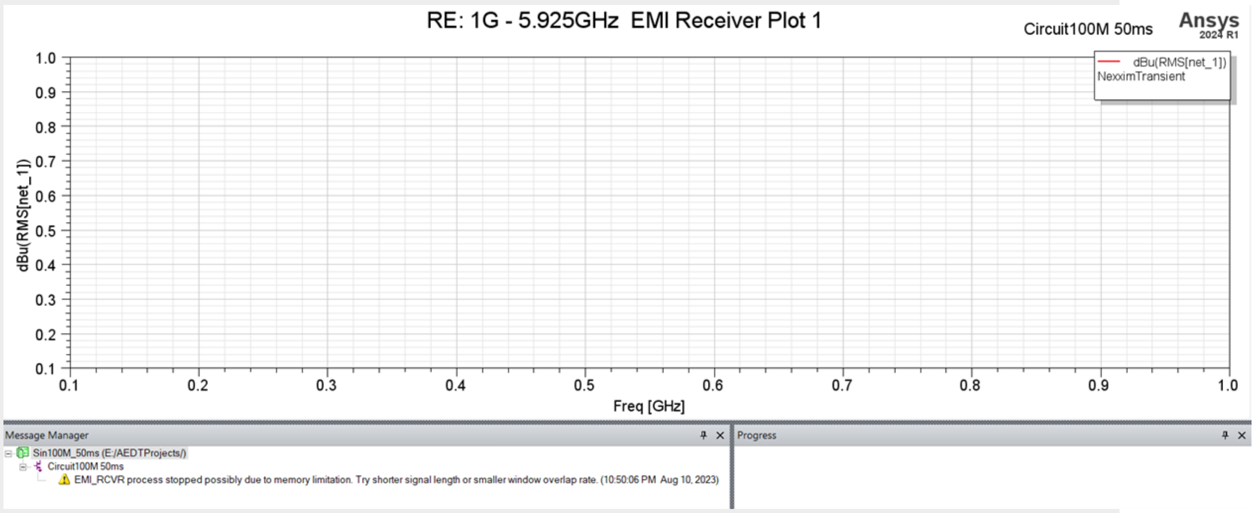
10. From the **Function** list, select **dBm** or **dBu**, typically, depending on signal strength (**<none>** is selected by default).



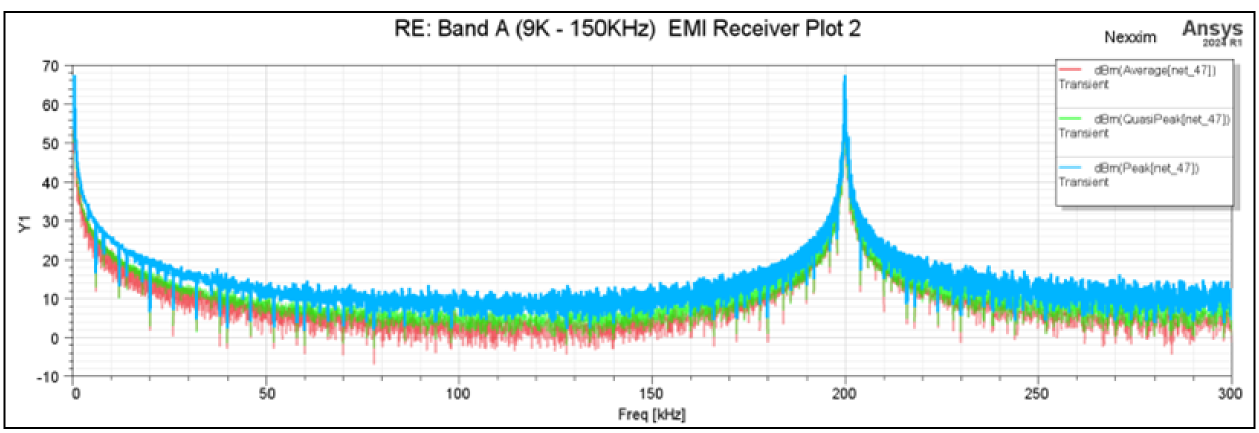
11. Click **New Report** to create the new **Transient EMI Receiver Plot *N***.



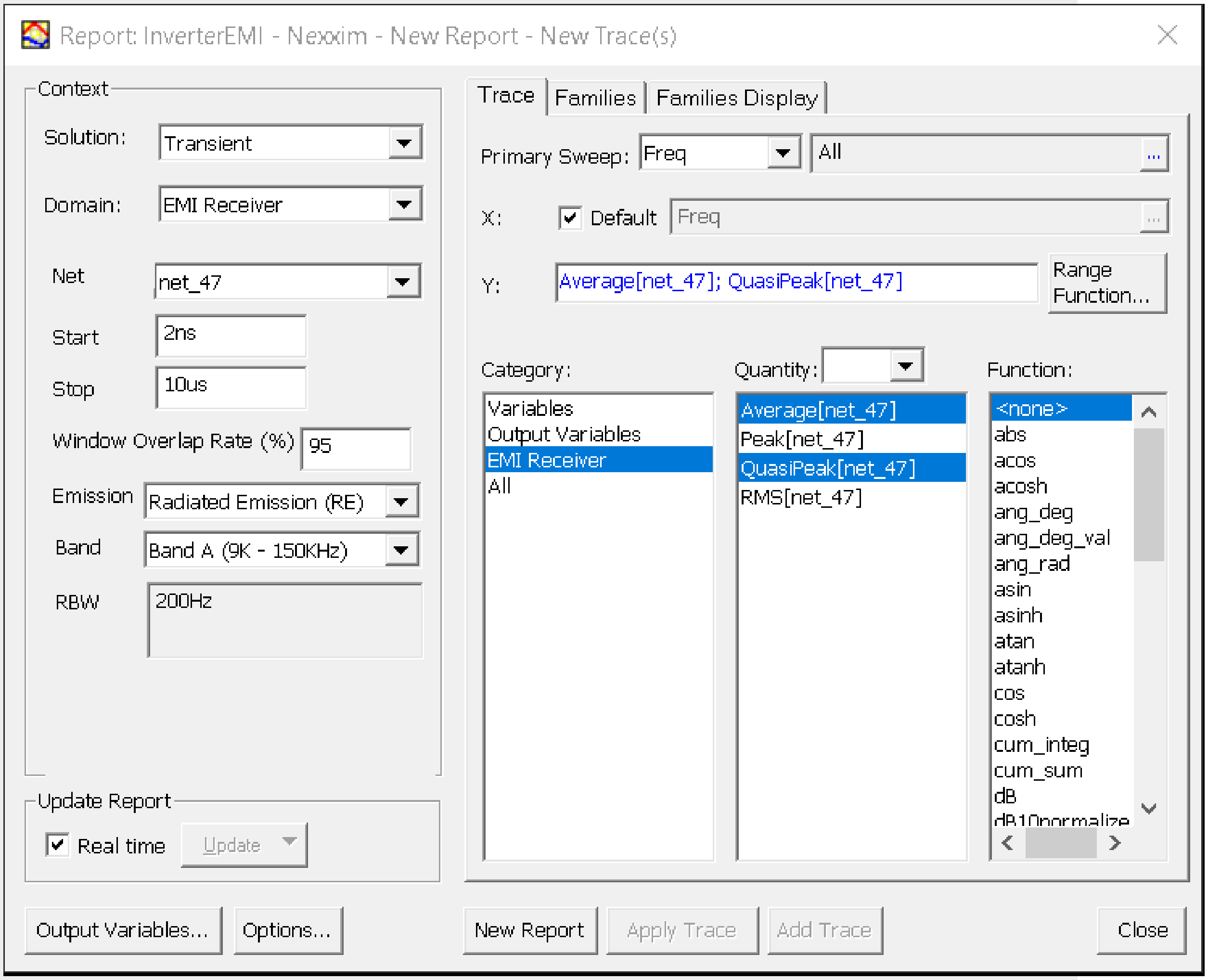
Note: Band selection, signal length, and window Overlap Rate are directly related to the computation time and memory complexity of EMI analysis. When the required memory exceeds the available memory, post processing could terminate with an error message. Please follow the directions in the message and try again with lower overlap rate and/or shorter signal length.



12. To close the **Report** window, click **Close**. If necessary, return to the **Report** window (i.e., from the **Project Manager** window, expand **Results**. Then right-click the appropriate report and select **Modify Report...**). Make changes to the selections in the **Report** window and click **Add Trace** to update the plot.



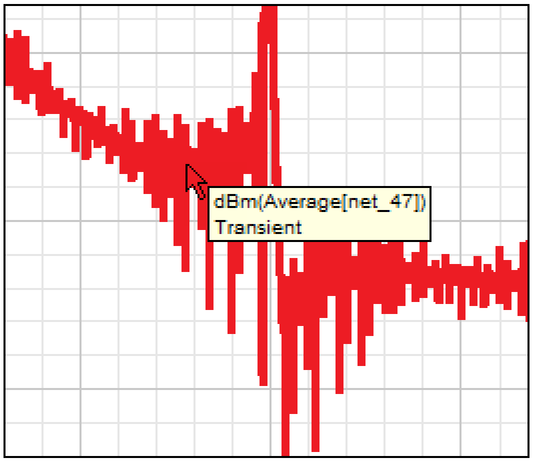
Note: After modifying the selections in the **Quantity** list, the **Function** list will revert to its default selection (i.e., **<none>**).



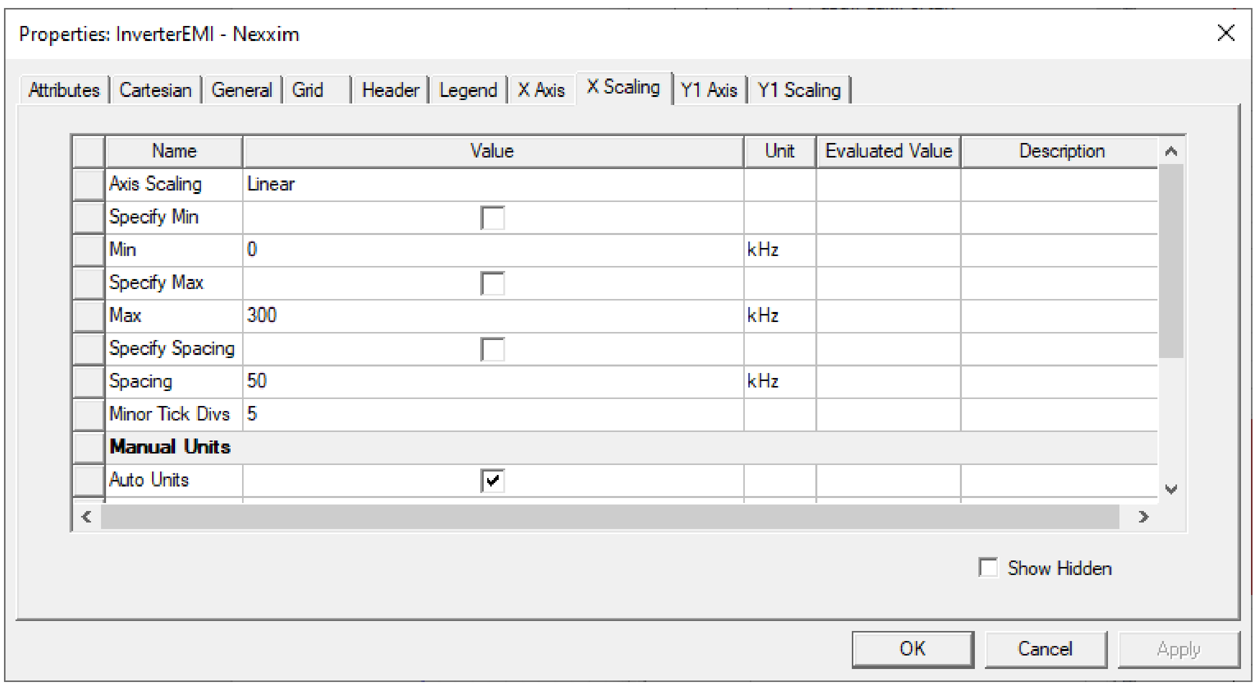
Selecting Log/Linear From the Properties Window

From the report's **Properties** window, users can change many settings. For example, complete the following steps to change **Axis Scaling** from **Linear** to **Log**.

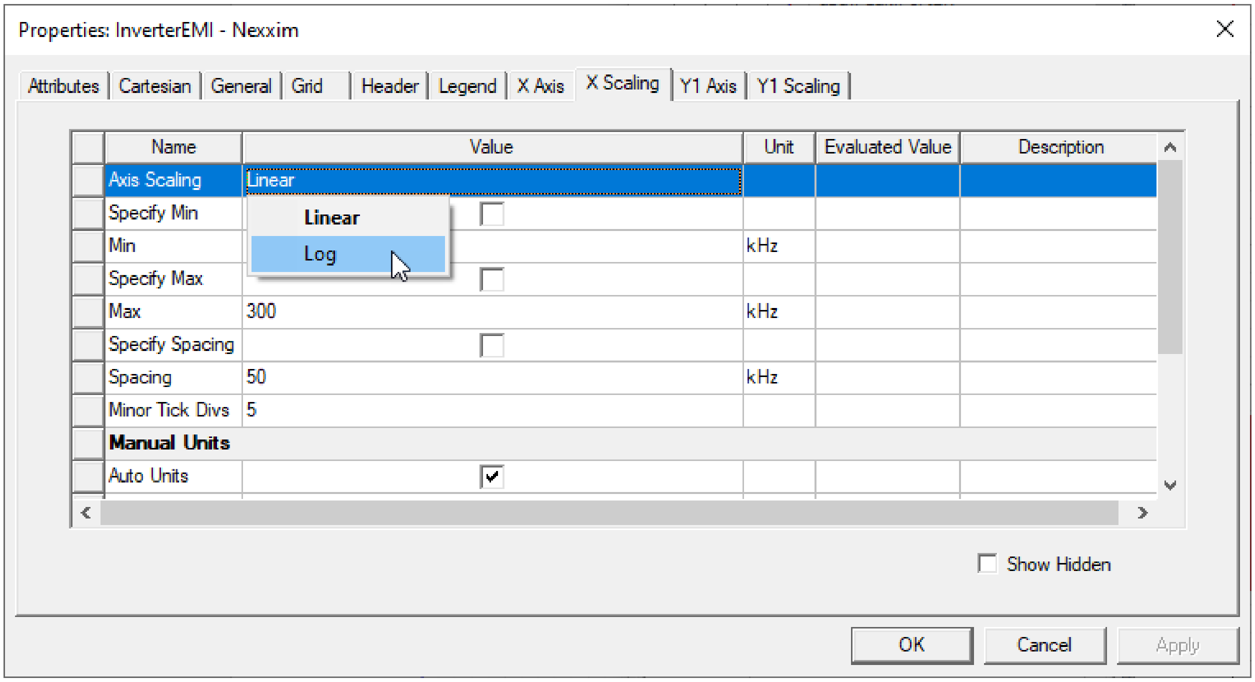
1. Double-click anywhere within the **Schematic Editor** to open the report's **Properties** window.



2. Navigate to the **X Scaling** tab.



3. From the **Axis Scaling** row, select **Log** from the **Value** drop-down menu (e.g., where **Linear** is currently).

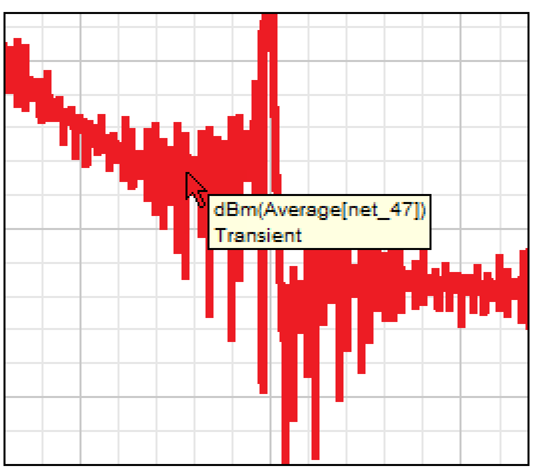


4. Click **Apply** to save the new parameters.
5. Click **OK** to close the **Properties** window.

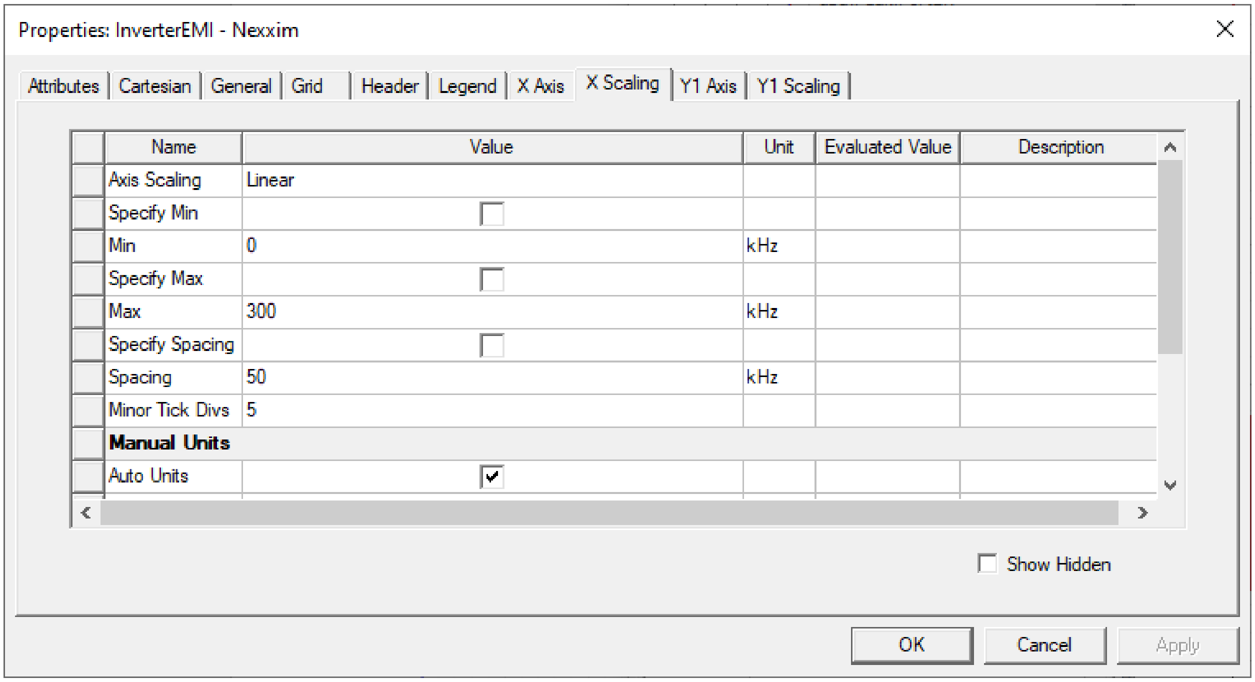
Selecting a New Minimum/Maximum From the Properties Window

Complete the following steps to modify the report's minimum/maximum resolution.

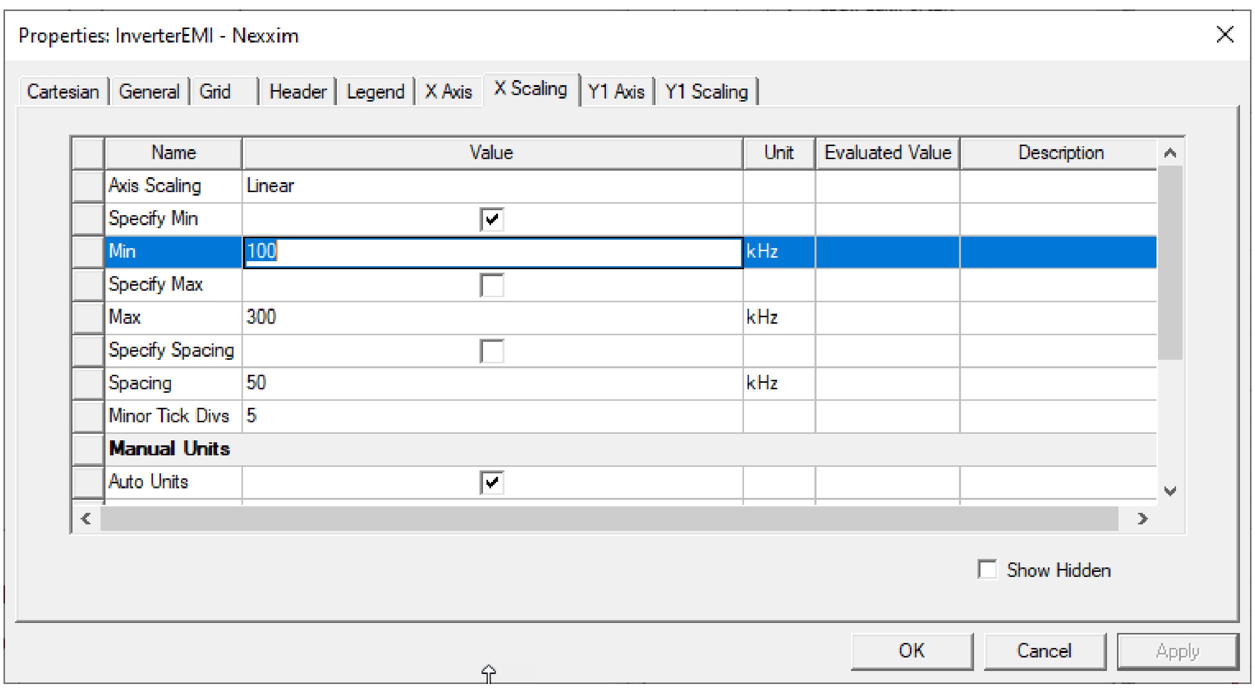
1. Double-click anywhere within the **Schematic Editor** to open the report's **Properties** window.



2. Navigate to the **X Scaling** tab.



3. Enter a new value in either the **Min** field, **Max** field, or both. Then check the corresponding boxes in the **Specify Min**/**Specify Max** fields.



4. Click **Apply** to save the new parameters.

5. Click **OK** to close the **Properties** window.

Computation of Traces Based UDO Calculations (2D and Circuit)

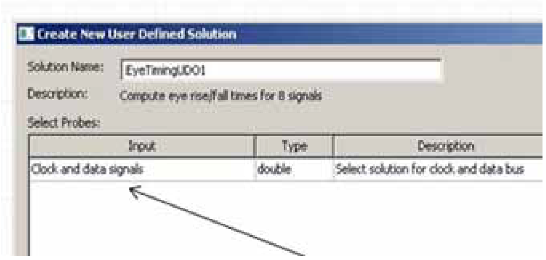
When traces that are based upon UDO outputs are computed, the data for probes is computed and passed to the UDO script for each design variation. Along with the probe data, the values of properties entered by user are also passed. The information about the UDO calculations that need to be computed is 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 uses the cached results instead of invoking the computation on UDO*.

Dimensions Reduction by UDO Calculations (2D and Circuit)

The probes in a UDO can have heterogeneous dimensions of data (e.g., one probe in a UDO can have data that is function of *n* intrinsic variable, while another probe in same UDO can have data that is function of m intrinsic variables, with *n* and *m* potentially being different). UDOs allow reducing any number of these intrinsic variables. In the previous example UDO calculations can be function of any number of intrinsic variables including not being function of any intrinsic variable at all. UDO calculations can also be a function 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 (2D and Circuit)

UDOs can also 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 having 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 end user to enter the probe information for each of those 10,000 quantities. Each dynamic probe is associated with a named probe that is entered by user, and information about solution, context and intrinsic variables is used from user selected probe, however multiple dynamic probes can be associated with same user selected probe. The dynamic probes are sampled 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, 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 on the UDO code.

user-defined Outputs: Python Script API (2D and Circuit)

A user-defined Output (UDO) extension is implemented as an IronPython script that defines a class with a specific name: **UDOExtension** which derives from a specific base class **IUDOPluginExtension** and implements its abstract methods.

UDO Extension IMPLEMENTATION

Optional Functions in IDO Extension Abstract Class

Data Types Used in Python Script

Working With Properties for UDO

Other Application Specific Classes Used in Python Scripts

user-defined Outputs: Messaging Methods

Using .NET Collection Classes and Interfaces in Python Scripts

UDO Extension Implementation (2D and Circuit)

The purpose, argument list and expected return types for each of the **IUDOPluginExtension** abstract methods, which the UDO author is expected to implement are shown in the following diagram.

- Import Statements
- UDOExtension Class
- IUDOPluginExtension Abstract Class

Import Statements (2D and Circuit)

The base class to be used and the types it uses in turn are contained in .NET assemblies. The use of these requires that the assemblies be imported into the UDO script: the following import statements should be added to the top of the python script:

from Ansys.Ansys.ModulePluginDotNet.Common.API import *

from Ansys.Ansys.ModulePluginDotNet.Common.API.Interfaces import *

from Ansys.Ansys.ModulePluginDotNet.UDO.API.Interfaces import *

from Ansys.Ansys.ModulePluginDotNet.UDO.API.Data import *

Optional Functions in IDO Extension Abstract Class (2D and Circuit)

The following functions, while a part of the IUDOExtension abstract class, have meaningful default implementations and are therefore optional. However, they can be overridden to take advantage of advanced functionality.

Validate(List<string> errorStringList,

Data Types Used in Python Script (2D and Circuit)

There are several types that you must use while authoring the python script. Some of them are used to pass data from UI to python script and provide interface for working with this data. Some are used to pass data from python script to UI.

To pass data from python script to UI the objects of the C# class must be created in python script using their C# constructors. The objects can then be set as functions return values or set to the output parameters using their API.

Working With Properties for UDO (2d and Circuit)

A property is the unit for collecting and using input on the user that is used to influence the UDO's Compute method. These are initially set up when the UDOs **GetInputUDSParams** method is called and are retrieved in the compute method.

There are 3 supported property types that could be used in the UDO script:

- INumberProperty to specify number properties (with unit support).
- IMenuProperty to allow the user to select from a list of options.
- ITextProperty to allow the user to enter text.

The IPropertyList type implements a collection for these properties.

IPropertyList Abstract class

IProperty Abstract class

INumberProperty Abstract class

ITextProperty Abstract class

IMenuProperty Abstract class

Other Application Specific Classes Used in Python Scripts (2D and Circuit)

This topic describes other classes used in Python scripts

Other Application Specific Classes Used in Python Scripts (2D and Circuit)

This topic describes other classes used in Python scripts:

- Constants Class
- UDSProbeParams Class
- UDSDynamicProbes Class
- QuantityInfo Class
- IProgressMonitor Abstract Class

Constants Class (2D and Circuit)

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

EPropType.PT_NUMBER
EPropType.PT_TEXT
EPropType.PT_MENU

Example:

paramType = Constants.kDoubleParamStr

propType = Constants.EPropType.PT_NUMBER

Other Application Specific Classes Used in Python Scripts (2D and Circuit)

This topic describes other classes used in Python scripts:

- Constants Class
- UDSProbeParams Class
- UDSDynamicProbes Class
- QuantityInfo Class
- IProgressMonitor Abstract Class

Constants Class (2D and Circuit)

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

EPropType.PT_NUMBER
EPropType.PT_TEXT
EPropType.PT_MENU

Example:

paramType = Constants.kDoubleParamStr

propType = Constants.EPropType.PT_NUMBER

Other Application Specific Classes Used in Python Scripts (2D and Circuit)

This topic describes other classes used in Python scripts:

- Constants Class
- UDSProbeParams Class
- UDSDynamicProbes Class
- QuantityInfo Class
- IProgressMonitor Abstract Class

Constants Class (2D and Circuit)

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

EPropType.PT_NUMBER
EPropType.PT_TEXT
EPropType.PT_MENU

Example:

paramType = Constants.kDoubleParamStr

propType = Constants.EPropType.PT_NUMBER

Other Application Specific Classes Used in Python Scripts (2D and Circuit)

This topic describes other classes used in Python scripts:

- Constants Class
- UDSProbeParams Class
- UDSDynamicProbes Class
- QuantityInfo Class
- IProgressMonitor Abstract Class

Constants Class (2D and Circuit)

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

EPropType.PT_NUMBER
EPropType.PT_TEXT
EPropType.PT_MENU

Example:

paramType = Constants.kDoubleParamStr

propType = Constants.EPropType.PT_NUMBER

Other Application Specific Classes Used in Python Scripts (2D and Circuit)

This topic describes other classes used in Python scripts:

- Constants Class
- UDSProbeParams Class
- UDSDynamicProbes Class
- QuantityInfo Class
- IProgressMonitor Abstract Class

Constants Class (2D and Circuit)

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

EPropType.PT_NUMBER
EPropType.PT_TEXT
EPropType.PT_MENU

Example:

paramType = Constants.kDoubleParamStr

propType = Constants.EPropType.PT_NUMBER

Using .NET Collection Classes and Interfaces in Python Scripts (2D and Circuit)

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 topic 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 are Python lists. You can also check for membership using 'in'. To get .Net Array and IList sizes 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 xrange(listSize)
        print sweepsNamesList[i]
```

Checking for membership:

```
if ‘Time’ in sweepsNamesList:
        doThis()
else:
        doThat()
```

For .NET IDictionary, 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 returns keys, .Net Dictionary returns .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 and time-consuming.

Example:

aPythonList = list(dotNetArray)

aPythonList = list(dotNetList)

aPythonDict = dict(dotNetDictionary)

user-defined Outputs: Messaging Methods (2D and Circuit)

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 Manager** window using the appropriate icon.

These functions can also be used for debugging purposes.

- **AddErrorMessage(string)**: Call this method to convey an error condition to the user.
- **AddWarningMessage(string)**: Call this method to convey a warning message: typically used for conditions that are not ideal but can be tolerated by the script.
- **AddInfoMessage(string)**: Call this method to convey an informational message to the user. This is the call to use when outputting messages for debugging purposes.

```
##################################################################

# Imports

##################################################################

from Ansys.Ansys.ModulePluginDotNet.Common.API import *

from Ansys.Ansys.ModulePluginDotNet.Common.API.Interfaces import *

from Ansys.Ansys.ModulePluginDotNet.UDO.API.Interfaces import *

from Ansys.Ansys.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 are simple doubles
    # 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. Only one double quantity is needed
    param1 = UDSProbeParams("probe1",
            "double quantity probe",
            Constants.kDoubleParamStr,
            "", "")
```

```
    udsParams.Add(param1)

    # Add the properties the user must supply
    # In this case, ask for a start/end range for
    # X parameters. Since defaults cannot be reasonably provided
    # as the sweep limits are unknown,
    # ask if the limits can 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
# There are no sweeps because they have been reduced.
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 was 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(sweeps[0], "probe1")

# 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 computing
# all the quantities.
        minVal = 0
        maxVal = 0
        avgVal = 0

# Check if a range computation needs to be performed
if useXRangeProp == "Yes":
seenAny = False
avgSum = 0
count = 0

# zip is used since sweep data must also be pulled in
# 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 is
# 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 there is only one item.
outData.SetDoubleQuantityData("min_val", [minVal])
outData.SetDoubleQuantityData("max_val", [maxVal])
outData.SetDoubleQuantityData("avg_val", [avgVal])


# The operation is complete.
return True
```

user-defined Outputs: Script Organization (2D and Circuit)

As described in the *Introduction* topic, the UDO scripts should all reside under the **UserDefinedOutputs** folder under either of the three library locations (system, user or personal).

Using Script Libraries

Additional .NET Assemblies

Using Script Libraries (2D and Circuit)

If you decide that you need base classes or additional data files to organize your UDOs better, 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 is automatically added to the system include path (any support script files from any Lib directory through the top level UserDefinedOutputs directory can be imported) .

Using Additional .NET Assemblies (2D and Circuit)

Because the UDO functionality uses IronPython, the full .NET ecosystem is accessible. 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
   - UserDefinedDocuments\Lib,
   - UserDefinedDocuments\a\Lib,
   - UserDefinedDocuments\a\b\Lib
   - UserDefinedDocuments\a\b\c\Lib
3. Add the following line to your python script
   - Import clr
   - clr.AddReference(“com.Acme.UDOLib”)
   - import com.Acme.UDOLib –or-- from com.Acme.UDOLib import * *etc*

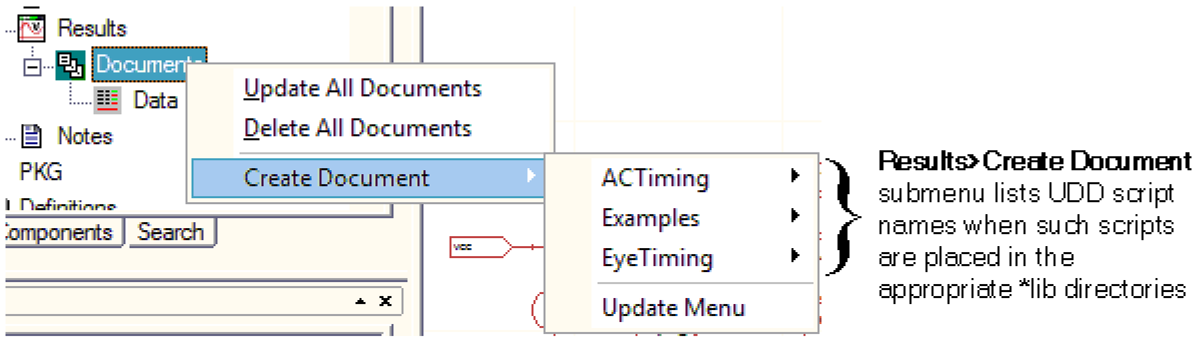
If for some reason you cannot place the .NET assemblies into a Lib directory under UserDefinedDocuments, you need to do a couple more steps before step 3 listed above.

Import sys

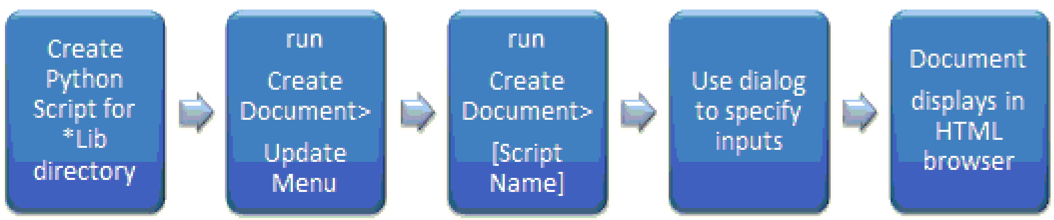
sys.path.append(“full path to your .NET assembly location”)

user-defined Documents (UDDs) (2D and Circuit)

user-defined documents (UDDs) are custom reports that you define through IronPython scripts. Once placed in a Lib directory, access the scripts via the **Create Document** command. The scripts describe a **Create User Defined Document** window that lets you specify trace and solution inputs. After you confirm your input selections, an xml, html and pdf document is generated. A web browser window opens to open the generated html file. The created document appears in the **Project Manager** window, under Results in the Documents folder.



The general UDD process flow is as follows.



The UDD python scripts must be placed in the **UserDefinedDocuments** directory under either of **syslib**, **userlib** or **Personallib** with any subdirectory structure needed. The Lib directory can contain python scripts that have common code that other scripts can use.

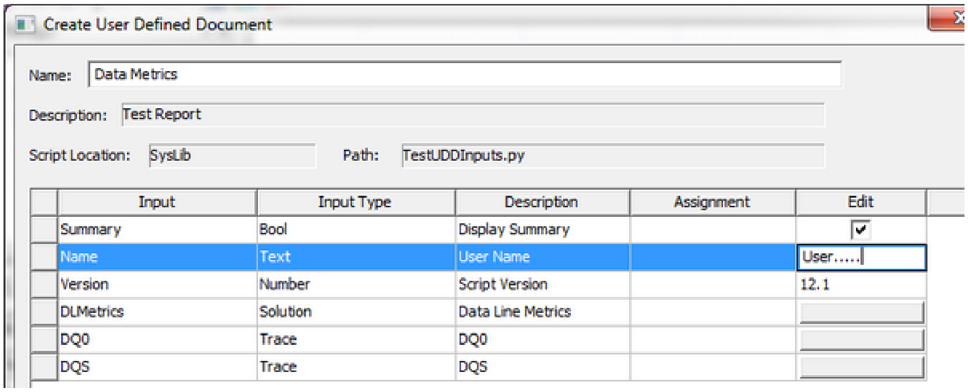
Use **Results>Create Document>Update Menu** to refresh the menu to include the new UDD scripts that have been copied to syslib, userlib or Personallib, or to exclude them if they have been deleted, after the launch of desktop.

The UDD scripts that are in syslib/UserDefinedDocuments, userlib/UserDefinedDocuments or Personallib/UserDefinedDocuments become available through the **Results >Create Document** menu.

Create user-defined Document window Inputs (2D and Circuit)

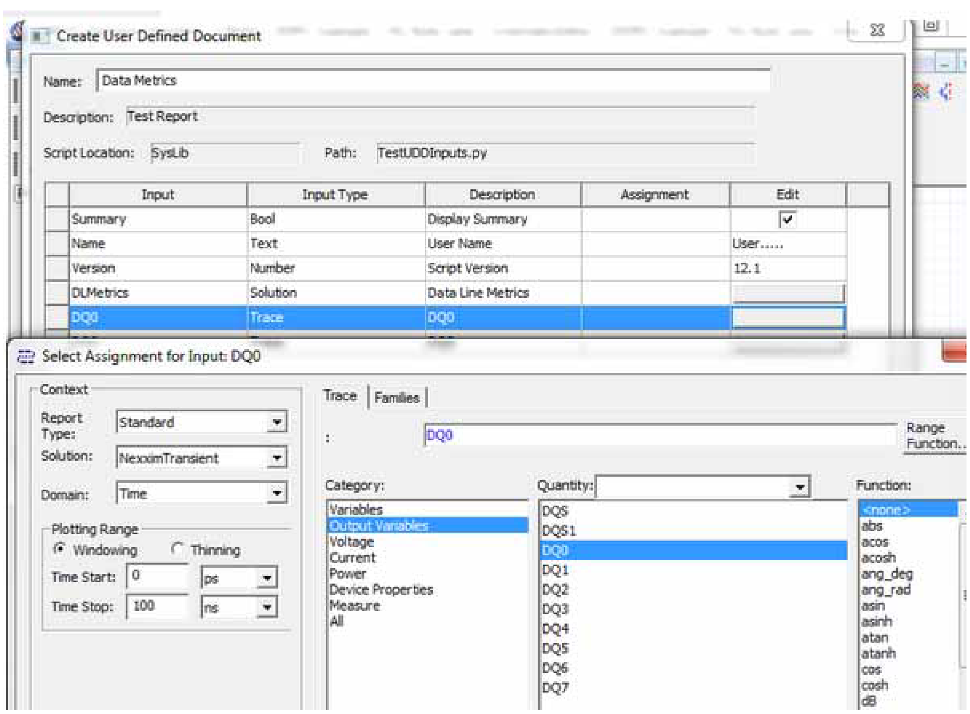
user-defined documents allow data from traces, solutions and report types as inputs. A UDD can specify the named inputs for which you select or enter the values in the **Create user-defined**

Document window that displays when you run **Results>Create Document>** *<scriptName>*.



Input Types can be of Boolean, number, text, trace or solution type. The boolean, number and text type can be given a default value that can be interactively overriden when the document is created or modified. For example, select a trace when you create or modify a UDD document. The trace data is available to the user and can be accessed on the python script.

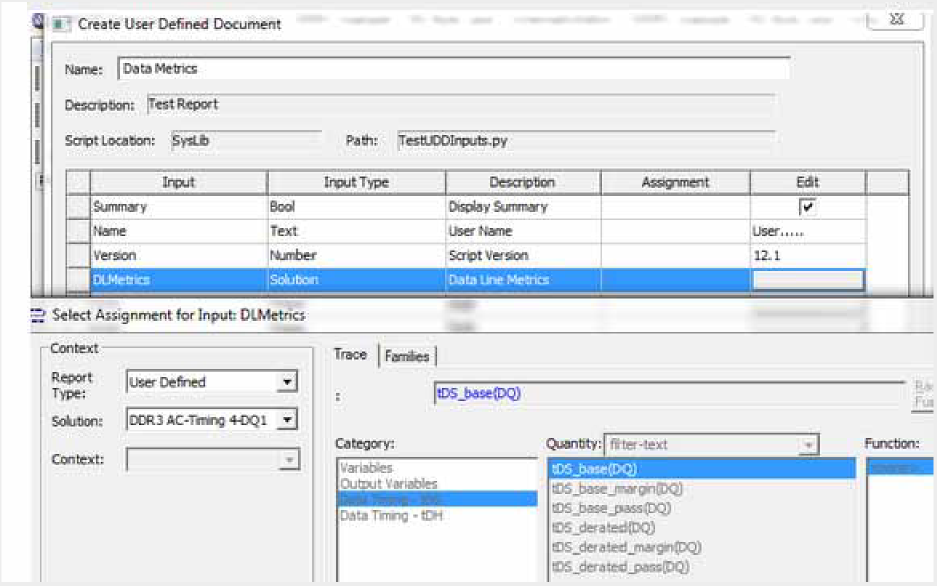
At the time of selection choose on the **Reporter** window, the report type (Standard, Eye Diagram, user-defined), solution name, context and the quantity for which you want the trace data.



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 on the python script.

At the time of selection choose on the reporter window, 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.

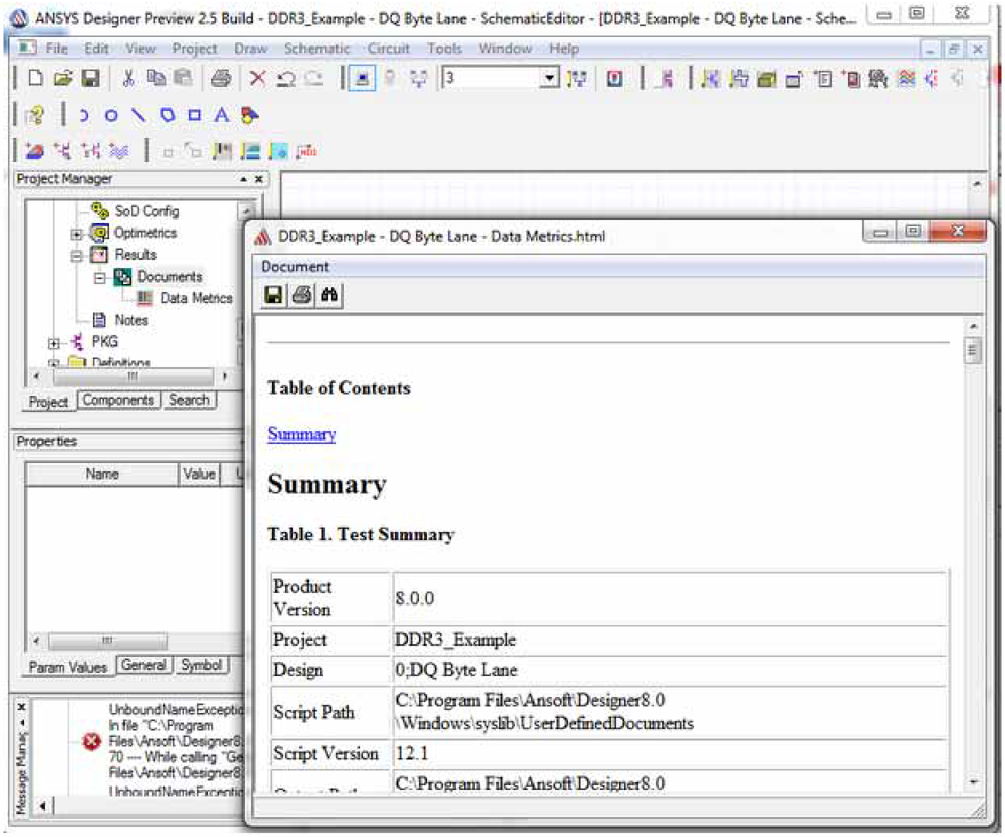
Note: The category/Quantity/Function portion of the window is unavailable for user input.



UDD Document Creation and Display (2D and Circuit)

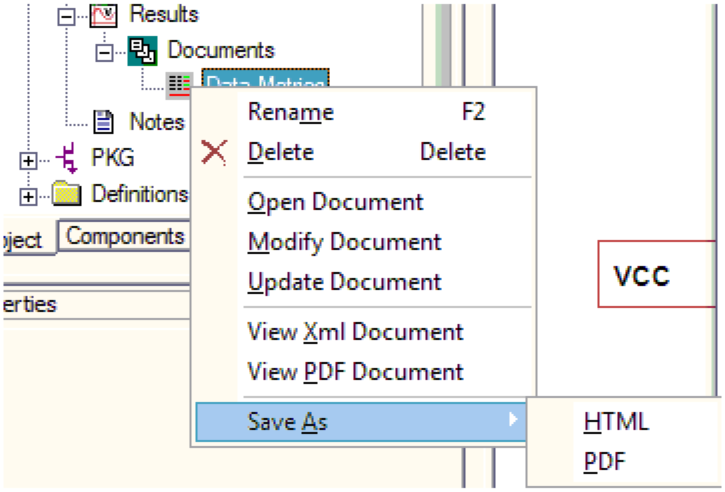
After all the input selections for a UDD are confirmed, based on the script, an xml, html and pdf document is generated based on the inputs provided by the user. (The xml, html and pdf generation is based on specific calls in the python script, which are explain in a following topic). A web browser window also opens to open the generated html file.

The created document is placed under a new folder named "Documents" under the "Results" folder. All documents that are created by the user for the design is placed under this folder.



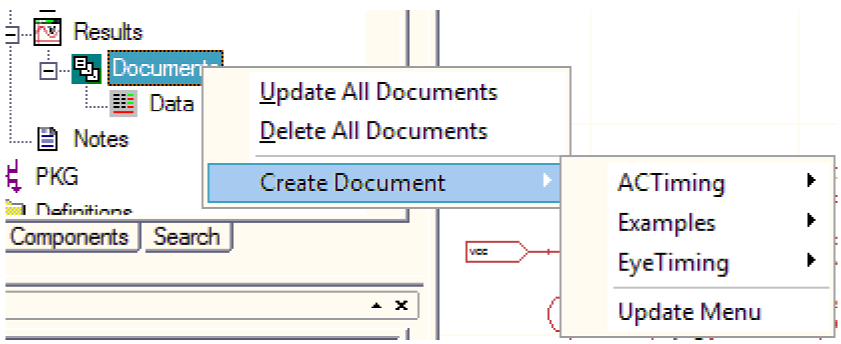
Managing Documents Listed in the Project Manager window Under Results (2D and Circuit)

In the **Project Manager** > **Results** > **Documents** folder, right-click a user-defined document and select **Rename** and **Delete**, among other choices. **Open Document** opens the web browser with the html document. **Modify Document** opens the setup window where change the selections for the input. Click **View xml Document** or **View PDF Document** to view the document in the appropriate format, or choose **Save As** to save the document in a different location.



Documents Folder shortcut menu

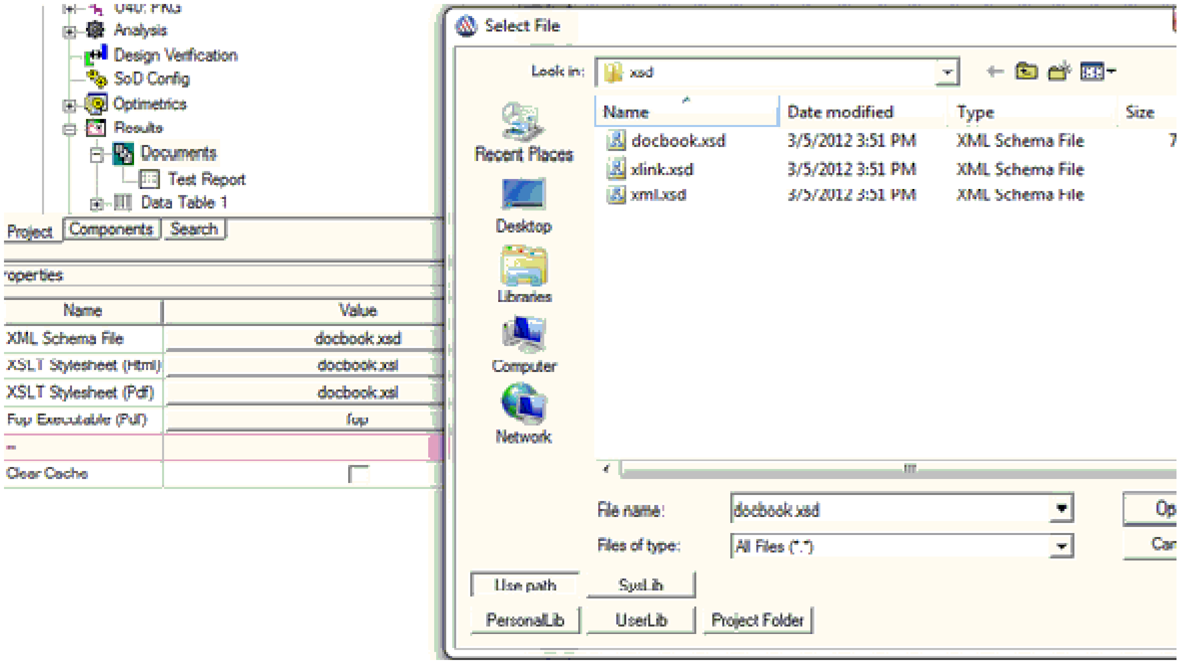
Right-click the Documents folder to **Update All Documents**, **Delete All Documents**, or **Create Document**.



Document Folder Property window

When the documents folder is selected, the Property window shows the following properties

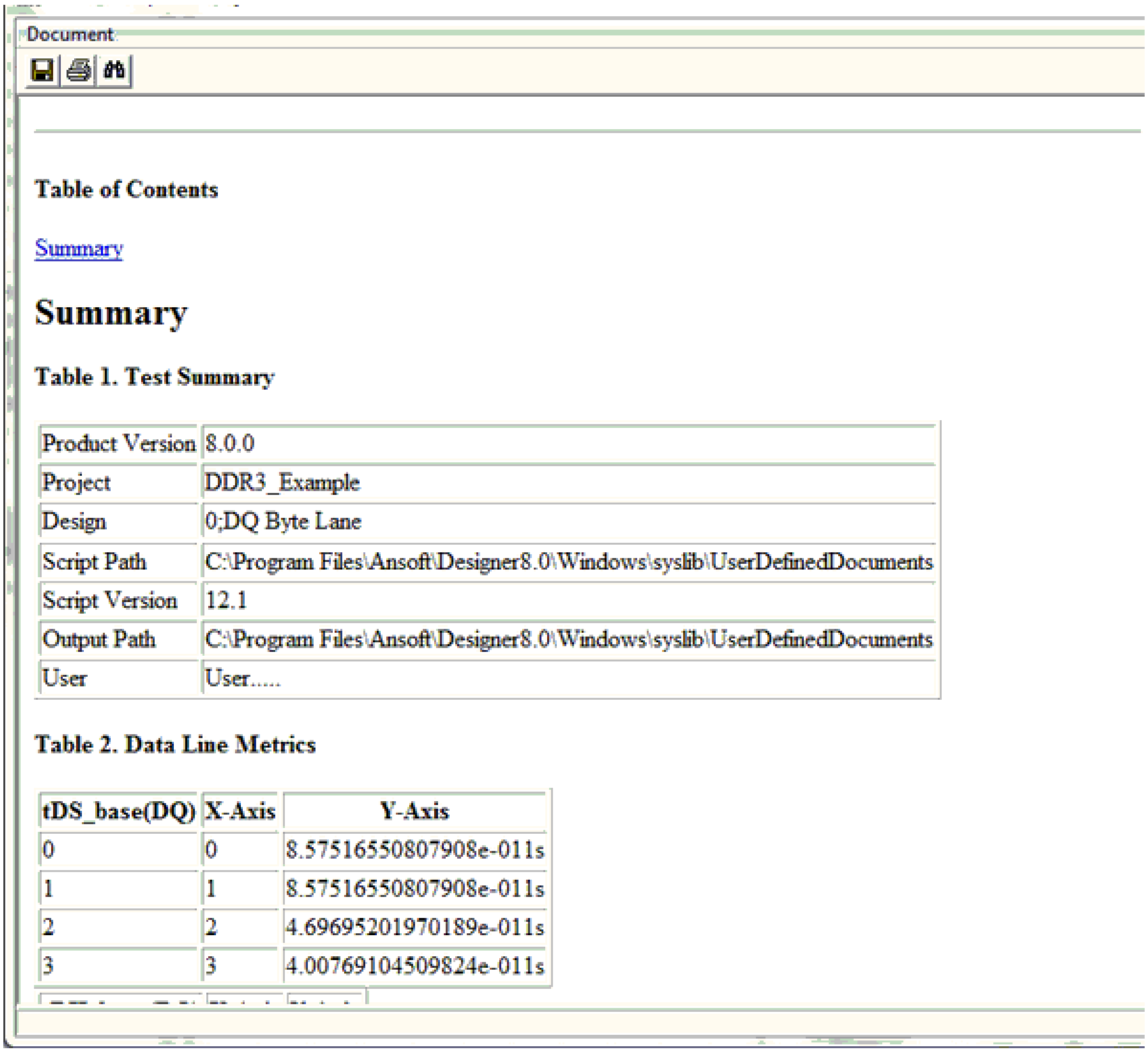
- xml Schema File - File path to the xml schema file.
- XSLT StyleSheet (Html) - File path to the XSLT stylesheet file used for Html generation.
- XSLT StyleSheet (Fo) - File path to the XSLT stylesheet file used for Pdf generation.
- Fop Executable (Pdf) - File path the Fop executable used for Pdf generation.
- Clear Cache - Clears the cached XSL transform object and forces creation of a new one. (The caching is done to save time during document generation, so subsequent generation or update of the document can use the cached transform object. But sometimes you may want to force a recompile of the document if you change the stylesheet).



The xml, HTML and PDF generation require the xml schema file and XSLT stylesheets to generate proper output. The PDF generation also 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 (2D and Circuit)

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 (2D and Circuit)

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 is automatically added to the system include path (any support script files from any Lib directory through the top level UserDefinedDefinitions directory can be imported).

The UDD functionality uses IronPython to provide 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\UserDefinedDefinitions\a\b\c\myudd.py and have a .NET assembly called com.Acme.UDDLib You can keep the .NET assembly under
   - UserDefinedDefinitions\Lib,
   - UserDefinedDefinitions\a\Lib,
   - UserDefinedDefinitions\a\b\Lib
   - UserDefinedDefinitions\a\b\c\Lib
3. Add the following line to your python script
   - Import clr
   - clr.AddReference(“com.Acme.UDDLib”)
   - import com.Acme.UDDLib –or-- from com.Acme.UDDLib import * etc

   If for some reason you cannot place the .NET assemblies into a Lib directory under UserDefinedDefinitions, 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 Definitions: Python Script API (2D and Circuit)

A user-defined Definition (UDD) extension is implemented as an IronPython script that defines a class with a specific name: **UDDExtension** which derives from a specific base class **IUDDPluginExtension** and implements its abstract methods.

Import Statements (2D and Circuit)

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.Ansys.DocGeneratorPluginDotNet.DocGenerator.API.Data
import *
from Ansys.Ansys.DocGeneratorPluginDotNet.DocGenerator.API.Interfaces
 import *
```

UDDExtension Class (2D and Circuit)

The UDD itself should be implemented as an IronPython class called **UDDExtension** which must derive on the **IUDDPluginExtension** abstract base class (on the **Ansys.Ansys.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 on their own base classes. The only requirement is that directly or indirectly, the UDD class must derive from **IUDDPluginExtension**.

Example:

```
def BaseClassUDD ((IUDDPluginExtension):
#base class implementation
…
def UDDExtension ((BaseClassUDD):
#UDD class implementation
…
```

> **Note:**
>
> All of the above text has been copied on the Help topic for the UDOs and modified for the UDDs. Since the UDDs are modelled after the UDOs, the usage is also similar.

IUDDPluginExtension Abstract Class (2D and Circuit)

Required functions:

The **IUDDPluginExtension** abstract class declares the following abstract methods that must be implemented in the UDDExtension class or one of its base classes. Not implementing any of these methods results in a run-time error and a non functioning UDD.

GetUDDName() : Return a string that is used as a prefix for all solution instances created using this UDD.

Example:

```
def GetUDDName(self):
return "MinMaxAvg"
```

GetUDDDescription() : Returns a description for the UDD, its purpose etc.

Example:

```
def GetUDDDescription(self):
```

```
return "Sample UDD”
```

ShowDefaultSetupwindow() : Returns True if the default window is to be shown. Return False if the user does not want the default window. In this case the user might want to implement/show a customized setup window.

Example:

```
def ShowDefaultSetupwindow(self):
return True
```

GetUDDInputParams(IUDDInputHelper inputHelper, List<UDDInputParams> uddInputs) : Returns the list of inputs parameters for the User Defined Document. Returns boolean: True on success, False on failure.

The supplied input parameters are used to populate details of the parameters to which the UDD user specifies value, specify the input names and their types.

inputHelper: This is the input helper object provides methods to gather information to define the inputs. This interface is explained in the Document Generator Interface document.

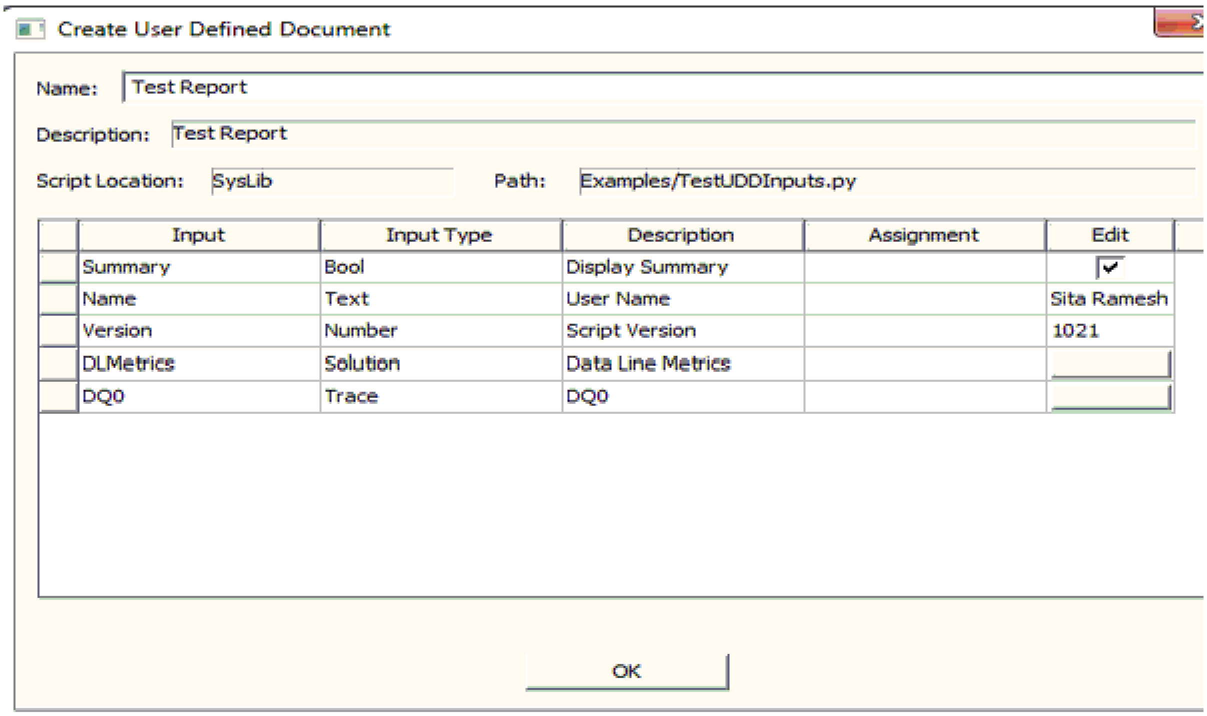
uddInputs: .NET list of UDDInputParams objects. The UDD script is expected to add one instance of UDDInputParams for each input definition it wants displayed. The UDD user can, when creating the UDD, assign a matching value to each such input.

Example:

```
def GetUDDInputParams(self, inputHelper, uddInputs)
# Boolean input
param1 = UDDInputParams("Summary","Display Summary",
Constants.kBoolTypeStr, True)
uddInputs.Add(param1)
# Text input
param2 = UDDInputParams("Name","User Name", Constants.kTextTypeStr ,
"Sita Koresh")
uddInputs.Add(param2)
# Number input
param3 = UDDInputParams("Version","Script Version",
Constants.kNumberTypeStr, 1021)
uddInputs.Add(param3)
# Solution input
param5 = UDDInputParams("DLMetrics","Data Line
Metrics",Constants.kSolutionTypeStr)
uddInputs.Add(param4)
```

```
# Trace input
param5 = UDDInputParams("DQ0","DQ0",Constants.kTraceTypeStr)
uddInputs.Add(param5)

return True
```

Based on the input params the following window is displayed when you click **Reports>Create Document**. The name and description of the UDD are also displayed in this window.



Generate(List<UDDInputData> uddInputs, IUDDGenerator generator, IProgressMonitor progressMonitor) : This is the main method which accesses the data on the uddInputs and generates the document.

uddInputs: The list of inputs that the user setup in the window. They are now available to query for data.

generator: This is the document generator object used to create different elements of the document like titles, topics, tables, images and write the data too. This interface is explained in the Document Generator Interface document.

progressMonitor : IProgressMonitor object. This can be used to set progress for long running calculations, check for user initiated abort etc.

Example:

```
def Generate(self, input, docgen, progMon):

# Gather data from inputs
boolinput = input[0].Data()
textinput = input[1].Data()
dblinput = input[2].Data()
# Get document root
docroot = docgen.GetDocumentRoot()

# Add Section
section1 = docroot.AddSection("Summary", "Overall Results ")

# Add a table
table1 = section1.AddTable("Test Summary")

#Add a table group with 2 columns
tgroup1 = table1.AddTableGroup(2)

# get desktop application
oApp = self.GetUDDAppContext()
if oApp != None:
oDesktop = oApp.GetAppDesktop()
if oDesktop != None:
# version number
version = oDesktop.GetVersion()
text1 = tgroup1.AddContent()
text1 .Add(0, "Product Version")
text1 .Add(1, version)

oProject = oDesktop.GetActiveProject()
if oProject != None:
projectname= oProject.GetName()
text1 = tgroup1.AddContent()
text1 .Add(0, "Project")
text1 1.Add(1, projectname)
```

```
oDesign = self.GetUDDDesignContext()
if oDesign != None:
designname = oDesign.GetName()
text1 = tgroup1.AddContent()
text1 .Add(0, "Design")
text1 .Add(1, designname)
# Provides a script path
scriptpath = docgen.GetScriptPath()
text1 = tgroup1.AddContent()
text1 .Add(0, "Script Path")
text1 .Add(1, scriptpath )

#Provides the script version
text1 = tgroup1.AddContent()
text1 .Add(0, "Script Version")
text1 .Add(1, str(dblinput ))

#Provides the output xml path
outputpath = docgen.GetOutputFilePath()
text1 = tgroup1.AddContent()
text1 .Add(0, "Output Path")
text1 .Add(1, outputpath )

#Provides the user information
text1 = tgroup1.AddContent()
text1 .Add(0, "User")
text1 .Add(1, textinput)

# Generate Xml output
docgen.Write(False)
# Generate Html output
docgen.WriteHTML()
# Generate PDF output
docgen.WritePDF()

return True
```

Optional functions:

SetupUDDInputParams(IUDDInputHelper inputHelper, List<UDDInputParams> uddInputs) : Displays a customized window and returns the user choices for the input params.

inputHelper: This is the input helper object provides methods to gather information to define the inputs. This interface is explained in the Document Generator Interface document.

uddInputs– .NET list of UDDInputParams objects with values for each of them. These can be the user choice for each input obtained through a custom window or some other non graphical assignment.

There is no way of assigning solution data to the input without the invocation of the reporter window, so trace and solution types of input with a custom window cannot be processed.

Example:

```
def SetupUDDInputParams(self, inputHelper, uddInputs)
uddwindow = BaseExampleUDDwindow()
if uddwindow.Showwindow() == Forms.windowResult.OK:
  # Boolean input
  param1 = uddwindow.GetInput("Summary")
  uddInputs.Add(param1)

  # Text input
  param2 = uddwindow.GetInput("Name")
  uddInputs.Add(param2)

  # Number input
  param3 = uddwindow.GetInput("Version")
  uddInputs.Add(param3)
```

HandleUDDEvents(List<string> eventTags) : The tags associated with the event is received

by plugin using this abstract class.

This method is the event handler for all link events set by the SetEventLink() method on a IUDDText. (See the definition of the IUDDText object in the Document Generator Interface document).

Example:

```
def HandleUDDEvents(self, uddLinks):
```

```
if uddLinks[0] == “Open” Report”:
  # Get Design Name
  oDesign = self.GetUDDDesignContext()
  if oDesign != None:
    oDesign.OpenReport(uddLinks[1])
return True
```

GetUDDSchema() : Returns the file path of the schema to validate the xml. This overrides the default schema used. Return string containing the full file path of the schema.

```
def GetUDDSchema(self):
return "C:\\Program
Files\\AnsysEM\\v242\\Win64\\common\\docbook\\schema\\xsd\\docbook.xs
d"
```

GetUDDStyleSheetForHtml() : Returns the file path of the style sheet used to generate the html document. This overrides the default stylesheet for html. Returns string containing the full file path of the style sheet.

```
def GetUDDStyleSheetForHtml(self):
return "C:\\Program Files\\AnsysEM\\v242\\Win64\\common\\docbook\\"
```

GetUDDStyleSheetForPdf() : Returns the file path of the style sheet used to generate the pdf document. This overrides the default stylesheet for pdf. Returns string containing the full file path of the style sheet.

Example:

```
def GetUDDStyleSheetForPdf(self):
return "C:\\Program
Files\\AnsysEM\\v242\\Win64\\common\\docbook\\xsl\\fo\\docbook.xsl"
```

GetFopExecutable() : Returns the file path of the fop executable used to generate the pdf document. This overrides the default stylesheet for pdf.Returns string containing the full file path of the fop executable.

Example:

```
def GetFopExecutable(self):
return "C:\\Program
Files\\AnsysEM\\v242\\Win64\\common\\ApacheFOP\\fop-1.0\\fop"
```

GetUDDAppContext() : Returns the UDD Owner (if set). This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the **Electronics Desktop** (**Electronics Desktop**) app.

GetUDDDesignContext() : Returns the UDD Owner (if set). This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the Design.

Data Types Used in Python Script (2D and Circuit)

There are several types that you must use while authoring the python script. Some of them are used to pass data from UI to python script and provide interface for working with this data. Some are used to pass data from python script to UI.

To pass data from python script to UI the objects of the C# class must be created in python script using their C# constructors. The objects can then be set as functions return values or set to the output parameters using their API.

Constants Class (2D and Circuit)

kTraceTypeStr : string constant used to specify an input of trace type

kSolutionTypeStr : string constant used to specify an input of solution type

kNumberTypeStr : string constant used to specify an input of number type

kTextTypeStr : string constant used to specify an input of text type

kBoolTypeStr : string constant used to specify an input of boolean type

kStandardReportStr : string constant to specify a standard report

kEyeDiagramReportStr : string constant to specify an eye diagram report

kUserDefinedReportStr : string constant to specify a user-defined report

kSweepDomainStr : string constant to specify the sweep domain

kTimeDomainStr : string constant to specify the time domain

UDDInputParams Class (2D and Circuit)

The objects of this class must be created in python script in the **GetUDDInputParams()** function and the **SetUDDInputParams()** function.

Attributes :

Input Name (string)

Input Description (string)

Input Type (Can be Boolean, Number, Text, Trace or Solution) (string)

BoolData (boolean)

DoubleData (double)

TextData (string)

ReportType (string)

SolutionName (string)

DomainName (string)

Constructors:

UDDInputParams(string name, string description, string type)

UDDInputParams(string name, string description, string type, bool data)

UDDInputParams(string name, string description, string type, double data)

UDDInputParams(string name, string description, string type, string data)

UDDInputParams(string name, string description, string type, string reportType, string solutionName, string domainName)

Property Accessors :

Name : Get/Set the name of an input

Description : Get/Set the 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

IProgressMonitor Abstract Class (2D and Circuit)

The object of this class is a progress monitor. It is used to display calculations progress in UI and check is the user has requested an abort of the computation.

When displayed in the application, each progress message has four items:

A task name

A sub-task name

The progress amount

A button to abort the task in progress.

All of this functionality and abort interaction is achieved using the following functions.

SetTaskName (string taskName)

SetSubTaskName (string subTaskName)

BeginTask (string name)

SetTaskProgressPercentage(int progressPercent)

CheckForAbort(): If the quantities being generated are computationally expensive, the UDO author can periodically call this method , then call EndTask with Fail and return False.

EndTask (bool passFail)

Example:

```
progMon.BeginTask("Process DQS")
progMon.SetSubTaskName("Compute UI segments")
progMon.SetTaskProgressPercentage(33)
progMon.SetSubTaskName("Compute the rest")
progMon.SetTaskProgressPercentage(100)
progMon.EndTask(True)
```

UDD Input Interfaces (2D and Circuit)

The Generate function takes in a list of inputs. These input interfaces allow the user to access data on 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 window.

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 window.

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 window.

IUDDInputTrace: This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

DoubleData() : Method used to return x and y double data as a IDictionary<double, double>

DoubleData(IDictionary<string, string> variation) : Method used to return x and y double data as a IDictionary<double, double>, given a variation.

ComplexData() : Method used to return x data and y complex data as a IDictionary<double, double[]>

ComplexData(IDictionary<string, string> variation) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a variation.

TextData() : Method used to return x data and y data as a IDictionary<string, string>

TextData(IDictionary<string, string> variation) : Method used to return x data and y data as a IDictionary<string, string>, given a variation.

VariableValues() : Method used to get a list of variations as a IList<Dictionary<string, string>>

IUDDInputSolution: This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

DoubleData(string name) : Method used to return x and y double data as a IDictionary<double, double>, given a quantity name.

DoubleData(string name, IDictionary<string, string> variation) : Method used to return x and y double data as a IDictionary<double, double>, given a quantity name and a variation.

ComplexData(string name) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a quantity name.

ComplexData(string name, IDictionary<string, string> variation) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a quantity name and a variation.

TextData(string name) : Method used to return x data and y data as a IDictionary<string, string> given a quantity name.

TextData(string name, IDictionary<string, string> variation) : Method used to return x data and y data as a IDictionary<string, string>, given a quantity name and a variation.

CategoryNames() : Method to return a list of category names in the solution as an IList<string>

QuantityNames(string category) : Method to return a list of quantity names in the solution as an IList<string>, given a category.

VariableValues() : Method used to get a list of variations as a IList<Dictionary<string, string>>

Examples:

```
def Generate(self, input, docgen, progMon):
# Getting the boolean dataset by the user
boolinput = input[0].Data()
# Getting the double dataset by the user
dblinput = input[1].Data()
# Getting the text dataset 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 on the trace
xydata = input[4].DoubleData()
```

user-defined Document Scripting Interface (2D and Circuit)

To access the UserDefineddocuments scripting object, use:

```
Set oModule = oDesign.GetModule("UserDefinedDocuments")
```

Once you have the scripting object, use the following methods:-

1. AddDocument ([in] VARIANT data, [in] VARIANT traces, [out, retval] BSTR* uniqueName)
   a. Takes a VARIANT data which defines the document.
   b. Takes a VARIANT trace data for the inputs in the document.
   c. Returns a unique name
2. EditDocument([in] BSTR originalName, [in] VARIANT modifiedData, [in] VARIANT modifiedTraces, [out, retval] BSTR* uniqueName)
   a. Takes the name of the original document.
   b. Takes a VARIANT data which defines the edited document.
   c. Takes a VARIANT trace data for the inputs in the document.
   d. Returns a unique name
3. RenameDocument([in] BSTR oldName, [in] BSTR newName)
   a. Takes the name of the original document.
   b. Takes the new name of the document.
4. DeleteDocument([in] BSTR name)
   a. Takes the name of the document to be deleted.
5. UpdateDocument([in] BSTR name)
   a. Takes the name of the document to be updated.
6. ViewHtmlDocument([in] BSTR name)
   a. Takes the name of the document to be viewed in HTML.
7. ViewPdfDocument([in] BSTR name)
   b. Takes the name of the document to be viewed as a PDF.
8. SaveHtmlDocumentAs([in] BSTR name, [in] BSTR saveTo)
   a. Takes the name of the document to be saved.
   b. Takes the file path to save the document as.
9. SavePdfDocumentAs([in] BSTR name, [in] BSTR saveTo)
   a. Takes the name of the document to be saved.
   b. Takes the file path to save the socument as.

10. GetDocumentNames([in] BSTR separator, [out, retval] VARIANT* names)
    a. 'separator' is used to convey the directory "level"
    b. Returns the names of documents.
11. GetDocumentDefinitionNames([in] BSTR separator, [out, retval] VARIANT* names)
    a. 'separator' is used to convey the directory "level"
    b. Returns the (file) names of doc definitions according to the files in various installation directories.
12. DeleteAllDocuments()
13. UpdateAllDocuments()

For 6, 7, 8, and 9, the document must have an existing, generated HTML or PDF.

The UserDefinedDocument Data Format (2D and Circuit)

To define a document in VB script:

Array("NAME:Test Report", (Name of the document)

"Test Report", (Description of the document)

"SysLib", (Location of the python script(Syslib, Userlib, PeronalLib etc)

"TestUDDReport", (Relative path of the script in the UserDefinedDocuments folder)

// Start of input definition //

Array("NAME:Inputs", (Document Inputs keyword)

// Solution input //

Array("NAME:DLMetrics", (Input name)

"Solution", (Solution Input Type)

"Data Line Metrics", (Input Description)

-1, (Solution ID)

-1), (Report ID)

// Trace input //

Array("NAME:DQ0", (Input name)

"Trace", (Trace Input Type)

"DQ0", (Input Description)

-1, (Solution ID)

-1), (Report ID)

// Text input //

Array("NAME:Name", (Input name)

"Text", (Text Input Type)

"User Name", (Input Description)

Array("Sita Ramesh")), (Default Value)

// Bool input //

Array("NAME:Summary", (Input name)

"Bool", (Boolean Input Type)

"Display Summary", (Input Description)

Array(true)), (Default Value)

// Number input //

Array("NAME:Version", (Input name)

"Number", (Number Input Type)

"Script Version", (Input Description)

Array(1021)))), (Default Value)

// Trace selection for the solution and trace inputs //

Array("NAME:DocTraces", (Document traces keyword)

// For input “DLMetrics” //

Array("NAME:DLMetrics", (Input name)

// Trace defintion similar to the UDO. This trace definition is a user-defined solution //

Array("user-defined",

"", "DDR3 AC-Timing 4-DQ1", Array("Context:=", ""), Array("Index:=", Array("All"), "Trise:=", Array("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"), "Data_ Rate:=", Array("Nominal"), "Length:=", Array("Nominal")), Array("Probe Component:=", Array ("")), Array())),

// For input “DQ0” //

Array("NAME:DQ0",

// Trace defintion similar to the UDO. This trace definiton is a Standard solution //

Array("Standard", "DQ0", "NexximTransient", Array("NAME:Context", "SimValueContext:=", Array(1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false, "0", "DP", _

false, "20000000", "DT", false, "0.001", "WE", false, "100ns", "WM", false, _

"100ns", "WN", false, "0ps", "WS", false, "0ps")), Array("Time:=", Array("All"), "Trise:=", Array(_

"Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"), "Data_Rate:=", Array ("Nominal"), "Length:=", Array("Nominal")), Array("Probe Component:=", Array(_

"DQ0")), Array()))

Python Script to Define Document (2D and Circuit)

To define a document in Python script:

```
[
"NAME:Test Report", "Test Report", "SysLib",
"Examples/TestUDDInputs",
[
"NAME:Inputs",
[
"NAME:DLMetrics","Solution", "Data Line Metrics", -1, -1
],
[
"NAME:DQ0", "Trace", "DQ0", -1, -1
],
[
"NAME:DQS", "Trace", "DQS", -1, -1
],
[
"NAME:Name", "Text", "User Name", ["Sita Ramesh"]
],
[
"NAME:Summary", "Bool", "Display Summary", [True]
],
[
"NAME:Version", "Number", "Script Version" [1021]
]
```

```
]
],
[
"NAME:DocTraces",
[
"NAME:DLMetrics",
[
"user-defined", "", "DDR3 AC-Timing 4-DQ1",
[
"Context:=" , ""
],
[
"Index:=" , ["All"], "Trise:=" , ["Nominal"], "Tfall:=" ,
["Nominal"], "Pulse_Width:=" ,
["Nominal"], "Data_Rate:=" , ["Nominal"], "Length:=" , ["Nominal"]
],
[
"Probe Component:=" , [""]
],
[]
]
],
[
"NAME:DQ0",
[
"Standard", "DQ0", "NexximTransient",
[
"NAME:Context", "SimValueContext:=" ,
[1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"20000000","DT",False,"0.0
01","WE",False,"100ns","WM",False,"100ns","WN",False,"0ps","WS",False
,"0ps"]
],
[
"Time:=" , ["All"],"Trise:=" , ["Nominal"],"Tfall:=" ,
["Nominal"],"Pulse_Width:=" ,["Nominal"],
"Data_Rate:=" , ["Nominal"], "Length:=" , ["Nominal"]
```

```
],
[
"Probe Component:=" , ["DQ0"]
],
[]
]
],
]
```

Sample Document Handler Script (2D and Circuit)

This script adds, edits, renames and deletes a document.

```
Set oModule = oDesign.GetModule("UserDefinedDocuments")
' Add a UDD
oModule.AddDocument Array("NAME:Test Report1", "Test Report",
"SysLib", _
"Examples/TestUDDInputs", Array("NAME:Inputs", Array
("NAME:DLMetrics", "Solution", _
"Data Line Metrics", -1, -1), Array("NAME:DQ0", "Trace", "DQ0", -1, -
1), Array("NAME:DQS",
"Trace", "DQS", -1, -1), Array("NAME:Name", "Text", "User Name",
Array("Sita Ramesh")), Array("NAME:Summary", "Bool", "Display
Summary", Array(true)), Array("NAME:Version", "Number", "Script
Version"))), Array("NAME:DocTraces", Array("NAME:DLMetrics", Array
("user-defined", "", "DDR3 AC-Timing 4-DQ1", Array("Context:=", ""),
Array("Index:=", Array("All"), "Trise:=", Array( "Nominal"),
"Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"),
"Data_Rate:=", Array( "Nominal"), "Length:=", Array("Nominal")),
Array("Probe Component:=", Array("")), Array())), Array("NAME:DQ0",
Array( _
"Standard", "DQ0", "NexximTransient", Array("NAME:Context",
"SimValueContext:=", Array( _
1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false, "0",
"DP", false, "20000000", "DT", false, "0.001", "WE", false, "100ns",
"WM", false, "100ns", "WN", false, "0ps", "WS", false, "0ps")), Array
("Time:=", Array("All"), "Trise:=", Array("Nominal"), "Tfall:=",
Array("Nominal"), "Pulse_Width:=", Array("Nominal"), "Data_Rate:=",
Array("Nominal"), "Length:=", Array("Nominal")), Array("Probe
Component:=", Array("DQ0")), Array())))
```

```
` Edit Document
oModule.EditDocument "Test Report1", Array("NAME:Test Report", "Test
Report", _
"SysLib", "Examples/TestUDDInputs", Array("NAME:Inputs", Array
("NAME:DLMetrics", _
"Solution", "Data Line Metrics", 1000001, 0), Array("NAME:DQ0",
"Trace", "DQ0", 32, _
2), Array("NAME:DQS", "Trace", "DQS", 32, 4), Array("NAME:Name",
"Text", "User Name", Array( "Sita Ramesh")), Array("NAME:Summary",
"Bool", "Display Summary", Array(true)), Array("NAME:Version",
"Number", "Script Version"))), Array("NAME:DocTraces", Array
("NAME:DLMetrics", Array( "User Defined", "Solution", "DDR3 AC-Timing
4-DQ1", Array("Context:=", ""), Array("Index:=", Array( "All"),
"Trise:=", Array("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_
Width:=", Array( "Nominal"), "Data_Rate:=", Array("Nominal"),
"Length:=", Array("Nominal")), Array("Probe Component:=", Array("")),
Array())), Array("NAME:DQ0", Array("Standard", "DQ1",
"NexximTransient", Array("NAME:Context", "SimValueContext:=", Array
(1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false,
"0", "DP", _
false, "20000000", "DT", false, "0.001", "WE", false, "100ns", "WM",
false, "100ns", "WN", false, "0ps", "WS", false, "0ps")), Array
("Time:=", Array("All"), "Trise:=", Array("Nominal"), "Tfall:=",
Array("Nominal"), "Pulse_Width:=", Array("Nominal"), "Data_Rate:=",
Array("Nominal"), "Length:=", Array("Nominal")), Array("Probe
Component:=", Array("DQ1")), Array())))

` Rename a UDD
oModule.RenameDocument "Test Report", "Test UDD Report"

' Update UDD
oModule.UpdateDocument "Test UDD Report"

' View Html
oModule.ViewHtmlDocument "Test UDD Report"

' View Pdf
oModule.ViewPdfDocument "Test UDD Report"
```

```
' Save Html
oModule.SaveHtmlDocumentAs "Test UDD Report",
"c:/AnsysProjects/Test.html"

' Save pdf
oModule.SavePdfDocumentAs "Test UDD Report",
"c:/AnsysProjects/Test.pdf"

‘ Delete UDD
oModule.DeleteDocument "Test UDD Report"
```

> **Note:**
>
> The product has to implement the GetModule call to create the UserDefinedDocument scripting object. For example: Check AltraSimDesign.cpp (function GetMgrIDispatch()).

Document Generator Interfaces (2D and Circuit)

This document briefly describes the API interfaces available in the document generator plugin.

(Ansys.Ansys.DocGeneratorPluginDotNet.dll)

Scripting objects available in the script for the Generate function

- oApp = self.GetUDDAppContext(

  Gets the application context

  Usage:- Gets the active project ad the version of the product

```
oDesktop = oApp.GetAppDesktop()
```

if oDesktop != None:

vr = oDesktop.GetVersion()

oProject = oDesktop.GetActiveProject()

- oDesign = self.GetUDDDesignContext()

  Gets the design context

  Usage:- Gets the design name.

oDesign = self.GetUDDDesignContext()

if oDesign != None:

nm = oDesign.GetName()

- IUDDGenerator interface

This interface available in the Generate method of the UDDPluginExtension.

This interface can be used to

1. Set the document output file path.
```
docgen.SetOutput("C:\\Examples\\DocumentOutput.xml")
```
2. Get the document root.
```
docroot = docgen.GetDocumentRoot()
```
3. Write out to the output file.
```
docgen.Write()
```
4. Write Html document
```
void WriteHTML();
```
5. Write PDF document
```
void WritePDF();
```
6. Load the Html transform object
```
void LoadHTMLTransform();
```
7. Load the cached PDF transform object
```
void LoadPDFTransform();
```
8. Get script path
```
string GetScriptPath();
```
9. Get output file path
```
string GetOutputFilePath();
```
- IUDDRoot interface

Calling GetDocumentRoot() on the IUDDGenerator interface provides you with the this interface. This interface can be used to

1. Add a new topic to the document. Provide a title.

```
section1 = docroot.AddSection("Section title")
```

2. Add a new topic to the document. Provide a title and subtitle

```
section1 = docroot.AddSection("Section title", “Section subtitle”)
```

3. Add a new title

```
section1 = docroot.AddTitle("Title")
```

4. Add a new subtitle

```
section1 = docroot.AddSubtitle("Subtitle”)
```

- IUDDSection interface

Calling AddSection() on the IUDDRoot interface provides you with the this interface. This interface can be used to

1. Set an ID for the topic for internal links.

```
section1.SetID("id")
```

2. Add a new table to the document. Provide a table title.

```
table1 = section1.AddTable("Table title")
```

3. Add a new image to the document. Provide an image title and a file path to the image file.

```
image1 = section1.AddImage("Image title")
```

4. Add text to the document.

```
text1 = section1.AddText("Random text........")
```

- IUDDImage interface

Calling AddImage() on the IUDDSection interface provides you with the this interface. On this interface 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. On this interface call the following methods

1. Set an ID for the text for internal links.

```
text1.SetID("id")
```

2. Set the emphasis attribute on the text

```
text1.SetEmphasis()
```

3. Set the quotes attribute on the text

```
Text1.SetQuotes()
```

4. Set the block quotes attribute on the text

```
text1.SetBlockquotes()
```

5. Set quotes on the text

```
Text1.SetQuotes()
```

6. Set the wordsize attribute on the text

```
text1.SetSize(size as an integer)
```

7. Set a link to an ID of any element to provide internal links

```
text1.SetLink("linkname")
```

8. Set an event link to handle an event. The HandleUDDEvents method should be implemented in the script to handle the event.

```
text1.SetEventLink("linkname")
```

- IUDDTable interface

Calling AddTable() on the IUDDSection interface provides you with the this interface. On this interface call the following methods

1. Set an ID for the table for internal links.

```
table1.SetID("id")
```

2. Set alignment can be "center", "left" and "right".

```
table1.SetAlignment("center")
```

3. Set the background color of the table

```
table1.SetBgColor(string bgcolor)
```

4. Set the frame type. Can be "all", "bottom", "top,", "sides", "topbot"

```
table1.SetFrame(string frame)
```

5. Add a table group and specify the number of columns. A table can have multiple table groups.

```
IUDDTableGroup table1.SetTableGroup(int columns)
```

- IUDDTableGroup interface

Calling AddTableGroup() on the IUDDTable interface provides you with the this interface. On this interface call the following methods

1. Set an ID for the table group for internal links.

```
tgroup1.SetID("id")
```

2. Set alignment information can be "center", "left" and "right".

```
tgroup1.SetAlignment("center")
```

3. Set the column width of a column given the index of the column and the required width. Width can be set in 2 ways.
   - Width can be set relative to 1. E.g Setting it to "2*" makes the column width double the width of the others.
   - If the entire table width is considered to be 99.99 units. Width can be a number relative to this.

```
tgroup1.SetColumnWidth(int index, string width)
```

4. Add a header to the table group

```
IUDDTableRow tgroup1.AddHeader()
```

5. Add a header with multiple rows to the table group. Takes number of sub rows.

```
IUDDTableRow tgroup1.AddHeader(int rows)
```

6. Add a row of content to the table group

```
IUDDTableRow tgroup1.AddContent()
```

7. Add content with multiple rows to the table group. Takes number of sub rows.

```
IUDDTableRow tgroup1.AddContent(int rows)
```

- IUDDTableRow interface

Calling AddHeader() & AddContent() on the IUDDTableGroup interface provides you with the this interface. On this interface call the following methods

1. Set an ID for the table row for internal links.

```
trow1.SetID("id")
```

2. Set alignment information can be "center", "left" and "right".

```
trow1.SetAlignment("center")
```

3. Set cell text. Can be cell content or header text. Takes a column index and a text string. It is added to the first row.

```
IUDDTextElement trow1.Add(int column, string text)
```

4. Set cell text. Can be cell content or header text. Takes a column index, row index and a text string. Takes in a row number because a table row can have multiple sub rows.

```
IUDDTextElement trow1.Add(int column, int subrow, string text)
```

5. Set cell content. Takes a column index and an int value. It is added to the first row.

```
IUDDTextElement trow1.Add(int column, int value)
```

6. Set cell content. Takes a column index, row index and a int value.

```
IUDDTextElement trow1.Add(int column, int subrow, string text)
```

7. Set cell text. Takes a column index and a double value. It is added to the first row.

```
IUDDTextElement trow1.Add(int column, double value)
```

8. Set cell text. Takes a column index, row index and a double value.

```
IUDDTextElement trow1.Add(int column, int subrow, double value)
```

9. Set cell text spanning 2 columns. Can be cell content or header text. Takes a sub row index , starting column index., ending column index and a text string.

```
IUDDTextElement trow1.AddSpanningcolumnst(int subrow, int
columnstart, int columnend, string text)
```

10. Set cell text. Can be cell content or header text. Takes a column index, starting sub row index, ending sub row index and a text string.

```
IUDDTextElement trow1.AddpanningRows(int column, int subrowstart,
int subrowend, string text)
```

- IUDDTableRow interface

Calling Add() on the IUDDTableGroup interface provides you with the this interface. On this interface 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.

- IUDDInputHelper interface

The GetUDDInputParams() and SetUDDInputPrams provide this interface to gather information on the UI. On this interface call the following methods

1. Returns a list of user-defined solution names.

```
IList<string> inputHelper.UDSSolutionNames()
```

2. Returns the list context names for a given user-defined solution.

```
IList<string> inputHeper.UDSContextNames(string solutionname)
```

3. Returns the file path of the script corresponding to the user-defined solution.

```
string inputHelper.UDSScriptFileName(string solutionname)
```

4. Returns the parameters defined in the user-defined solution.

```
PropList UDSParameters(string solutionName)
```

5. Returns a dictionary of probe names and their corresponding expression, for a user-defined solution.

```
IDictionary<string, string> UDSProbes(string solutionName)
```

6. Returns a dictionary of variables and their nominal values.

```
IIDictionary<string, string> UDSVariableValues(string solutionName)
```

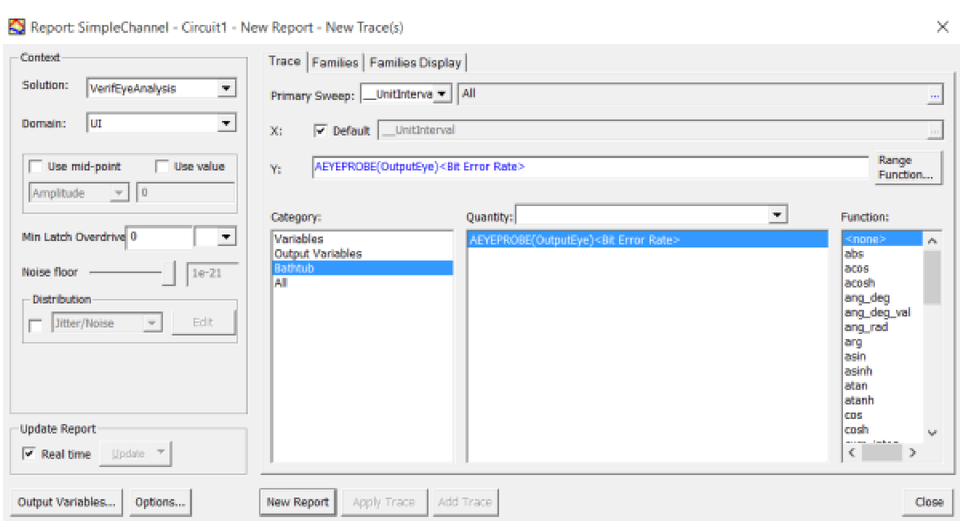
Technical Notes for 2D and Circuit Reports

Electronics Desktop post-processing allows you to define all types of plots for simulations prior to running an analysis. Using the reporter window, define plots prior to simulation for all types of reports, including VerifEye, QuickEye, AMI, and Transient PMF. All reports are populated using the eye probes and AMI probes in a design, so set up all types of displays before an analysis.

The following topics describe how to take advantage of additional Electronics Desktop post-processing tools in order to generate various reports and view simulation results.

Unit Interval and Amplitude Values for Circuit Bathtub Curves

You can specify values for the Unit Interval or Amplitude while plotting bathtub curves for a QuickEye or VerifEye analysis. The controls allow you to choose between two check boxes for **Use mid-point** or **Use value**. The menu allows selection of **Amplitude** or **UI**.

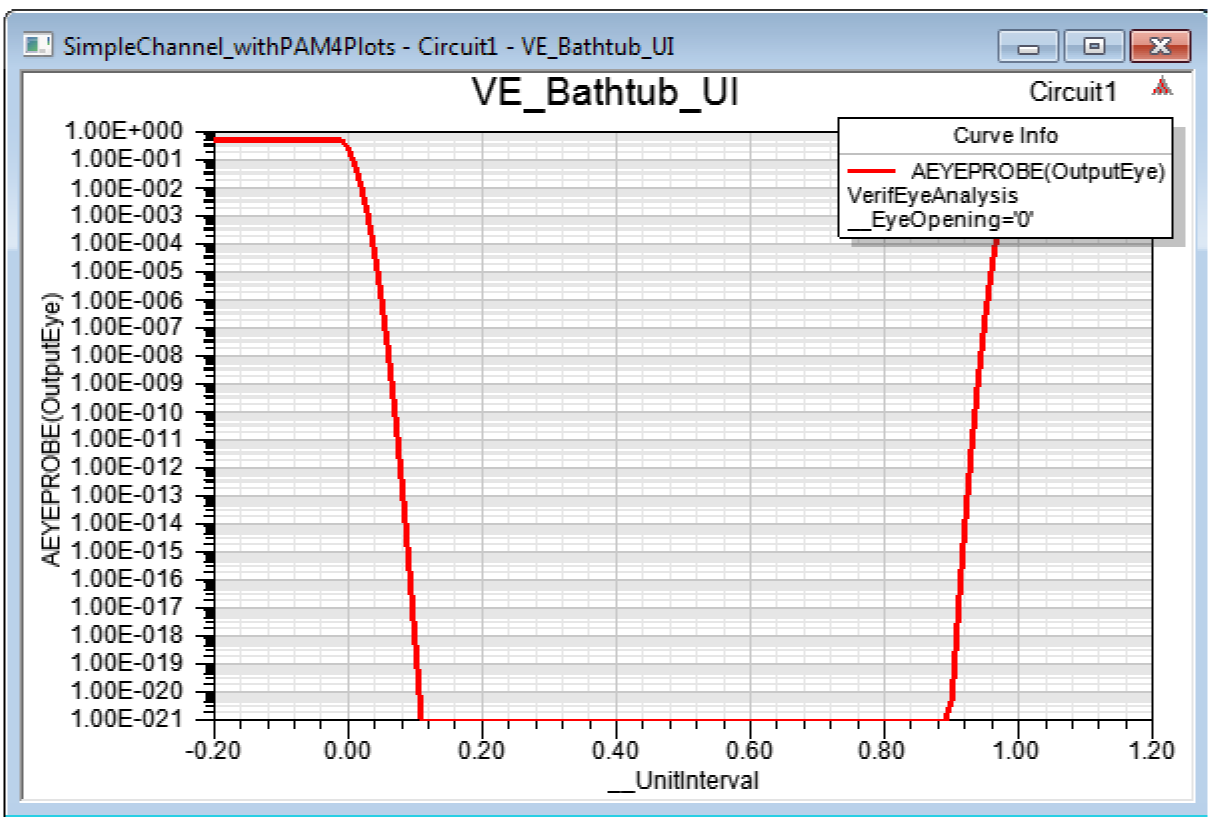


- When the **Use value** group box is checked, the edit window is enabled, allowing you to enter an Amplitude or UI value at which to see the bathtub curve. Electronics Desktop interpolates the data to plot a bathtub curve at that Amplitude or UI.

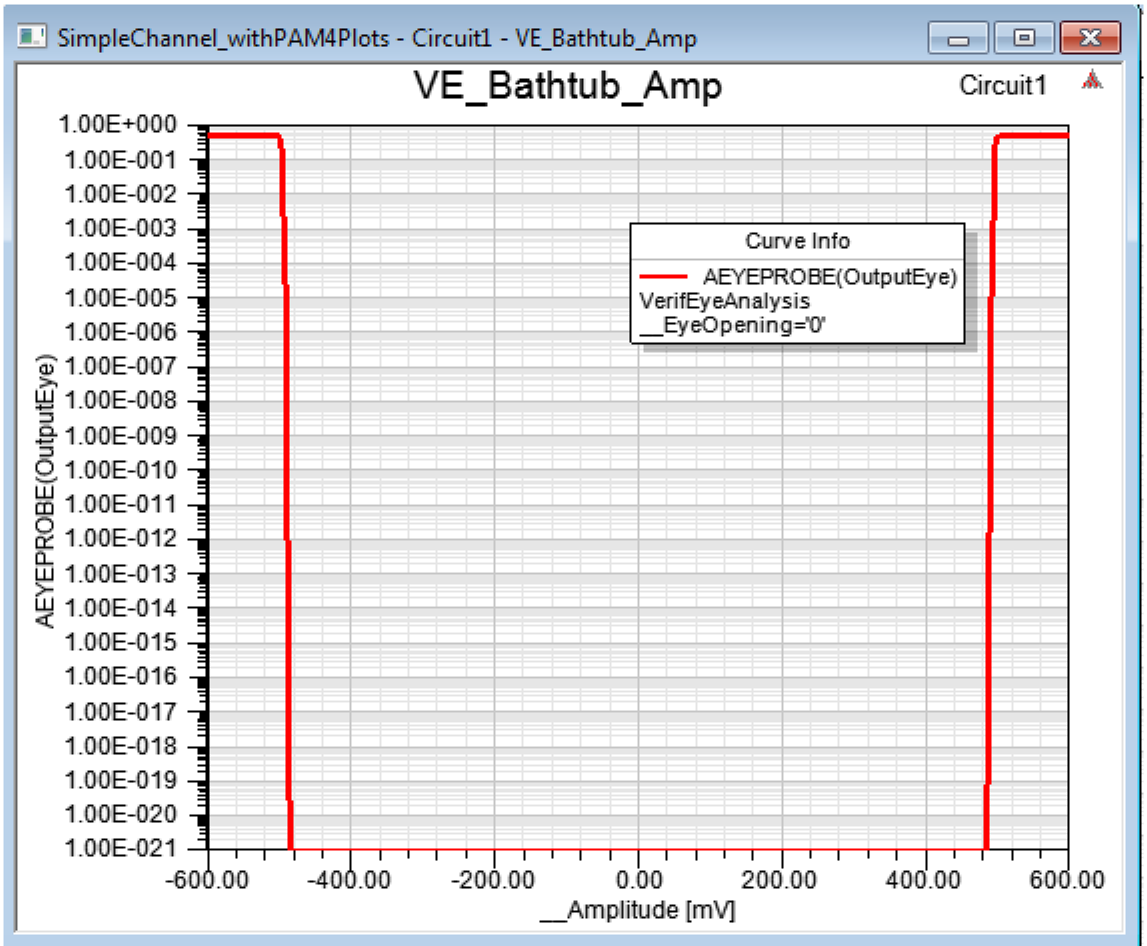
- The **Use midpoint** check box can also be enabled. Select **Amplitude** or **UI** on the menu.
- By default, the **Use midpoint** box is checked, and the midpoint of Amplitude is used as the default sweep value for the plot.
- Bathtub noise floor range is e-16 to ~ e-21.
- Default data length is set at 2^{18} or around $2.5x10^5$.
- BitPattern is PBRS and "random bit pattern generation".

> **Note:** With QuickEye, AMI, and Transient plots, when bathtub curves can be plotted and txrj is present, a warning message is displayed if the number of bits is less than 2.5e5.

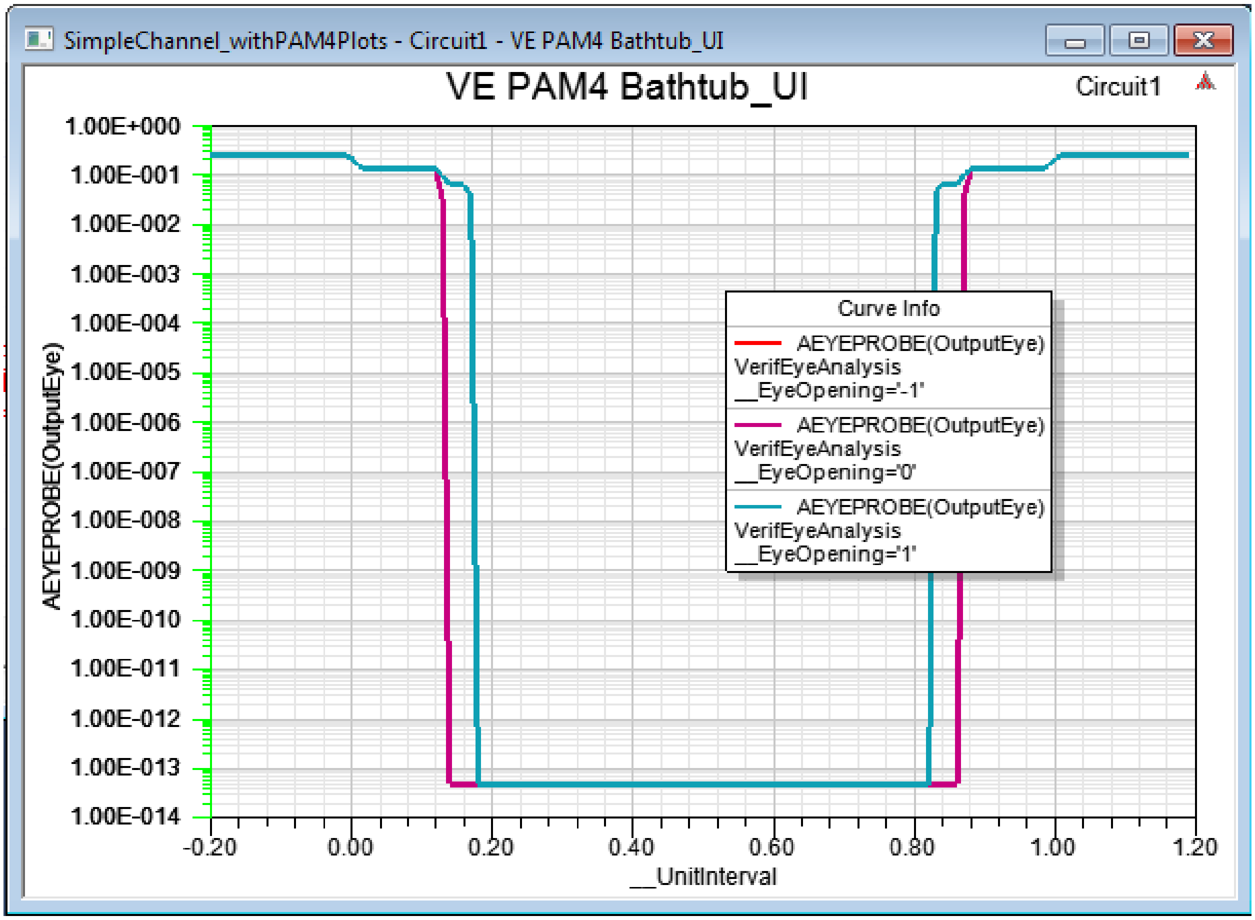
The following shows a VerifEye plot with the default settings and NRZ transmission (X-axis is the UI):



Following is a VerifEye plot centered on the UI with NRZ (X-axis is the Amplitude):

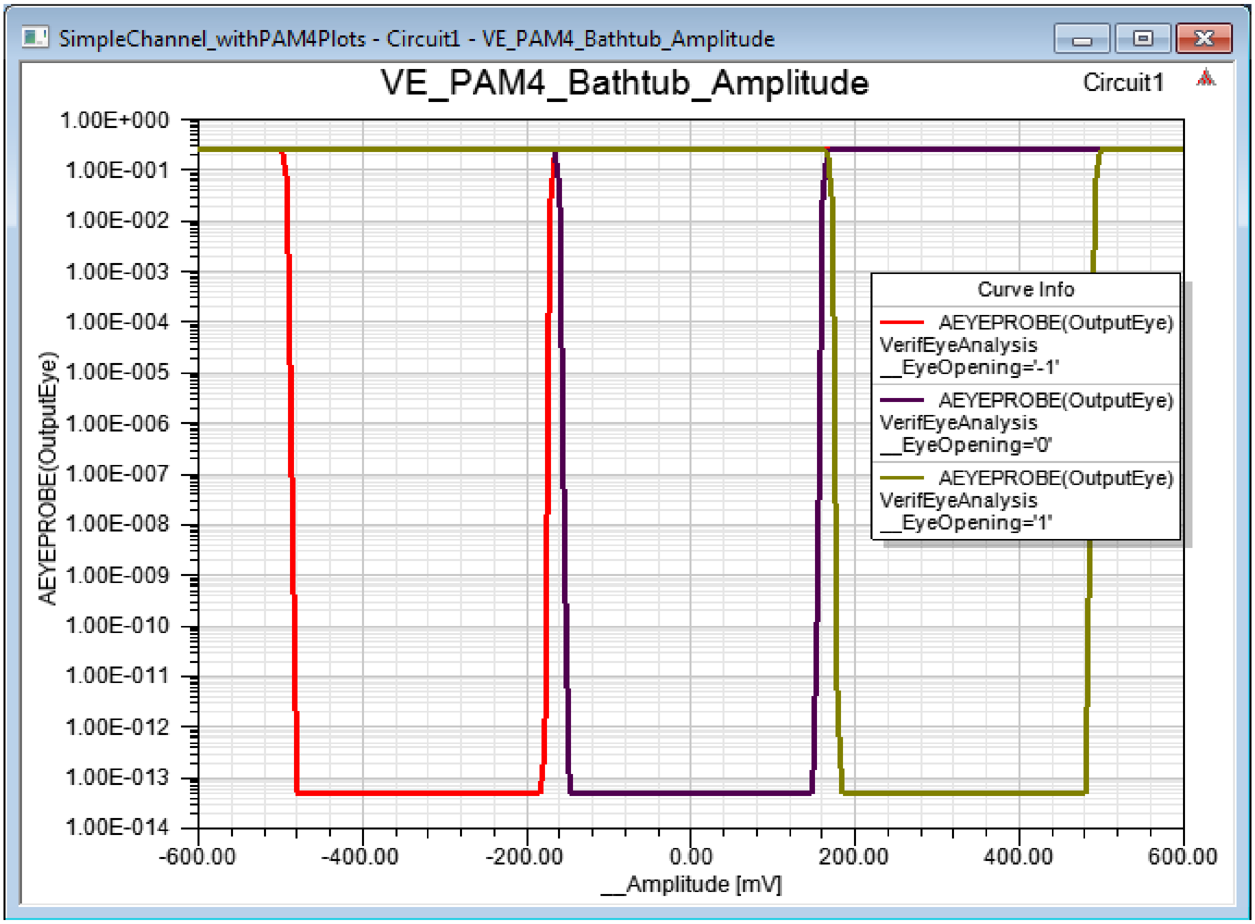


Following is a VerifEye PAM-4 plot centered on the Amplitude (X-axis is the UI):



- The PAM-4 eye has three levels. The Symbol error rates for each of the three sub-eyes is shown in a different trace.

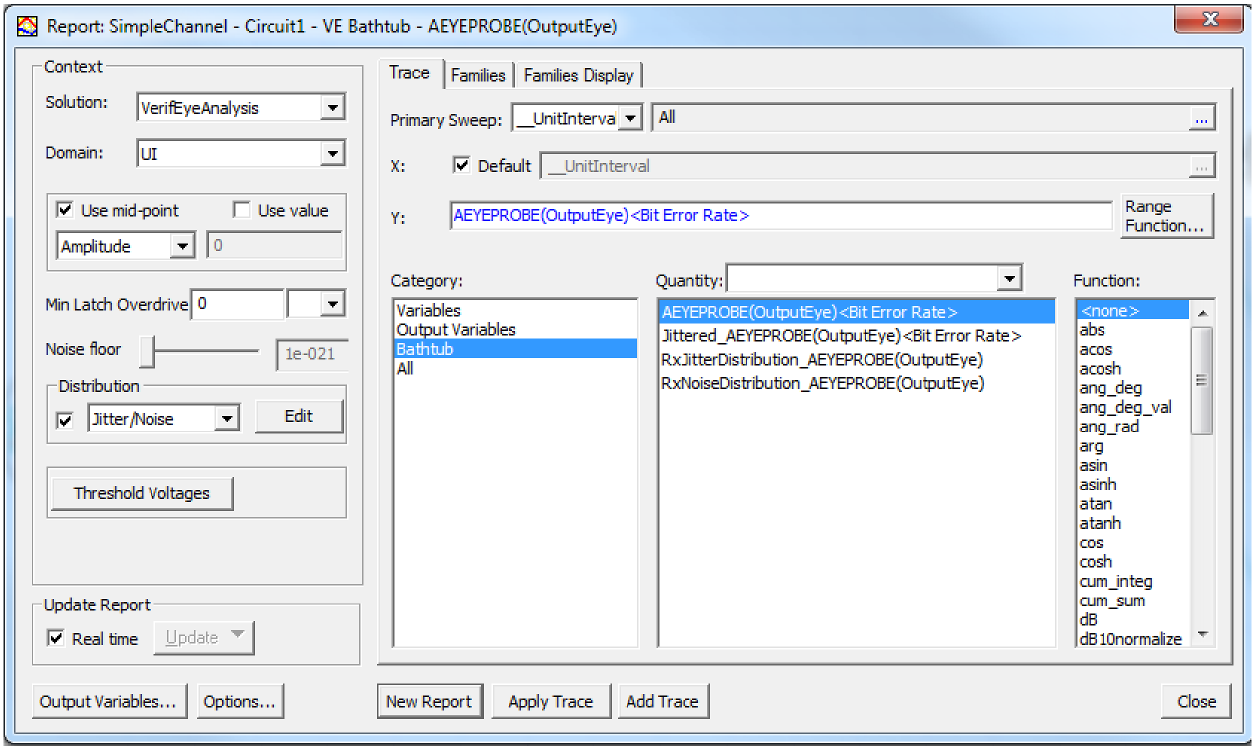
Following is a VerifEye PAM-4 plot centered on the UI (X-axis is the Amplitude):



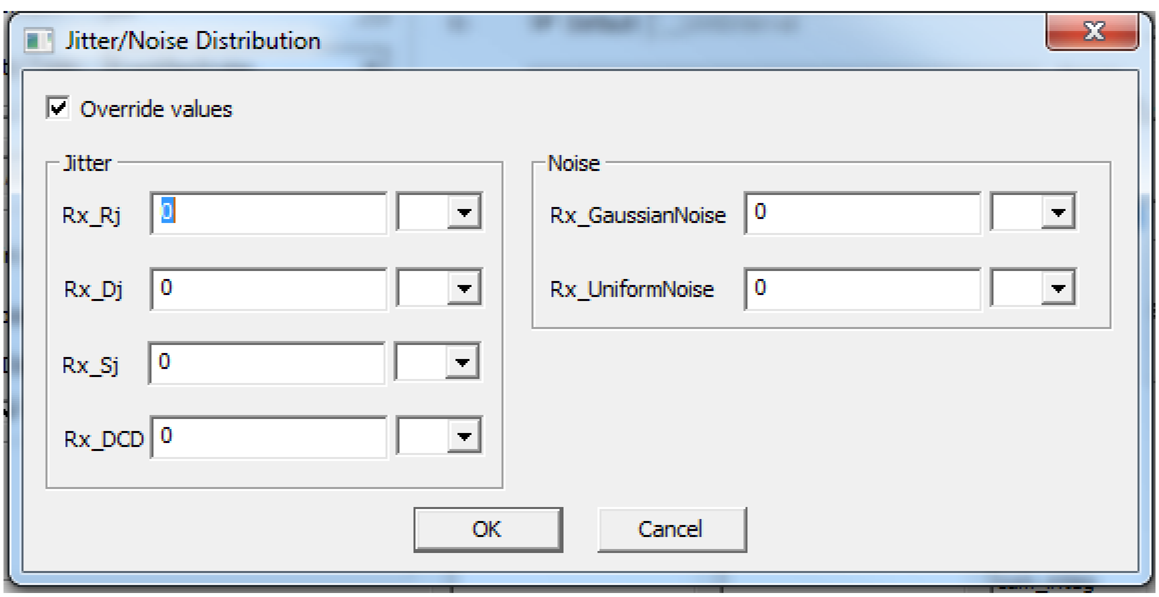
The above shows the symbol error rates for each of the three are depicted as three separate traces at different voltage levels.

Apply Receiver Jitter and Noise Controls Simultaneously (2D and Circuit)

You can apply distributions of receiver jitter (variation in timing) and noise (variation in voltage) to the simulated data, using the controls in the **Distribution** group box of the reporter window. To view the added jitter or noise distribution, a VerifEye, QuickEye, or AMI bathtub plot is used. For AMI, the **Distribution** group box is available with the "Eye After Probe" quantity.



- Click the check box in the **Distribution** group box and select **Jitter/Noise** on the drop-down menu. Then click **Edit** to open the **Jitter/Noise Distribution** window.

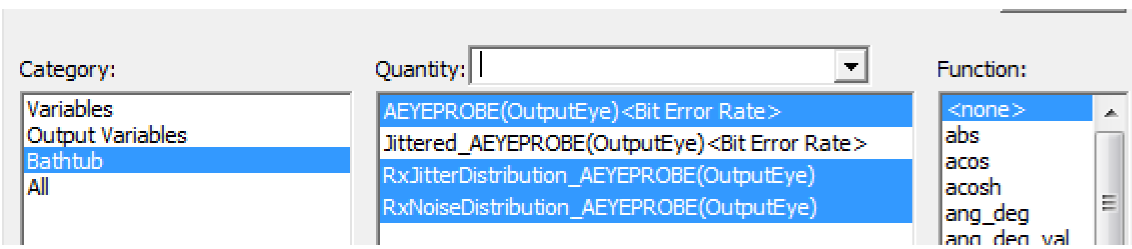


- Add a Gaussian Jitter distribution in the **Rx_Rj** field, giving the standard deviation. Specify the time units.

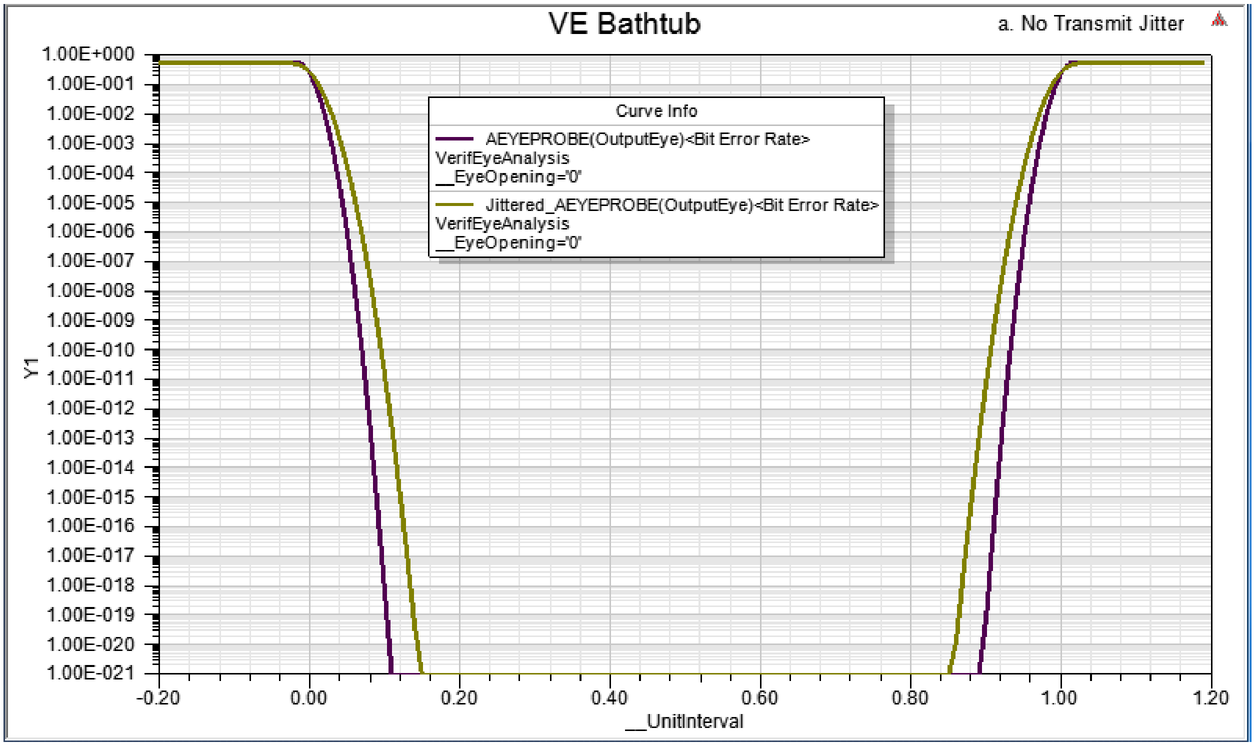
- Add a Uniform Jitter distribution in the **Rx_Dj** field, giving the half-width of the distribution in time, and specify the time units.
- Add a sinusoidal jitter (**Rx_Sj**), giving the amplitude of the distribution in time units.
- Add a clock duty cycle distortion (**Rx_DCD**), giving one-half the peak-to-peak variation of the distortion in time units.
- Add a Gaussian Noise distribution in the **Rx_GaussianNoise** field, giving the standard deviation, and set the voltage units.
- Add a Uniform Noise distribution in the **Rx_UniformNoise** field, giving the half-width of the distribution, and specify the voltage units.
- Click **OK** to close the window.

The parameter names and definitions are as defined in the IBIS specification. See "AMI Receive Jitter Parameters " on page 11-153.

When Jitter/Noise has been enabled, four quantities are available for plotting: the bathtub with no jitter (AEYEPROBE(OutputEye)<Bit Error Rate>), the bathtub with the jitter and noise applied (Jittered_AEYEPROBE(OutputEye), the distribution of the jitter (RxJitterDistribution_ AEYEPROBE(OutputEye), and the distribution of the noise (RxNoiseDistribution_AEYEPROBE (OutputEye).



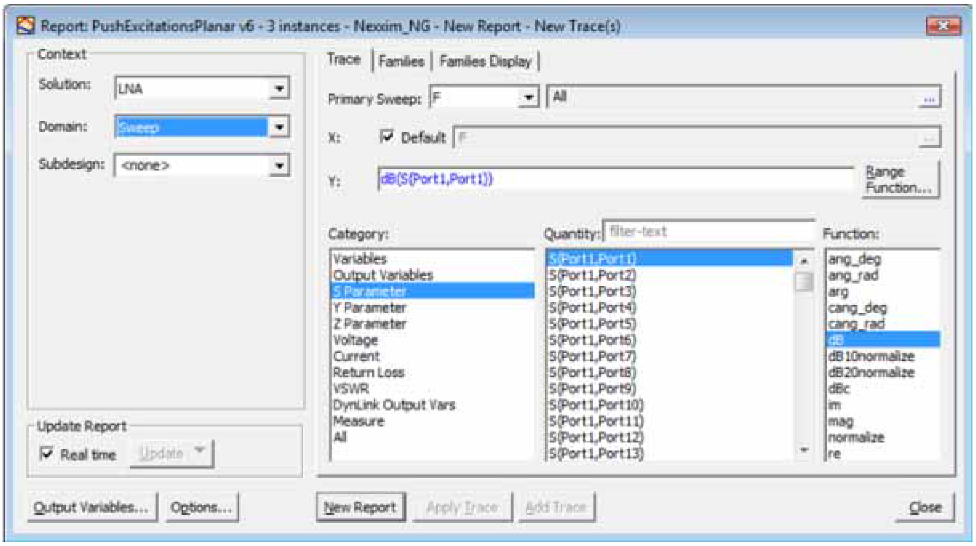
The Bathtub plot shows some example results:



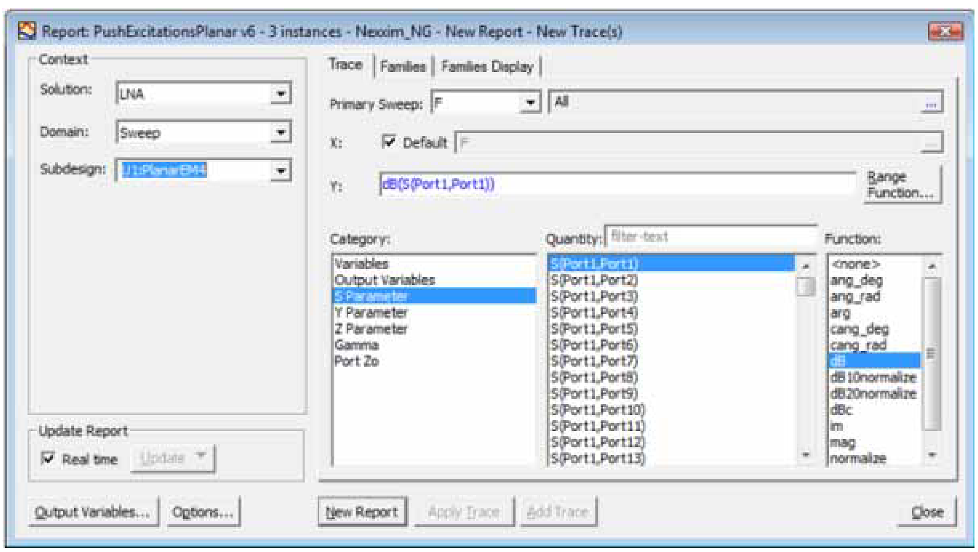
The AEYEPROBE(OutputEye) trace shows the BER curve before the jitter is applied (UI=1ns). The Jittered_AEYEPROBE(OutputEye) trace shows the BER curve after a random receive jitter (**Rx_Rj**) with standard deviation 100ps is applied.

EM Subdesign Plotting on the Top-Level Design (2D and Circuit)

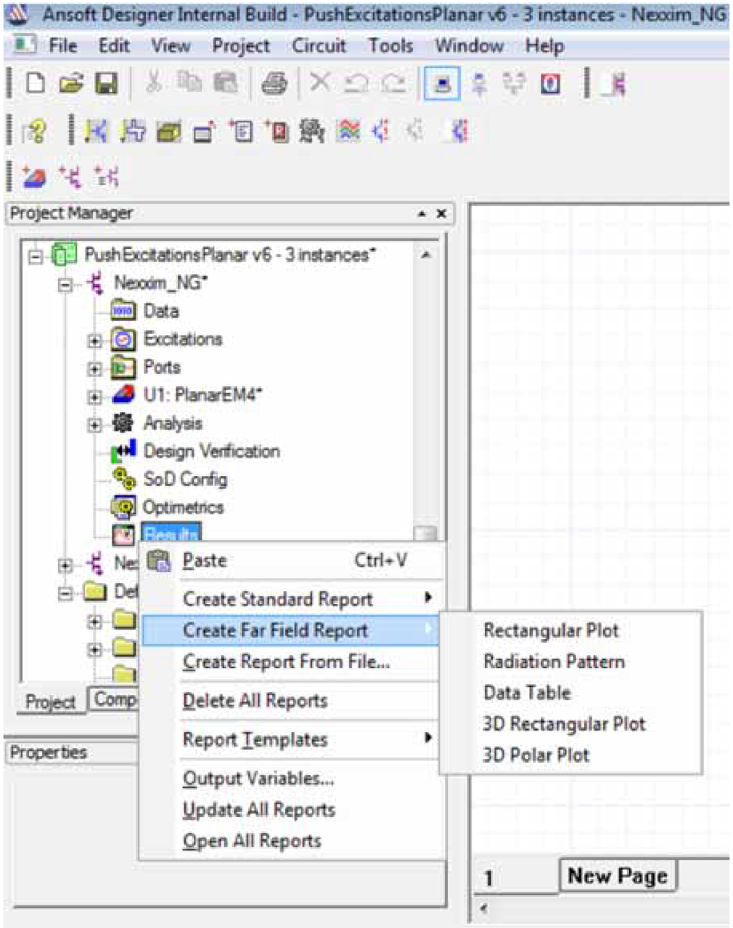
For a Nexxim project with Planar EM subdesigns, plot reports on the EM subdesigns directly on the top-level Nexxim design. When the reporter window is invoked on the top-level Nexxim design, it contains a drop-down menu in the **Context** group box that can specify the **Subdesign**. When the **Subdesign** selection is set to the <none> default, the domain, categories and quantities shown in the reporter window are related to the top level design.



But when the **Subdesign** selection is set to an EM subdesign, the reporter window populates with **Category** and **Quantity** values on the subdesign. This allows you to plot the EM responses on the top-level reporter window, rather than editing the subdesign and opening the reporter window there.

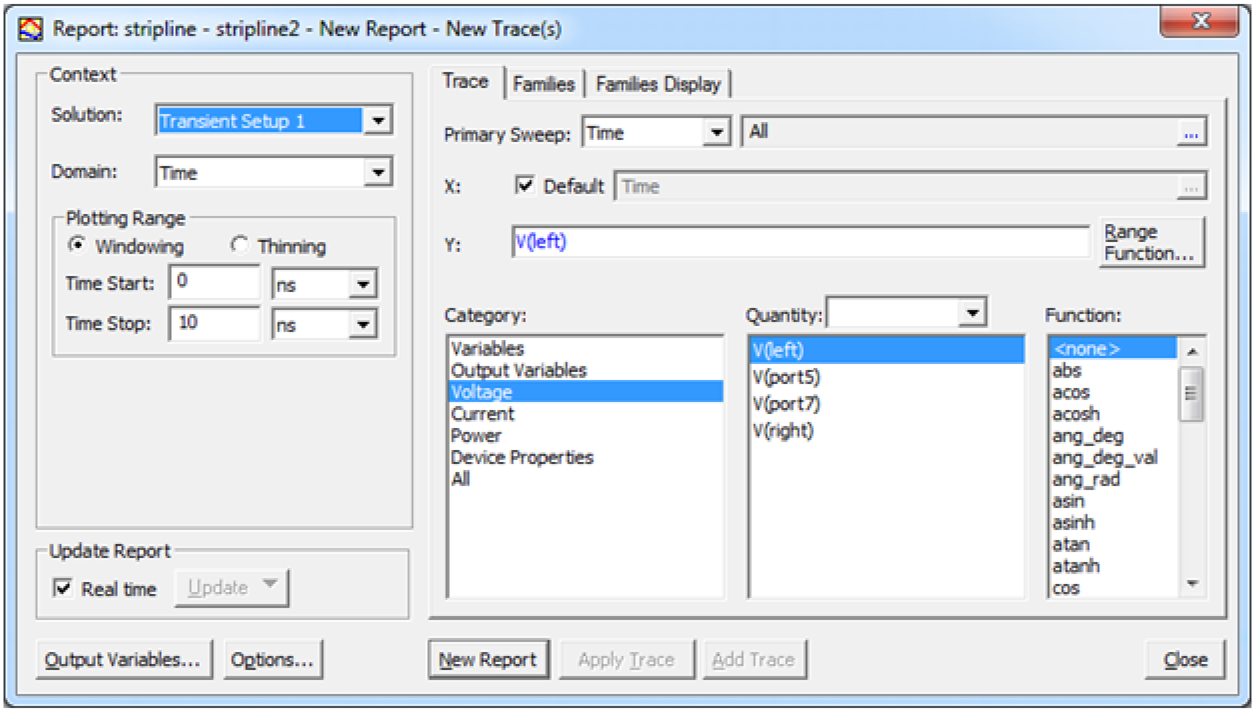


Additionally, plot Far Field reports from the planar EM subdesign.

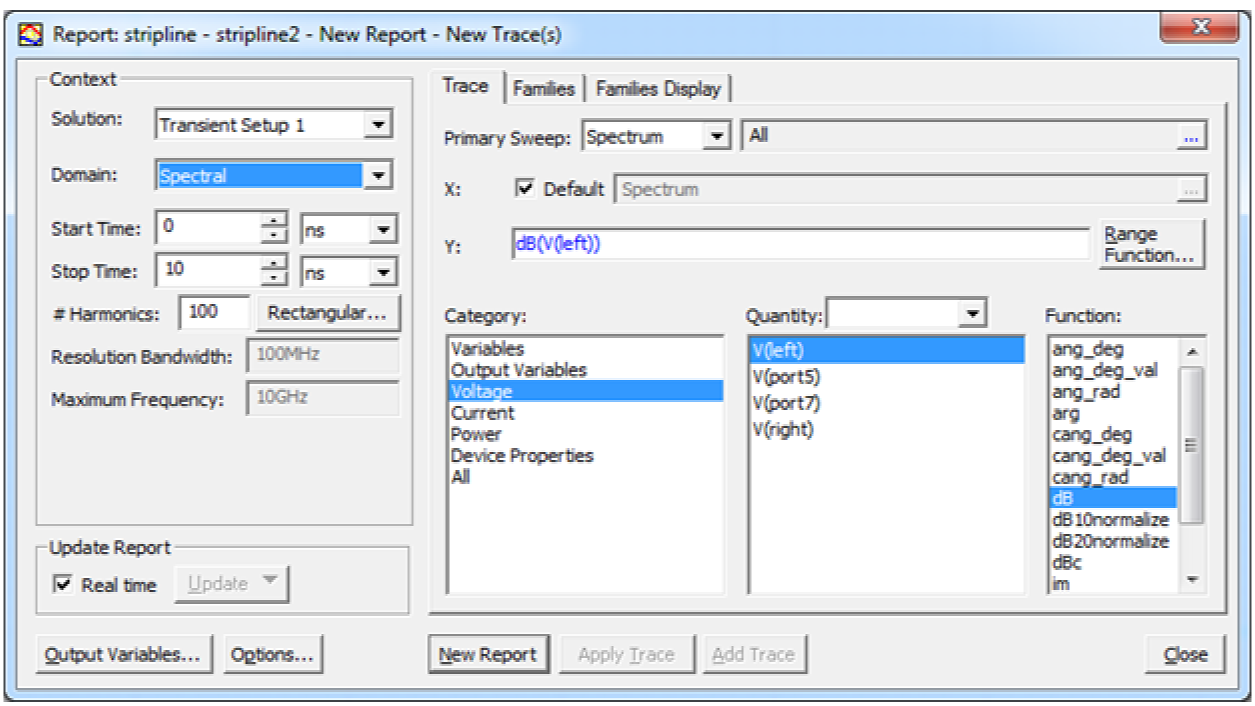


Post Processing for Transient in Layout

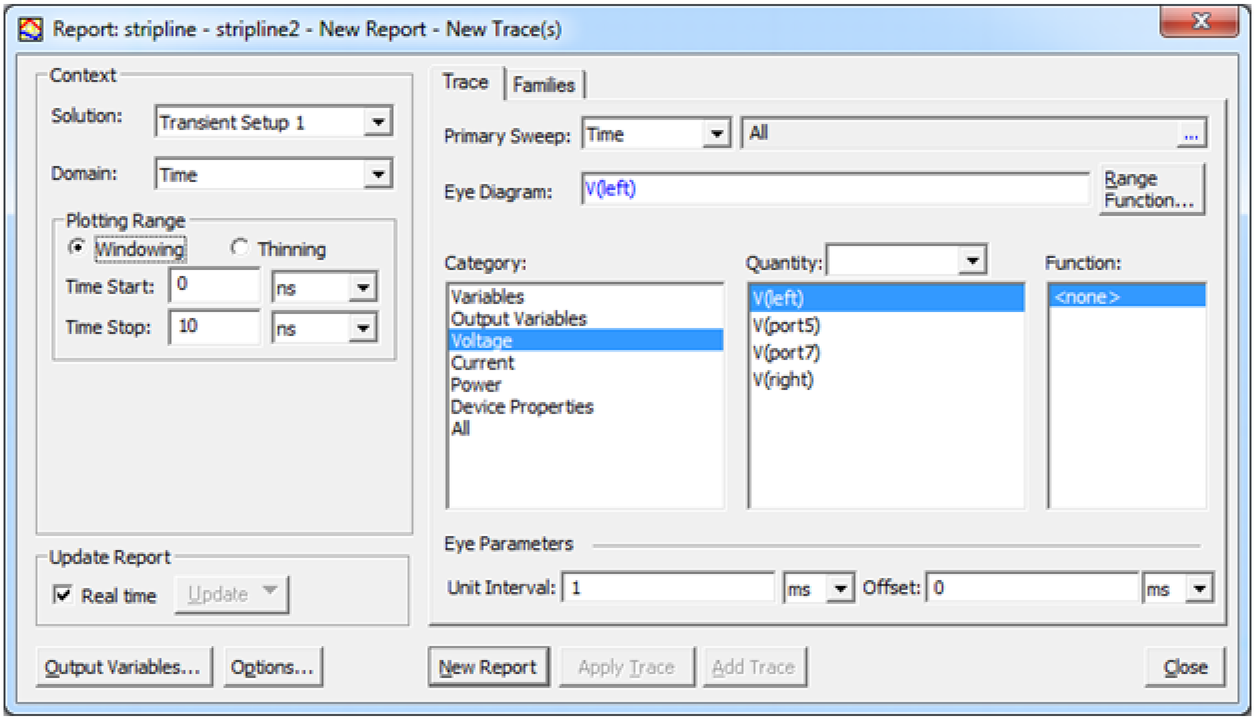
Transient results include voltage, currents, and power quantities in the Time domain and the Spectral domain. In the Time domain, the user can set a range to plot a window of the data or thin the data displayed in the plot.



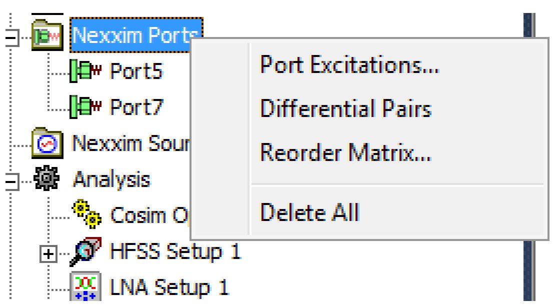
In the spectral domain, set the Start/Stop time and the number of harmonics, then plot spectral plots using the voltage, time, and power quantities.



You can also plot eye diagrams in Time domain.

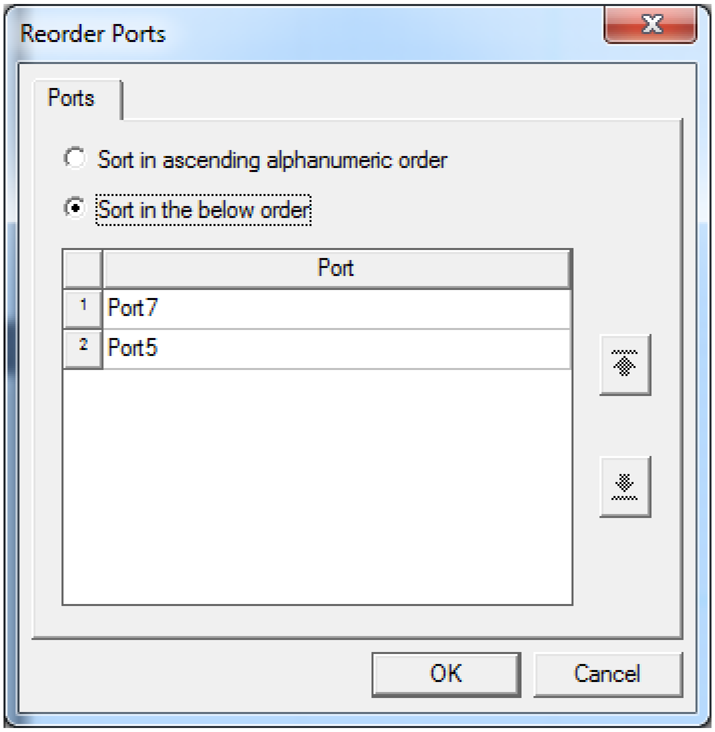


You can also use the post-processing window to reorder ports, set differential pairs, and port excitations for Interface Ports.

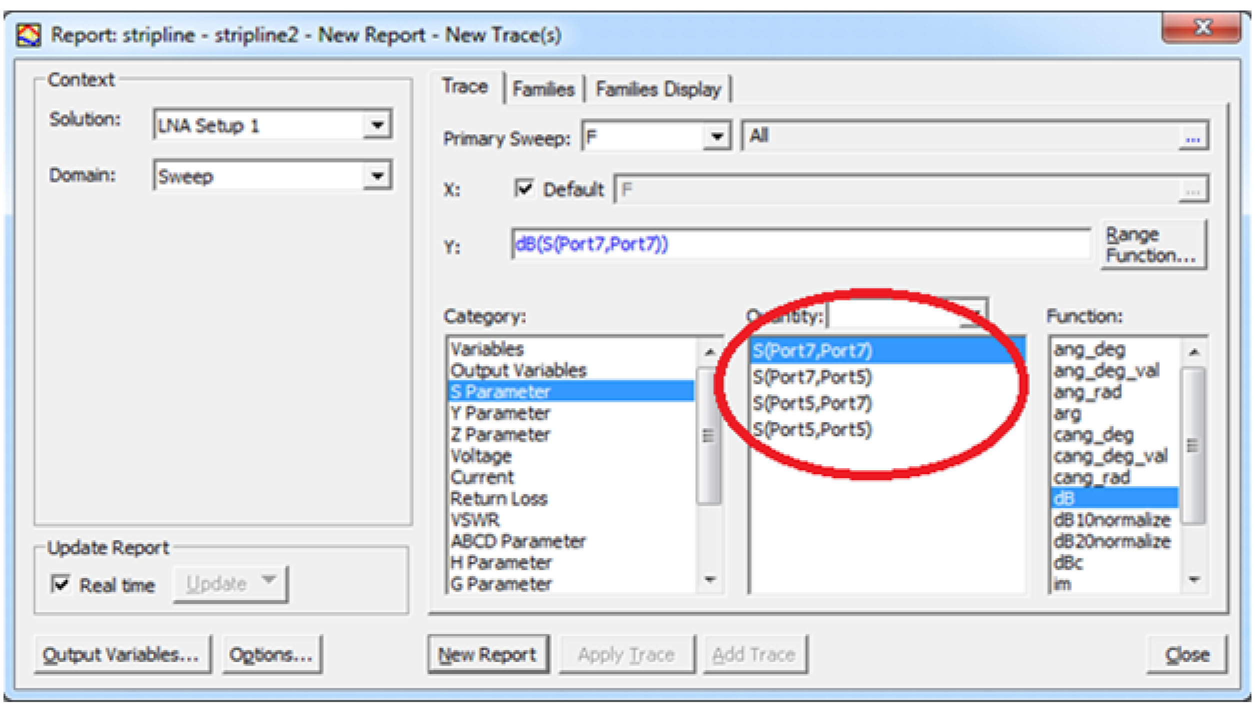


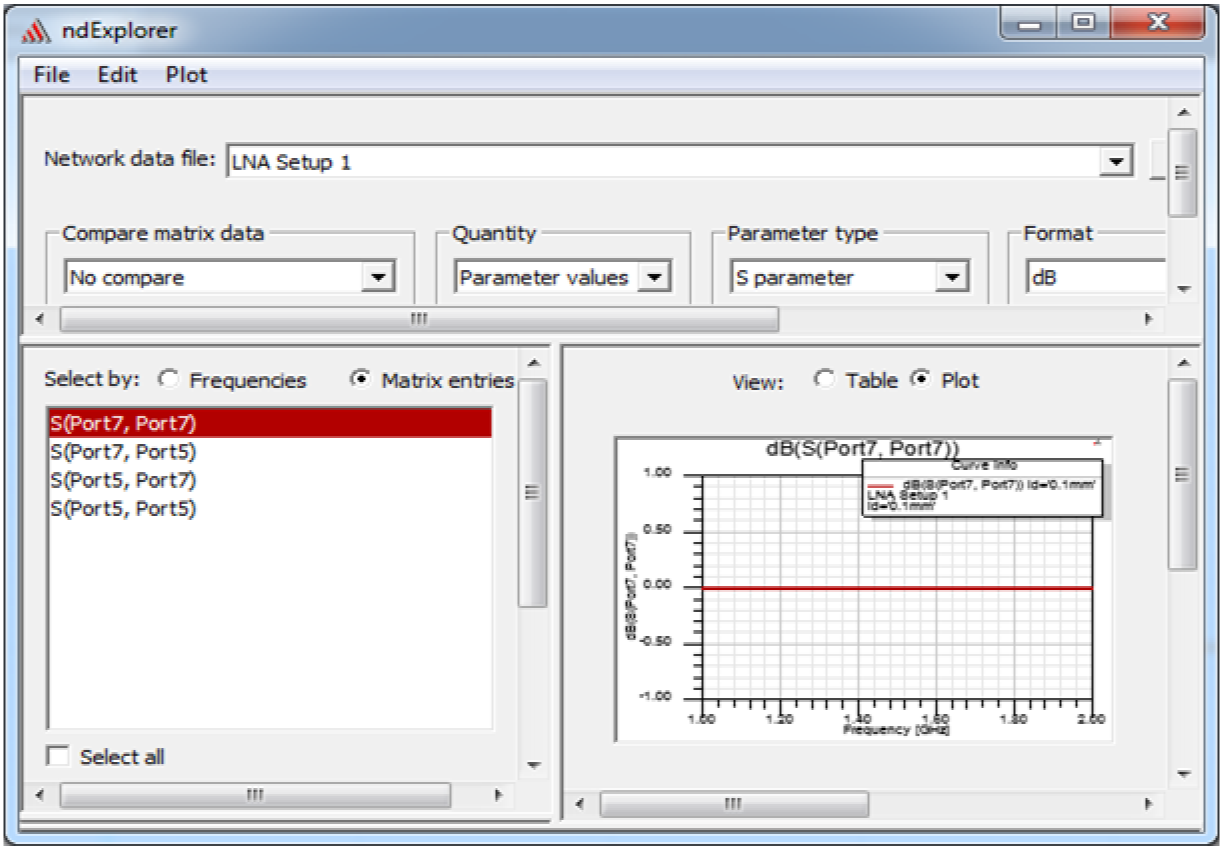
Reorder Nexxim Interface Ports

The user choose to reorder the ports in alphanumeric order or in any custom order of their choice. This order is used to reorder the results matrix and is displayed on the matrix data page, by the Network Data Explorer, and in the Reporter window.



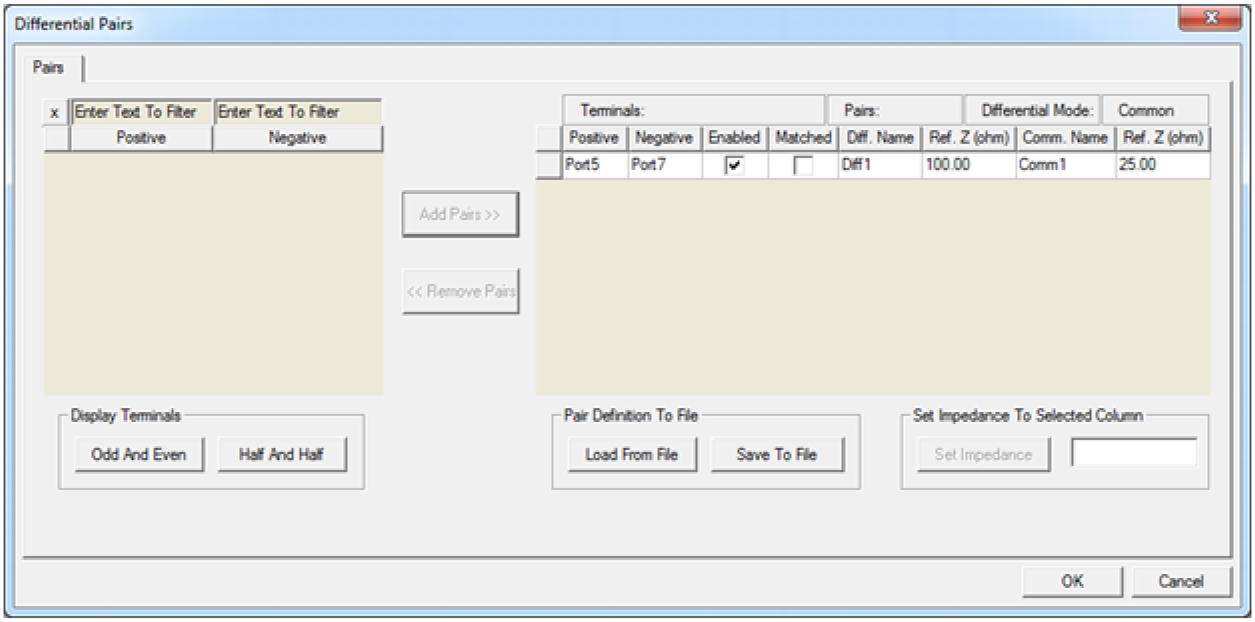
Here is the order as it is displayed in the Reporter and the Network Data Explorer.



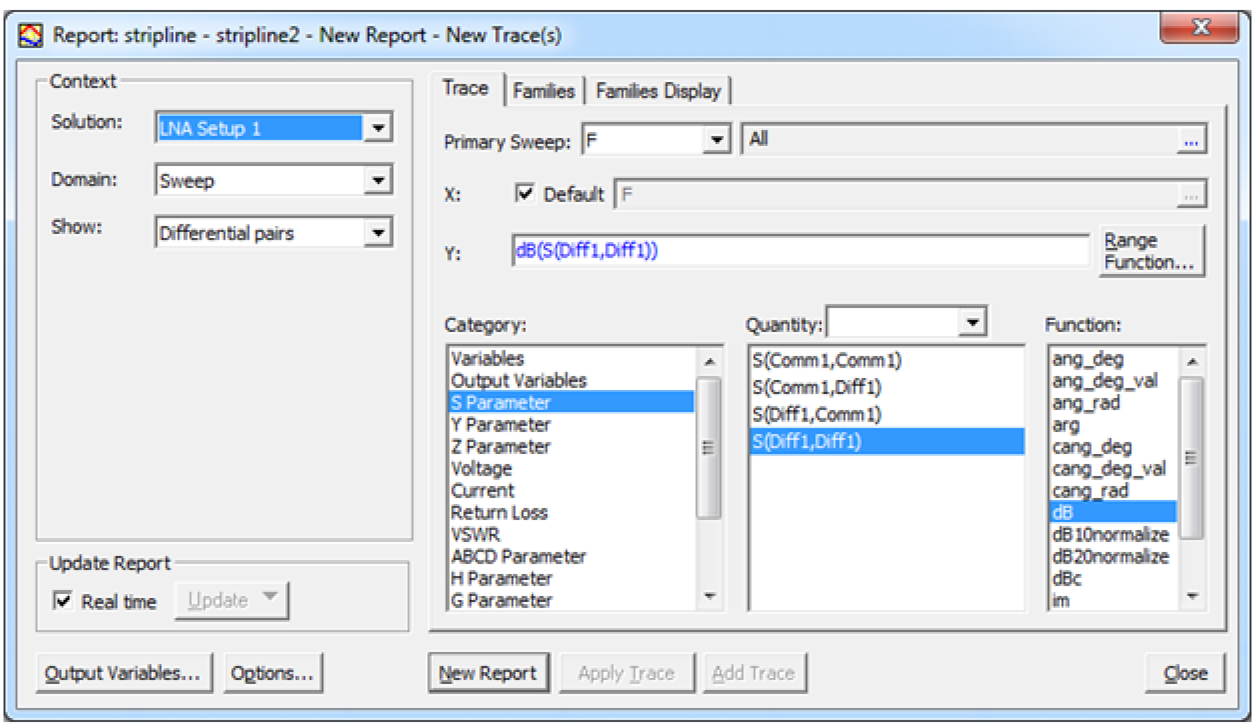


Differential Pairs window

You can set differential pairs for the Interface Ports through the Differential Pairs window. The values can then be accessed in the Reporter window.

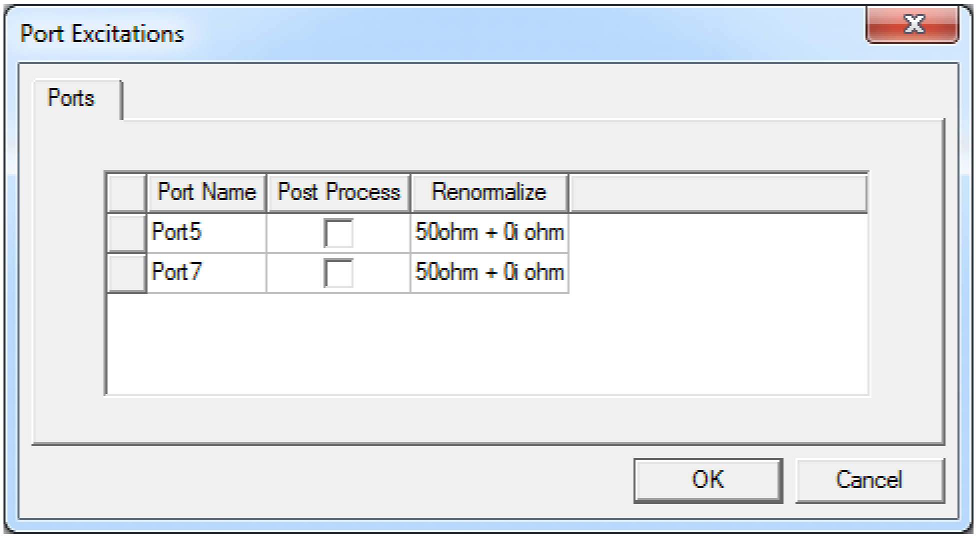


Here is the Reporter window with the differential quantities.



Port Excitations window

You can set the renormalization impedance for the Interface Ports using the Port Excitations window.



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).

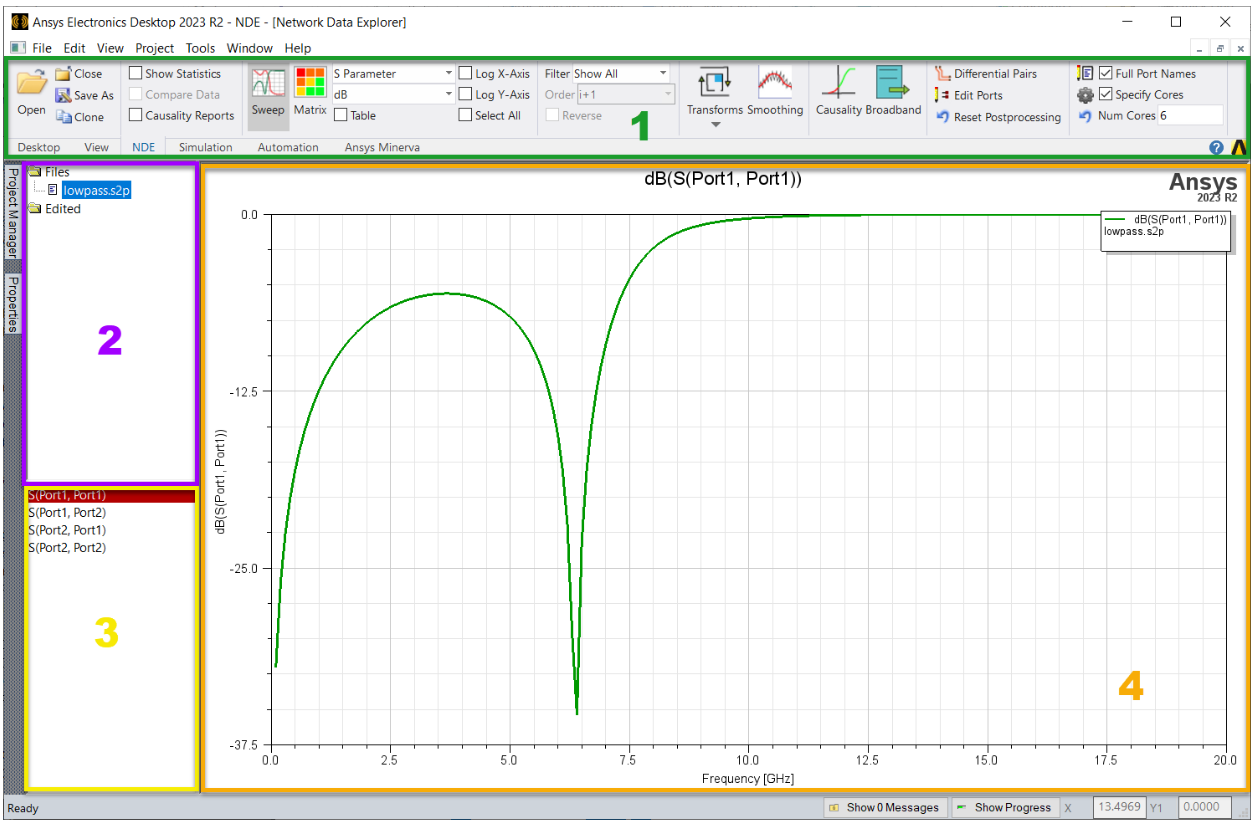
For more information, consult the SIwave and HFSS 3D Layout help.

19 - 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.

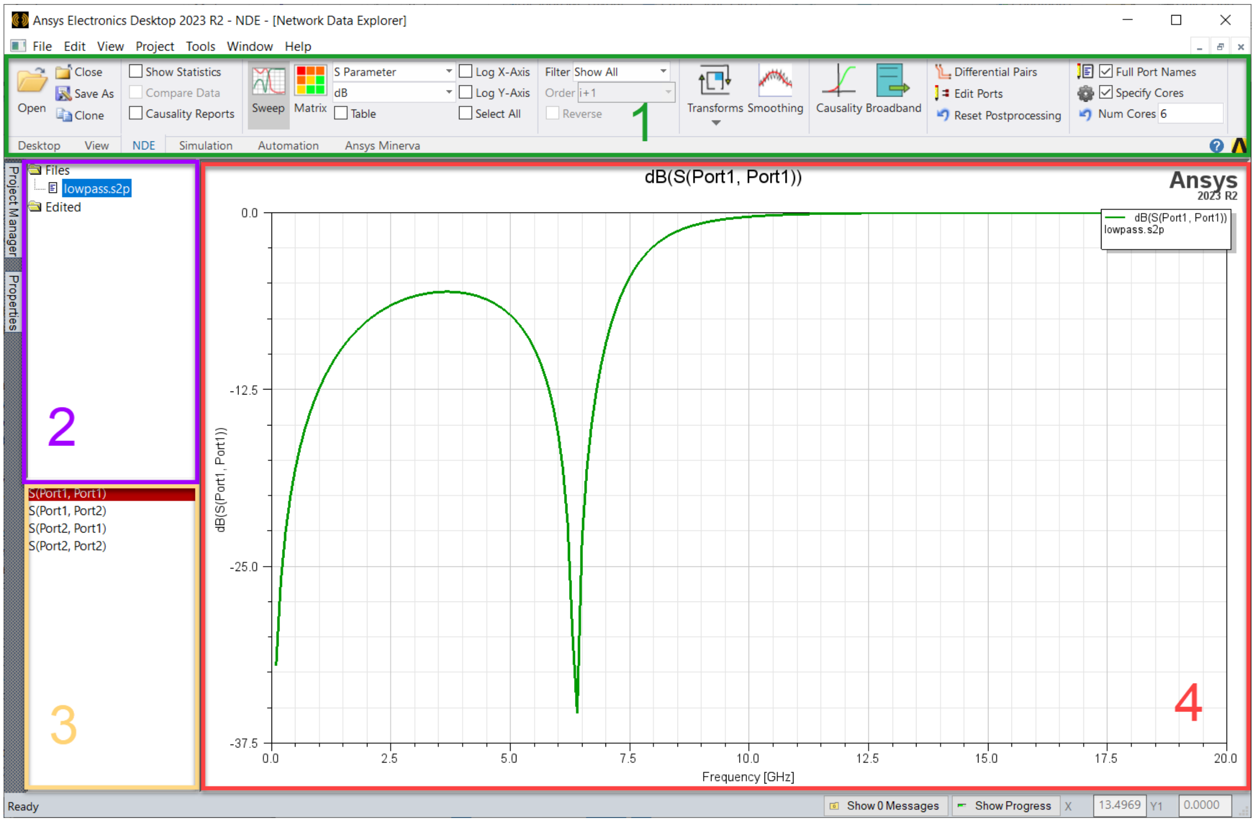


Network Data Explorer Overview

The Network Data Explorer window is divided into the following panes:

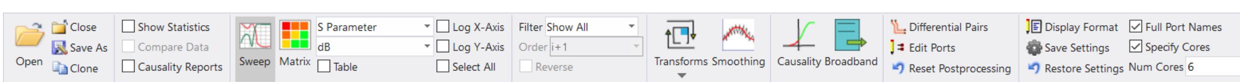
1. **NDE ribbon** – allows you to perform many functions of the Network Data Explorer.
2. **Network Data Selection pane** – allows you to select a network data file.
3. **Cell and Frequency Selection pane** – allows you to narrow your selection.
4. **Data View pane** – displays data in table or plot 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.



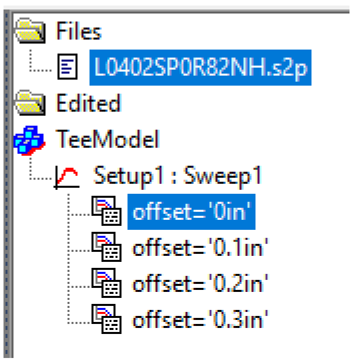
On this ribbon, you can control:

- 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.

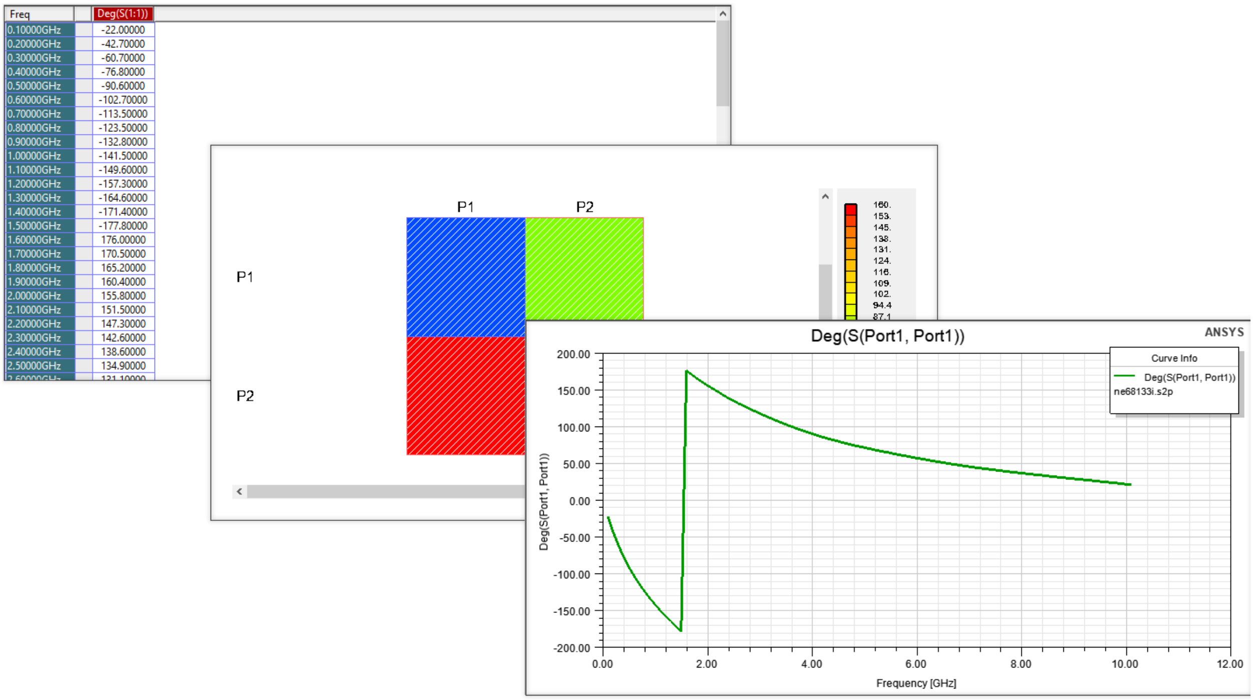
Use the **Shift** and **Ctrl** keys to select and display data sets from multiple files simultaneously. Multiple data sets can be selected to display multiple traces in sweep plots, but not in tables. Click the **Files** icon to select all data sets under **Files**. Click the **Edited** icon or a design/setup 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:

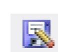
Exporting SYZ Data

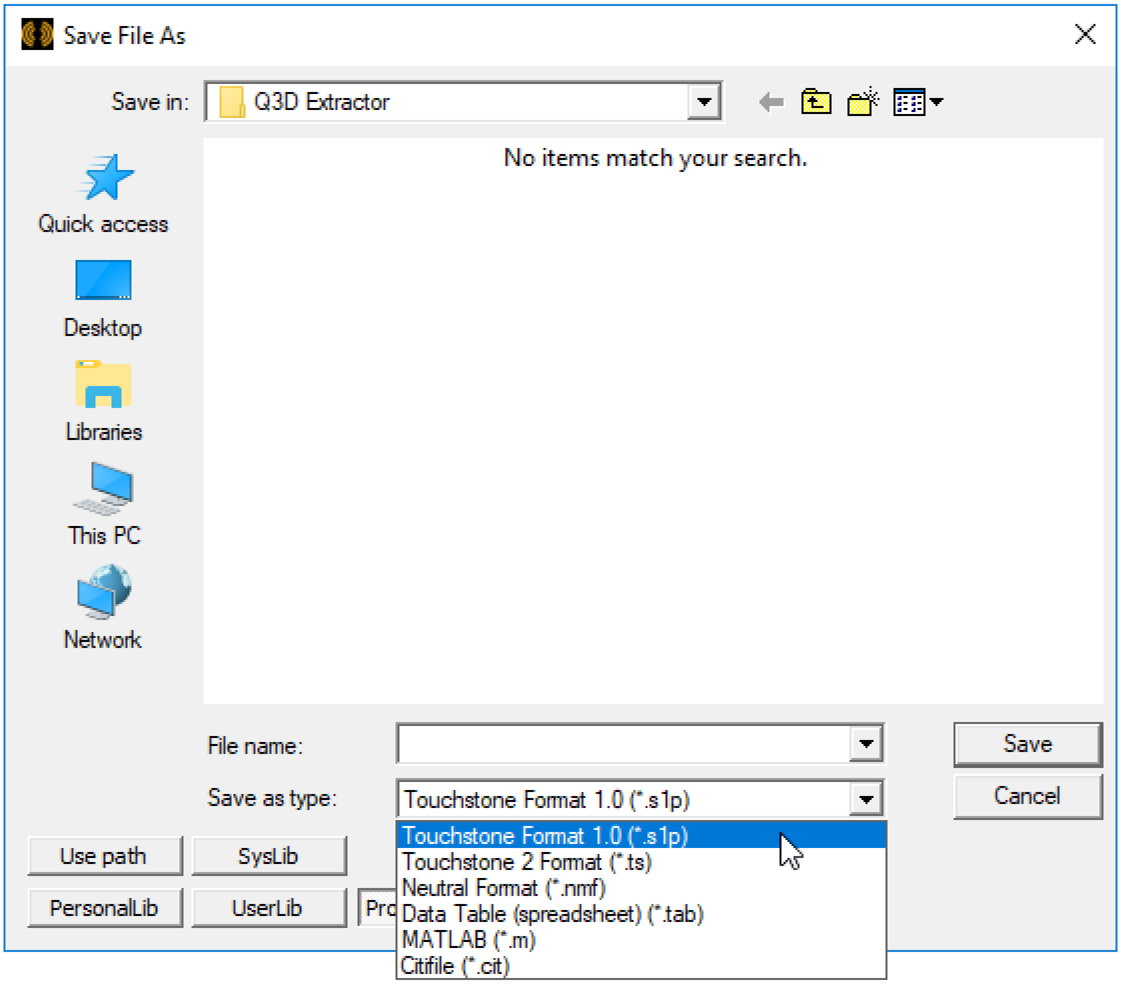
Exporting Macro Model

Creating an NPort Model

Scripting for Network Data Explorer

Exporting SYZ Data

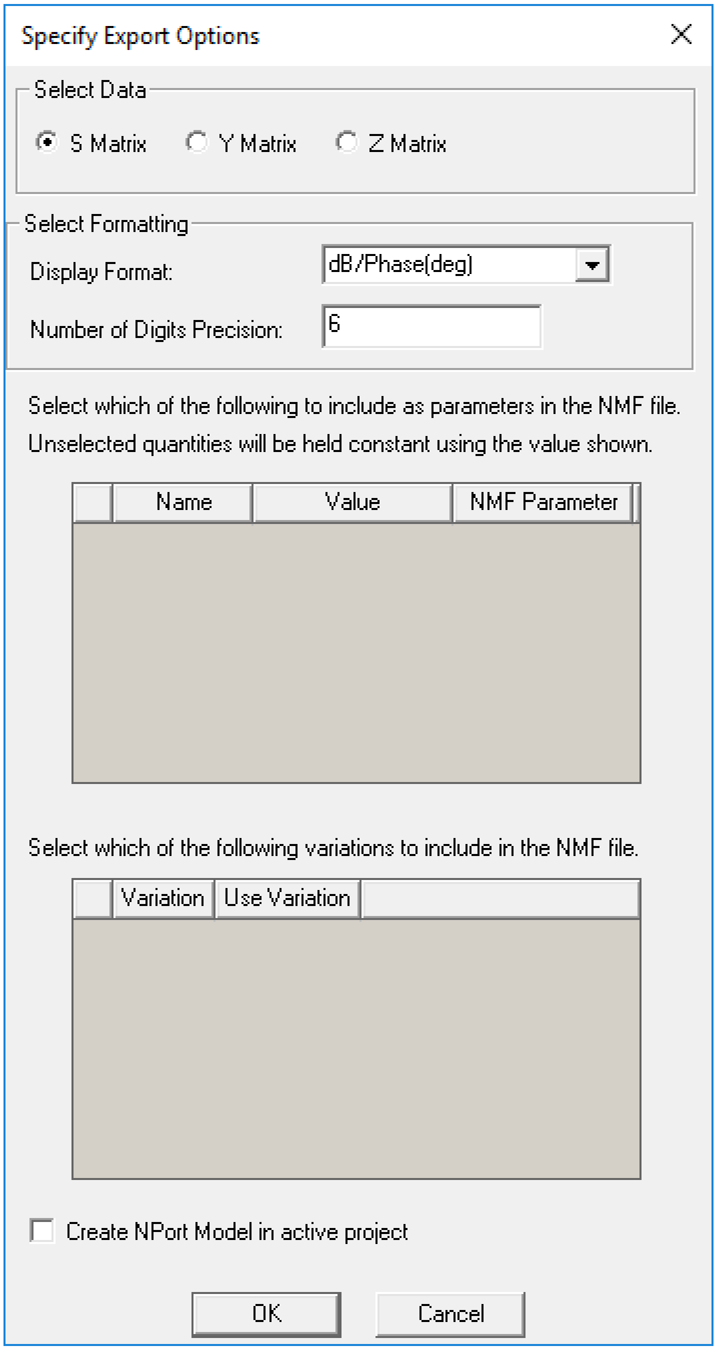
To export SYZ data from within Network Data Explorer, click the **Save As** icon  on the **NDE** ribbon. The **Save File As** window appears.

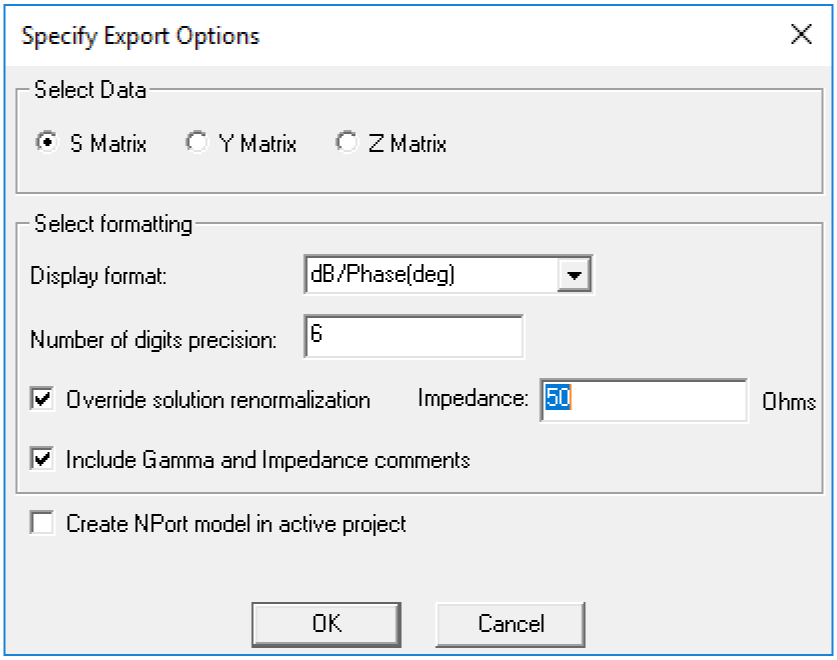


You can export data in any of six file types:

- 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.


Specify Export Options
Select Data
S Matrix
Y Matrix
Z Matrix
Select Formatting
Display Format:
dB/Phase(deg)
Number of Digits Precision:
6
Select which of the following to include as parameters in the NMF file.
Unselected quantities will be held constant using the value shown.
Name
Value
NMF Parameter
Select which of the following variations to include in the NMF file.
Variation
Use Variation
Create NPort Model in active project
OK
Cancel



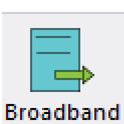
Depending on the type of export file, different options appear. However, all file types allow you to:

- Select from **S Matrix**, **Y Matrix**, and **Z Matrix** data.
- Select the **Display Format**.

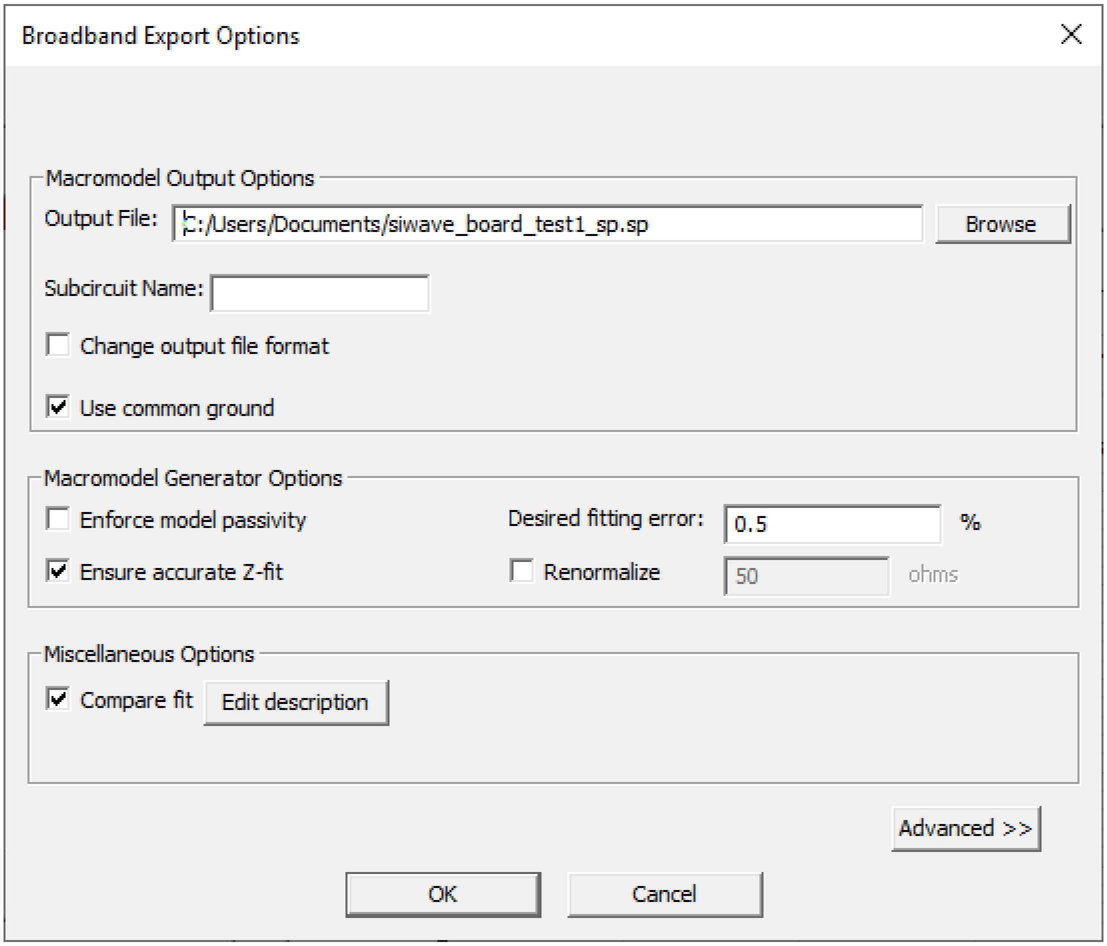
Some types allow you to create an NPort Model in the active project.

Exporting Macro Model

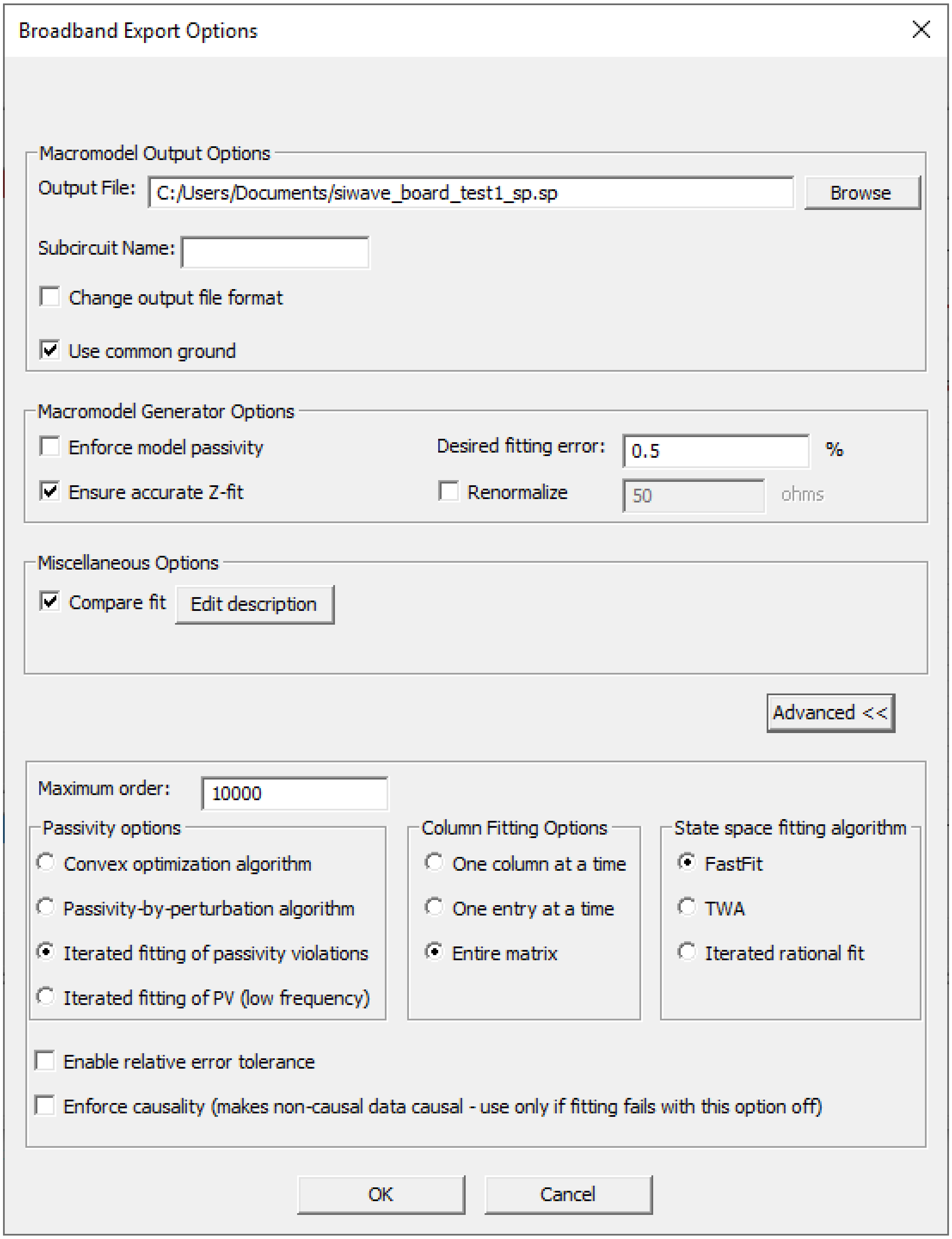
Network Data Explorer lets you export macro model data. To export data, click the **Broadband** icon on the **NDE** ribbon.



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.

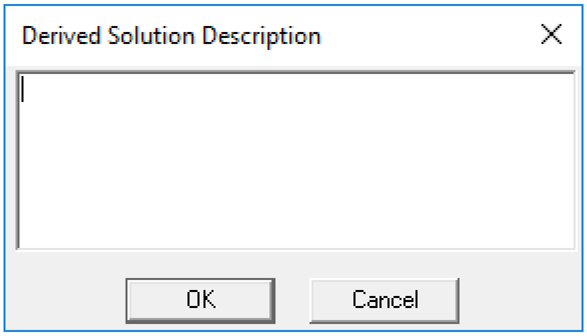
Macromodel Generator Options include:

- **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.

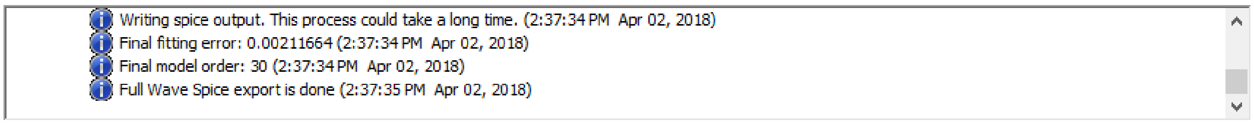
- **State Space Fitting Algorithm** – Allows you to select FastFit, TWA, Iterated rational fitting.
  - **FastFit** (default) – FastFit is the Ansys-proprietary method for state-space fitting. Network Data Explorer uses FastFit for calculating the state-space matrices from the network data. The FastFit 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

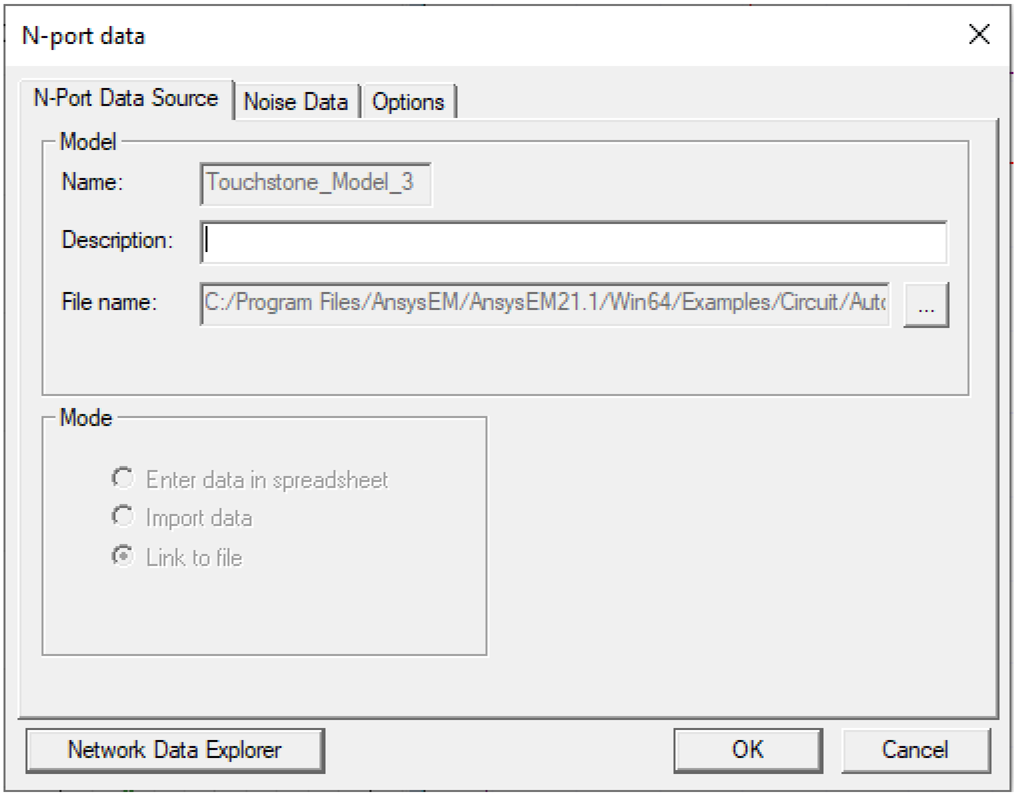
Touchstone, Neutral Format, and Citifile exports allow you to select an option to **Create NPort Model in active project**.

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.

State Space N-Port Data Source Tab

The **N-Port Data Source** tab is used to name the N-port device and specify the source for the data, typically an external file.



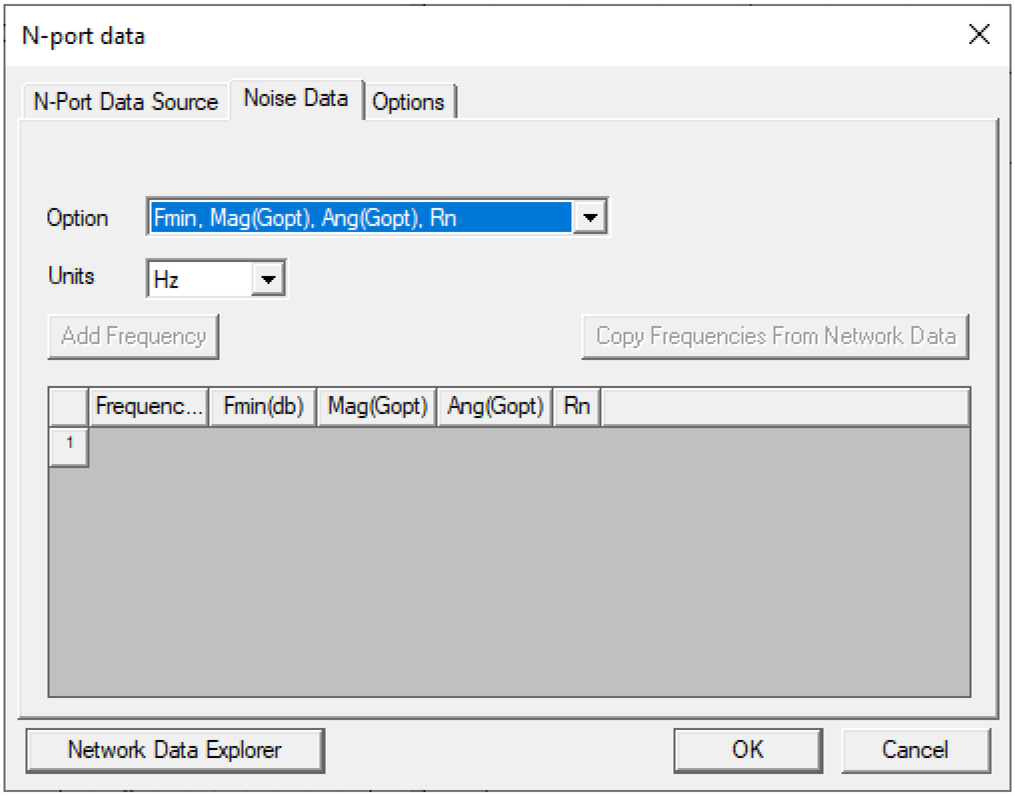
- **Name** — Displays the name of the imported solution as it appears in the **Project Tree**.
- **Description** – Used to set or change the description of the imported solution. To change the description, click in the box , then type the description you want.
- **File Name** – Used to display or change the file name of the imported solution. To change the file name, browse (...) for a file.
- **Mode** – The network solution data is read from an external **Link to file** at analysis time.
- **Network Data Explorer** allows you to view solution data by opening the Network Data Explorer.

> **Note:**
>
> **Network Data Explorer** is not activated for State-Space models.

When the **N-Port Data Source** options have been entered, click **OK** or select another tab.

State Space N-Port Data Noise Data Tab

Click the **Noise Data** tab to enter frequency-dependent noise parameters.

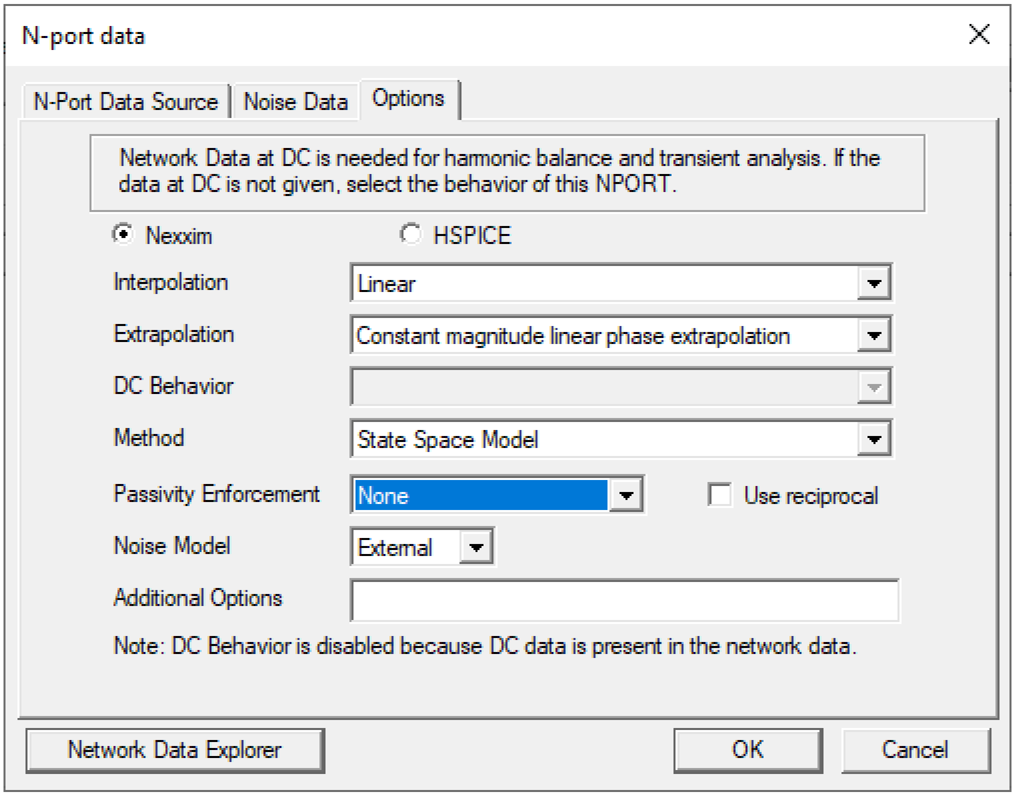


- The **Option** field selects the units for the noise parameters:
- The **Units** field selects the units of frequency: **Hz**, **KHz**, **MHz**, **GHz**, **THz**, or **rps**.
- click **Copy Frequencies from Network Data** to create a list of frequencies that correspond to the frequencies in the N-port data file.
- click **Add Frequency** to add a new frequency to the end of the list.

Enter the noise data for each frequency in the list, then click **OK** or select another tab.

State Space N-Port Data Options Tab

The **Options** tab is used to select the behavior of the selected NPort.

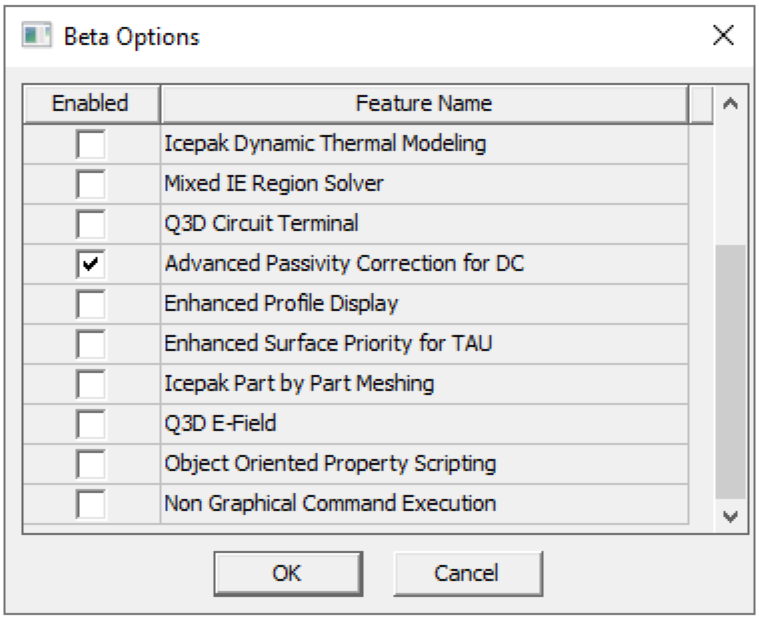


The following controls are available:

- **Nexxim/HSPICE** — Sets the NPort type. The selections for Interpolation, Extrapolation, DC Behavior, Method, and Passivity Enforcement depend on the NPort type selected.
- **Interpolation** — Specifies the interpolation method for data points in the frequency range of the Touchstone data.
- **Extrapolation** – Specifies the extrapolation method for data points outside the frequency range of the Touchstone data.
- **DC Behavior** — Specifies behavior of the N-Port at DC (zero frequency).
- **Method** – Specifies the method used to convert frequency domain data to the time domain. Choice of methods is available only for Nexxim Nports.
- **Passivity Enforcement** — Performs a passivity check on the S-parameter data and attempts to correct non-passive data. The options are None, IFPV, IFPVLF, Convex Optimization, and Passivity by Perturbation.

  Passive devices may dissipate or temporarily store energy, but never generate energy. Use the drop-down menu to select the method for enforcing passivity. Passivity enforcement is available only for Nexxim NPorts. See Passivity Checking and Enforcement. For details on passivity checking and enforcement, see S-Parameter Technical Notes.

When applying IFPVLF, the passive DC fit to Scattering Parameter Data (S-data) can deteriorate to the point the inaccuracies of the (passive) fit to the corresponding Impedance Data (Z-data) at zero frequency (DC) and low-frequency may cause inaccurate time-domain steady state simulations. If preserving an accurate passive DC fit to S-data is important, like it is in power electronics applications, e.g., with DC-DC converters and power supply Voltage Regulator Module (VRM)), then click the **Advanced Passivity Correction for DC** beta option to enhance the passive DC fit to S-data using IFPVLF. The **Advanced Passivity Correction for DC** is located in the **Beta Options** in the Global Options.



- **Use reciprocal** – Computes the inverse, or reciprocal, at each frequency so the N-port can be used for de-embedding. See Deembedding S-Matrices.
- **Noise Model** – Specifies noise model for DC and frequency-domain analyses:
  - **None** – No noise analysis
  - **Internal** – Use internal noise model
  - **External** – Use noise data from Touchstone file if present, else use internal noise model
- **Additional Options** — Allows you to add model-level options for Nexxim or HSPICE.

When the N-port Options have been entered, click **OK** or select another tab.

- **Network Data Explorer** – Allows you to view solution data by opening the Network Data Explorer.

When the N-port Options have been entered, click **OK** or select another tab.

Scripting for Network Data Explorer

Scripting is available for each Network Data Explorer export method, which means a script can be recorded to duplicate the export process.

Network Data Explorer can be invoked in the following contexts, and scripting is available from the Project Context and the Design Instance Context (simulation setup):

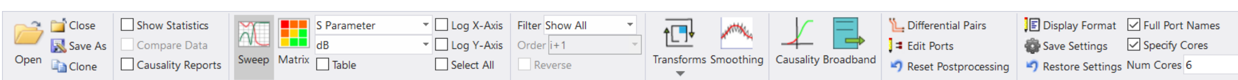
- **Project Context** – In the context of a project, you can open a touchstone file and then export.
- **Design Instance Context** – In the context of a solution (RCM from the simulation setup in a design), you can export the corresponding network data solution.

When there is no design available, export functionality and scripting are not available.

For more information, see the Circuit 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

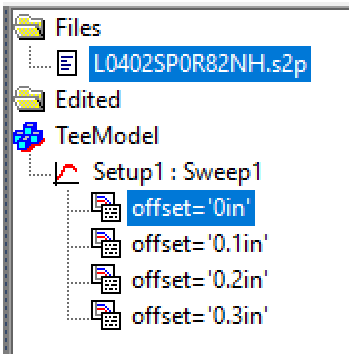
Editing Port Properties

Defining Differential Pairs and Displaying Mixed Mode Parameters

Resetting All Port Properties

Network Data Explorer Data Sources

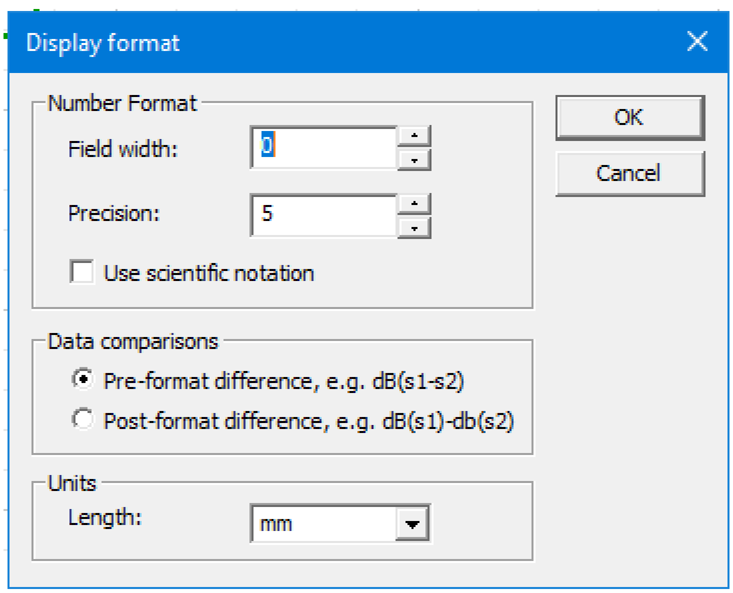
Network Data Explorer allows you to easily view all data sources in the **Network Data Selection** pane.



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.

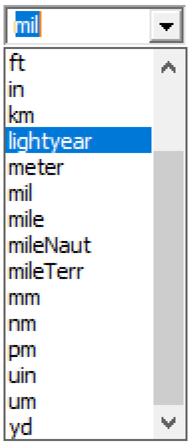


The **Number Format** options allow you to specify **Field Width** (the minimum number of characters used to display a number) and **Precision** (the number of decimals to display). You can also check the **Use scientific notation** check box, if desired.

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.

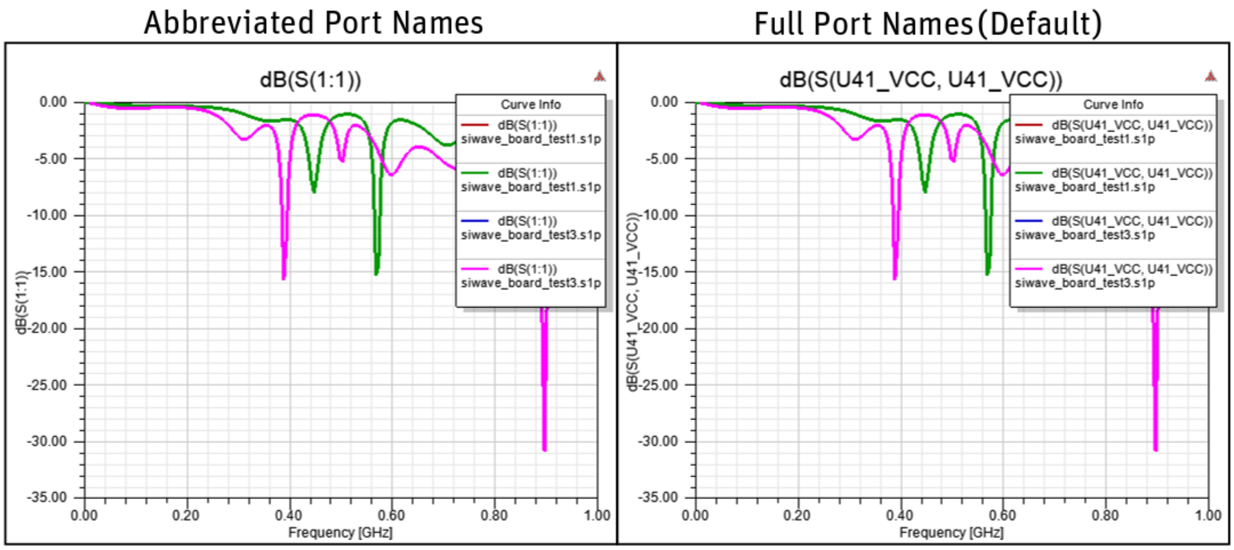
The **Units** option allows you to specify the **Length** unit (the unit used to display and interpret length values). The default is mm.



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

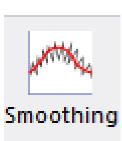
To save field settings as the default, click the **Save Settings** icon  on the **NDE** ribbon. The next time Network Data Explorer is opened, the chosen settings will be selected by default.

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.

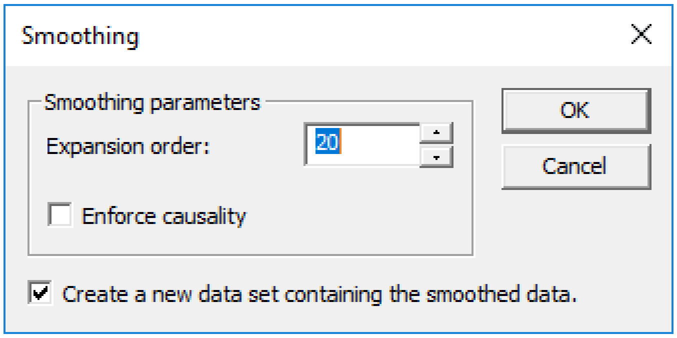
Smoothing

To access data smoothing options:

1. Click the **Smoothing** icon on the **NDE** ribbon.

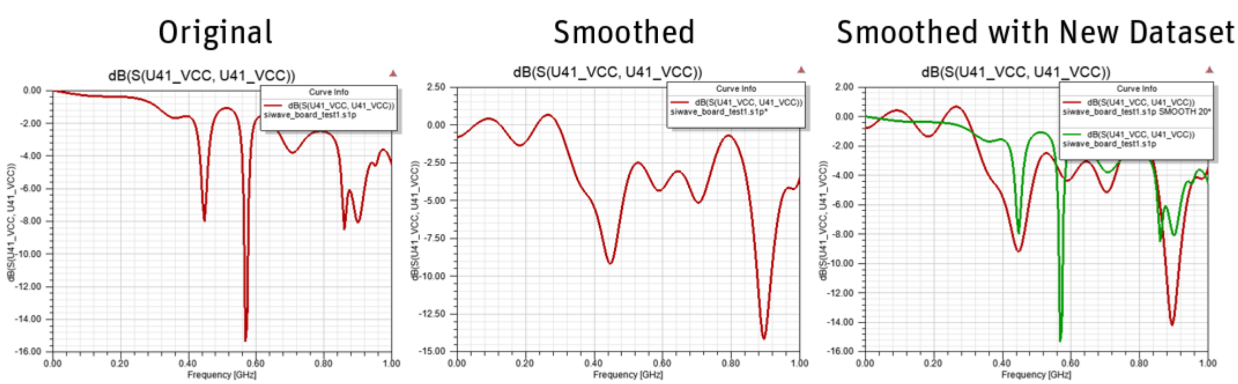
   

   The **Smoothing** window appears.



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**.

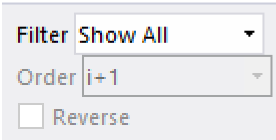
The Data View Pane updates. The following image shows a plotted result with **Create a new data set containing the smoothed data** unchecked and checked.



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:

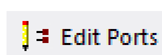
- **Show all** – display all available cells. There are n-squared choices.
- **Return loss** – show S(i, i). There are n choices.
- **Insertion loss** – show S(i, i+1). There are n choices.
- **Lower triangle** – show S(i, j) for all j < i. There are n(n-1)/2 choices.

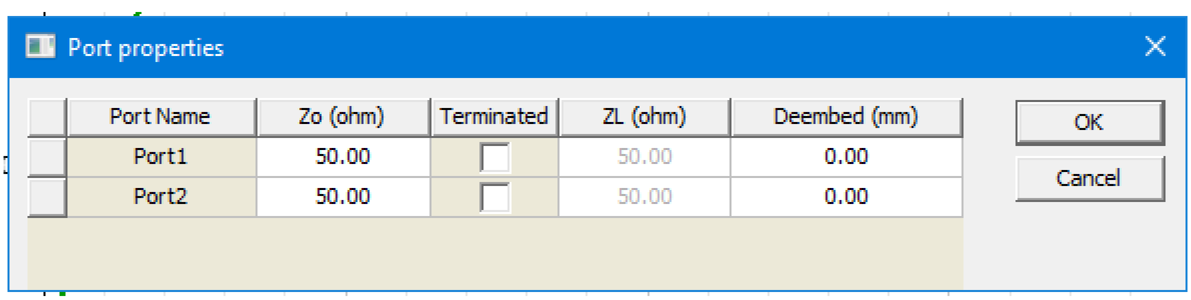
Three pin arrangements are recognized:

- 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.

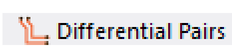
To save changes, click **OK**.

Displaying Mixed-Mode Parameters using Differential Pairs

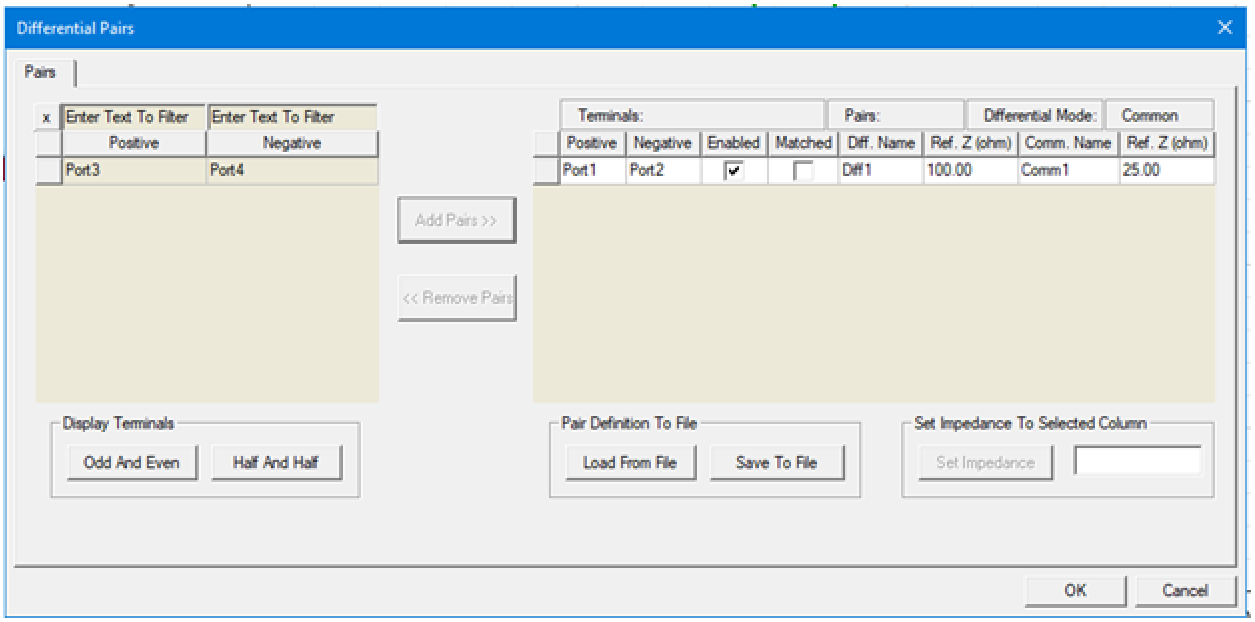
Network Data Explorer displays mixed-mode parameters when differential pairs are both defined and activated.

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**.

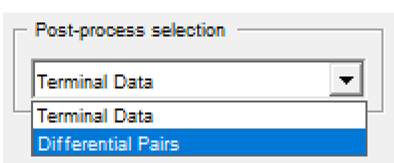


The **Differential Pairs** window appears.



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.

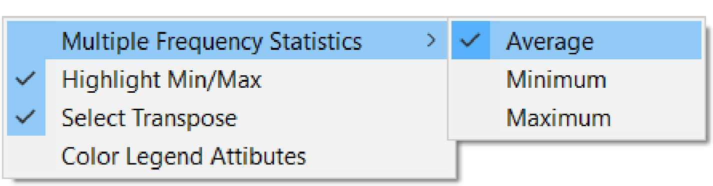
Network Data Explorer commands unique to the context menus are:

- 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.

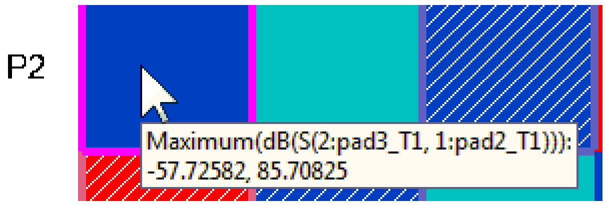
This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



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.

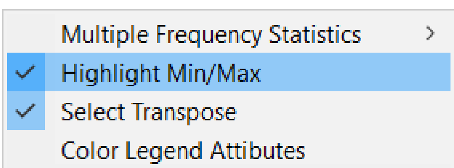
The selected information also appears in a tool-tip when you hover the cursor over a cell.



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.

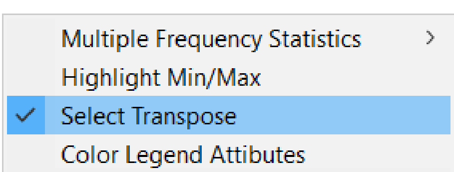
This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



Select Transpose

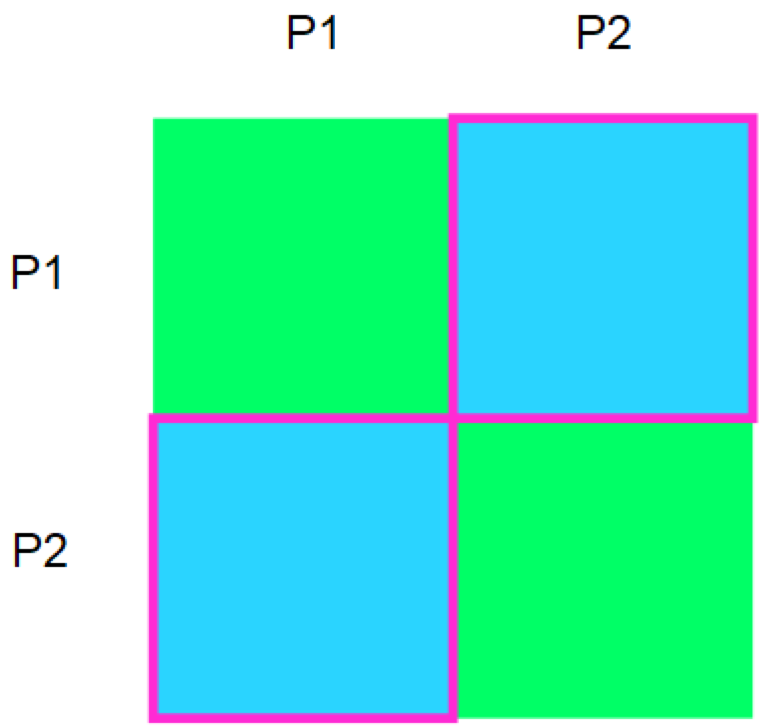
The **Select Transpose** menu option determines whether transpose cells are highlighted along with selected cells.

This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



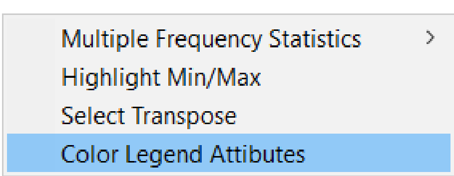
Transpose cells are highlighted in Table or Plot view, as shown below.

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

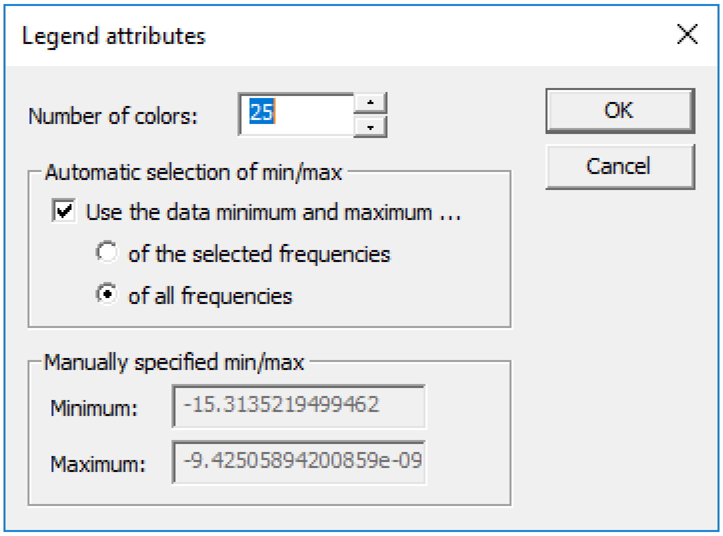


Color Legend Attributes

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.



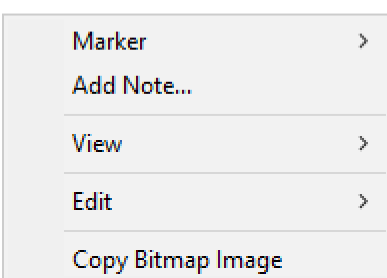
Options include:

- **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.

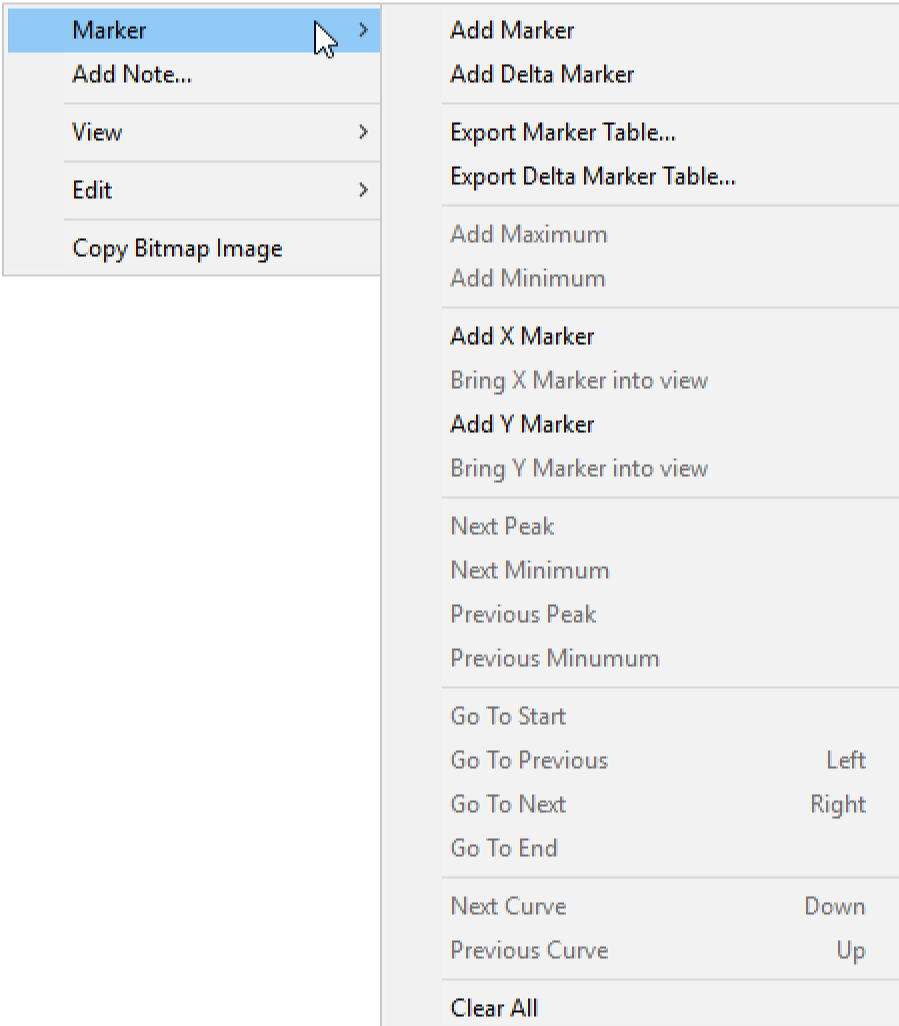
Using a standard range across all frequencies permits you to quantitatively compare plots, and ndExplorer remembers legend settings for each data-type and display-format pair.

Matrix Entries Plot Menu

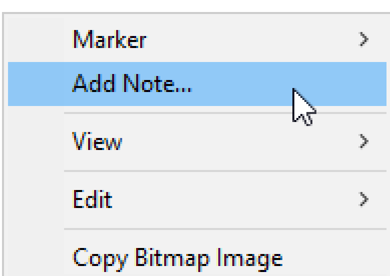
On the **Matrix** plot, several new right-click menu options appear.



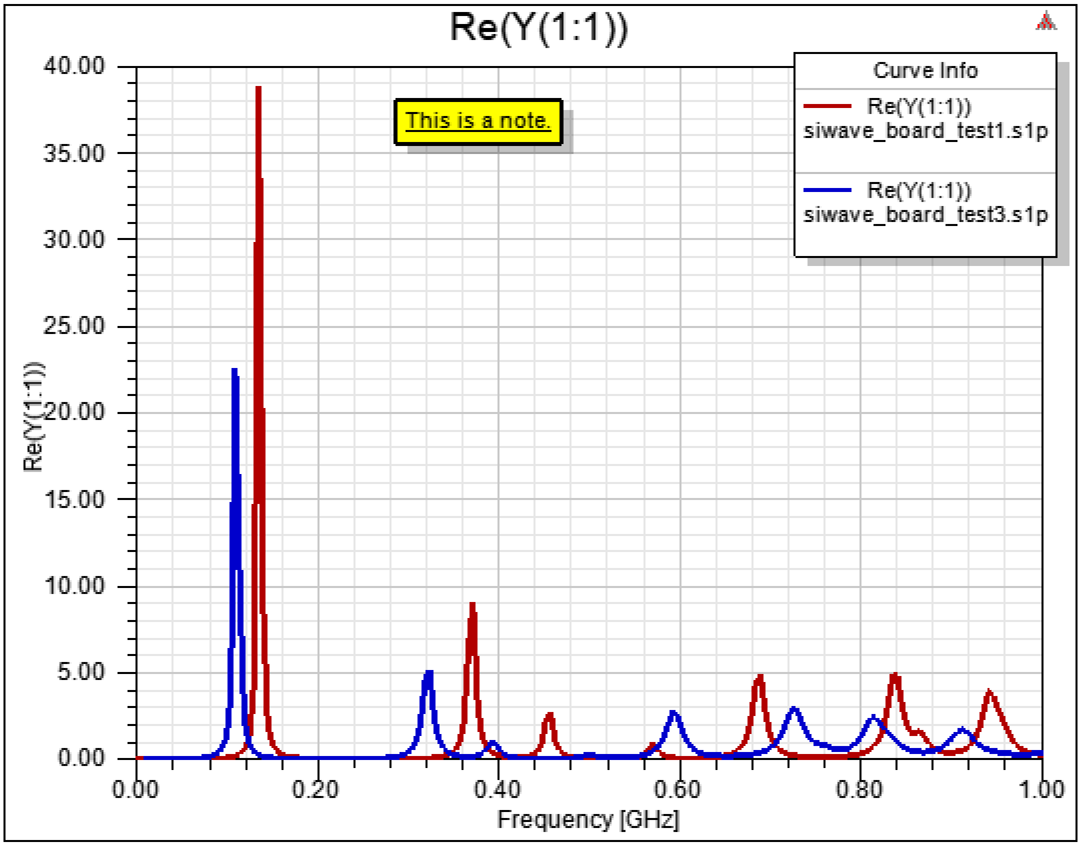
The **Marker** sub-menu provides commands for adding markers to plots.



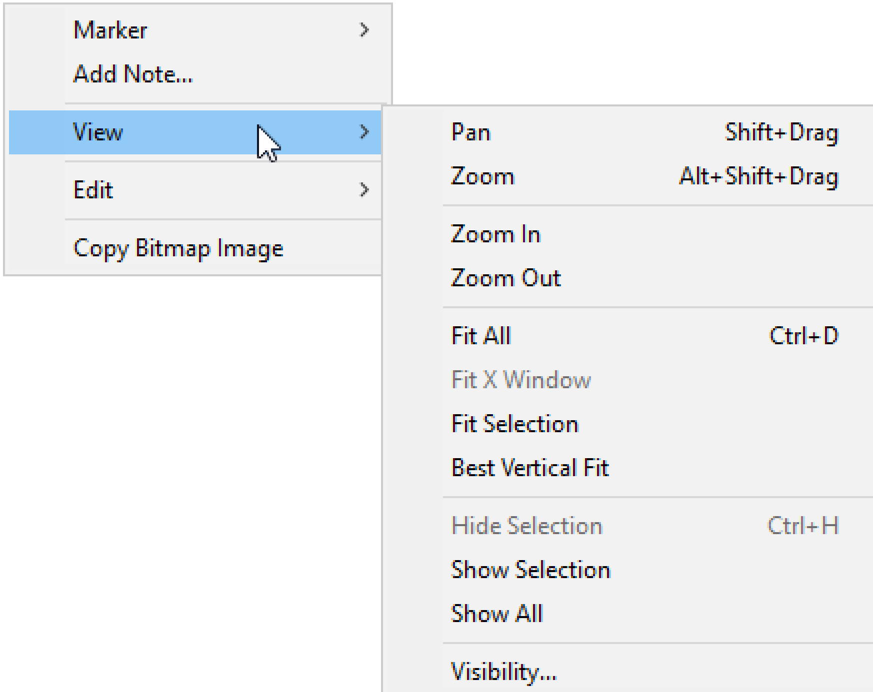
The **Add Note** menu option allows you to add a note to the plot.



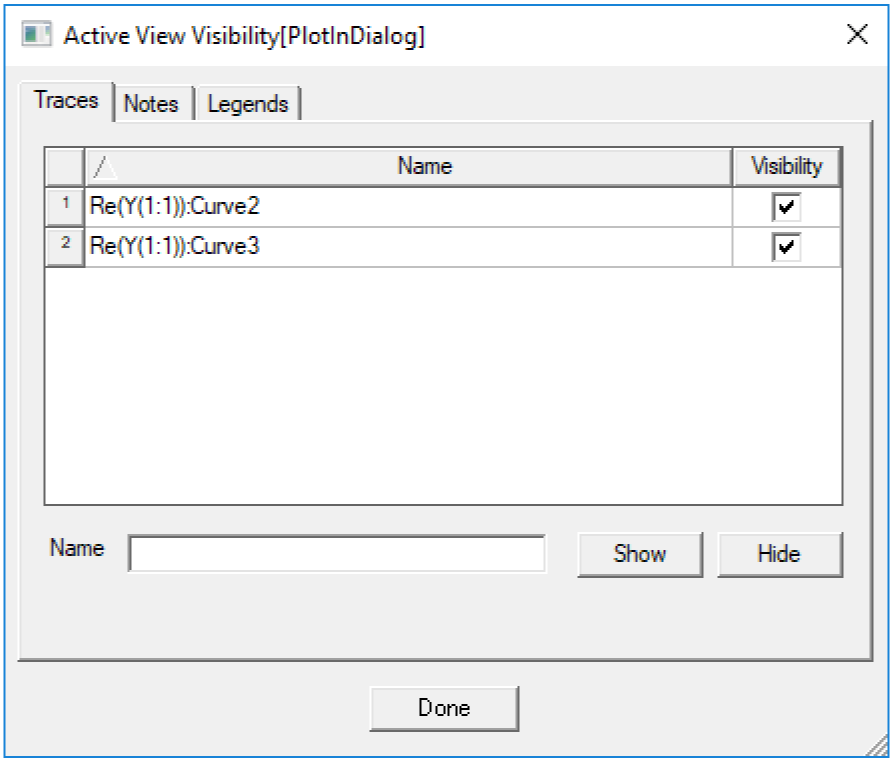
The note appears at the location you right-clicked. You can click and drag the note to a new location, or double-click the note to change its color and font.



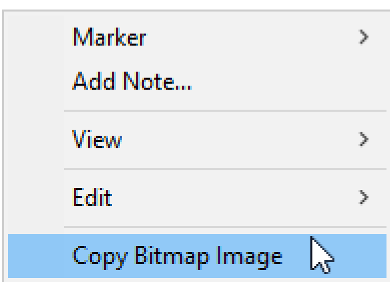
The **View** sub-menu provides commands for viewing, panning, zooming, and fitting elements to your plot.



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

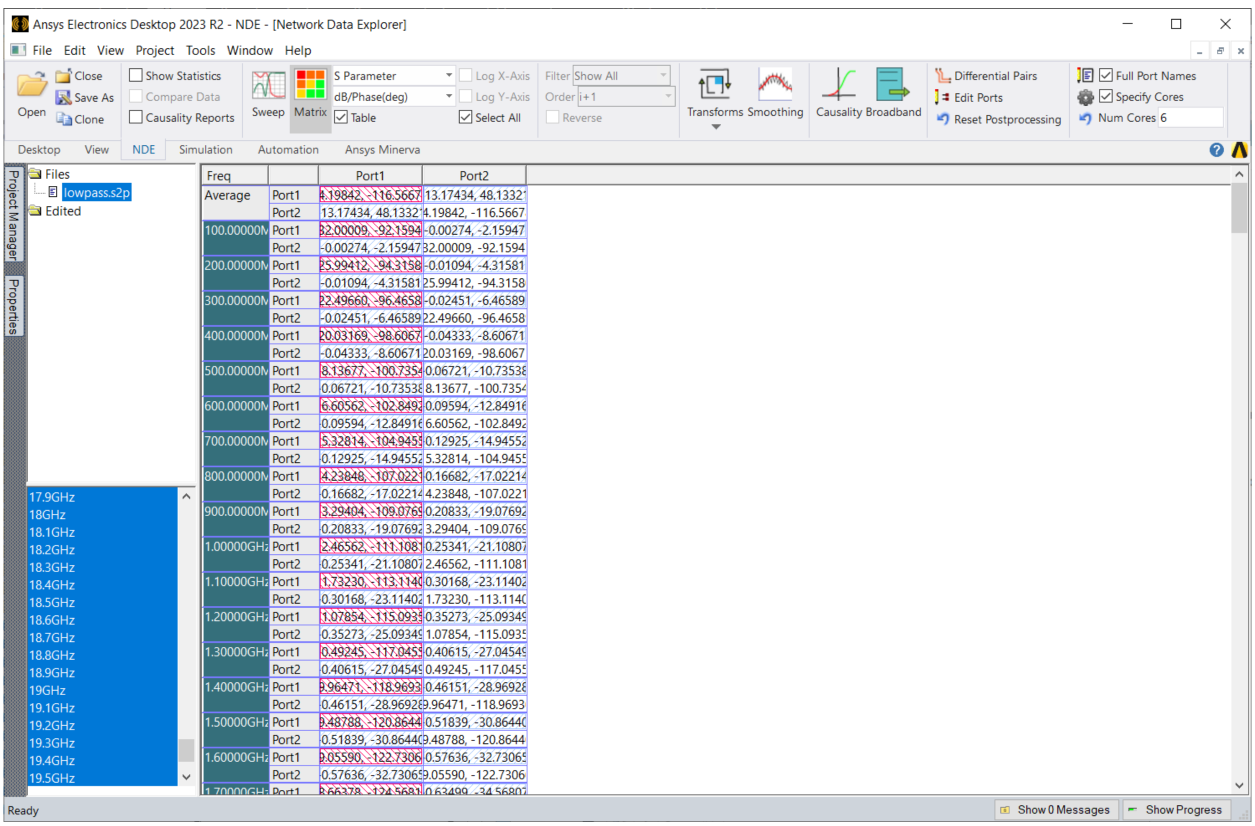
Network Data Explorer allows you to view data and modify various aspects of the display, including color plots, color coding, viewing across frequencies, and displaying individual statistics. This section provides examples of Network Data Explorer capabilities:

- Viewing the S, Y, or Z Matrix for a Frequency
- Viewing a Color-coded Matrix Plot
- Displaying a Cell Graph Across All Frequencies

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.

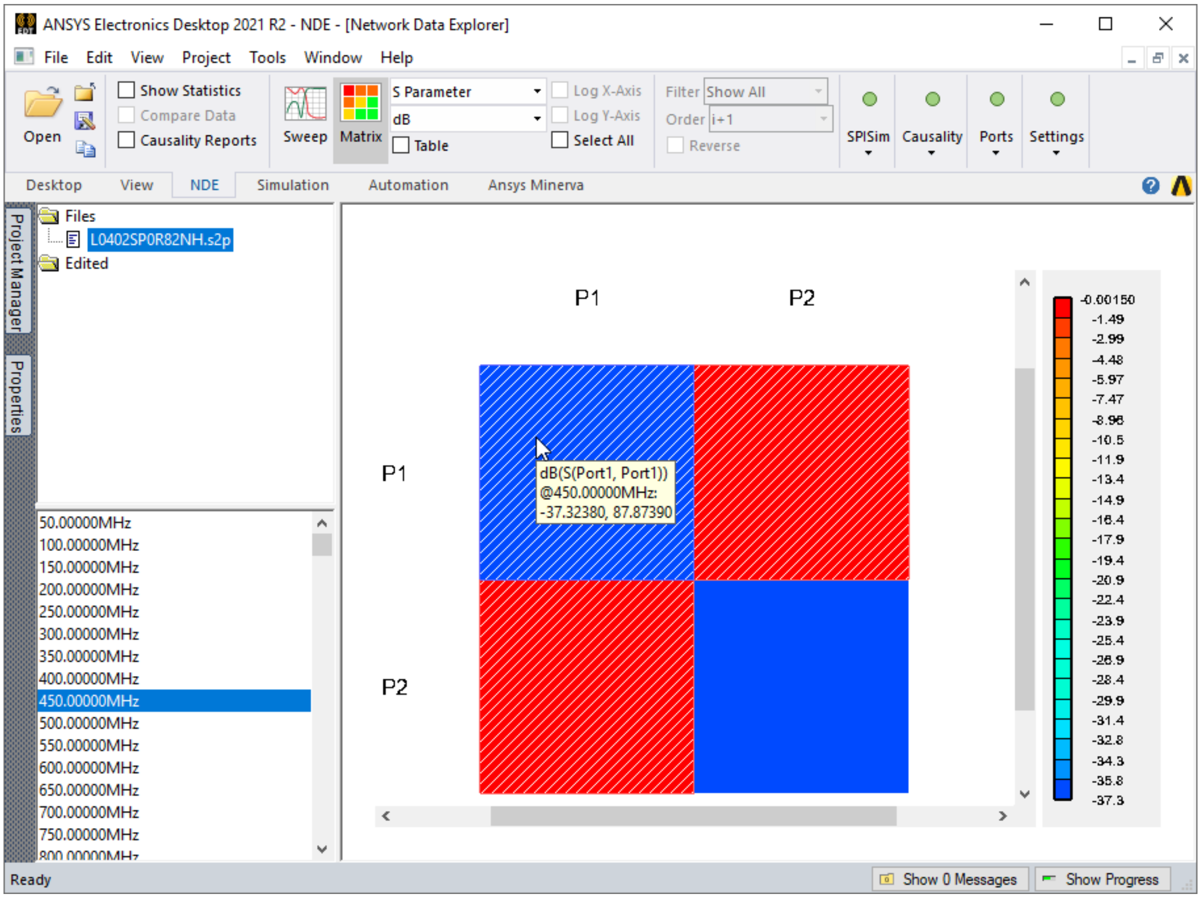
Selected cells appear in solid blue. If Select Transpose is enabled, the transpose is highlighted in blue as well.

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.

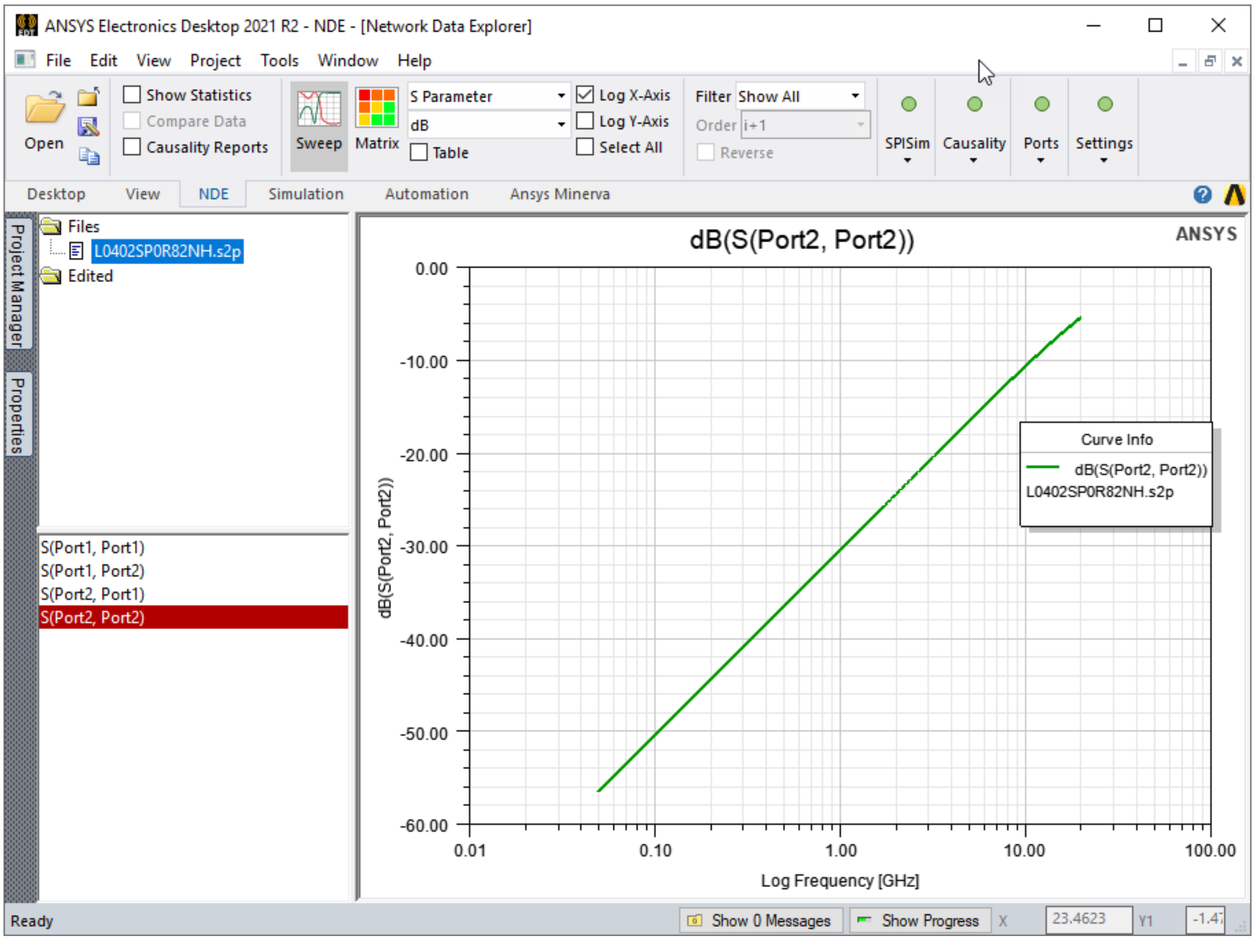
Click any cell to select it. Selected cells appear with a pink outline. If Select Transpose is enabled, the transpose is selected in pink as well.

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.



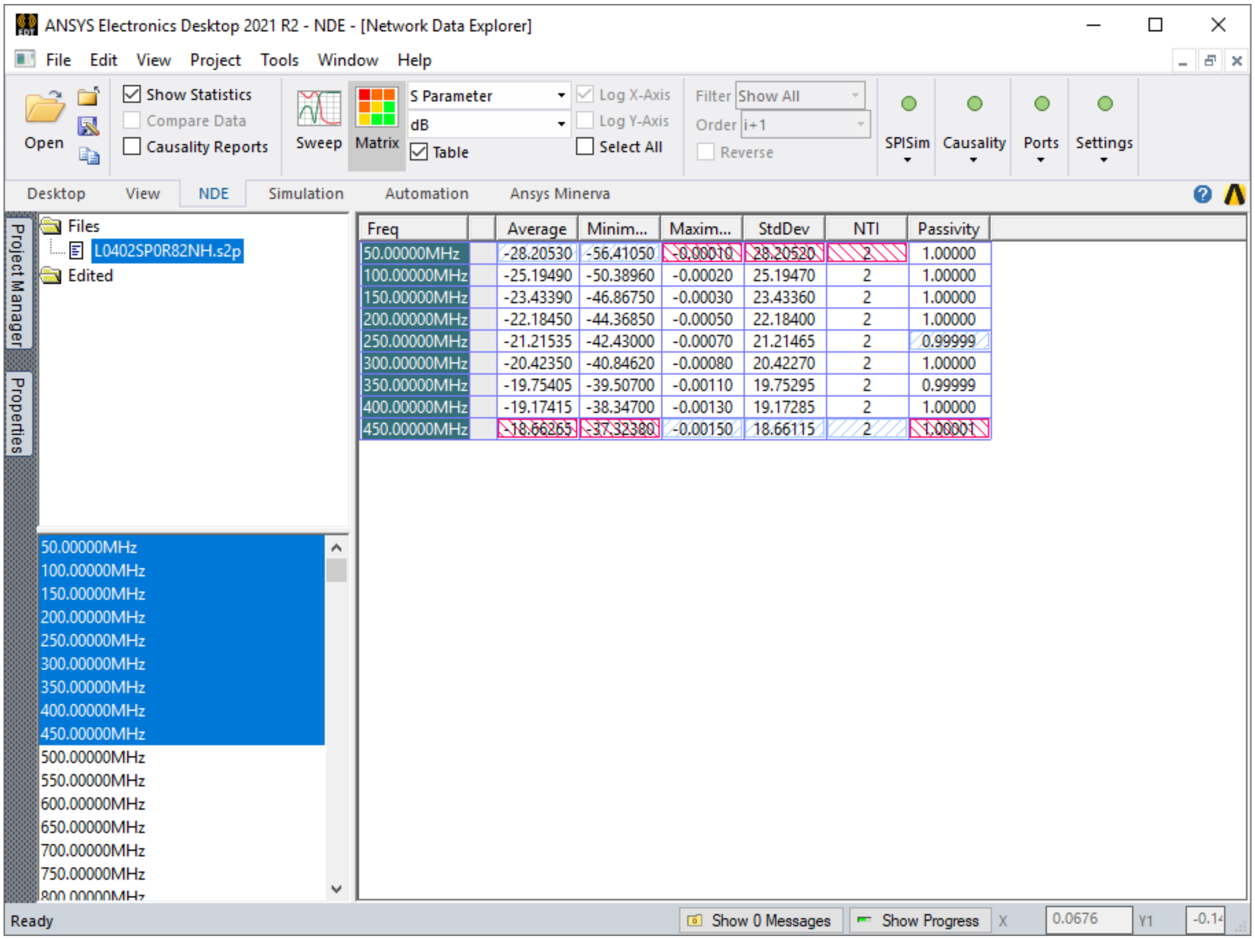
5. To add a log scale to the Y-axis, click **Log Y-Axis** on the **NDE** ribbon.
6. Right click to see a right click menu with other commands.

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.

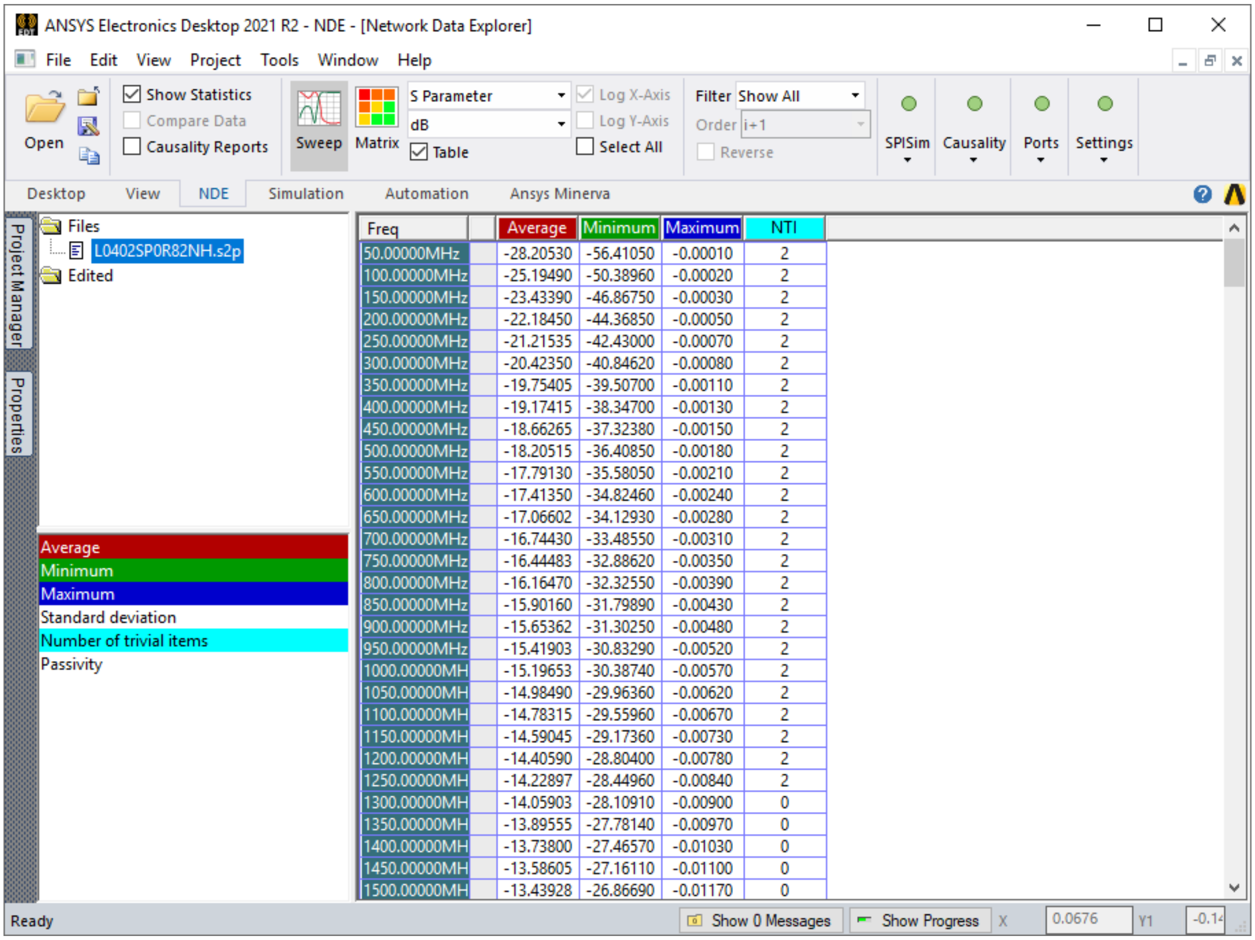
Hover the cursor over a cell to view information about it.

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).

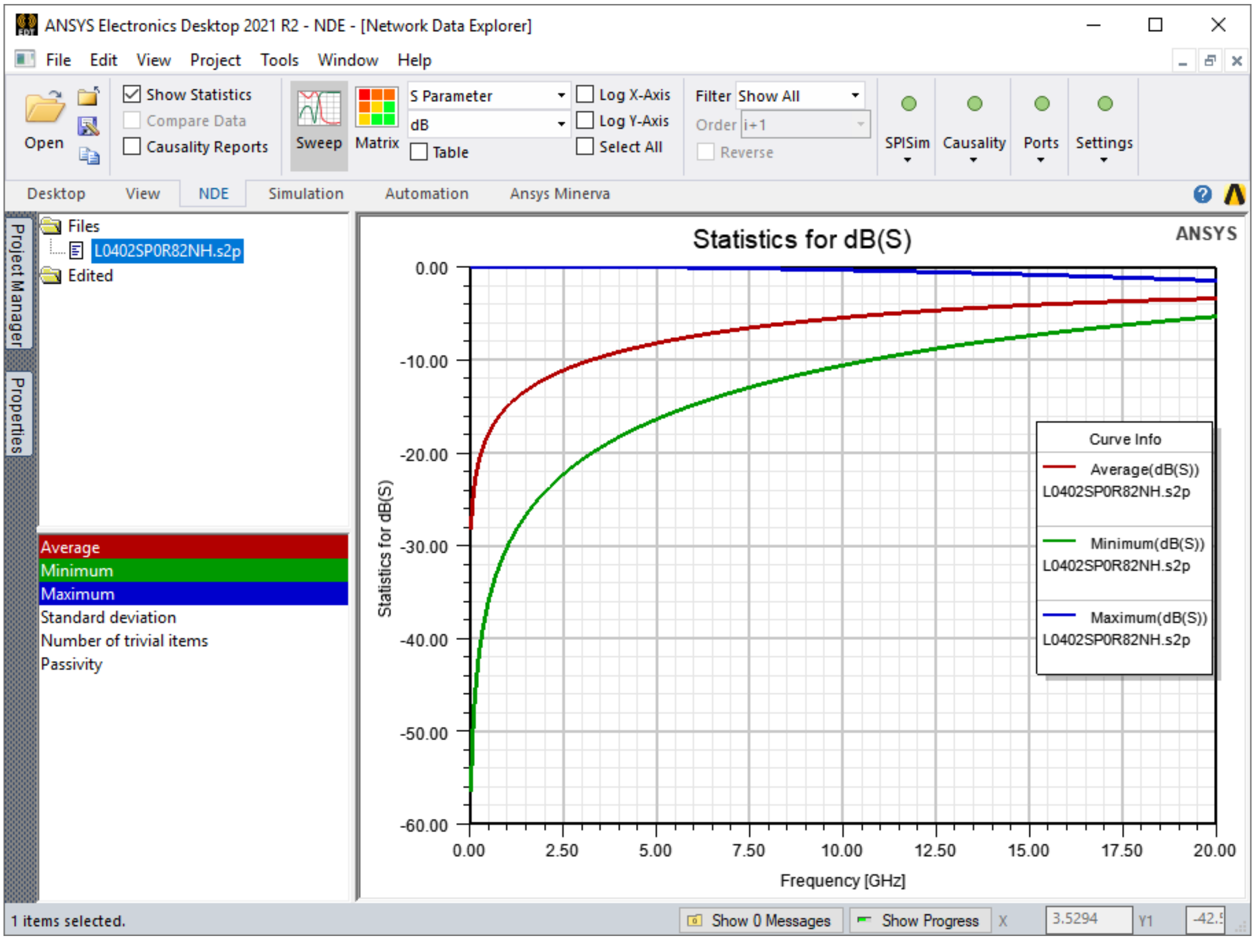
For S-parameters, the **Number of Trivial Items (NTI)** includes all values of 0 and 1. For other data and data comparisons, only values of 0 are counted as trivial.

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.

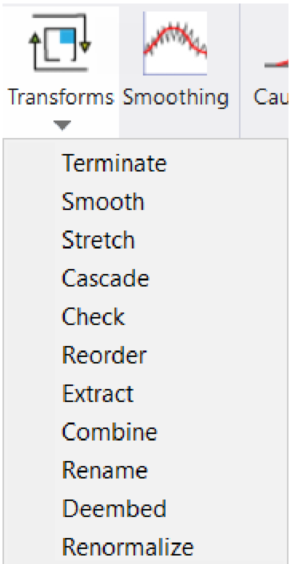
SPISim Transforms in Network Data Explorer

Network Data Explorer offers several transform functions to process s-parameters. These transform correspond to transforms available in SPISim.

NDE transforms are available for Microsoft Windows. Linux users can find similar functions in SPISim. To open SPISim, click **Tools** > **SPISim** on the toolbar.

To use a transform:

1. Click **Transforms** on the **NDE** ribbon.



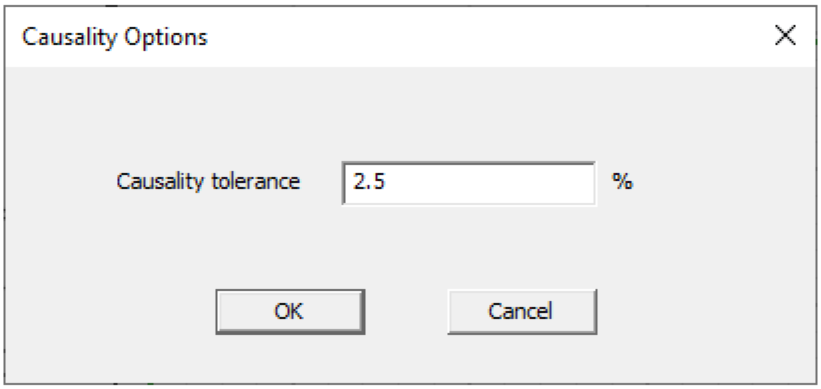
2. Choose a transform:
   - **Terminate** - opens the termination dialogue. Choose which ports to terminate and enter the impedance to terminate them with.
   - **Smooth** - see Smoothing.
   - **Stretch** - stretches given *.snp files to a specified unit length.The given .snp must be “homogeneous”, e.g. a transmission line.
   - **Cascade** - cascades different stages' *.snp files together into one *.snp files. Use this dialog to specify the correct source port ordering so that input files are properly reordered. If necessary, use **Reorder** first. Specified*.snp files can have different sampling points. Samples are synced automatically.
   - **Check** - checks s-parameter qualities, based on IEEE P370 TG3 S-Parameter quality checking algorithms.
   - **Reorder** - reorders s-parameter ports.
   - **Extract** - extracts subset of original s-parameter's ports to form sub-matrices.
   - **Combine** - combines samples from various *.snp files of the same number of ports.
   - **Rename** - renames ports. In SPISim, the original file is directly overwritten.
   - **Deembed** - removes s-parameters though a reciprocal method.
   - **Renormalization** - renormalizes port impedance to different values. The same value can be given to multiple reference Z cells at once.
3. In the top panel of the transform window, specify the input data. Selected *.snp files from NDE are provided automatically. Drag and drop to add more files
4. In the lower panel, configure the transform's settings.
5. Click **OK**.

Causality Checking and Plots

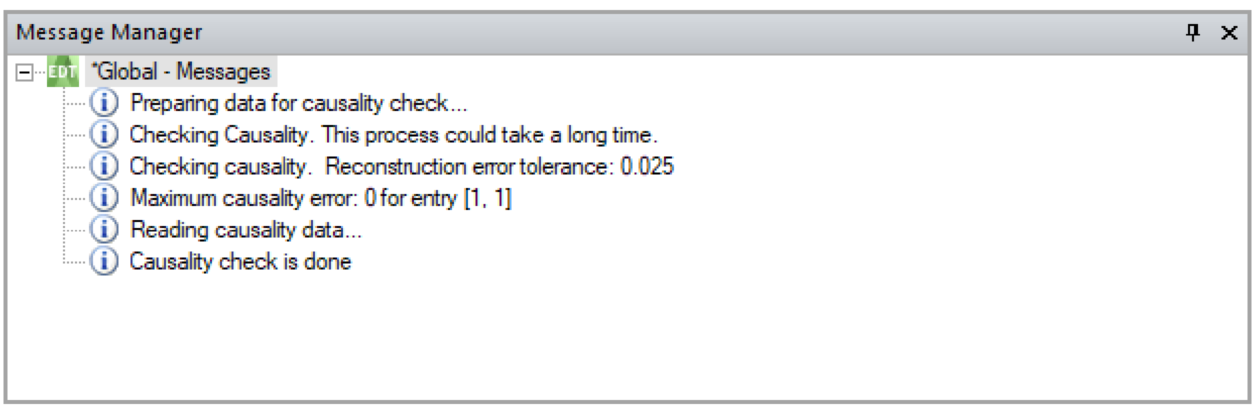
Network Data Explorer can perform a causality check on S-parameter data from any source (solution or file), and provide plots of the results in various formats.

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.

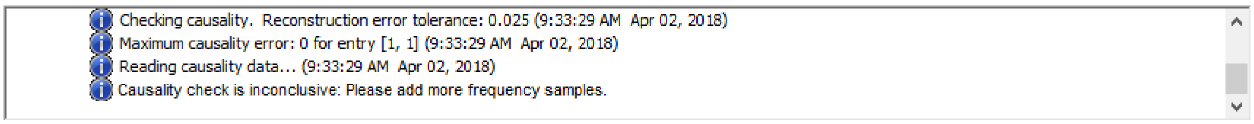


When the check completes, the Message Manager updates to display a summary of results.

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.

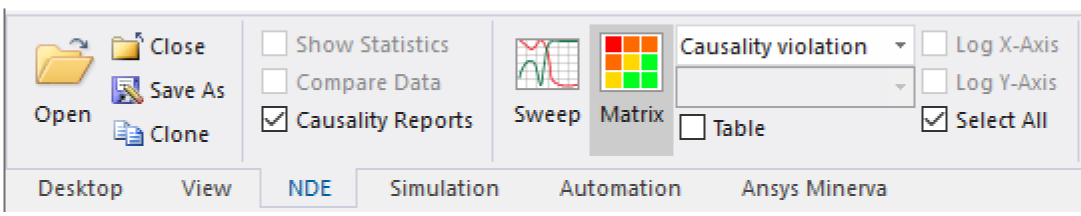
Maximum Causality Error – The maximum causality error for all port pairs and all frequencies, along with the matrix indices (port numbers) where the maximum noncausality occurs. A noncausal response is one where all matrix entries can be conclusively analyzed, and at least one entry exceeds the causality 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.

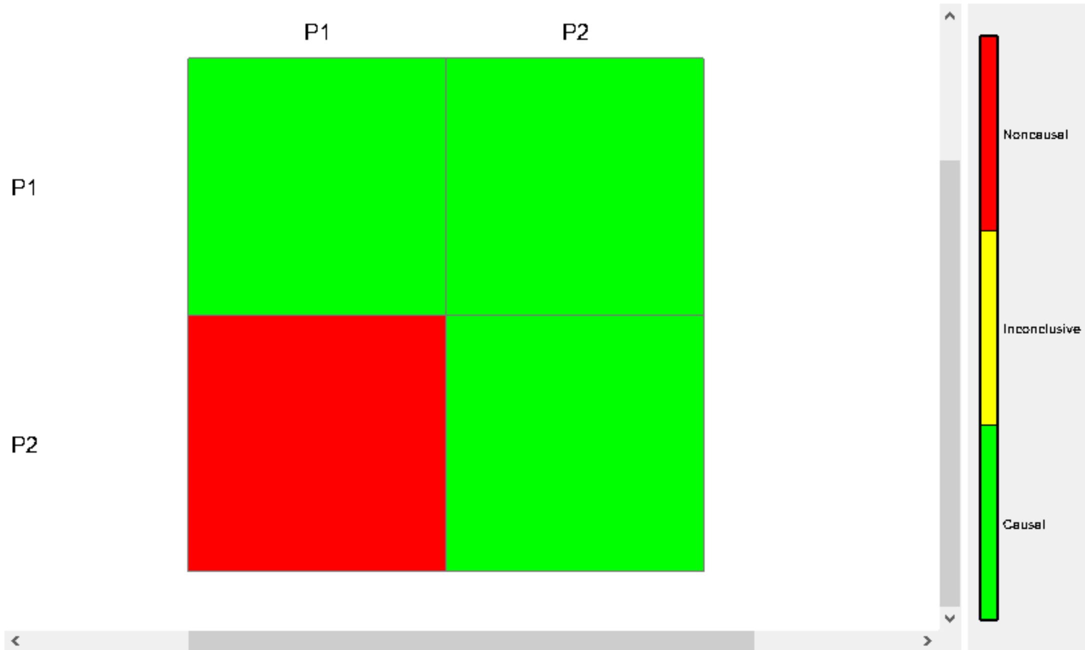


To see the plotted results of the causality check:

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.



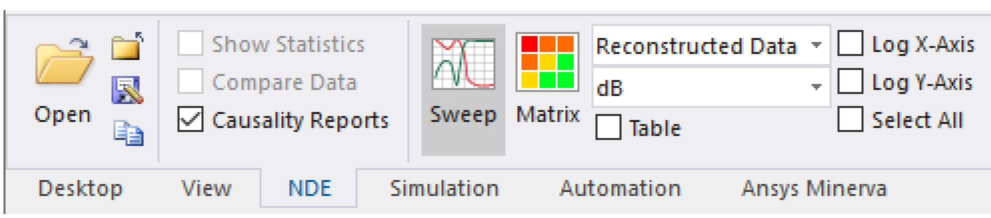
In this plot, the cells go from (Port 1, Port 1) at the upper-left area to (Port N, Port N) at the lower-right corner. The result shows the causality over all frequencies in the data. In this example, the matrix is asymmetric, so that S12 is noncausal, while S11, S21, and S22 are causal.

To see the details for each frequency, click the **Table** check button on the **NDE** ribbon.

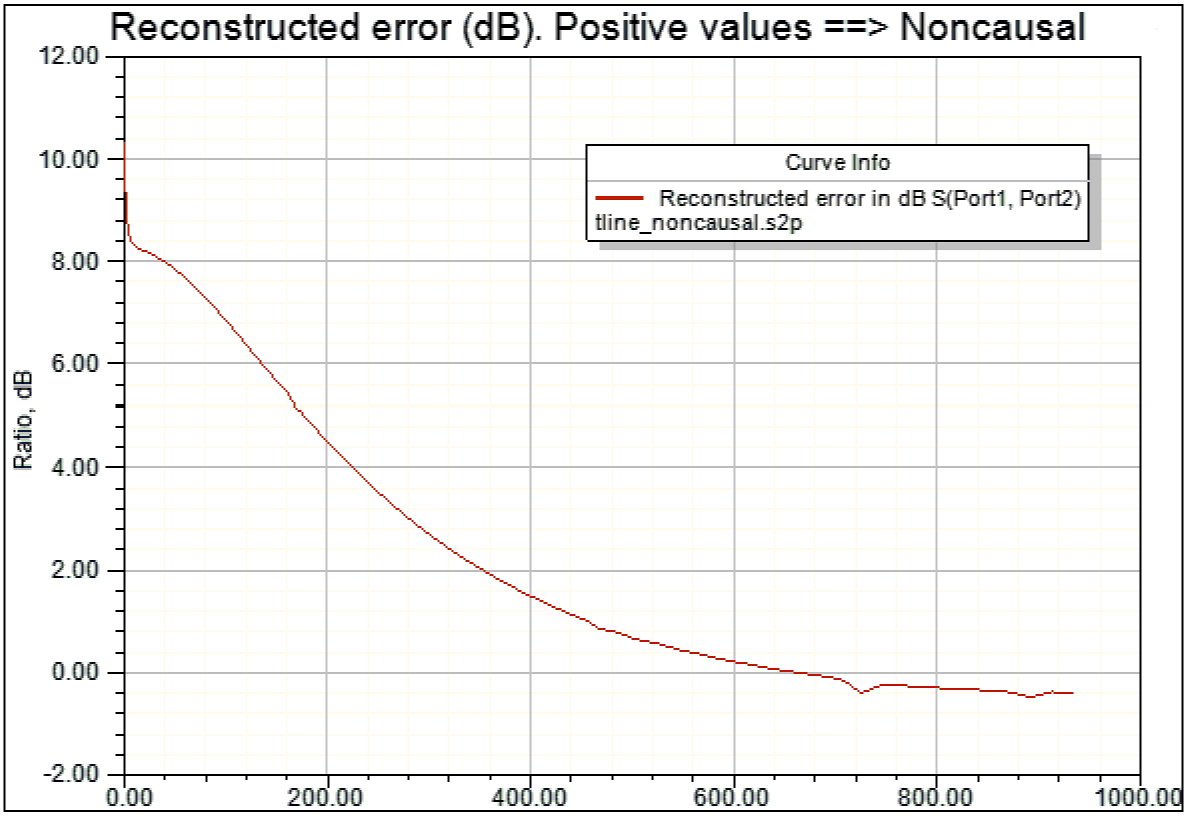
Freq		Port1	Port2	Port1	Port2
0.10000GHz	Port1	0.01715, 0.00000	0.00050, 0.00000	0.01493, 0.00000	0.00098, 0.00000
	Port2	0.53111, 0.00000	0.00579, 0.00000	0.11471, 0.00000	0.00832, 0.00000
0.20000GHz	Port1	0.01367, 0.00000	0.00059, 0.00000	0.00736, 0.00000	0.00052, 0.00000
	Port2	0.27237, 0.00000	0.00394, 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.

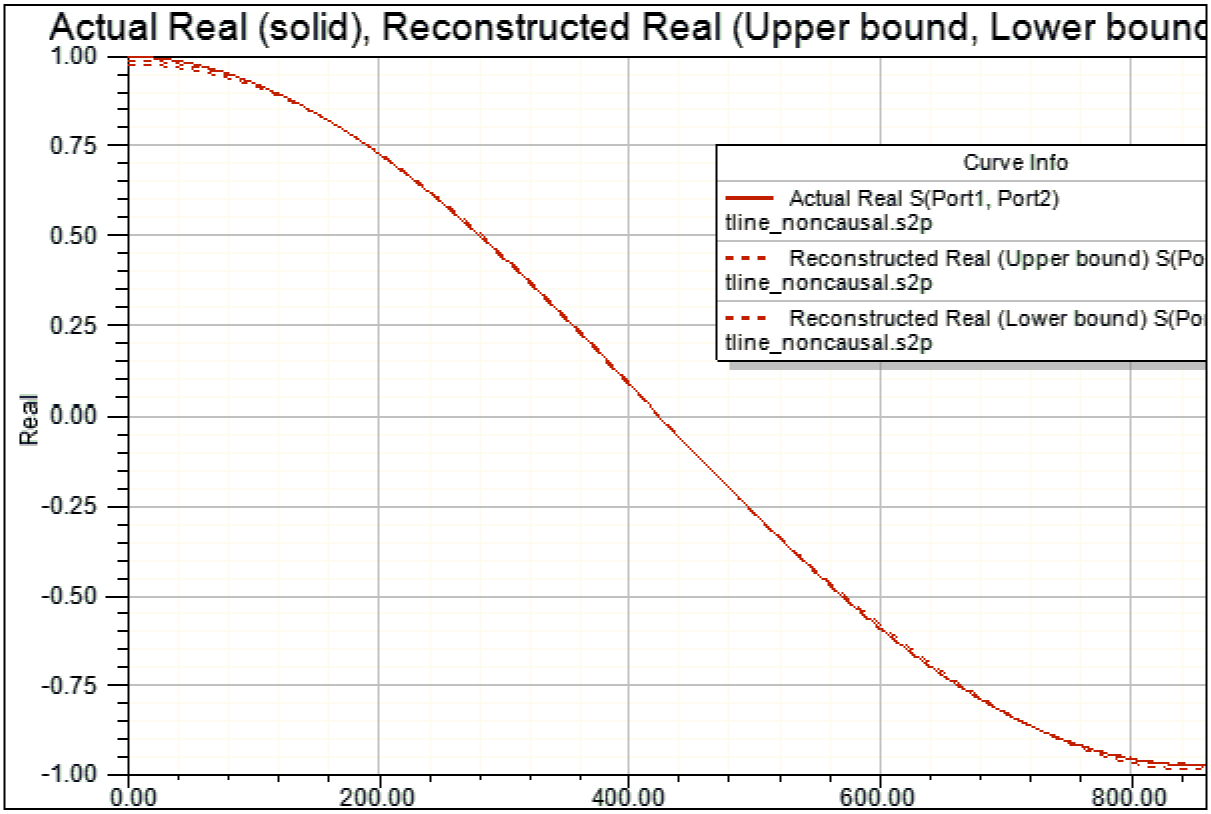


The plot appears, showing the reconstruction error at each frequency divided by the tolerance.



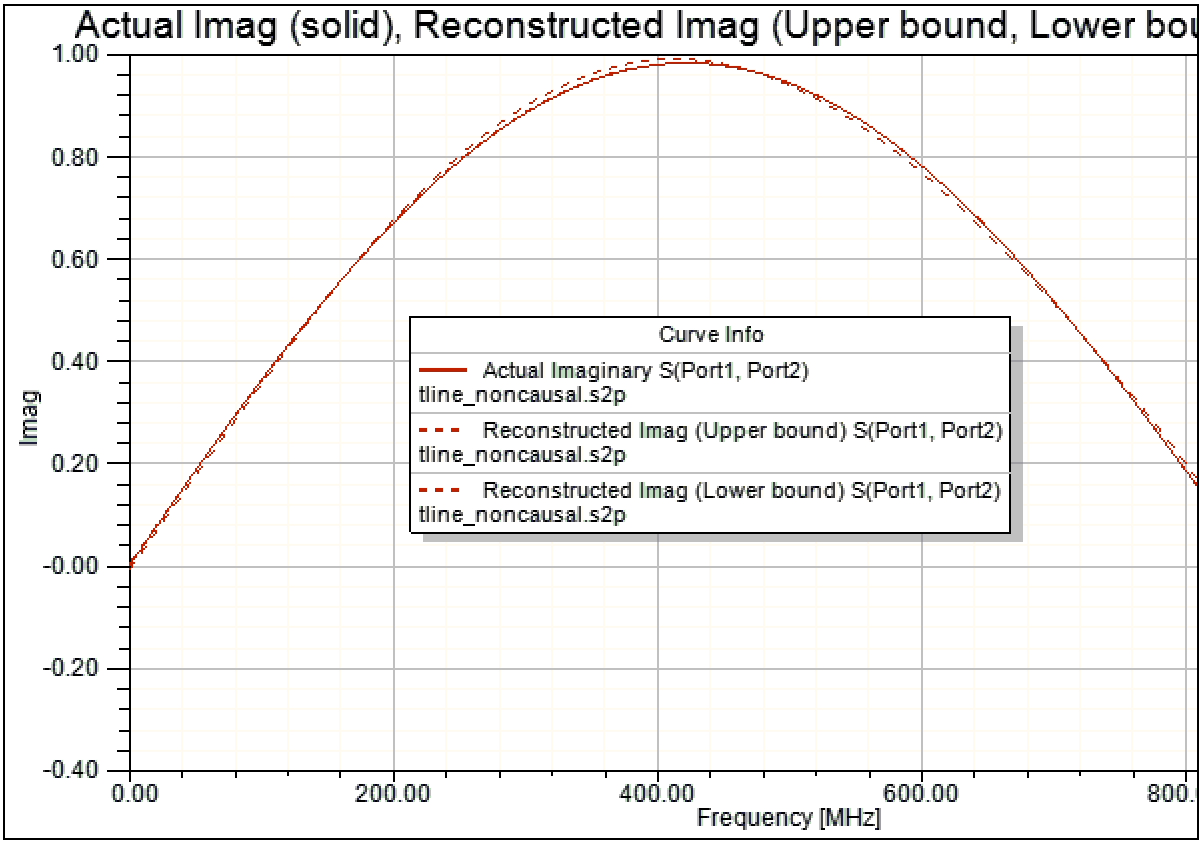
The reconstruction error ratio for parameter S12 is positive for frequencies less than about 680MHz, indicating a broad range of noncausal behavior.

4. To compare the real part of the reconstructed data to the real part of the actual data, set the **Format** to **Real**.



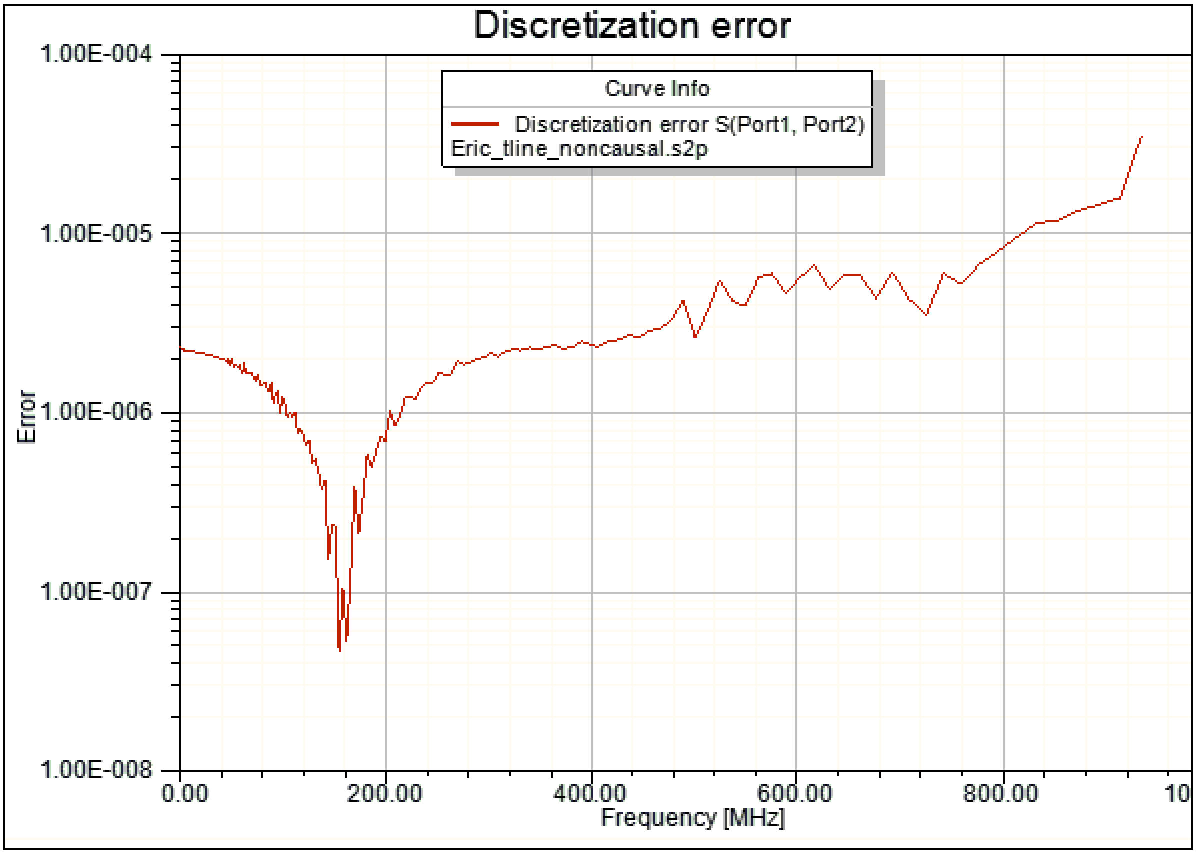
For a causal frequency response, the actual data (solid line) will be within the upper and lower bounds of the reconstruction (dotted lines) at all frequencies.

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**.



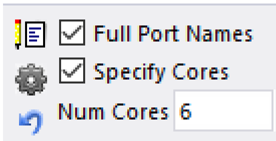
The discretization error is the error that is due to the fact that the data are available only at discrete frequencies rather than for a continuous spectrum. A discretization error near or greater than the causality tolerance renders the causality check inconclusive. Data at more frequencies could reduce 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.

20 - Nexxim Design Examples

Nexxim is a simulator for high-speed, high-density circuit designs. Nexxim provides analyses in the time and frequency domains for all levels of projects; on the characterizing of individual transistor models to the analysis of large-scale integrated devices and communication channels. The following topics provide examples of circuit analysis using Nexxim.

This topic assumes that you are already familiar with **Electronics Desktop** and Nexxim basics discussed in the online Getting Started Guides. Work through the sequence of design topics described in this topic to learn to:

- Load schematic example designs into **Electronics Desktop** (**Electronics Desktop**) for analysis using Nexxim.
- Set up and run the applicable analyses.
- Display simulation results using the **Electronics Desktop** reporting functions.

The examples are grouped by focus:

- Nexxim RF
- Nexxim SI

Nexxim RF

Nexxim RF focuses on radio-frequency designs and provides analyses in the time and frequency domains for all levels of RF projects, from individual component models to large-scale transmitter/receiver systems. The following examples demonstrate Nexxim analyses with RF issues:

Transient Analysis with Sweeps

Transient Analysis Method Option

Harmonic Balance Analysis Example

Oscillator and Phase Noise Analyses

TV Noise and PXF Analyses

Loadpull Analysis Example

Envelope Analysis Example

Time Domain Reflectometer Example

Nonlinear Loadpull Amplifier Example

X-Parameter Example

Nexsys Timestep Example

Nexsys MPSK Example

System Frequency Domain Analysis: Receiver Circuit

System Frequency Domain Analysis: Mixer Data

DC-IV Characteristics Transistor Example

Nexxim SI

Nexxim SI provides powerful tools for analyzing Signal Integrity problems, and also provides analyses in the time domain for analyzing channel responses and displaying eye diagrams and other useful reports. The following examples demonstrate Nexxim analyses with SI issues:

Eye Analysis Example

Transmit Jitter Demonstration

CTLE Gain with USB3.0 Parameters

CTLE Gain with USB3.0 Parameters from a File

CTLE Gain with PCIe3.0 Parameters

CTLE Gain with PCIe3.0 Parameters from a File

CTLE using a Generic Rational Transfer Function

PCIe3.0 Crosstalk Example

AMI Analysis Example

"HDMI Filter Demonstration " on page 20-232

External Step Response Example

Transient Parallel Bus Speedup Demonstration

Time Domain Reflectometer Example

Transient Analysis with Sweeps

In this example you runs transient analysis of a simple mixer circuit:

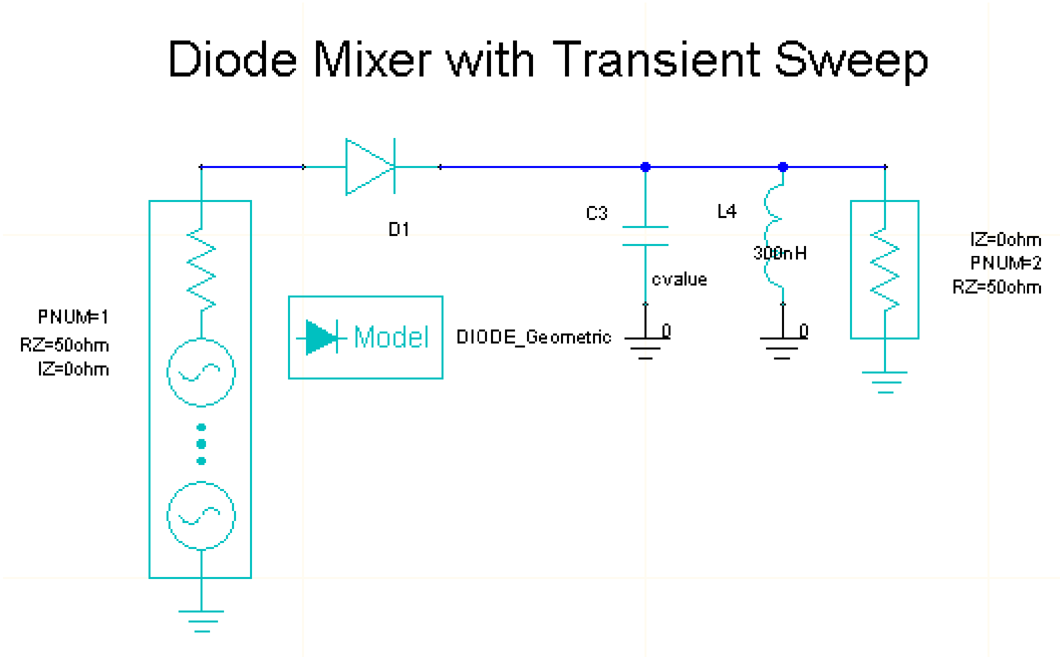
- Open a parameterized schematic circuit on the Examples directory.
- Run a transient analysis simulation at a single value of the variable, and view the results.
- Run transient analysis over several sweeps of the variable, and view the results.

Open the Example Diode Mixer Project

Open the diode mixer project on the directory of Nexxim example projects.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Mixers** directory, then select the file **SimpleMixerwithSweep.aedt**. Click **Open**.

The **Schematic Editor** window displays the diode mixer circuit.

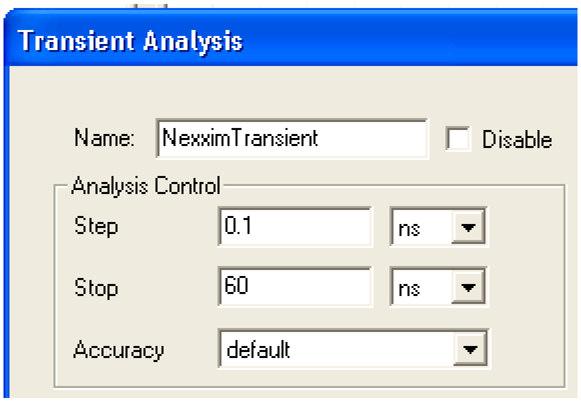


3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

Perform a Transient Analysis of the Mixer

First, let us perform a transient analysis of this netlist to show the time-domain behavior of the mixer. Later, run a sweep of a circuit parameter.

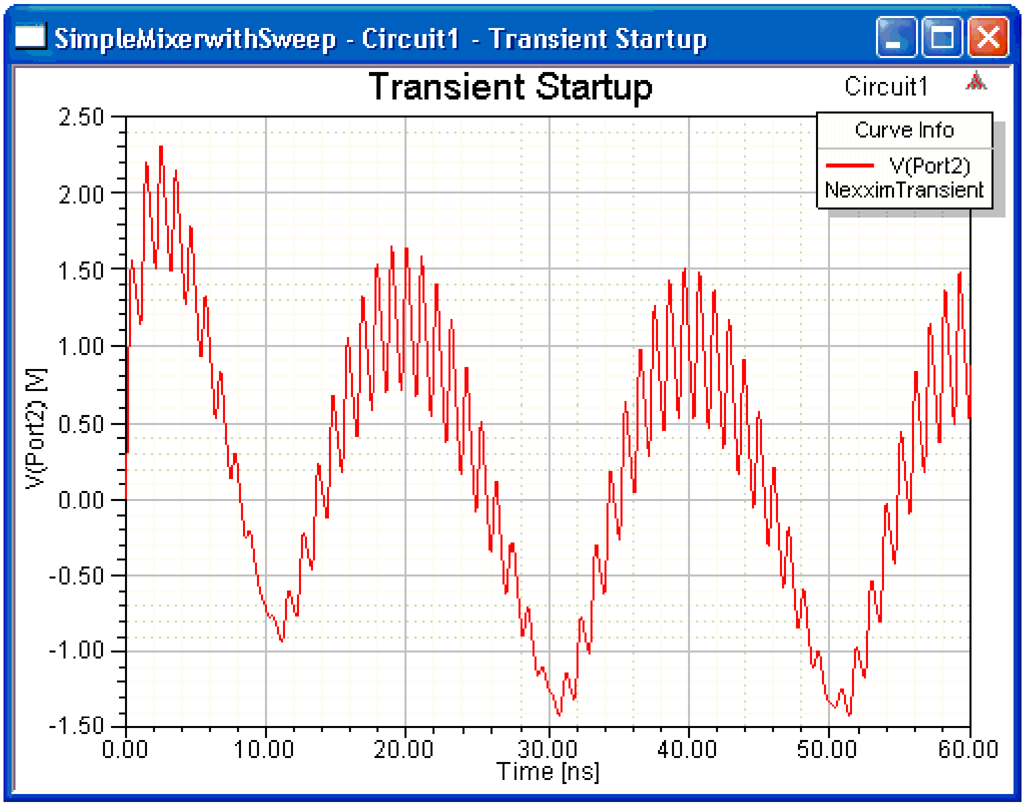
1. Expand the Analysis icon and double-click the **Nexxim Transient** setup:



This setup specifies an analysis from time 0 to 60 ns with a maximum timestep of 0.1 ns.

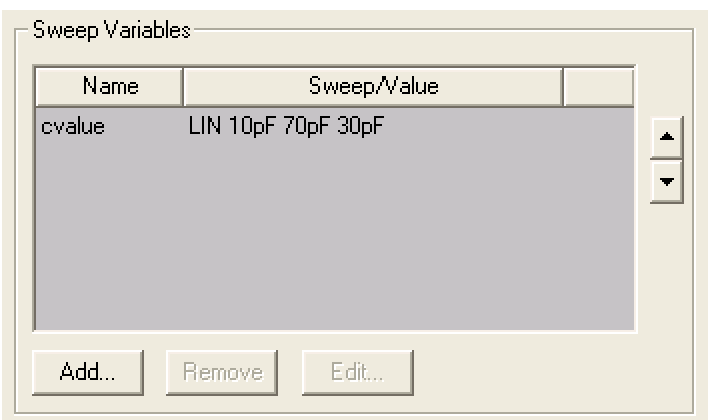
2. To run the transient analysis, right-click the **Nexxim Transient** and select **Analyze**. A progress window briefly appears at the lower-right to indicate that the transient analysis is being performed. When the analysis is completed, the Message Manager window at the lower-left shows the name of the project, errors or warnings if any, and analysis statistics such as CPU time.
3. To view a graph of the results, click the Results icon and select the **Transient Startup** report.

The **Reports** window opens with the graph of the transient analysis:



Perform a Transient Analysis with Parameter Sweep

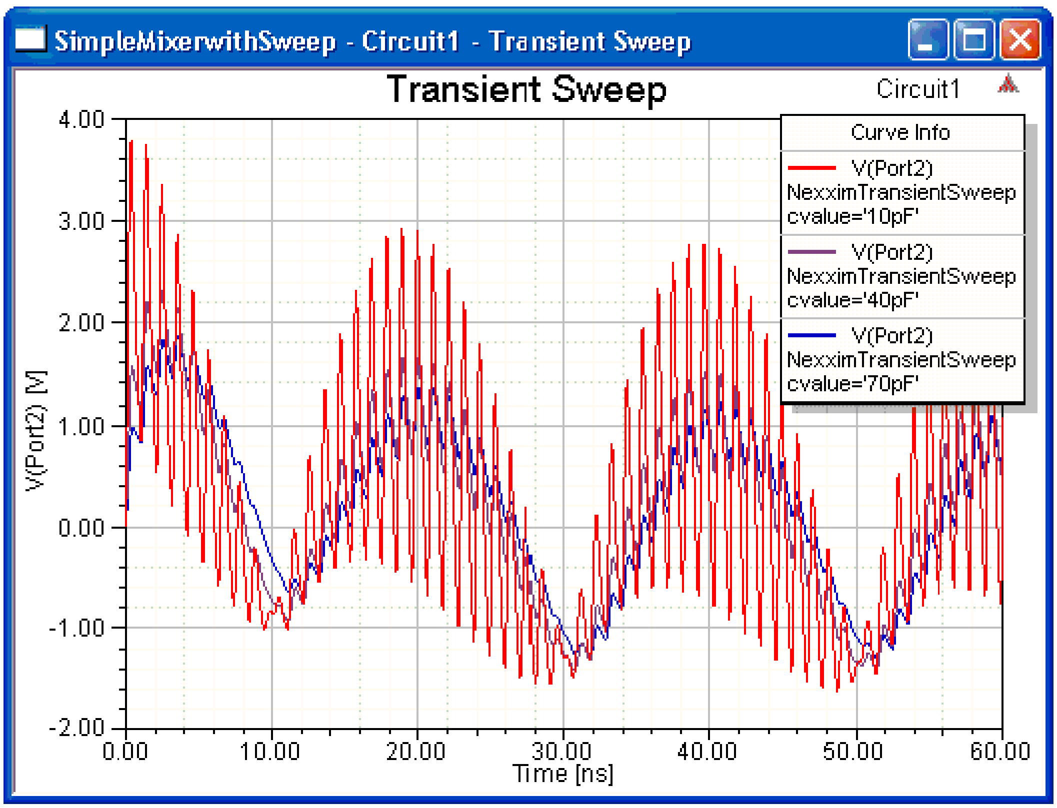
1. Click the Capacitor and verify that **cvalue** is the value of the **C** property.
2. Expand the Analysis icon and double-click the **Nexxim Transient Sweep** setup. Verify that the setup contains a sweep of **cvalue**:



This setup specifies a linear sweep of **cvalue** from 10pF to 70pF with a step of 30pF. The swept values is 10pF, 40pF, and 70pF.

3. To run the transient sweep, right-click the **Nexxim Transient Sweep** and select **Analyze**. A progress window opens at the lower-right to showeach sweep as transient analysis is being performed. When the analysis is completed, the Message Manager window at the lower-left shows the name of the project, errors or warnings if any, and analysis statistics such as CPU time.
4. To view a graph of the results, click the Results icon and select the **Transient Sweep** report.

The **Reports** window opens with the graph of the transient analysis with all sweep values:



Related Topics:

Nexxim Transient Analysis

Nexxim Variable Sweep

Transient Analysis Method Option RF

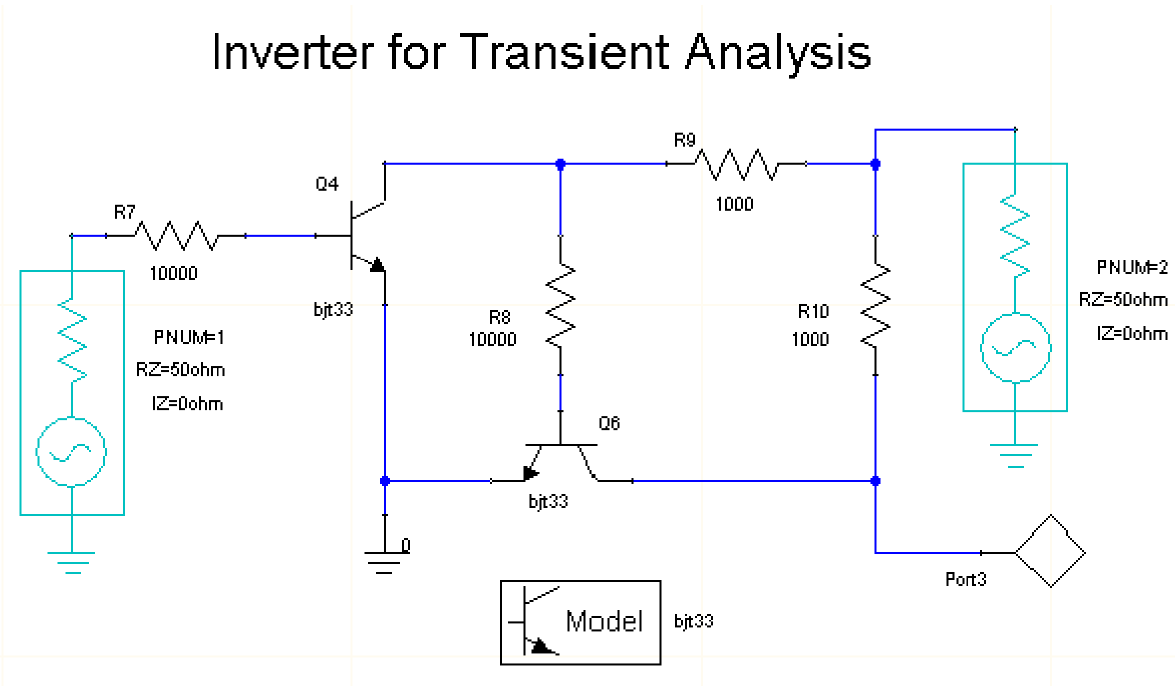
This topic describes the use of the **Method** option to select numerical methods for Transient Analysis. The example used is a BJT inverter circuit. You can open the inverter circuit schematic on the Examples directory.

Open the Example Inverter Project

Load the BJT inverter project on the directory of Nexxim example projects.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Amplifiers** directory, then select the file **BJT_Inverter.aedt**. Click **Open**.

The **Netlist Editor** window opens to open the BJT inverter circuit schematic:



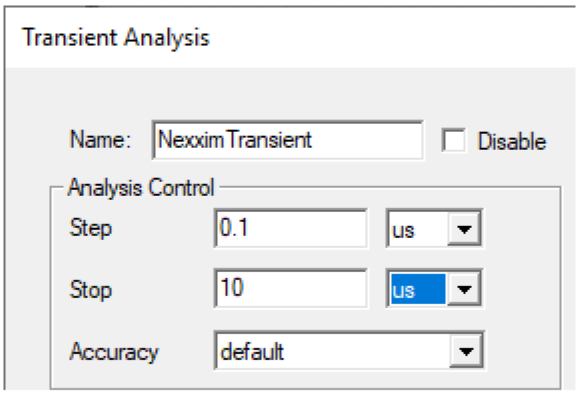
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project.
4. Click **OK**.

Perform a Transient Analysis of the Inverter

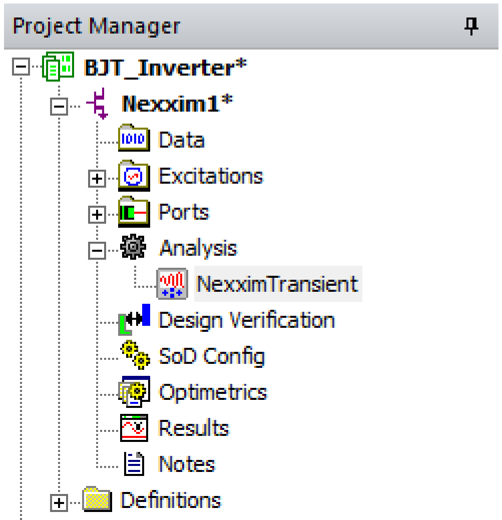
Add a solution setup before analyzing the inverter circuit.

1. From the **Circuit** menu, select **Add Nexxim Solution Setup ...** > **Transient Analysis**. A window opens to specify the parameters for the transient analysis.
2. Set the **Step** value to 0.1 microseconds, using the drop-down menu to select the unit (us). This is the maximum step size.
3. Set the **Stop** value to 10 microseconds (us), to define the length of the simulation. Since the frequency of the input sinusoid is 1 MHz, ten microseconds should contain ten cycles.

The **Analysis Control** group box should look like the following:



4. From the **Transient Analysis** window, click **OK**.
5. Click the **Projects** tab of the **Project Manager** window, and expand the project icon by clicking on the plus sign. Expand the **Nexxim1** icon, the **Analysis** icon. The newly solution setup, named **NexximTransient** if it is the first, or **NexximTransient***n* for the *n*th one you create, is now available.



6. Click that solution setup to select it, then right-click to display a drop-down. From that drop-down, click **Analyze** (or click the **Analyze** icon on the toolbar).

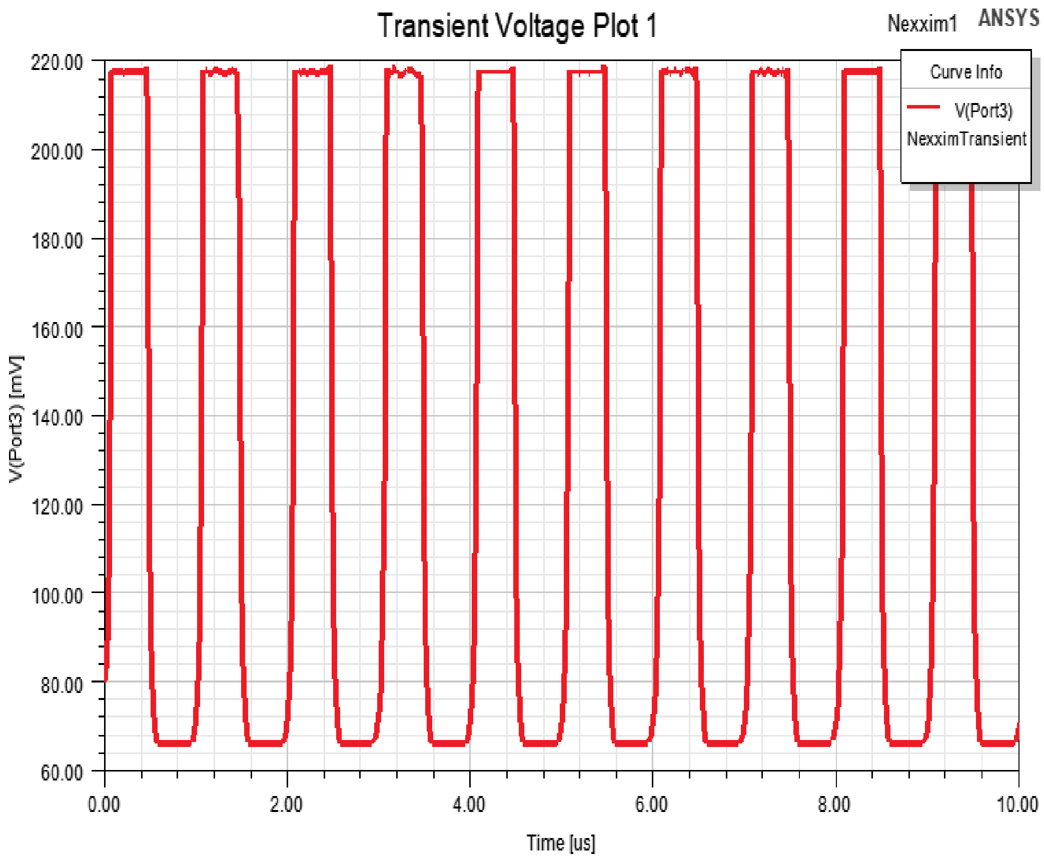
A progress window briefly appears at the lower right to indicate that the transient analysis is being performed. When the analysis is completed, the Message Manager window at the lower-left shows the name of the project, errors or warnings if any, and analysis statistics such as CPU time.

7. From the **Project Manager** window, locate the **Results** icon. Right-click the **Results** icon, then select **Create Standard Report** > **Rectangular Plot**. The **Create Report** window opens with the **Trace** tab selected.

   The X-axis displays the time values. For the Y-axis, select **Voltage** as the Category, **V (Port*n*)** as the Quantity, and **<none>** as the Function.

   > **Note:** The Port number (**Port3** in our example) and the other node names (**net_ *n***) may have different numbers in your circuit. However, there should be only one port voltage to select.

8. Click **New Report**.
9. Click **Close** on the **Report** window. The **Reports** window displays the graph of the result:

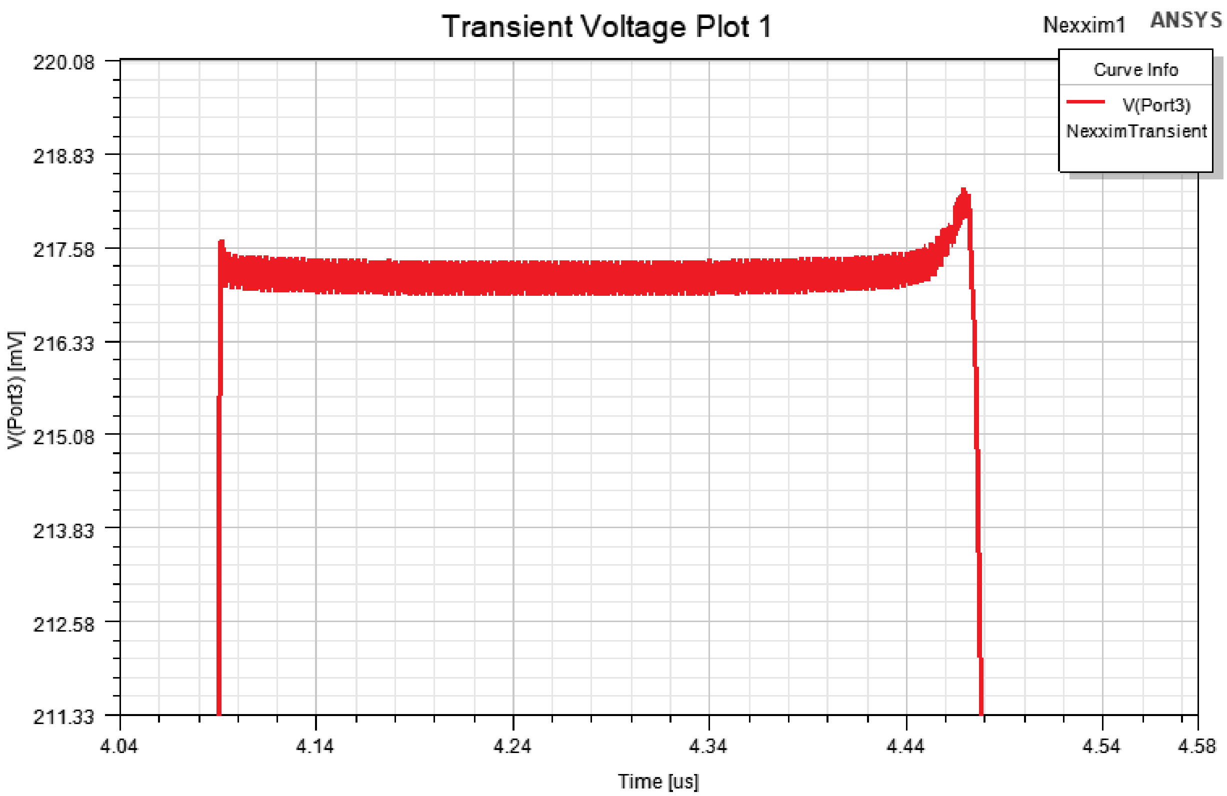


The BJT inverter has converted the sinusoidal input to a square wave output. Note the slightly ragged appearance at the peaks of the wave. Let us look more closely at one of these segments.

10. Right-click in the **Reports** window, then select **View** > **Zoom In**. Click directly above and the left of one of the wavetops, then **click+drag** to include the wavetop. Then release the mouse button. The graph changes to show the zoomed-in view.

> **Note:** To restore the full image (not zoomed in), right-click in the graph and select **Fit All**.

10. You may have to repeat this operation once or twice more until you see an image like the following:



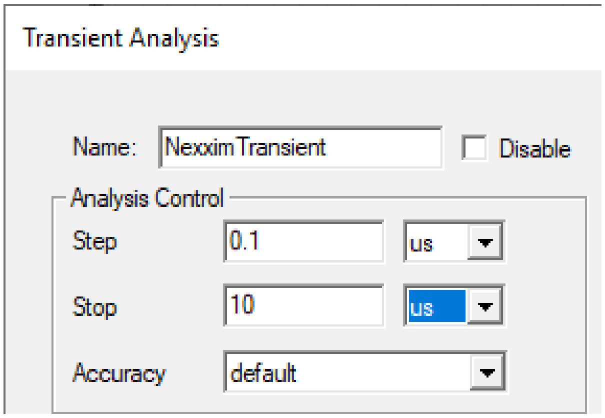
This tiny irregularity (approximately 1.0 mV) is called “trapezoidal ringing.” It is produced by the trapezoidal rule method of numerical integration, and is not the true circuit behavior. The trapezoidal rule is the Nexxim default numerical integration method. Numerical ringing commonly occurs with the trapezoidal rule method at points in a waveform with sharp corners or discontinuities such as this square wave.

12. Leave the **Reports** window open with the zoomed-in trace.

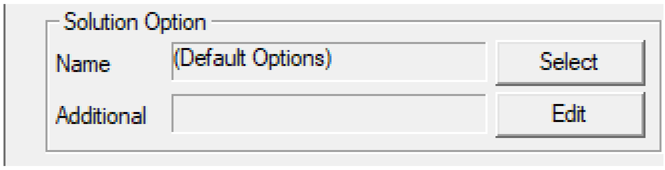
Use the NDF2 Method Option for Transient Analysis

For cases where the exact circuit behavior is important, Nexxim offers an integration method, NDF2, that avoids numerical ringing. Here is how to select the **Method=NDF2** option for Transient Analysis.

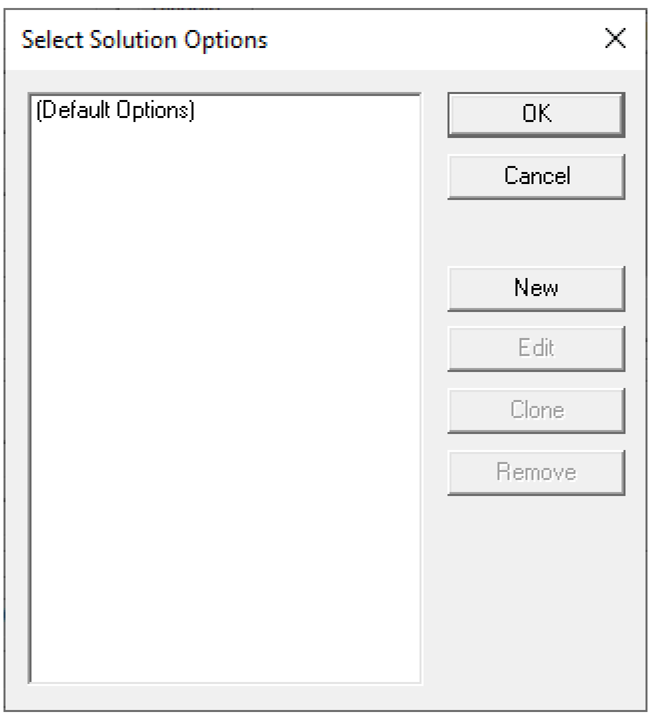
1. From the **Project Manager** window, expand the **Project Tree** and **Analysis** folder. Then double-click the Transient analysis setup to reopen the **Transient Analysis** window.
2. To perform the same transient analysis, the **Step** and **Stop** values should be set up the same as before.



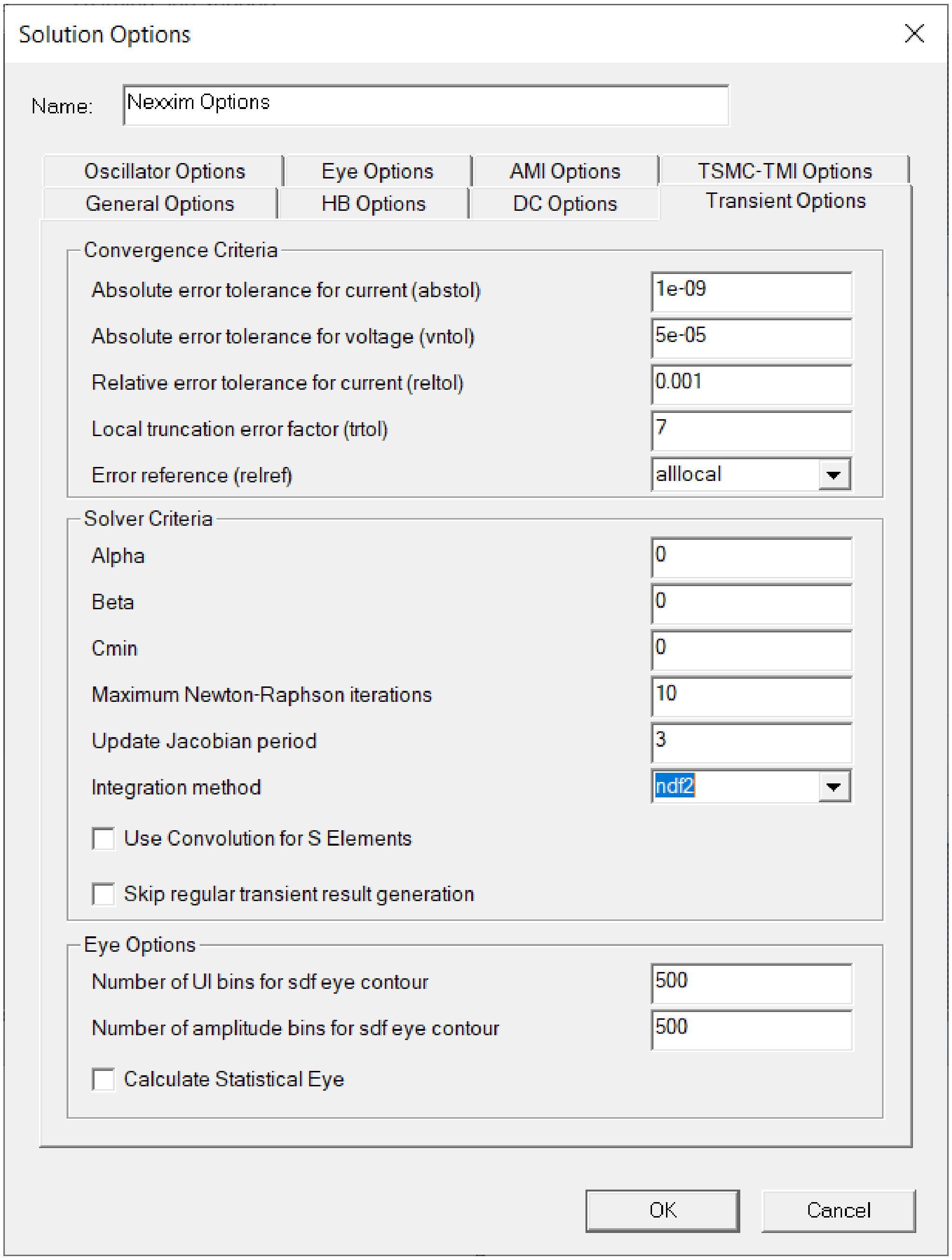
3. Click **Select** in the **Solution Options** group box:



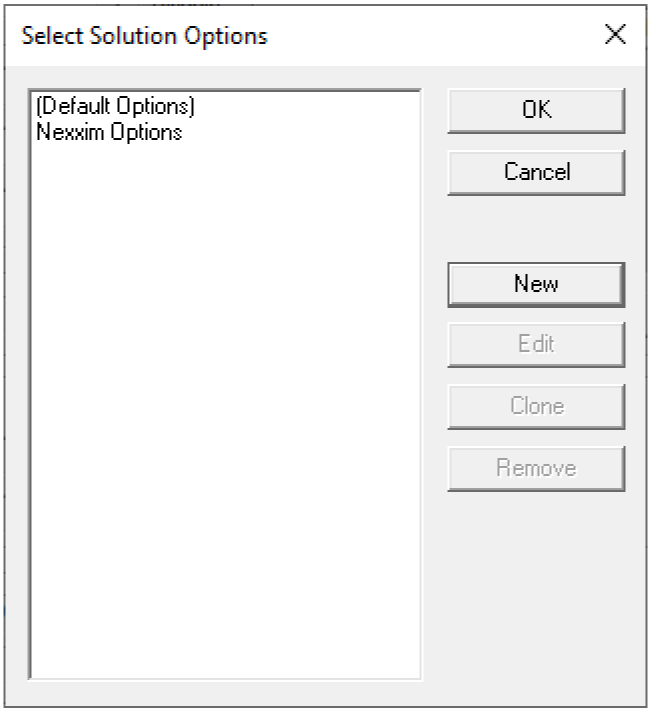
The **Select Solution Options** window opens:



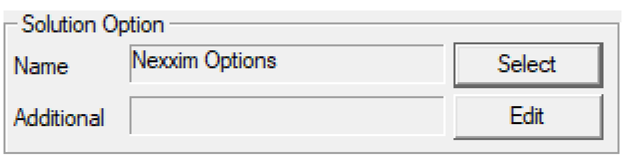
4. Click **New**. The **Solution Options** group box opens, with the **Transient Options** tab already selected.



5. The default **Integration method** is **trapezoidal**. From the **Integration method** window, select **ndf2** on the drop-down menu (for more information see Transient Analysis Options).Then click **OK** on the **Solution Options** window to reopen the **Select Solution Options** window.



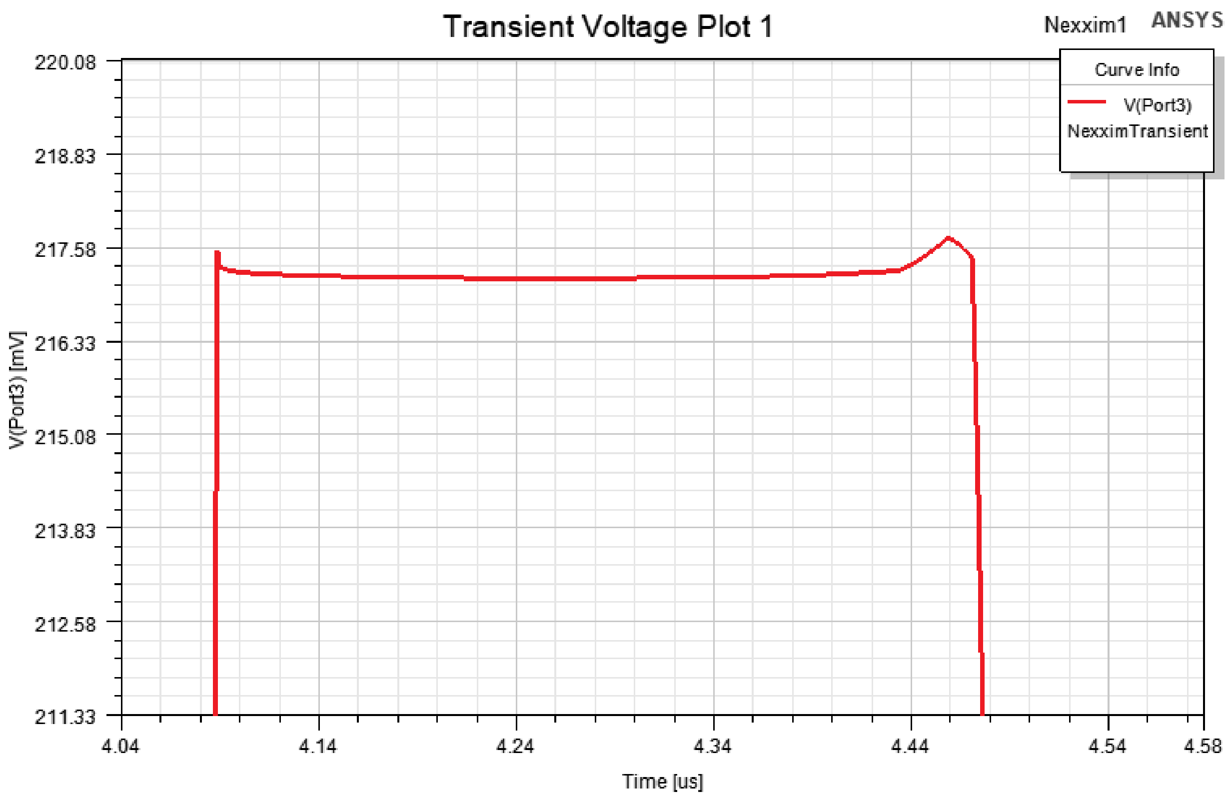
6. Click the name of the new set of options you just defined (**NexximOptions** in our example). Click **OK** to reopen the **Transient Analysis** setup window, now showing the name of the new option ("Nexxim Options" in our example) set in the **Solution Options** group box:



7. Click **OK**.
8. To run the modified transient analysis, open the **Analysis** icons, right-click the solution setup "Transient" you just created, and select **Analyze**. This runs the analysis with the options you have set up.

A progress window briefly appears at the lower right to indicate that the transient analysis is being performed. When the analysis is completed, the Message Manager window at the lower-left shows the name of the project, errors or warnings if any, and analysis statistics such as CPU time.

The transient analysis report is redrawn with the new result:



The simulation response using NDF2 does not exhibit the numerical ringing with the trapezoidal rule.

For more information, see Nexxim Transient Analysis.

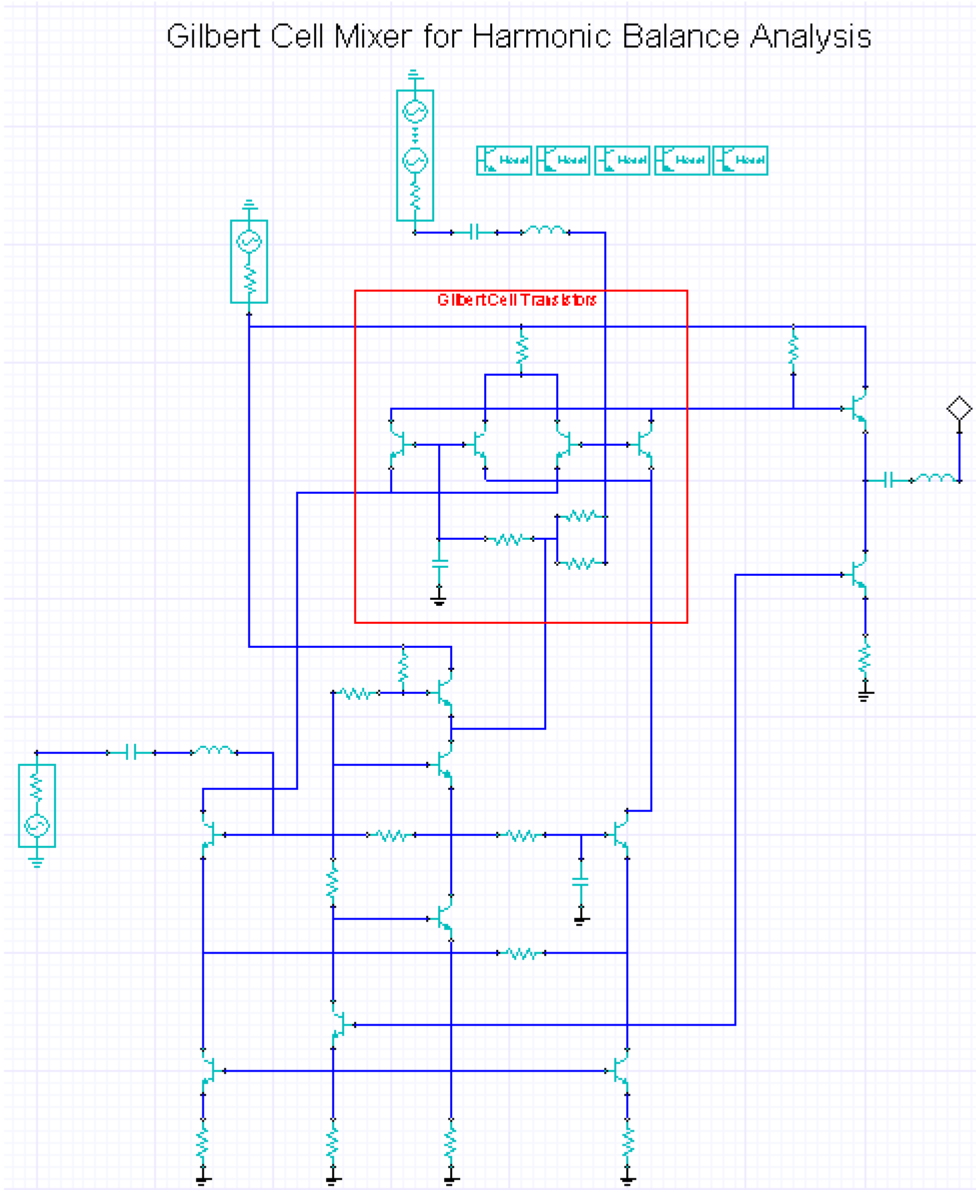
Harmonic Balance Analysis Example

The Gilbert Cell mixer is a well-known design. It is a differential-pair mixer, with low conversion loss when implemented with active nonlinear elements. This example compares Nexxim harmonic balance analysis with transient analysis. In this example, a local oscillator frequency **VLO** is the carrier for one or two RF signals, **VRF1** and **VRF2**.

Open the Gilbert Cell Mixer Schematic

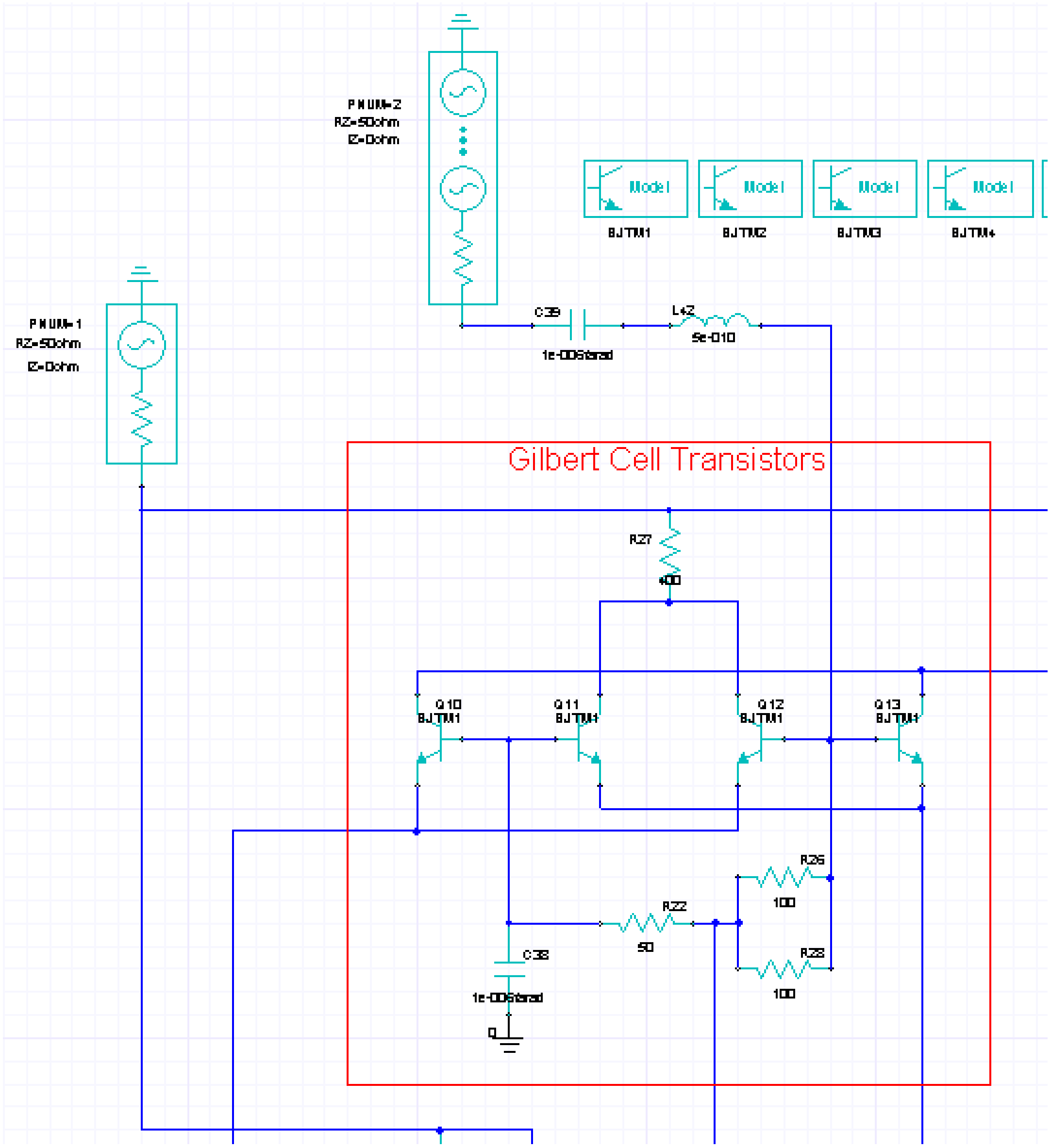
Open the Gilbert Cell Mixer project on the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Mixers** directory, then select the file **GilbertCellMixer.aedt**. Click **Open**.
3. The schematic appears in the design window.

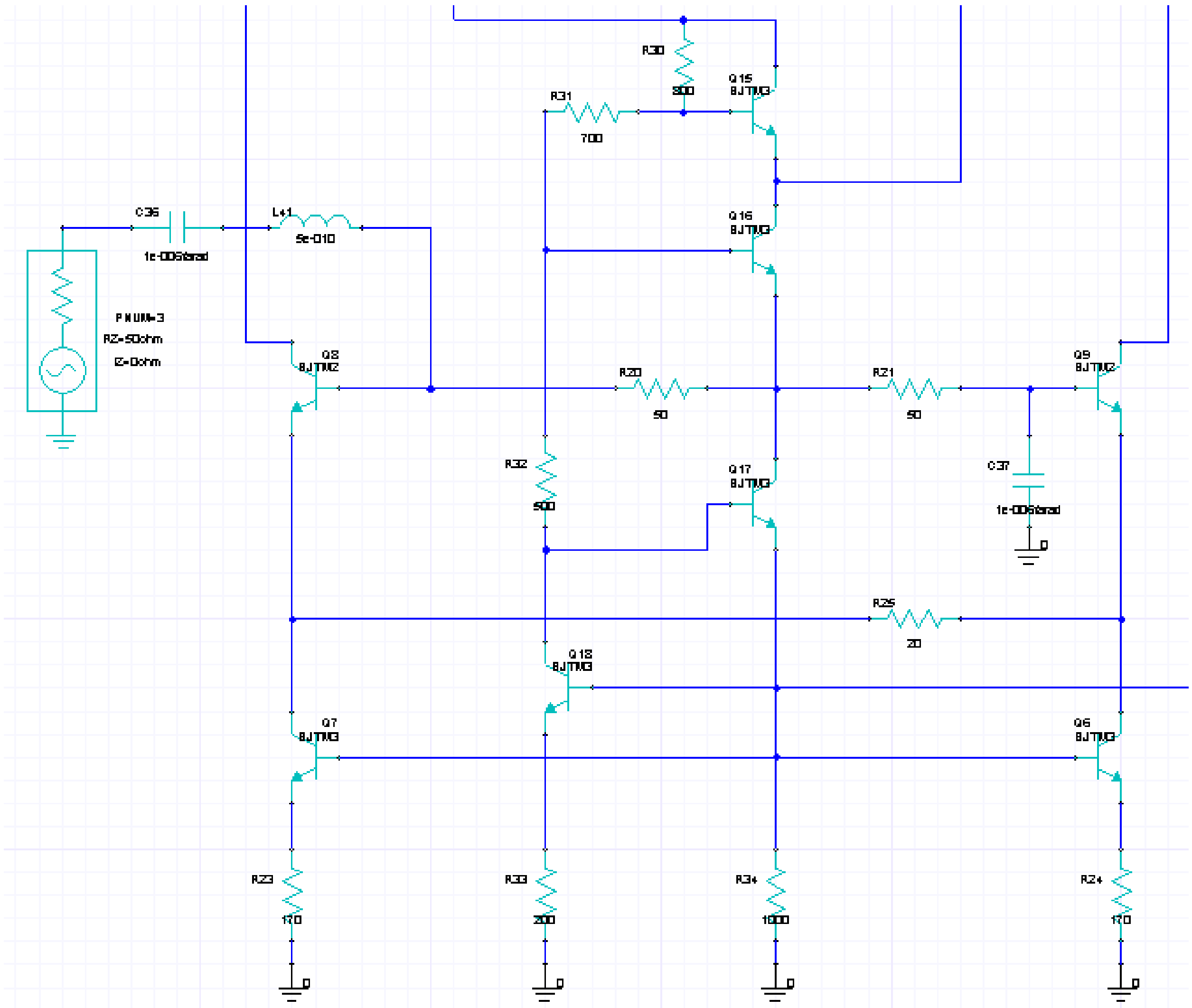


4. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

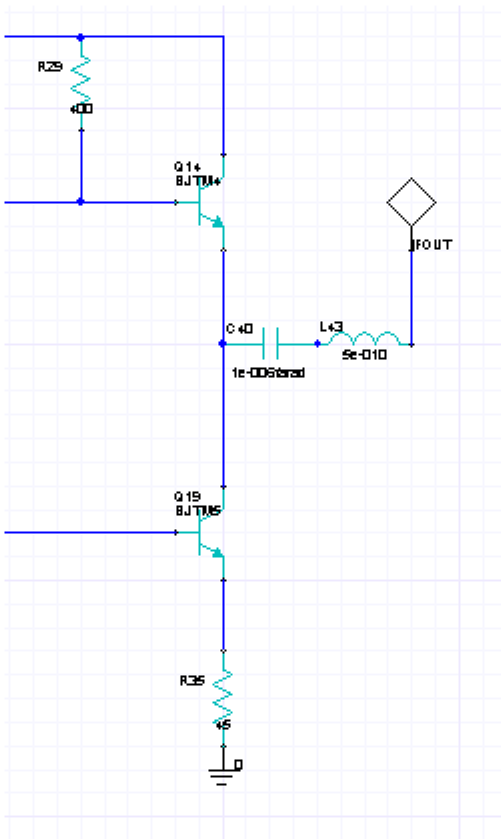
The core Gilbert Cell employs four NPN transistors. This is the portion of the schematic with the Gilbert cell. This block also shows two of the four input ports. Port1 includes a DC voltage source **VSRC1** for all the transistors in the circuit. Port2 includes the two sinusoidal voltage signals **VRF1** and **VRF2**.



Here is the mixer **VLO** input. Port 3 includes the **VLO** sinusoidal voltage source.



This is the mixer output.



The circuit has three input ports and one output port.

- Port 1 (PNUM=1) contains a 5V DC voltage source **VSRC1** common to all the transistors.
- Port 2 (PNUM=2) contains two sinusoidal voltage sources, the RF inputs **VRF1** and **VRF2**.
- Port 3 (PNUM=3) contains one sinusoidal voltage source, the local oscillator input **VLO**.
- Port 4 is the output port.

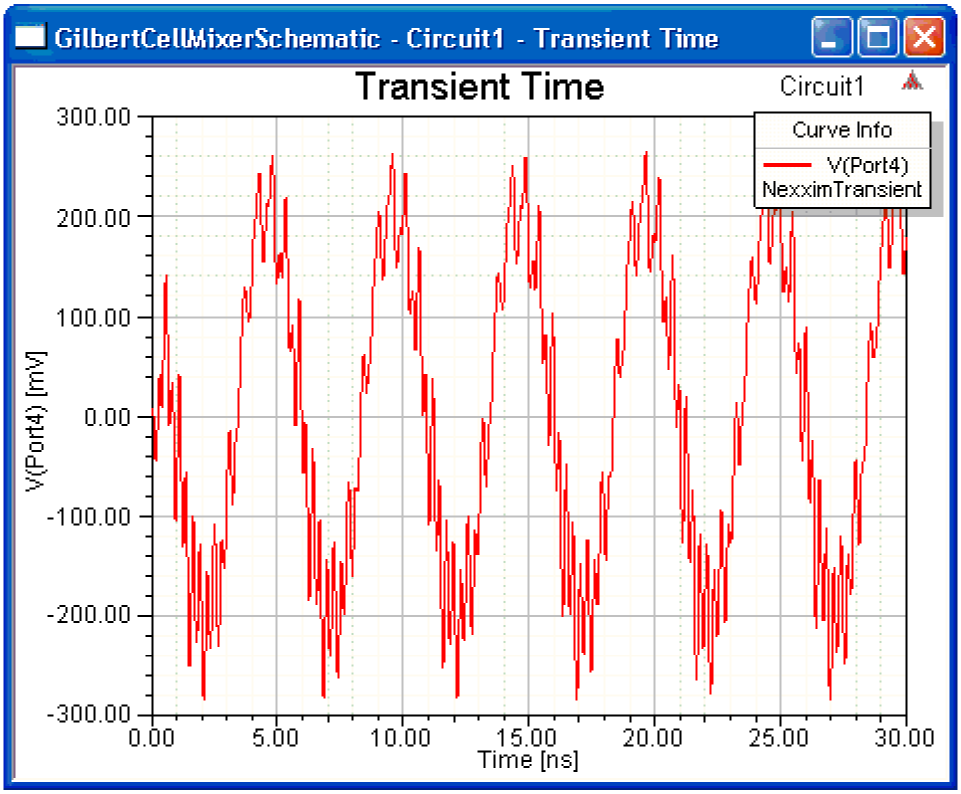
Analyze the Gilbert Cell Mixer

1. From the Project group box, expand the **Analysis** icon. Two analyses have been set up, **Nexxim Transient** and **HBnTone**. Sources **VSRC1**, **VLO**, and **VRF1** are enabled for both analyses. Source **VRF2** is enabled for **HBnTone** only.
2. To run both analyses, right-click the **Analysis** icon, and click **Analyze**.

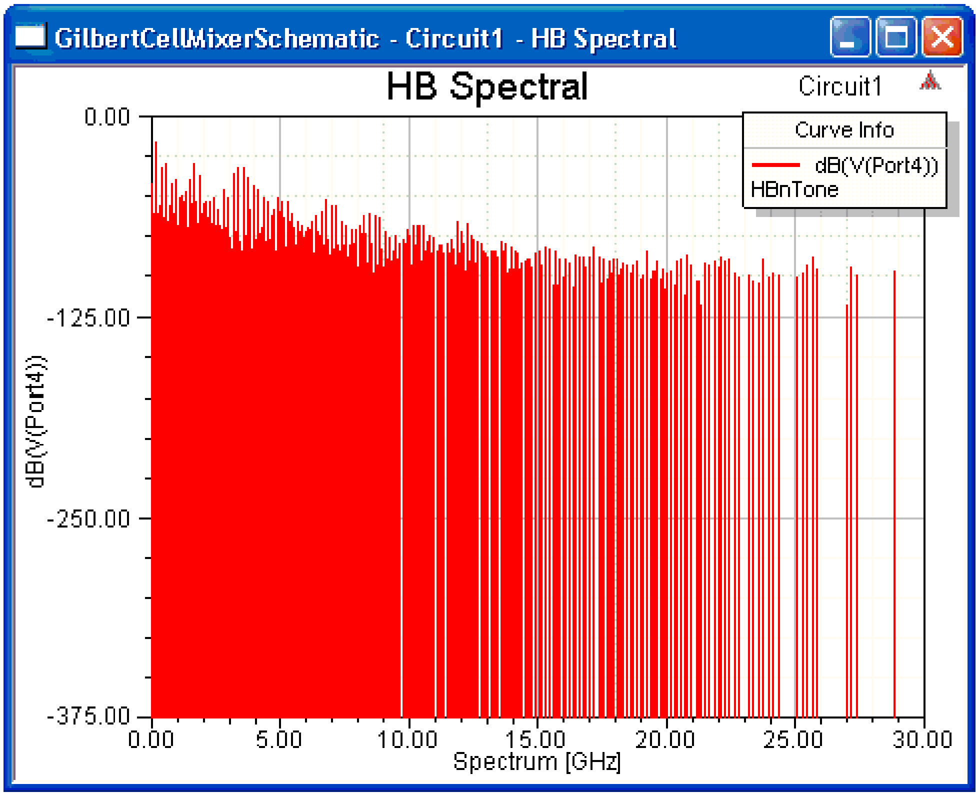
A progress window briefly appears at the lower right to indicate that the transient analysis is being performed. A longer-duration progress window indicates that the harmonic balance analysis is being performed. When both analyses are completed, the Message Manager window at the lower-left shows the name of the project, errors or warnings if any, and analysis statistics such as CPU time.

View the Time and Spectral Domain Results

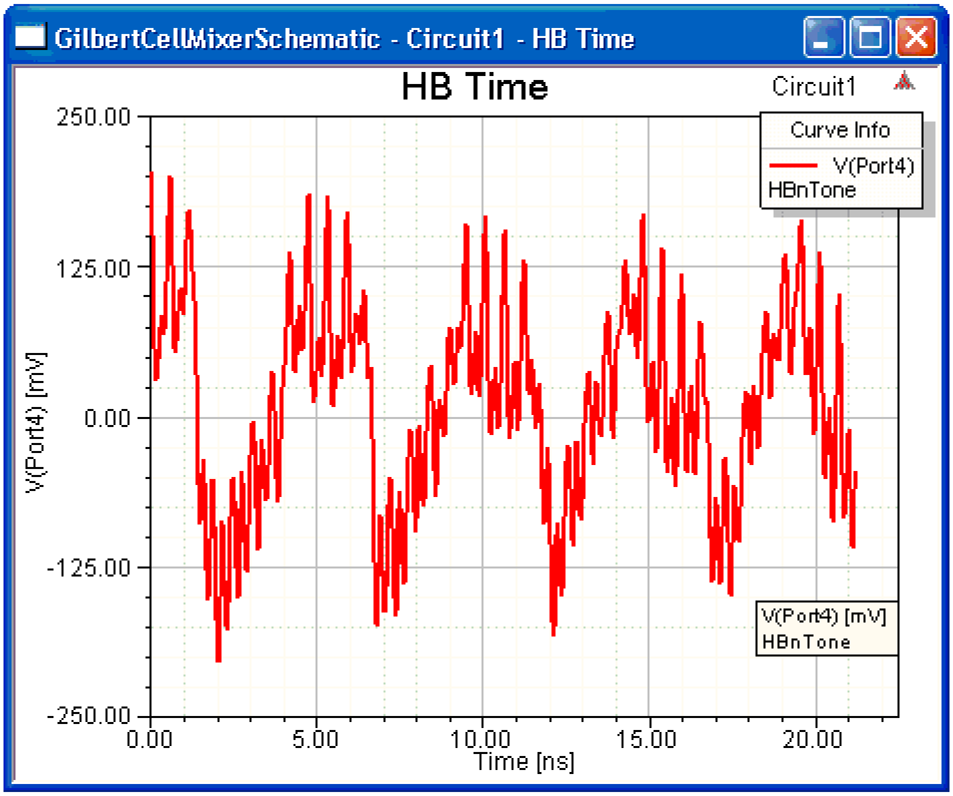
To view the result of transient analysis, expand the reports icon and click the **Transient Time** report. The **Reports** window displays the time domain waveform for the mixer:



3. Double-click the **Spectral** report. The **Results** window displays the frequency domain data:



4. For comparison with the transient analysis result, view the harmonic balance analysis results in the time domain. In the **Results** tree, double-click the **HB Time** report. The **Results** window displays the time domain transformation:



For more information, see Nexxim Harmonic Balance.

Oscillator and Phase Noise Analysis

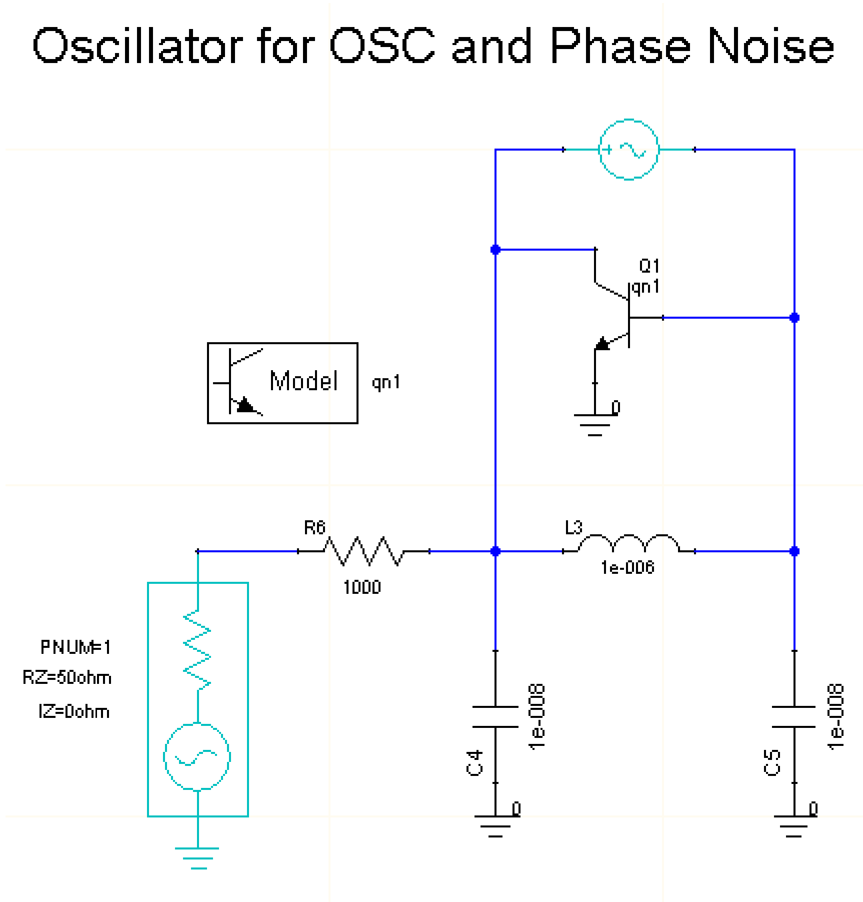
In this example you run a Nexxim oscillator analysis to find the frequency of a Colpitts oscillator circuit , then verify the initial estimate using a resonant frequency search. Next, you run a phase-noise analysis to determine the behavior of the oscillator around its dominant oscillation point.

Open the Oscillator Schematic

Open the example Oscillator Schematic project on the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Oscillators** directory, then select the file **ColpittsOscillator.aedt**. Click **Open**.

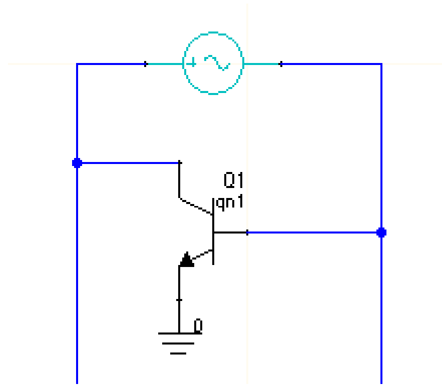
The oscillator schematic appears in the design window:



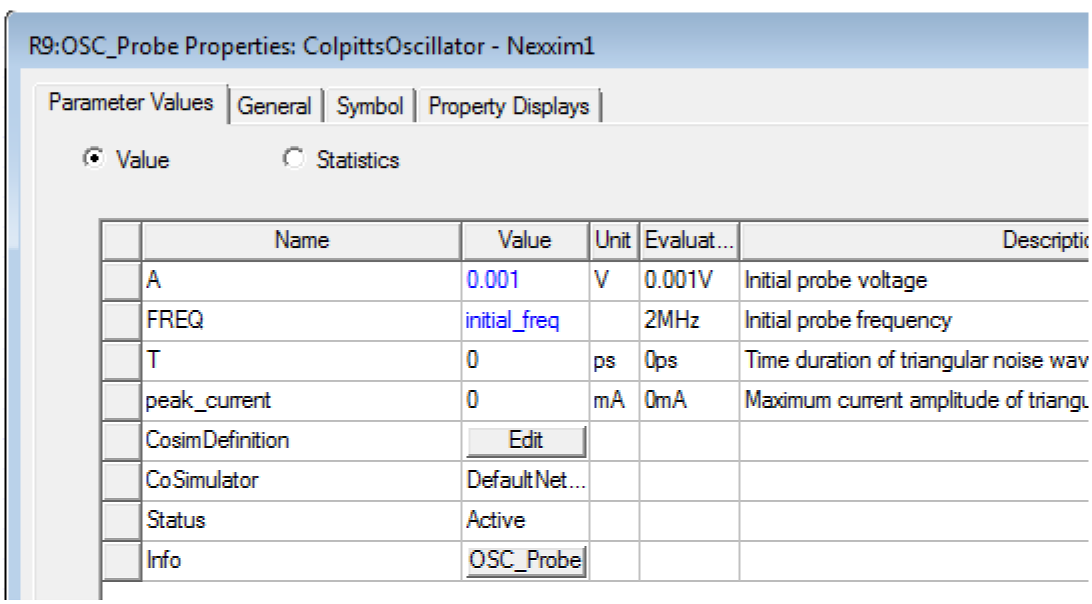
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

Perform an Oscillator Analysis

1. To enable oscillator analysis, an oscillator probe has been added to the schematic above the bipolar transistor Q1. Here is that portion of the schematic :



- Click the oscillator probe to display its properties on the **Param Values** tab of the **Properties** window.
- The **Param Values** tab for the oscillator probe should have the values shown here:

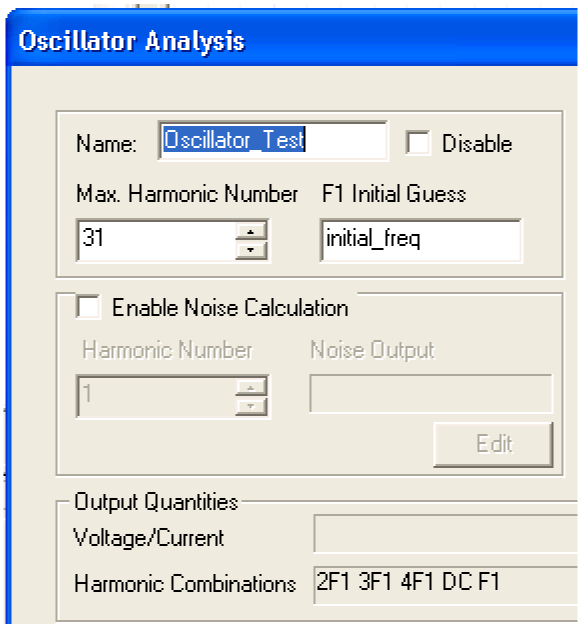


	Name	Value	Unit	Evaluat...	Descriptic
	A	0.001	V	0.001V	Initial probe voltage
	FREQ	initial_freq		2MHz	Initial probe frequency
	T	0	ps	0ps	Time duration of triangular noise wav
	peak_current	0	mA	0mA	Maximum current amplitude of triangu
	CosimDefinition	Edit			
	CoSimulator	DefaultNet...			
	Status	Active			
	Info	OSC_Probe			

- Parameter **A** is set to 0.001. Parameter **A** is the initial voltage used for the initial guess at the oscillating frequency. Press **Enter** to move to the next parameter.
- Parameter **FREQ** is set to local variable **initial_freq**. The initial value is **2Mhz** as shown above. Click **OK** to close the **Properties** window.
- Parameters **T** and **peak_current** are specific to transient analysis and do not affect oscillator analysis, so they are not set for this example.

2. Select the **Projects** tab in the Project window. Expand the **Analysis** icon in the **Project Manager** window, right-click the **Oscillator_test** solution setup.

3. The **Oscillator Analysis** window opens:



- **Oscillator_test** is the name of the setup.
- **31** is the **Max. Harmonic Number**. This specifies the highest harmonic to use in the harmonic balance calculations that underlie the oscillator analysis.
- Variable **initial_freq** is the **F1 Initial Guess**. The local variable sets the initial frequency estimate for both the oscillator probe and the analysis setup:
- Click **OK** to close the **Oscillator Analysis** window.

4. Expand the **Analysis** icon in the Project window, right-click the **Oscillator_test** solution setup, and select **Analyze** on the menu. The oscillator analysis runs. The **Progress** window displays a red line to show the progress of the analysis.

```
analysis:osc(info): lowest oscillation frequency is 2.28274e+006 Hz
(1:09:46 PM Mar 25, 2017)
```

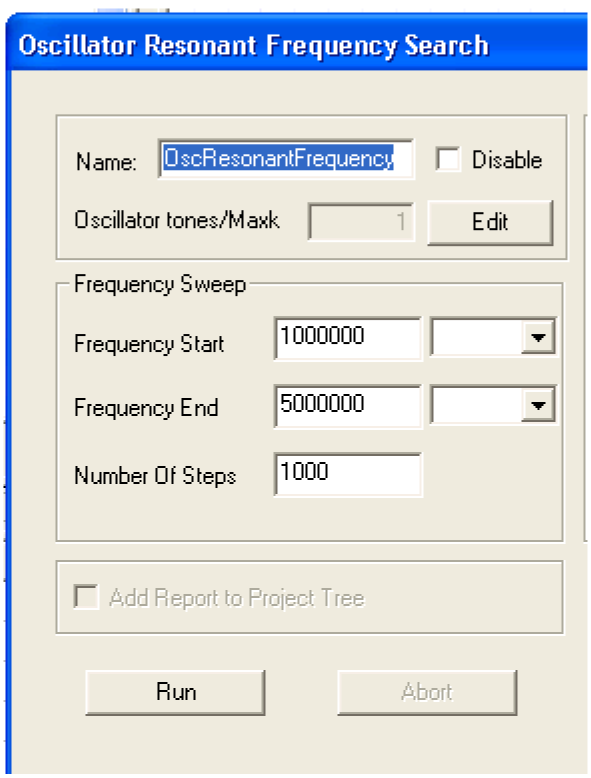
- The **analysis:osc** message shows the final oscillating frequency, 2.28274 MHz.
- Other messages regarding the initial guess and other analysis operations are written to the log file.

Perform a Resonant Frequency Search

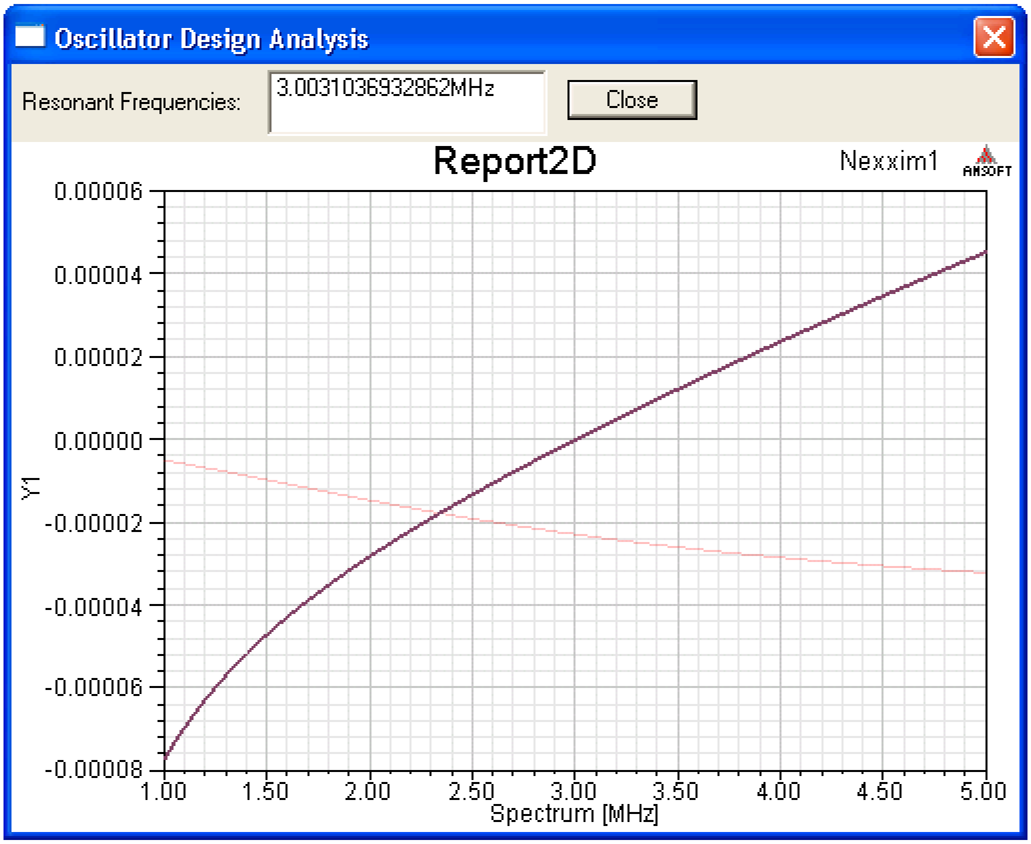
Resonant frequency search performs a limited form of oscillator analysis to obtain a rough estimate of the resonant frequency. With resonant frequency search, only the initial guess phase of oscillator analysis is performed. Resonant frequency search uses the frequency sweep method rather than the transient method used in the full oscillator analysis. The frequency sweep method operates on highly linearized versions of the circuit elements. Compare the result obtained by the resonant frequency search with the result of the fully nonlinear oscillator analysis obtained earlier in this example.

> **Note:**
>
> Resonant frequency search is a separate analysis only in Nexxim running under the Schematic Editor. With netlist designs, the equivalent of resonant frequency search can be set up using oscillator command entries and options. For more information see .OSC Resonant Frequency Search Statement.

1. Right-click the Analysis icon and select **Add Nexxim Solution Setup > Oscillator Resonant Frequency Search** to open the **Oscillator Resonant Frequency Search** setup window.
3. In the **Frequency Sweep** group box, **Frequency Start** is 1MHz, **Frequency End** is 5MHz, and N**umber of Steps** is 1000. The values are displayed in decimal form:



4. Click **Run**. The analysis runs directly on the window window. The Progress window briefly opens to show the progress of the analysis.
5. When the simulation has completed successfully, the **Oscillator Design Analysis** window opens to open the result:

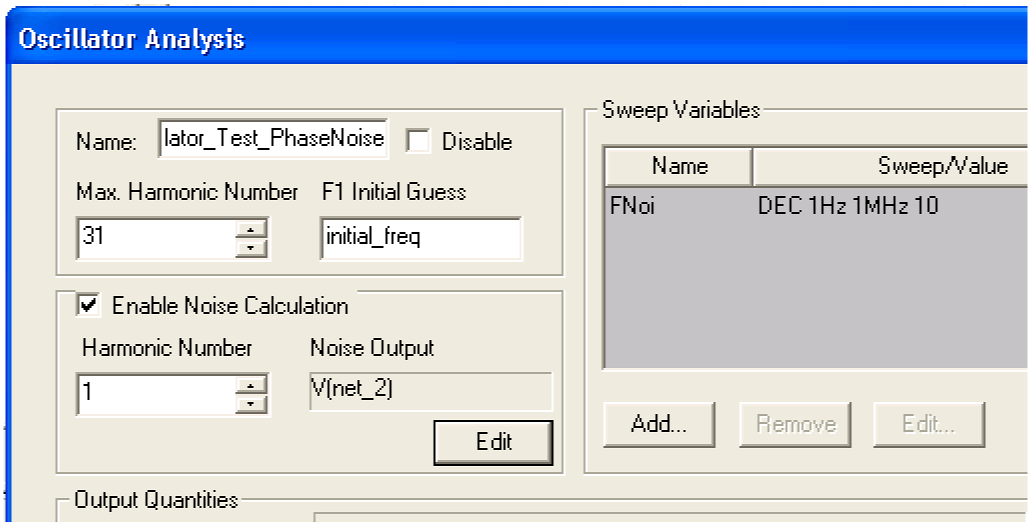


6. The resonant frequency occurs where the imaginary part of the probe current Im[I (Probe1)] (the darker trace) crosses the zero line on the Y-axis. This frequency displays in the **Resonant Frequencies** field at the top of the window. Because of the limitations of resonant frequency search, the estimated resonant frequency is about 30% higher than the frequency calculated by the full oscillator analysis.
7. Click **Close** to close the **Oscillator Design Analysis** window.
8. Click **OK** to close the **Oscillator Resonant Frequency Search** window.

Perform a Phase Noise Analysis

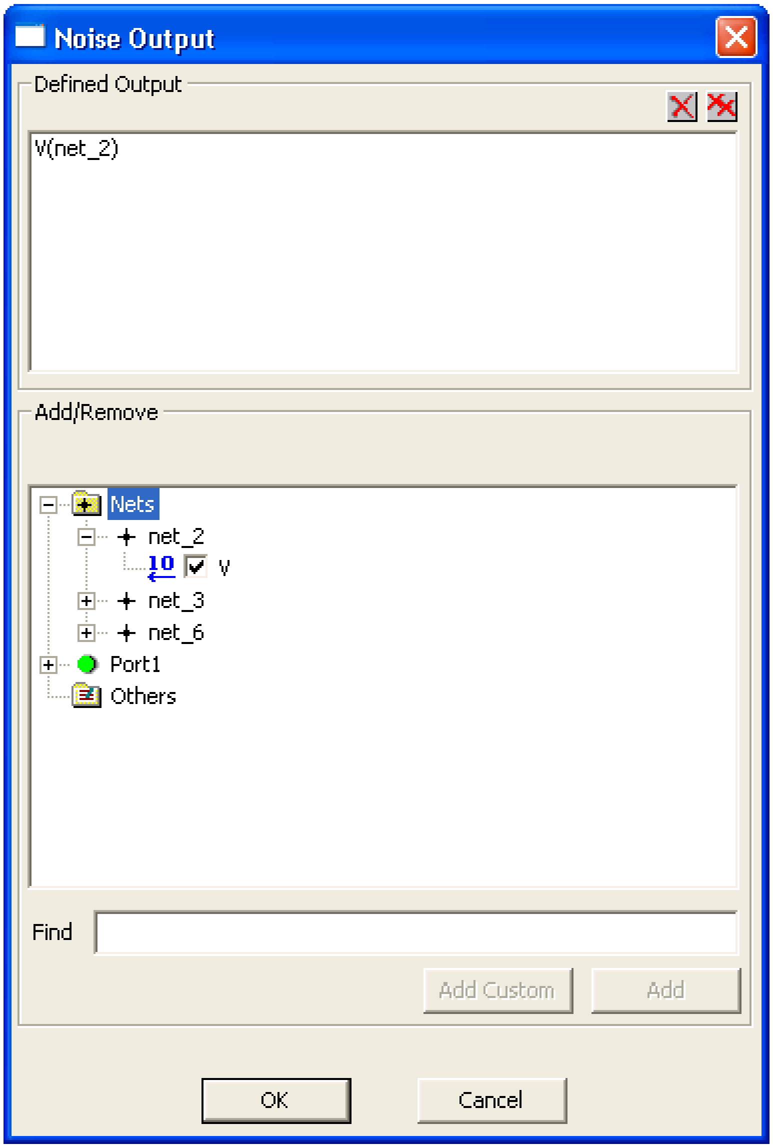
The next exercise is to run phase noise analysis on the oscillator circuit. Nexxim uses the calculated oscillating frequency as the fundamental around which phase noise is computed.

1. Expand the **Analysis** icon in the Project window and double-click the **Oscillator_Test_PhaseNoise** solution setup to open the **Oscillator Analysis** window.



- The **Enable Noise Calculation** check box is checked.
- The default **Harmonic Number** is 1, the fundamental oscillating frequency.
- In the **Sweep Variables** group box, a sweep of the noise frequency Fnoi has been set up. The sweep is a **Decade count**, from 1 Hz to 1 MHz using a count of 10.

- Click **Edit** under the **Noise Output** field. The **Noise Output** window opens:



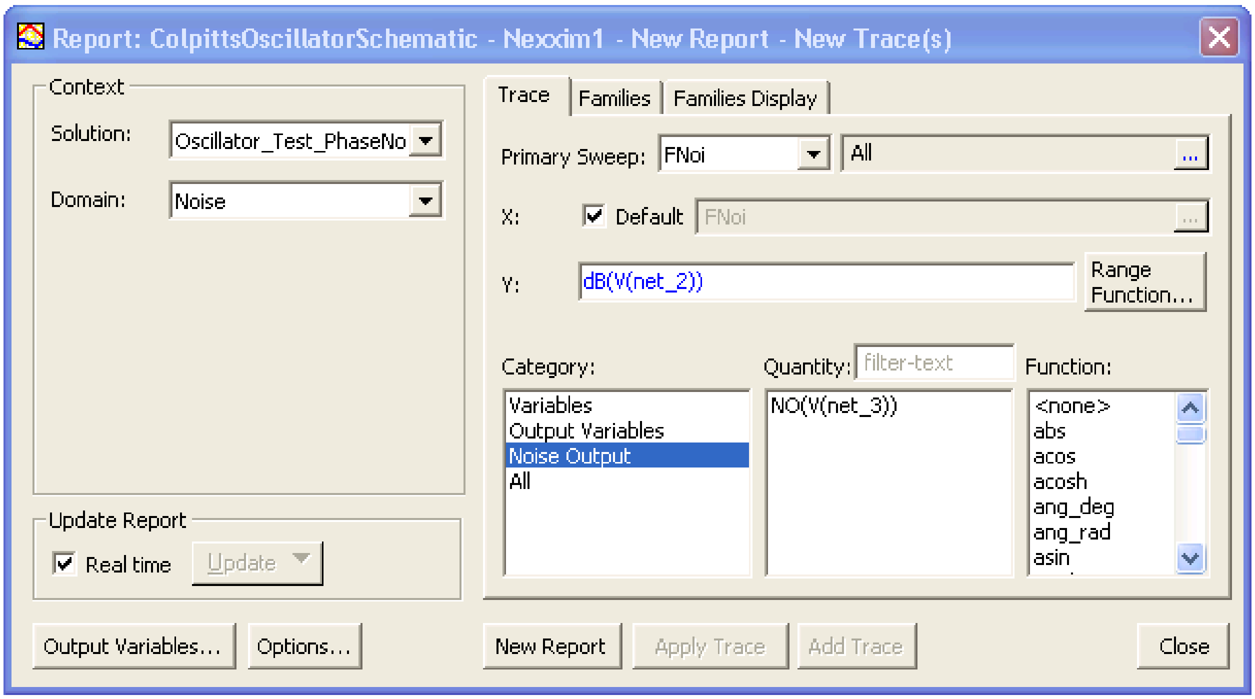
- Expand the **Nets** icon, and then expand the **net_2** icon. The **V** group box under **net_2** is selected as the output. The **Defined Output** group box shows **V(net_2)**.
- Click **OK** to close the **Noise Output** window and return to the **Oscillator Analysis** window.
- Click **Finish** to close the **Oscillator Analysis** window.

2. Expand the **Analysis** icon in the Project window, right-click the **Oscillator_Test_ PhaseNoise** solution setup, and select **Analyze** on the menu. The oscillator analysis runs. The **Progress** window displays a red line to show the progress of the analysis.

3. When the analysis has completed, the Message window reports the oscillator RMS jitter:

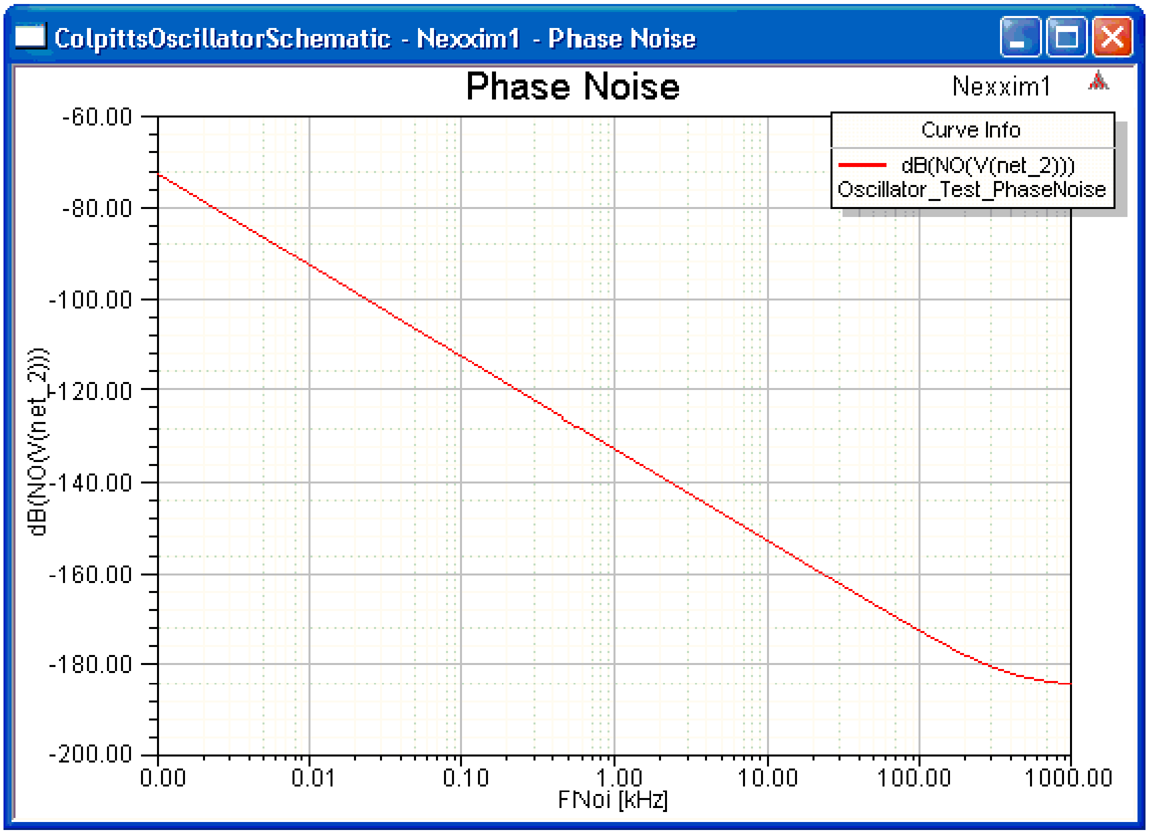
```
analysis:osc(info): Oscillator root mean square jitter is
5.57352e-014 seconds (7.99402e-007 radians) per oscillation
```

4. Right-click the **Results** icon in the **Project Manager** window and select the **PhaseNoise** report setup. The **Report** window opens:

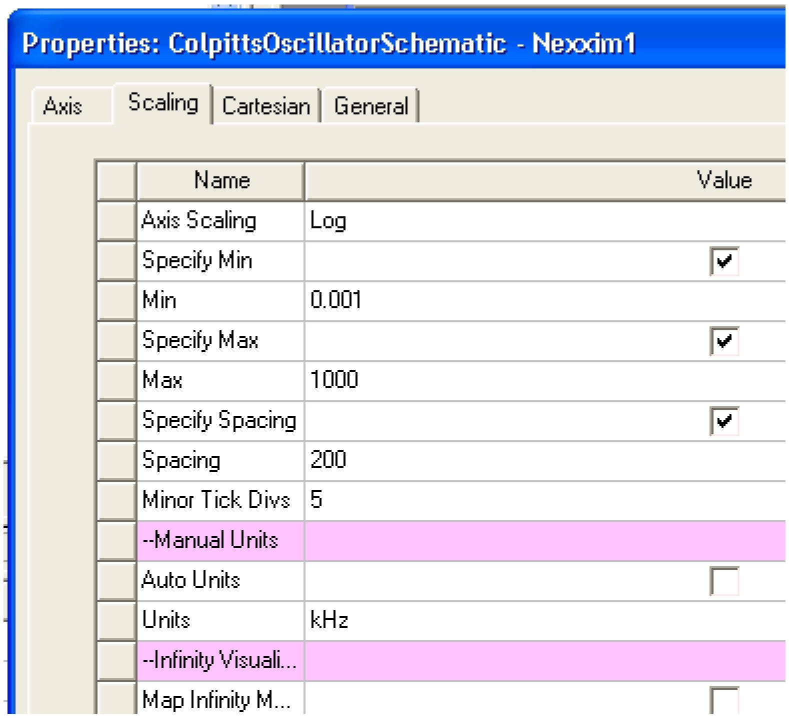


- The **Solution** is **Oscillator_Test_PhaseNoise**.
- **Noise** is the **Domain**.
- **Noise Output** is the **Category**.
- The X-axis is the sweep of noise frequency **Fnoi**.
- the Y-axis is the noise output **dB(V(net_2))**.
- Click **New Report** to generate the report.
- Click **Close** to close the **Report** window.

5. The Report window opens with a graph of the result:



6. The X-axis has been modified to show the double-logarithmic form that is commonly used for displaying phase noise. To see the modifications, double-click the X-axis label “Fnoi [MHz]” to open the **X-Axis Properties** window. Select the **Scaling** tab:



- **AxisScaling** is **Log** scaling (default is **Linear**). **Min** is 0.001, **Max** is 1000, and **Minor Tick Divs** to 5. The **Auto Units** option is deselected, and the **Units** are **Khz**.
- Click **OK** to close the **X-Axis Properties** window.

7. Close the Project when you are finished viewing the graph.

For more information, see Nexxim Oscillator Analysis.

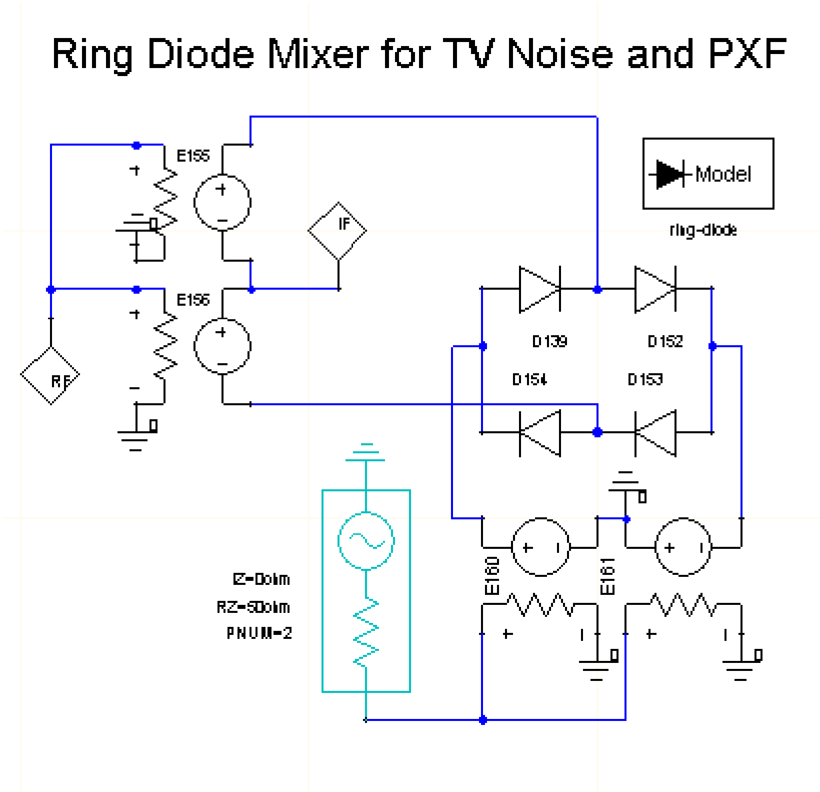
TV Noise and PXF Analyses

In this example you perform time-varying noise analysis and periodic transfer function analysis on a ring diode mixer circuit.

Open the Ring Diode Mixer Schematic

Open the example on the file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Navigate to **Electronics Desktop** > **Circuit** > **RF_Microwave** > **Mixers** and select **Ring_ Diode_Mixer.aedt**. Then click **Open** to open the mixer schematic in the design window.

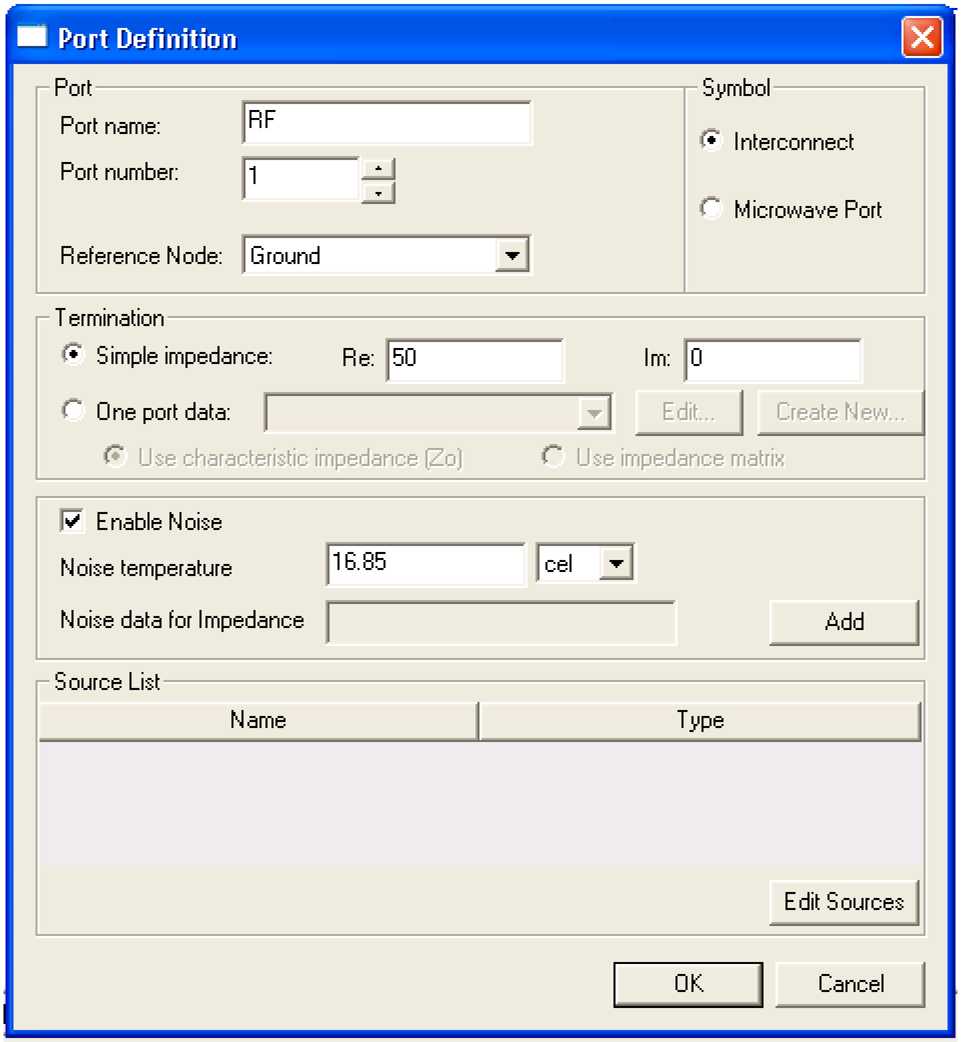


3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

Perform a Time-Varying Noise Analysis

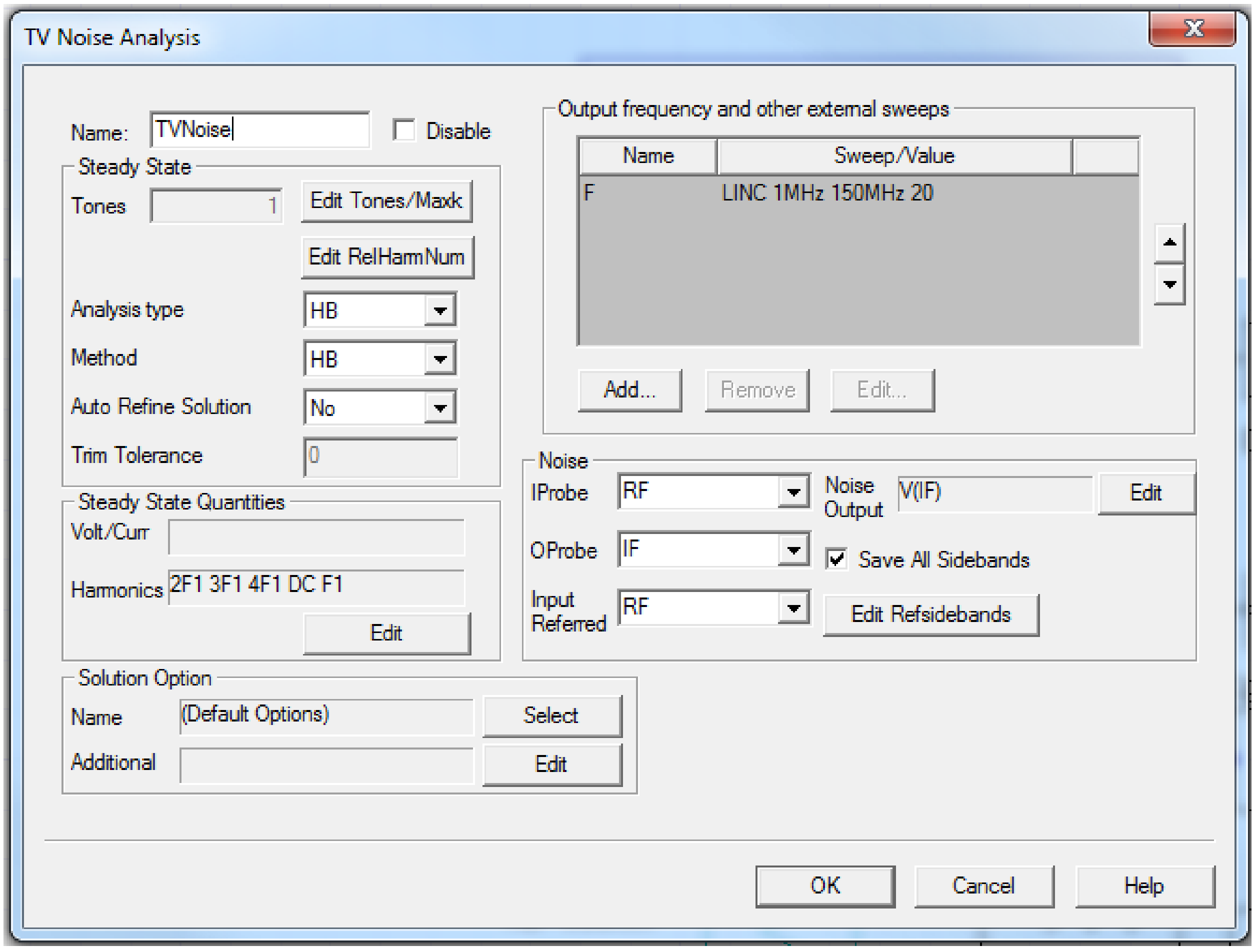
To enable periodic time-varying noise (TV noise) analysis, the input port definition should include noise data.

1. Right-click the **RF** interface port at the left of the schematic and select **Edit Port** on the menu. The **Port Definition** window opens:

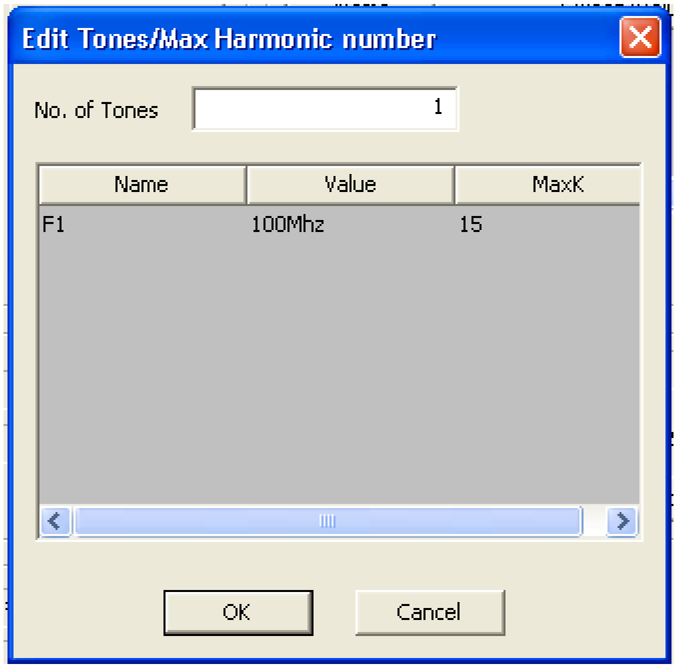


- Verify that the **Enable Noise** check box is checked as shown in the illustration above.
- Click **OK** to close the **Port Definition** window.

2. Expand the Analysis icon and double-click the **TVNoise** analysis setup. The **TV Noise Analysis** window opens:

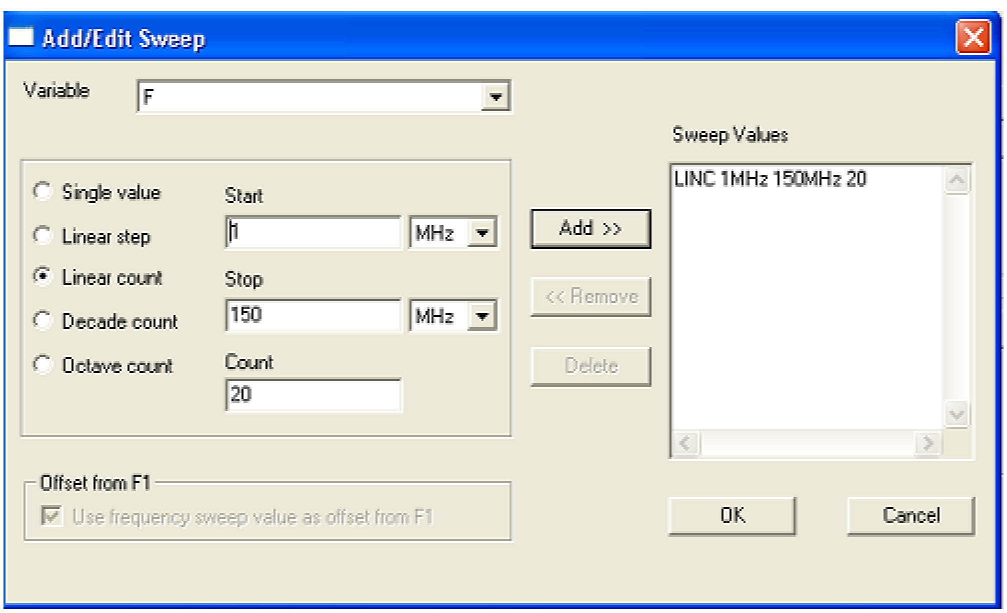


3. Verify that the **Analysis Type** is HB, the **Method** is HB, **Auto Refine Solution** is **No**, and the **Trim Tolerance** field is inactive (0).
4. Click **Edit Tones/Maxk** in the **Steady State** group box.
5. The **Edit Tones/Max Harmonic Number** window opens:



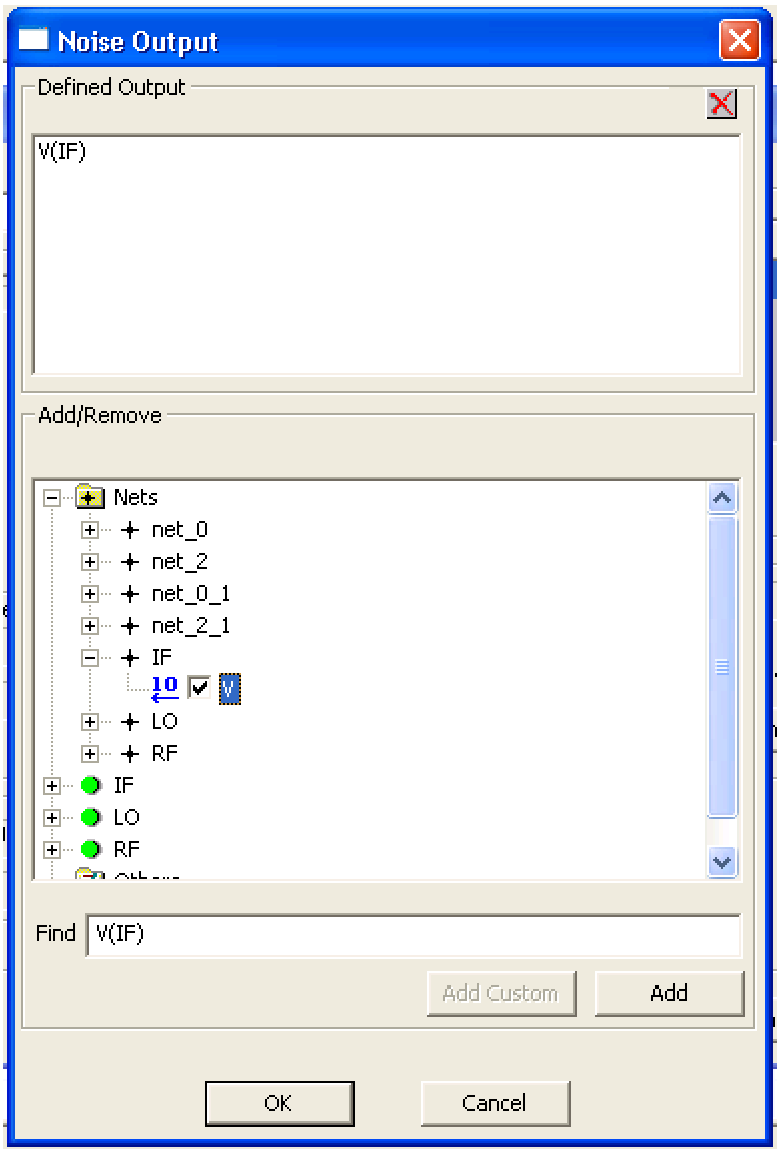
- **T** he **No. of Tones** is **1**.
- F1 is the frequency for the harmonic balance calculations. The **Value** field is **100MHz**.
- **15** is the **MaxK** field. **MaxK** specifies the highest harmonic to use in the harmonic balance calculations that underlie the TV Noise analysis.
- Click **OK** to close the **Edit Tones/Max Harmonic Number** window and return to the **TV Noise Analysis** window.

6. In the **Output frequency and other external sweeps** group box, click **Edit** to open the **Add/Edit Sweep** window.



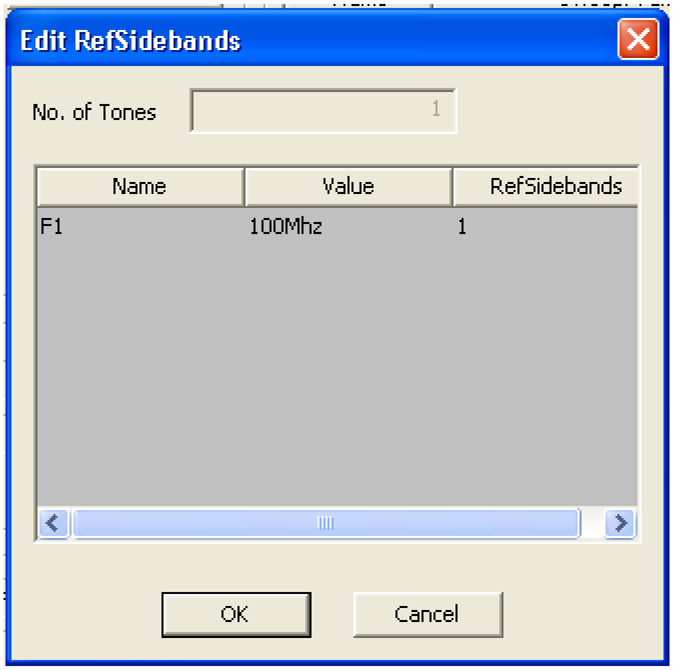
- Ensure **F** is showing in the **Variable** field.
- Select **Linear Count**.
- Enter **1** in the **Start** field. Use the drop-down menu to select **MHz** as the unit.
- Enter **150** in the **Stop** field. Use the drop-down menu to select **MHz** at the unit.
- Enter **20** in the **Count** field.
- Click **Add**. The sweep specification appears in the **Sweep Values** display.
- Click **OK** to close the **Add/Edit Sweep** window and return to the **TV Noise Analysis** window.

7. In the **Noise** group box, select **RF** aand **IF** on the **IProbe** and **OProbe** drop-down menus.
8. Click **Edit** on the **Noise Output** field to open the **Noise Output** window:



- Expand the **Nets** icon, the **IF** icon, then click the **V** group box as shown in the preceding figure. Click **Add**. The entry V(IF) appears in the **Defined Output** field.
- Click **OK** to close the **Noise Output** window and return to the **TV Noise Analysis** window.

9. Selecting a noise output enables the **Input Referred** field. Select **RF** on the drop-down menu.
10. Click the **Save All Sidebands** group box. click **Edit Refsidebands** to view the sidebands settings:



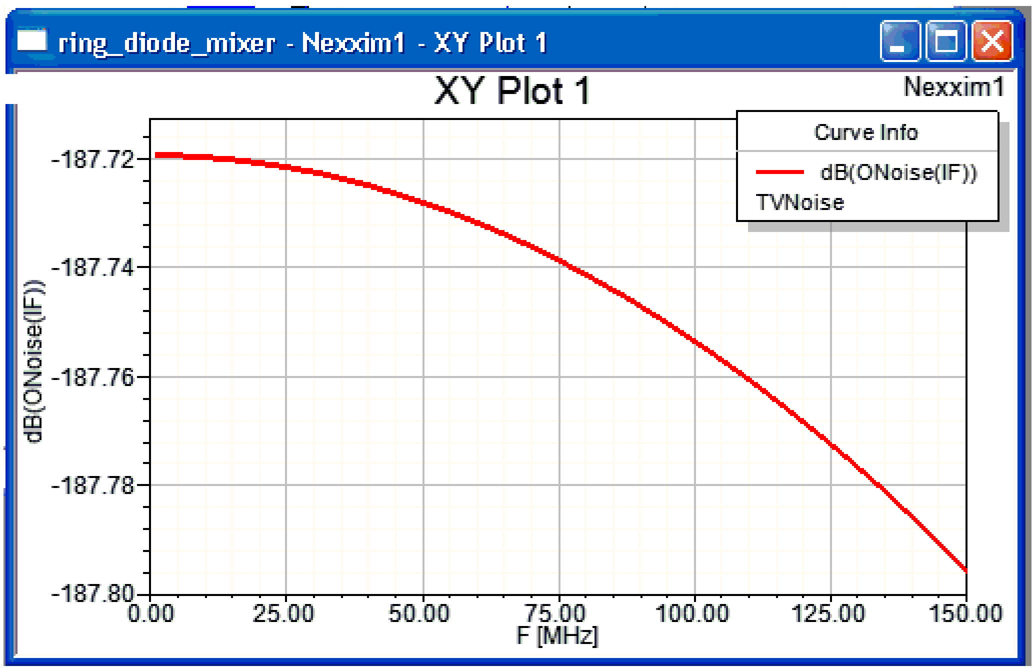
11. Click **OK** to close **EditRefSidebands** window.
12. Click **OK** to close the **TV Noise Analysis** window.
13. Expand the **Analysis** icon in the Project window, right-click the **TVNoise** solution setup, and select **Analyze** on the menu. The TV noise analysis runs. The **Progress** window displays a red line to show the progress of the analysis.

When the analysis has completed, the results are available, including: the Noise Figure (NF) and the Power Conversion Gain (CG).

1. Right-click the **Results** icon in the **Project Manager** window and select **Create Standard Report > Rectangular Plot**.
2. The **Report** window opens with the **Trace** tab selected.
   - Select **TVNoise** on the **Solution** drop-down.
   - Select **Noise** on the **Domain** drop-down.
   - Select **Noise** in the **Category** group box.

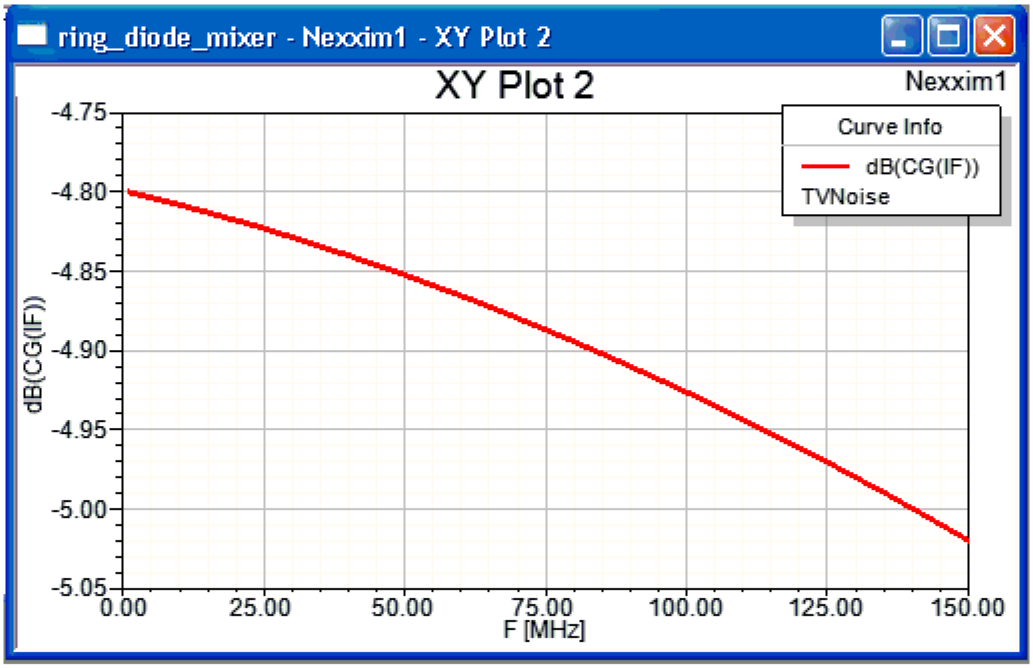
- Select **Onoise(IF)** in the **Quantity** list.
- Select **dB** in the **Function** group box.
- Click **New Report**.
- Click **Close** to close the **Report** window.

3. The Report window opens with a graph of the result:



4. Right-clickce more on the **Results** icon in the **Project Manager** window and select **Create Standard Report > Rectangular Plot**.
5. The **Report** window opens with the **Trace** tab selected.
   - Select **TVNoise** on the **Solution** drop-down.
   - Select **Noise** on the **Domain** drop-down.
   - Select **Power Conversion Gain** in the **Category** list.
   - Select **CG(V(IF))** in the **Quantity** list.
   - Select **dB** in the **Function** group box.
   - Click **New Report**.
   - Click **Close** to close the **Report** window.

6. The Report window opens with a graph of the result:



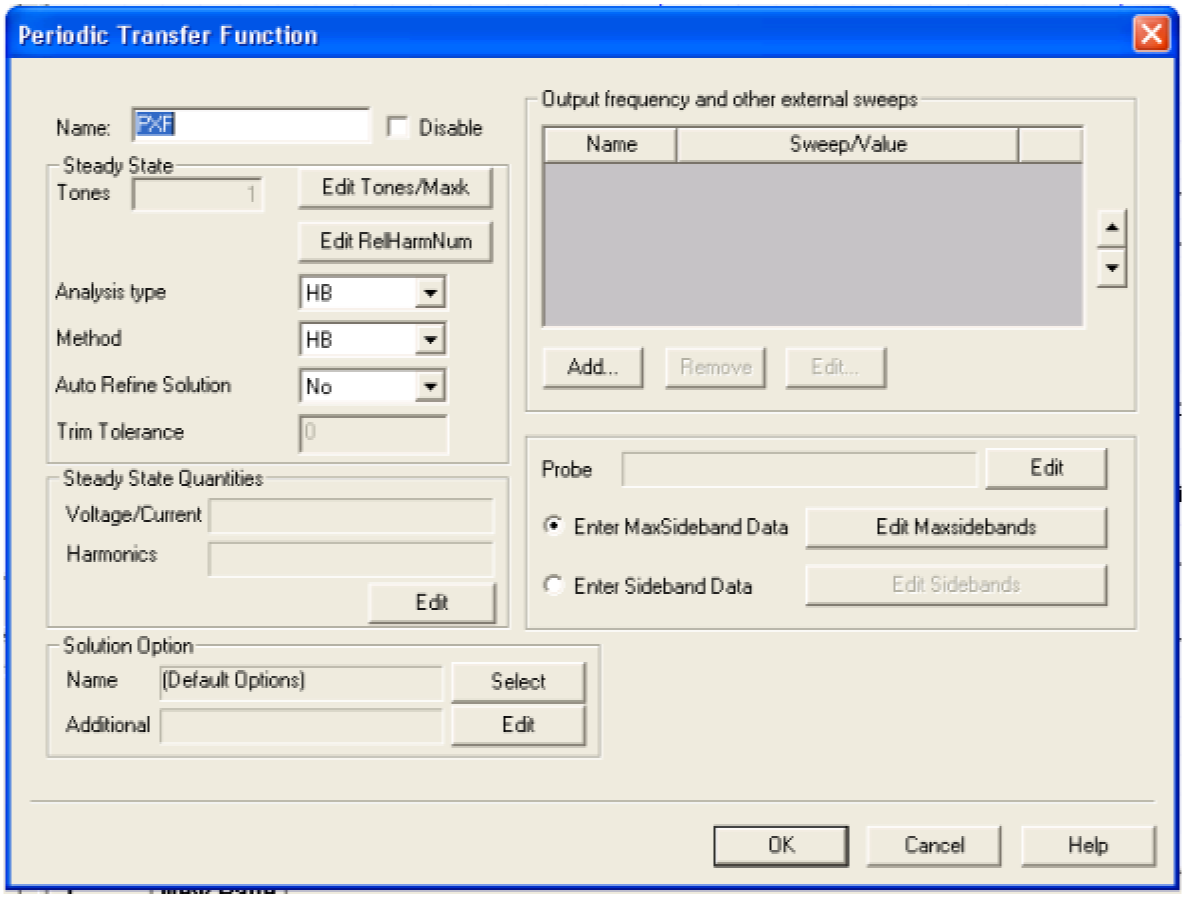
Perform a Periodic Transfer Function Analysis

Periodic transfer function (PXF) analysis computes the small-signal transfer function from selected voltage and current sources in the circuit to a specified single output at a frequency that can be swept. A steady-state HB or OSC analysis first computes the periodic or quasi-periodic operating point, on which the small-signal PXF analysis is calculated. In Nexxim, PXF analysis is an extension of time-varying noise analysis.

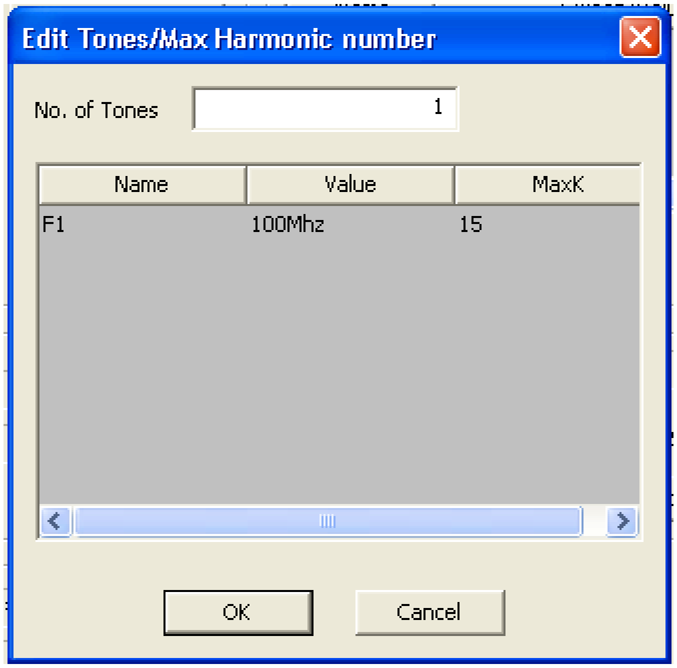
Use the ring diode mixer to demonstrate periodic transfer function analysis.

1. From the **Circuit** menu, select **Add Nexxim Solution Setup** > **Periodic Transfer Function (PXF)**.

2. The Periodic Transfer Function setup window opens:

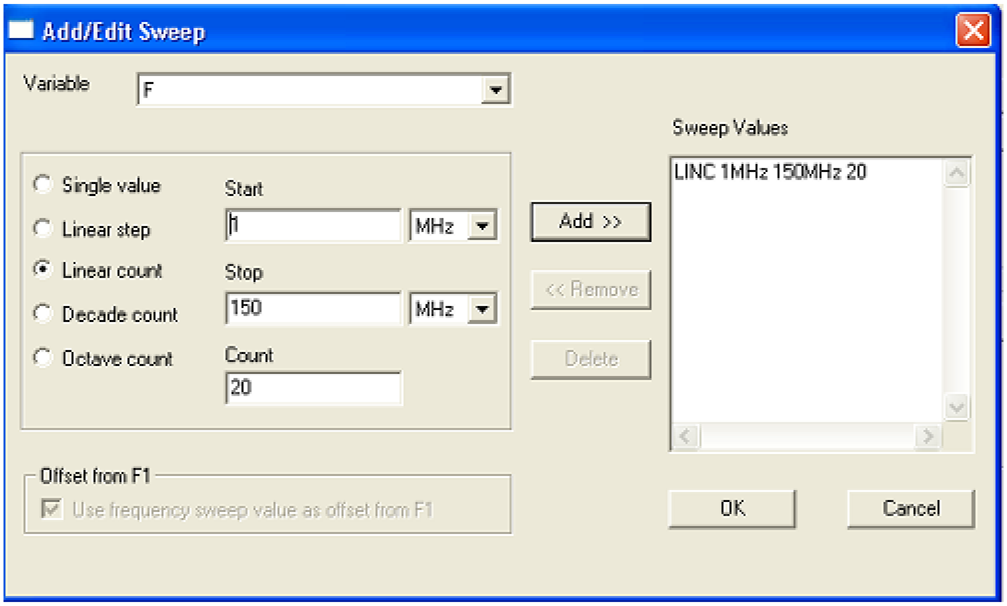


3. Verify that the **Analysis Type** is HB, the **Method** is HB, **Auto Refine Solution** is No, and the **Trim Tolerance** field is inactive (0).
4. Click **Edit Tones/Maxk** in the **Steady State** group box.
5. The **Edit Tones/Max Harmonic Number** window opens:



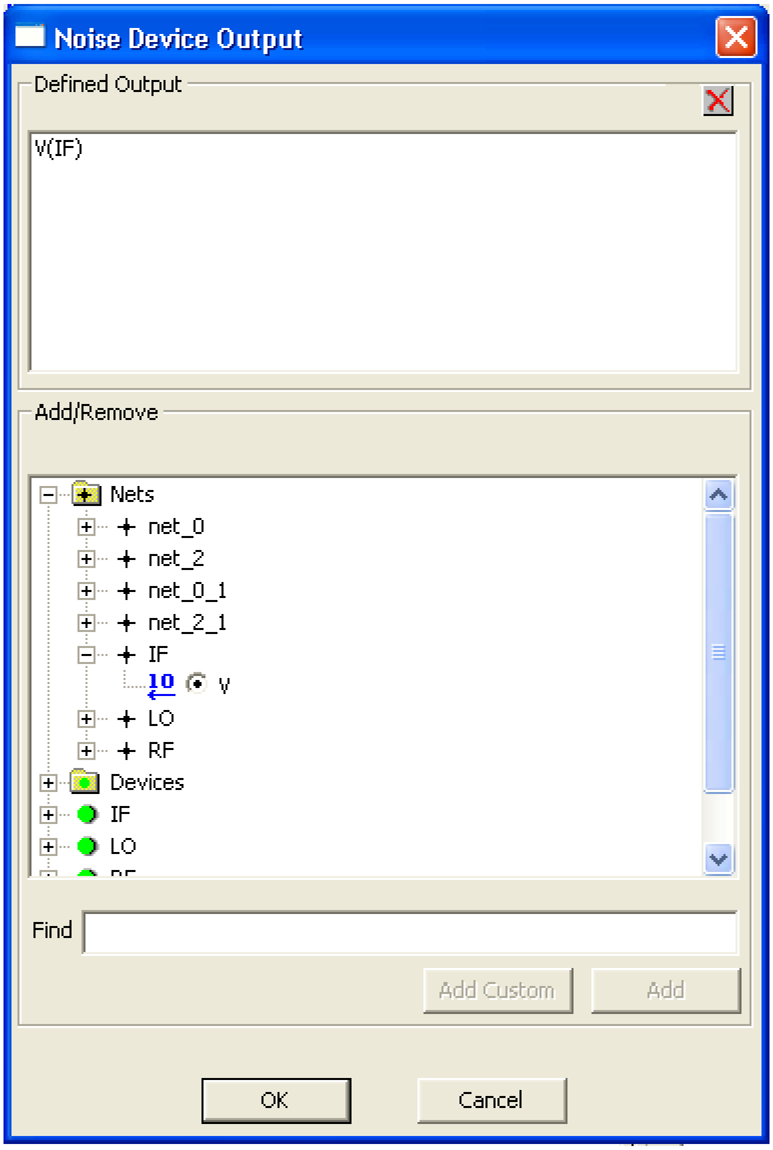
- Leave the default of **1** as the **No. of Tones**.
- Click the F1 line and click again on the **Value** field. Enter **100MHz** in the **Value** field. The value sets the frequency for the harmonic balance calculations.
- Enter **15** in the **MaxK** field. This specifies the highest harmonic to use in the harmonic balance calculations that underlie the Periodic Transfer Function analysis.
- Click **OK** to close the **Edit Tones/Max Harmonic Number** window and return to the **Periodic Transfer Function Analysis** window.

6. In the **Sweep Variables** group box, click **Add** to open the **Add/Edit Sweep** window.



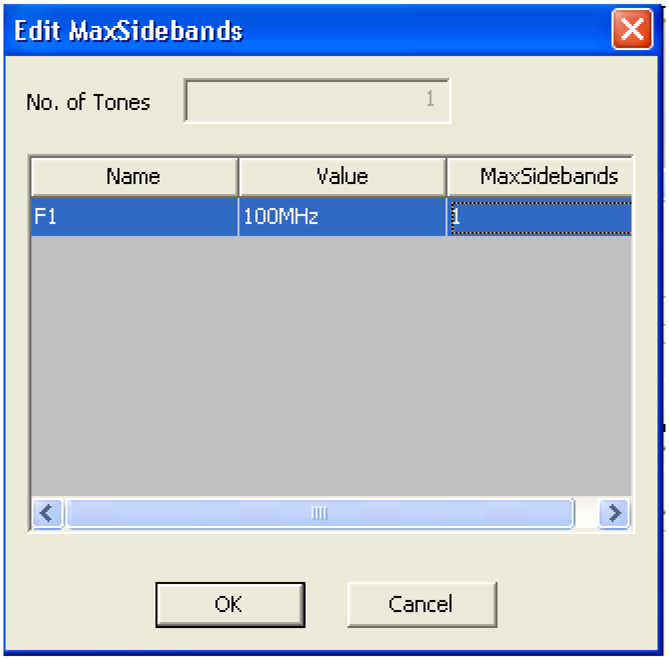
- Ensure **F** is showing in the **Variable** field.
- Select **Linear Count**.
- Enter **1** in the **Start** field. Use the drop-down menu to select **MHz** as the unit.
- Enter **150** in the **Stop** field. Use the drop-down menu to select **MHz** at the unit.
- Enter **20** in the **Count** field.
- Click **Add**. The sweep specification appears in the **Sweep Values** display.
- Click **OK** to close the **Add/Edit Sweep** window and return to the **Periodic Transfer Function Analysis** window.

7. Click **Edit** on the **Probe** field to open the **Noise Device Output** window:



- Expand the **Nets** icon, the **IF** icon, then click the **V** group box as shown in the preceding figure. Click **Add**. The entry V(IF) appears in the **Defined Output** field.
- Click **OK** to close the **Noise Device Output** window and return to the **Periodic Transfer Function Analysis** window.

8. Click **Edit Maxsidebands** to open the **Edit MaxSidebands** window:



- The F1 and 100MHz values are automatically inserted in the window.
- Click the **MaxSidebands** field and enter **1**.
- Click **OK** to close the **Edit MaxSidebands** window and return to the **Periodic Transfer Function Analysis** window.
- Click **OK** to close the **Periodic Transfer Function Analysis** window.

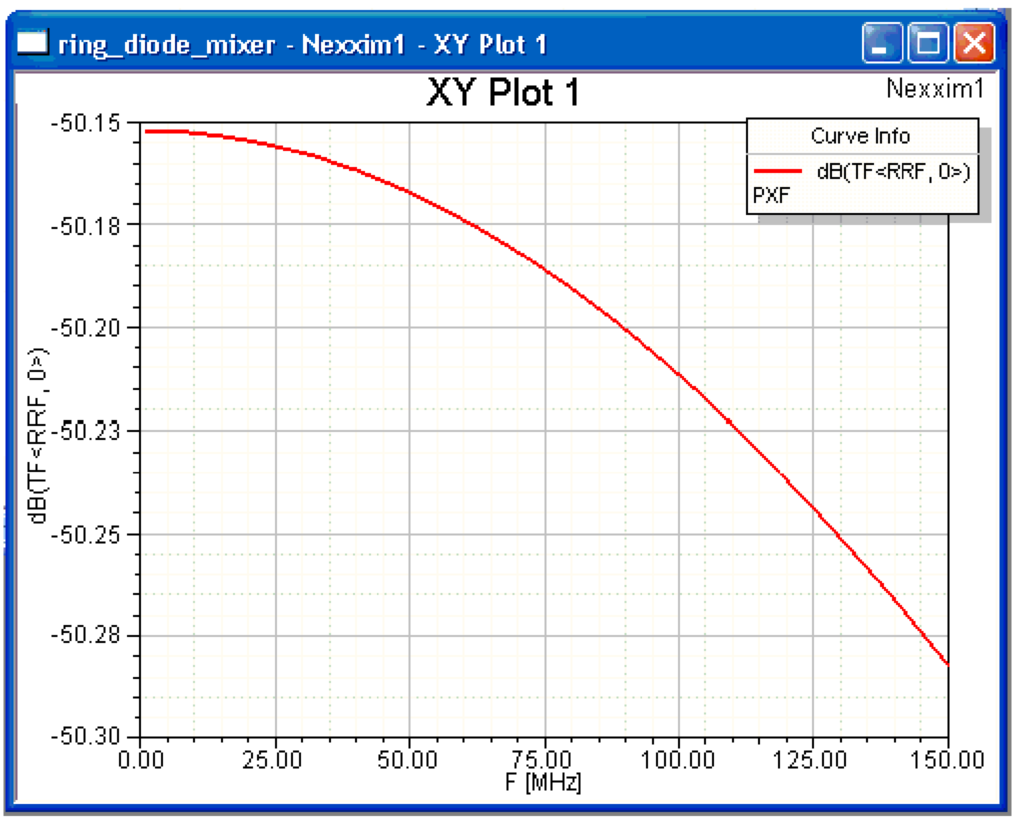
9. Select the **Projects** tab in the Project window. Expand the **Analysis** icon and right-click the **Periodic Transfer Function** solution setup. Select **Browse Netlist** from the menu. The Netlist Editor opens to show the netlist that has been created for the circuit and the solution setup. Here is the solution setup:

```
* end toplevel circuit
.TV_NOISE
+ LIN 20 1000000 150000000
+ PROBE=[v(if)]
+ SS_MAXK=[15]
+ SS_TONES=[100000000]
+ MAXSIDEDAND-[1]
+ SS_ANALYSIS=HB
+ METHOD=HB
+ AUTO_REFINE_SOLUTION=No
+ PXF=1

.end
```

10. Right-click the **Periodic Transfer Function** solution setup and select **Analyze** on the menu. The PXF analysis runs. The Progress window displays a red line to show the progress of the analysis.

When the analysis has completed, the results are available. Compare the PXF analysis result to the Power Conversion Gain (CG) on the TV Noise analysis.

11. Right-click the **Results** icon in the **Project Manager** window and select **Create Standard Report > Rectangular Plot**.
12. The **Report** window opens with the **Trace** tab selected.
    - Select **PXF** on the Solution drop-down.
    - Select **Transfer Function** on the **Domain** drop-down.
    - Select **Transfer Function** in the **Category** list.
    - Select **TF(RRF,0)** in the **Quantity** list.
    - Select **dB** in the **Function** group box.
    - Click **New Report**, then click **Close** to close the **Report** window and generate the report.



13. You can verify that this graph is identical to the Conversion Gain graph on the TV Noise analysis.
14. Close the Project when you are finished viewing the graph.
15. Exit **Electronics Desktop** by selecting **Exit** on the **File** drop-down, or by clicking the X at the upper-right of the window.

For more information, see Nexxim Time-Varying Noise Analysis.

Loadpull Analysis Example

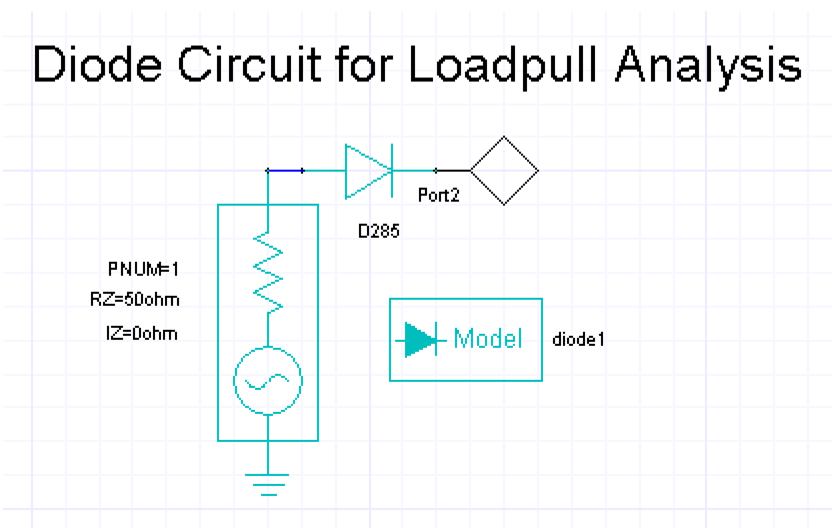
This example demonstrates how to set up and perform a loadpull analysis using a circuit that consists of just a single diode. For more information see Loadpull Analysis.

Open the Loadpull Schematic

Open the Loadpull Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Amplifiers** directory, then select the file **Nexxim_Load_Pull_Schematic.aedt**. Click **Open**.

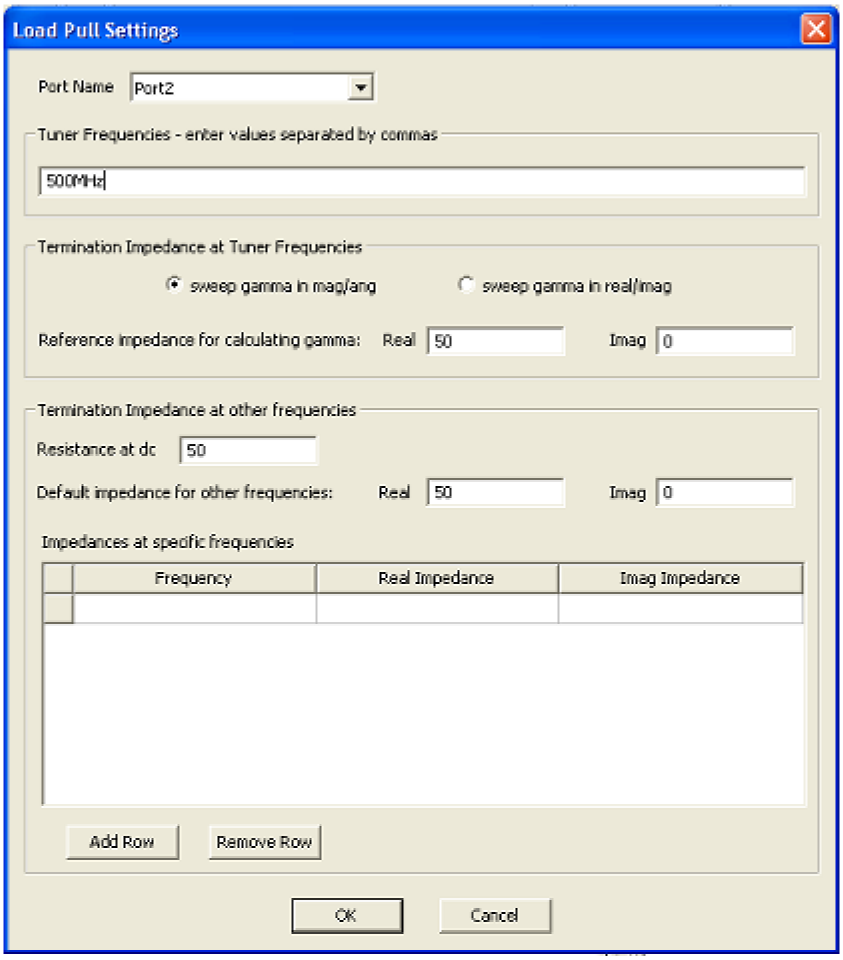
The schematic appears in the design window:



3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

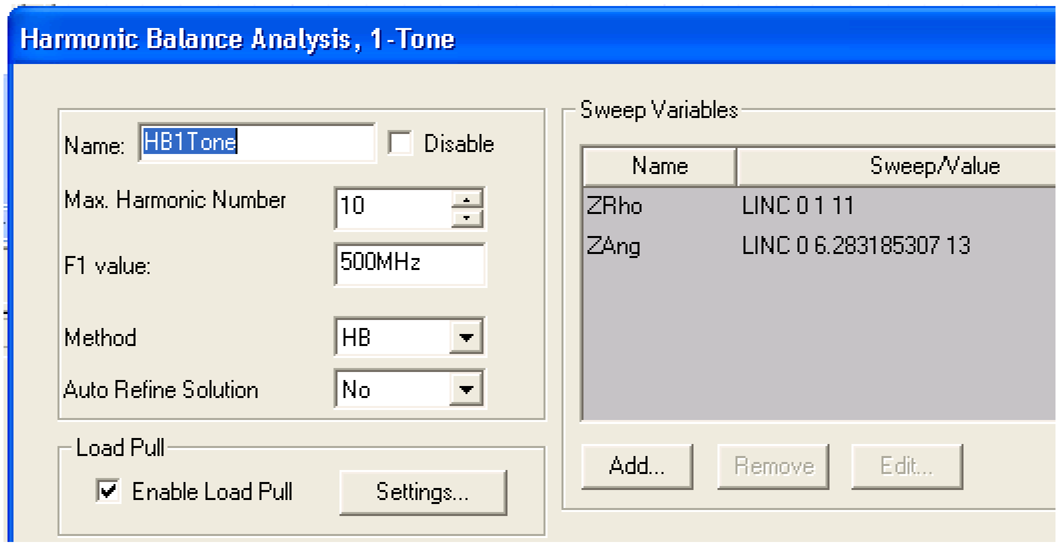
Set Up a Loadpull Analysis

1. Click the **Circuit** drop-down menu on the menu and select **Add Nexxim Solution Setup,** then choose **Harmonic Balance(1-Tone**) to open the **Harmonic Balance Analysis, 1-Tone** setup window.
   - In the upper-left corner of the window, select **10** as the **Max Harmonic Number**.
   - Set the **F1** value to **500MHz**.
   - Verify that the **Method** is **HB** and **Auto Refine Solution** is **No**.
   - Click **Enable Load Pull**. The **Load Pull Settings** window opens:



   - Choose **Port2** on the **Port Name** drop-down menu.
   - Enter **500MHz** as the **Tuner Frequency**.
   - Verify that the termination impedances at the tuner frequency and at frequencies other than the tuner frequency are the defaults (50 Ohm), and that **sweep gamma in mag/ang** is selected.

- Click **OK** to close the **Load Pull Settings** window and return to the **Harmonic Balance Analysis, 1-Tone** setup window. The completed window should look like the following:

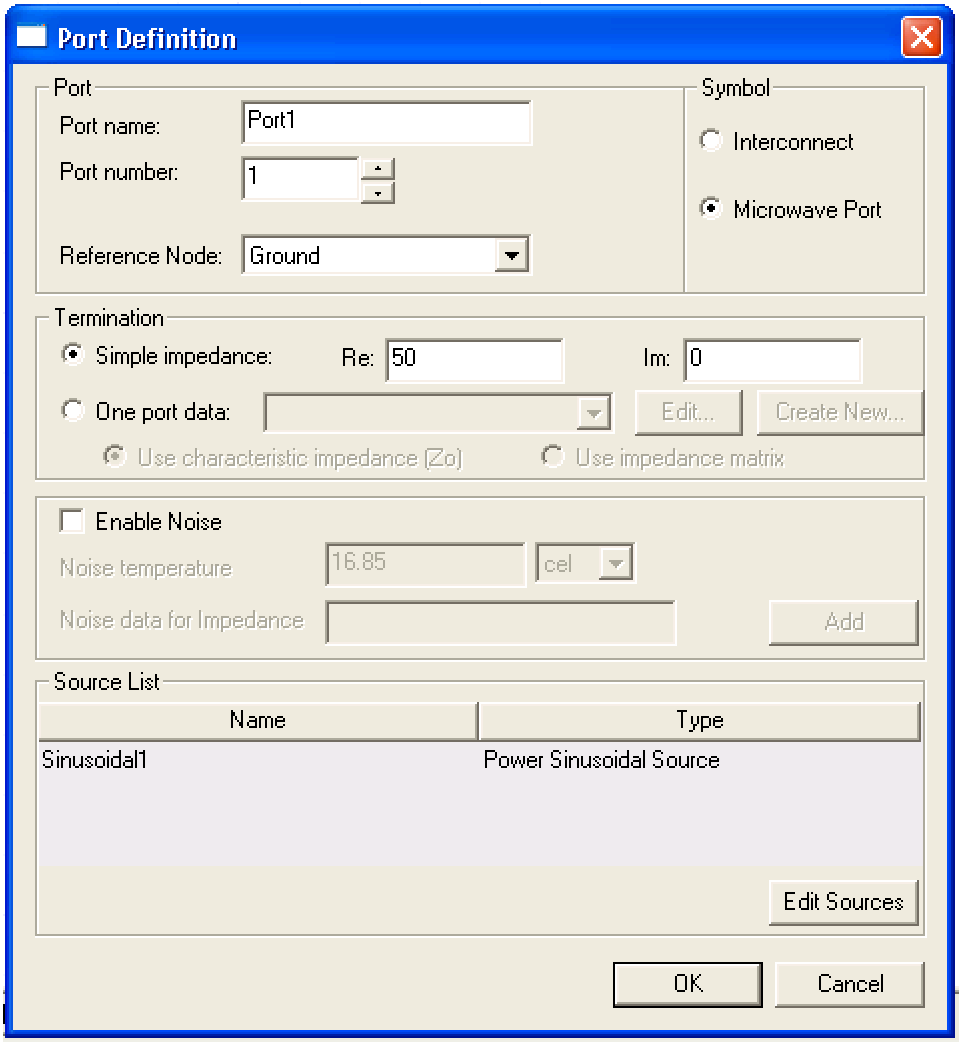


- Note that sweeps have been set up for **ZRho** and **ZAng**. These variables set the output load. Sweeping the output load generates the loadpull data.
- Click **OK** to close the **Harmonic Balance, 1-Tone** setup window.

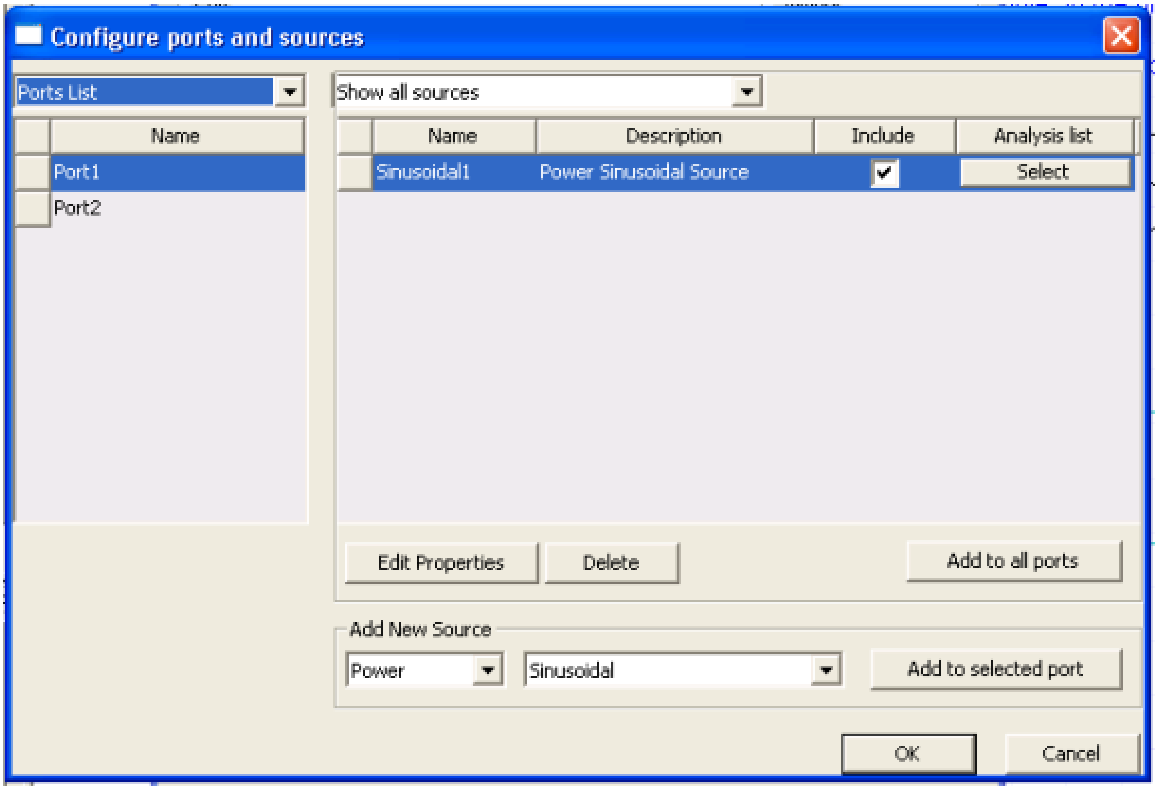
Check the Port1 Source

Nexxim allows you to configure any number of ports and sources, and associate each source to run only for certain analyses. This example has only one source and only one analysis.

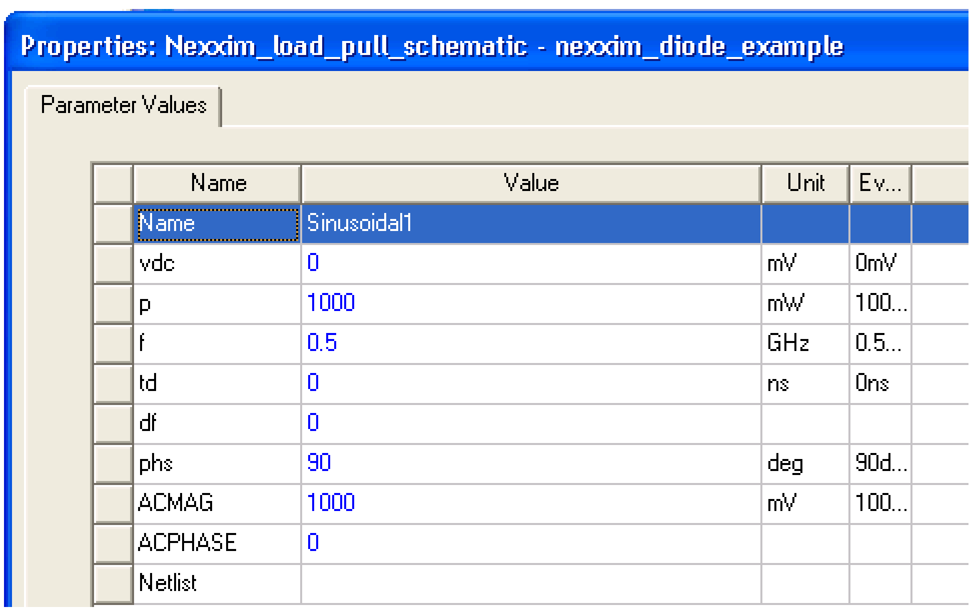
1. Right-click Port 1 (PNUM=1), and select **Edit Port** on the menu. The **Port Definition** window opens:



2. Port 1 has been set up as a sinusoidal power source. To review the setup, click **Edit Sources** to open the **Configure ports and sources** window.

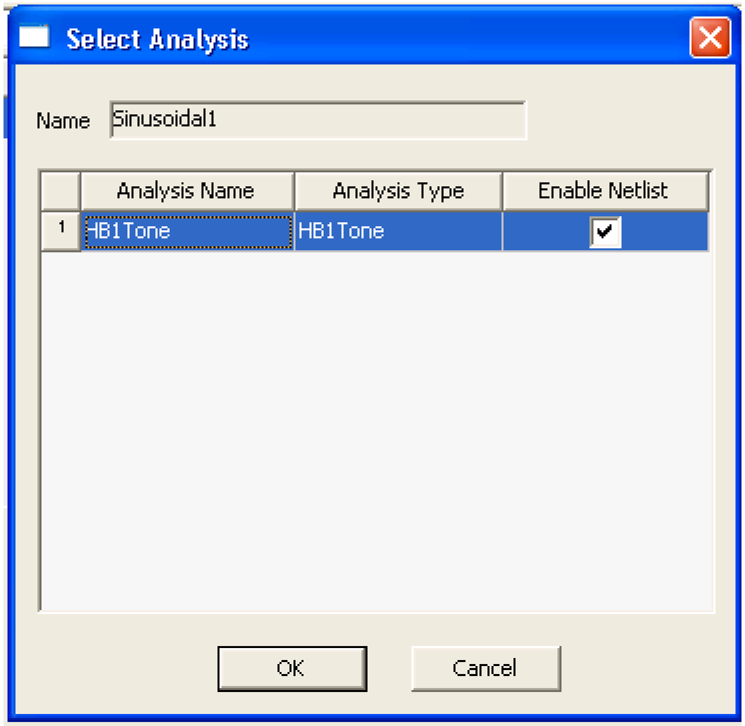


3. To view the source definition, click Edit Properties. The **Properties** window opens:



- Verify that the frequency (property **f**) is 0.5 GHZ (500MHz).
- Click **OK** to close the **Source Selection** window and return to the **Port Definition** window.

4. Click **OK** to close the **Properties** window and return to the **Configure ports and sources** window.
5. Click **Select** in the Analysis list column. The **Select Analysis** window opens:

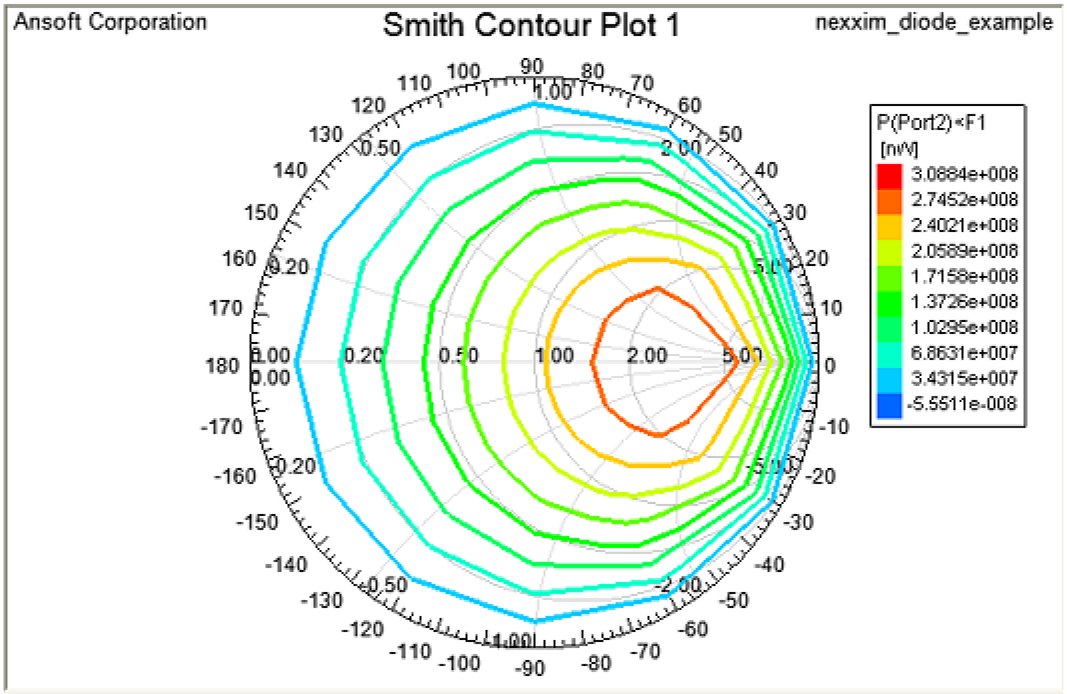


Verify that the source is enabled for the harmonic balance loadpull analysis netlist. Click **OK** to return to the **Configure ports and sources** window. Click **OK** to return to the **Port Definition** window. Click **OK** to close the **Port Definition** window.

Run the Loadpull Analysis and View the Results

1. Right-click the **HB1Tone** setup icon and select **Analyze**. The Progress window displays a red line to show the analysis as it performs the sweeps.
2. When the analysis is complete, right-click on the **Results** icon in the **Project Manager** window and select **Create LoadPull Report > Smith Contour Chart**.
3. The **Report** window opens with the **Trace** tab selected.
   - Verify that **HB1Tone** is the **Solution** and **LoadpullContour** is the **Domain**.
   - Select **Power** as the **Category**, **P(Port2)<F1>** as the **Quantity**, and **<None>** as the **Function** (loadpull results are automatically set in watts).
   - Click **New Report**, then click **Close**.

4. The Report window opens with a Smith Chart showing the contour plots of the sweep results:



5. Close the Report window and the Project when you have finished viewing the graph.

Envelope Analysis Example

This example demonstrates how to set up and perform an envelope analysis using an amplifier circuit. For more information see Envelope Analysis.

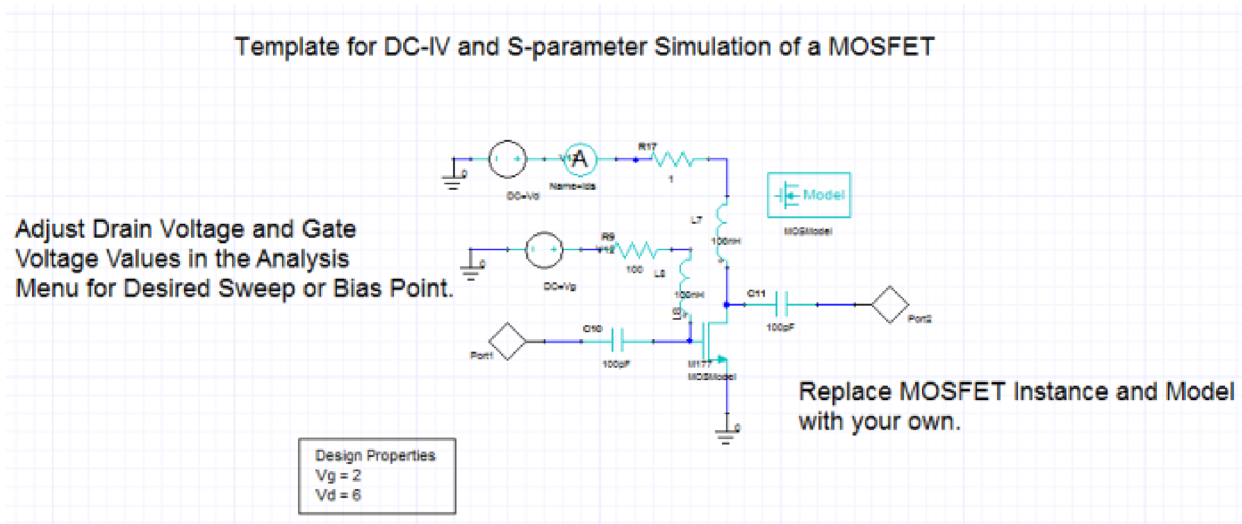
Open the Amplifier Netlist

Envelope analysis allows us to evaluate the performance of the amplifier when it is stimulated by the IQ-data source.

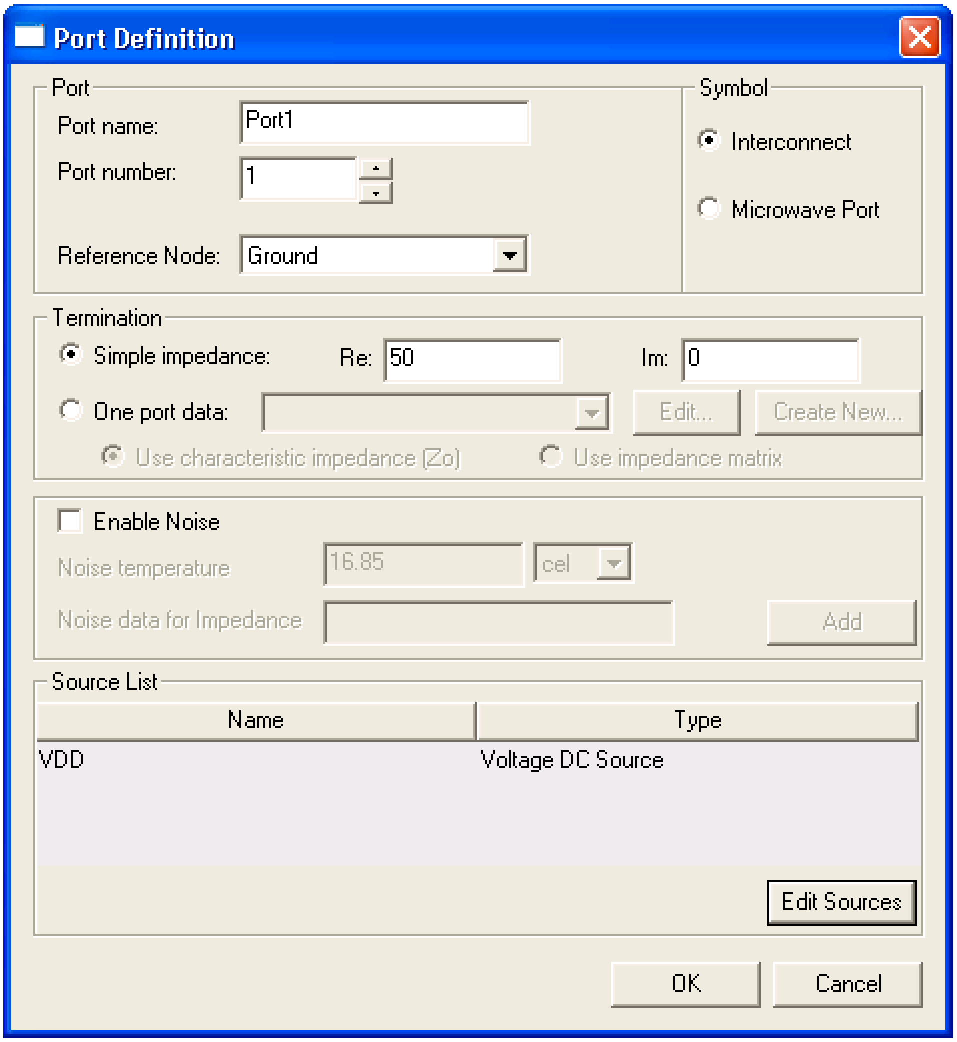
Open the Amplifier project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Amplifiers** directory, then select the file **DifferentialAmplifier_Envelope.aedt**. Click **Open**.

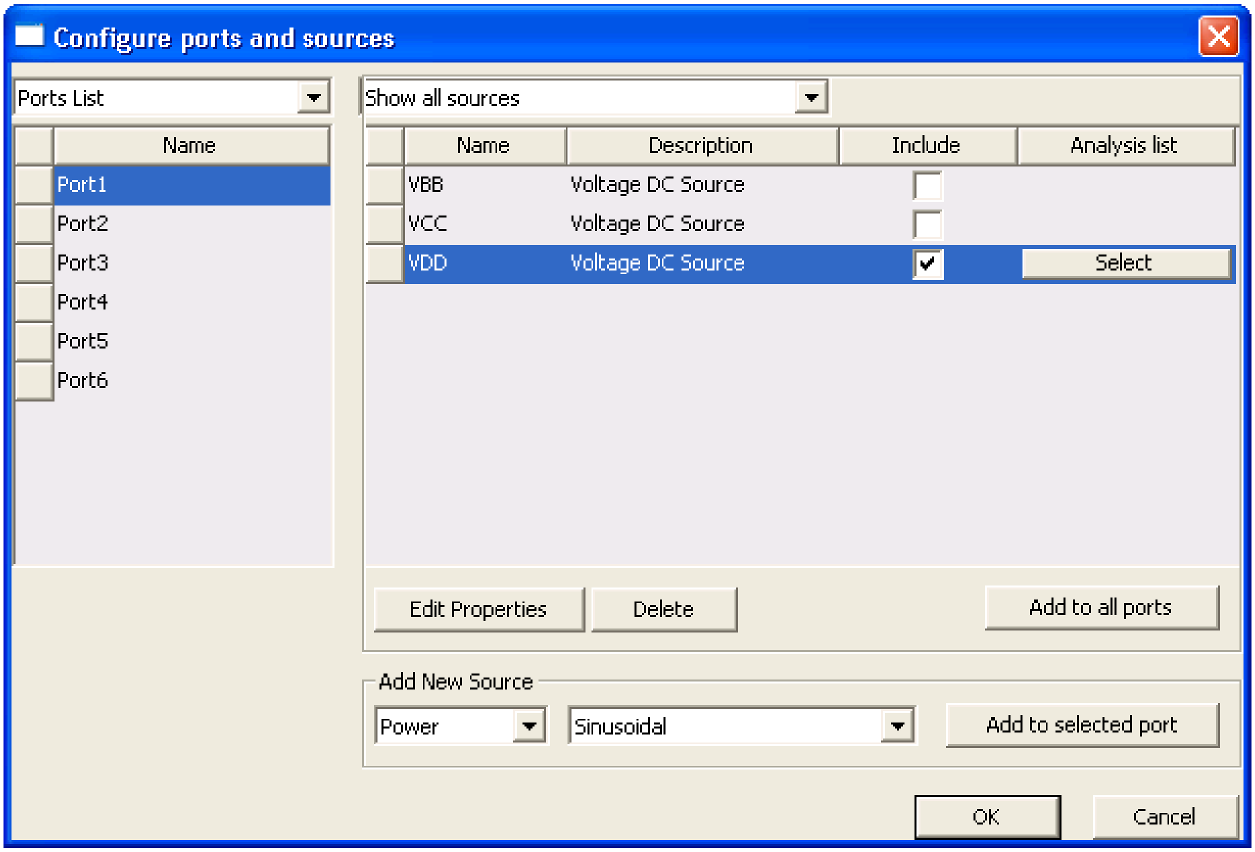
The circuit opens in the **Schematic Editor** window.



3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.
4. The differential amplifier consists of two transistors and seven resistors. The setup includes two VCVS sources and the IQ data source.
5. Select **Port 1** (or any port), right-click it, and select **Edit Port** to open the Port Definition window.

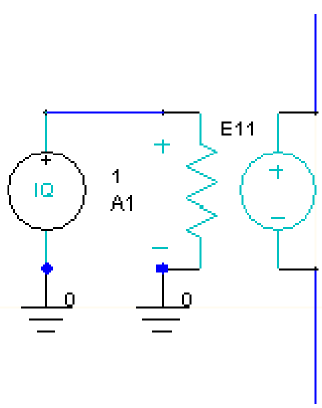


6. The **Source List** shows the voltage source connected to the port. Click **Edit Sources** to view the configuration window:

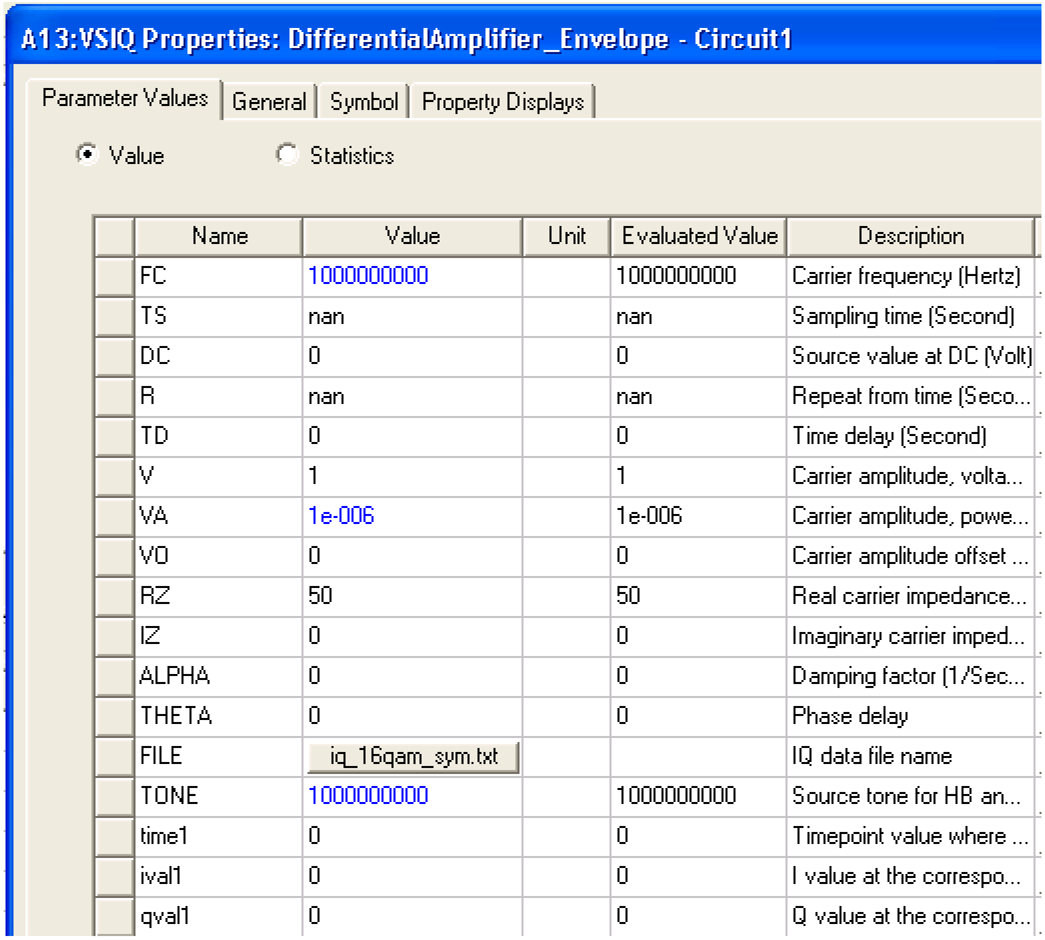


The window shows that six ports and three DC sources are defined. Click the ports to verify that Port 1 receives VDD (-3V), Ports 2 and 3 receive VBB (1.5V), Ports 4 and 5 receive VCC (15V). Port 6 is the output.

7. Here is the input portion of the schematic with the IQ Source:



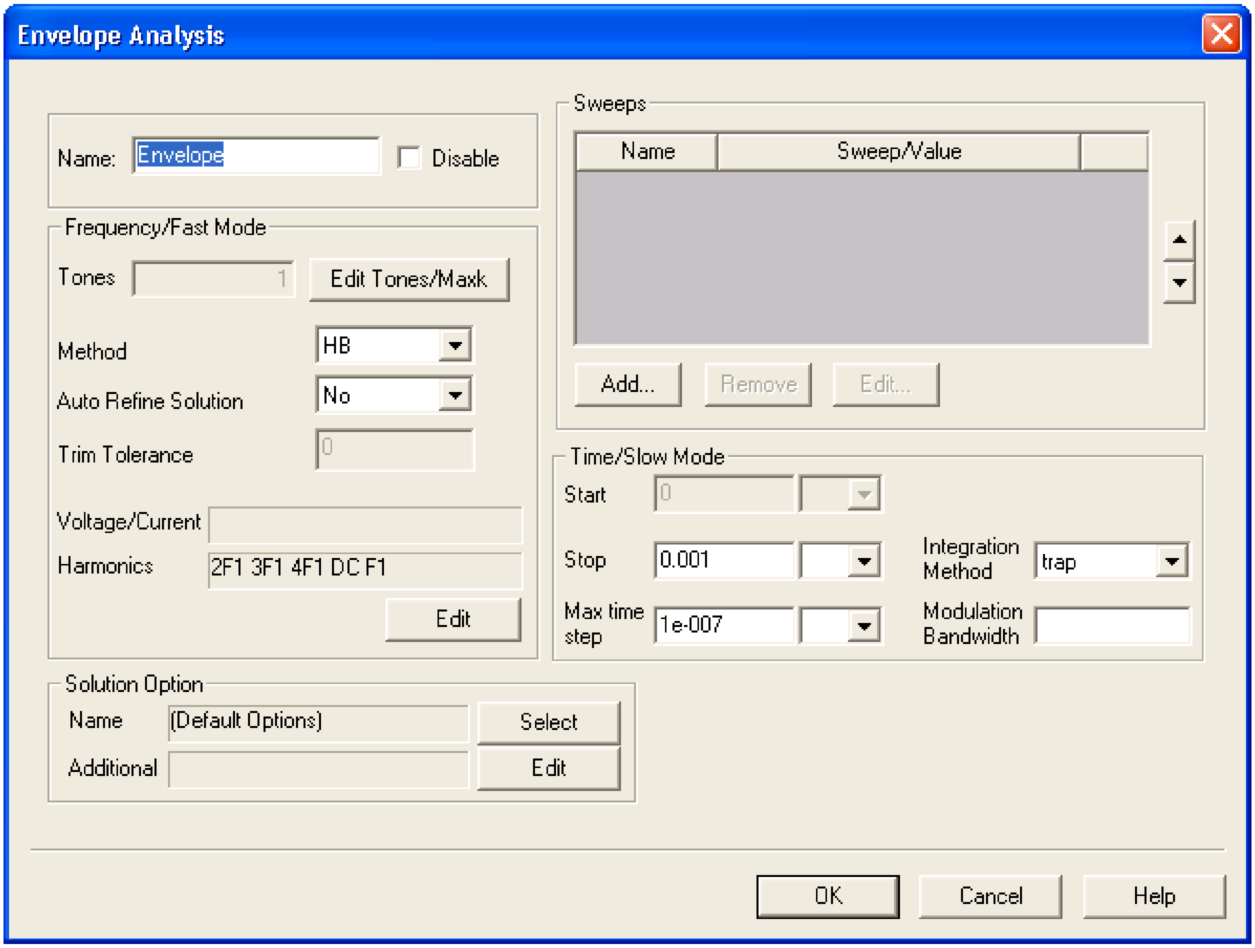
8. Select the IQ source and display its properties:



Name	Value	Unit	Evaluated Value	Description
FC	1000000000		1000000000	Carrier frequency (Hertz)
TS	nan		nan	Sampling time (Second)
DC	0		0	Source value at DC (Volt)
R	nan		nan	Repeat from time (Seco...
TD	0		0	Time delay (Second)
V	1		1	Carrier amplitude, volta...
VA	1e-006		1e-006	Carrier amplitude, powe...
VO	0		0	Carrier amplitude offset ...
RZ	50		50	Real carrier impedance...
IZ	0		0	Imaginary carrier imped...
ALPHA	0		0	Damping factor (1/Sec...
THETA	0		0	Phase delay
FILE	iq_16qam_sym.txt			IQ data file name
TONE	1000000000		1000000000	Source tone for HB an...
time1	0		0	Timepoint value where ...
ival1	0		0	I value at the correspo...
qval1	0		0	Q value at the correspo...

The carrier frequency FC is set to 1GHz, and the TONE parameter sets harmonic balance to use this frequency. The FILE parameter references the IQ data file, **iq_16qam_sym.txt**, in the Examples directory. You can open the data file with a text editor to view the time and data points.

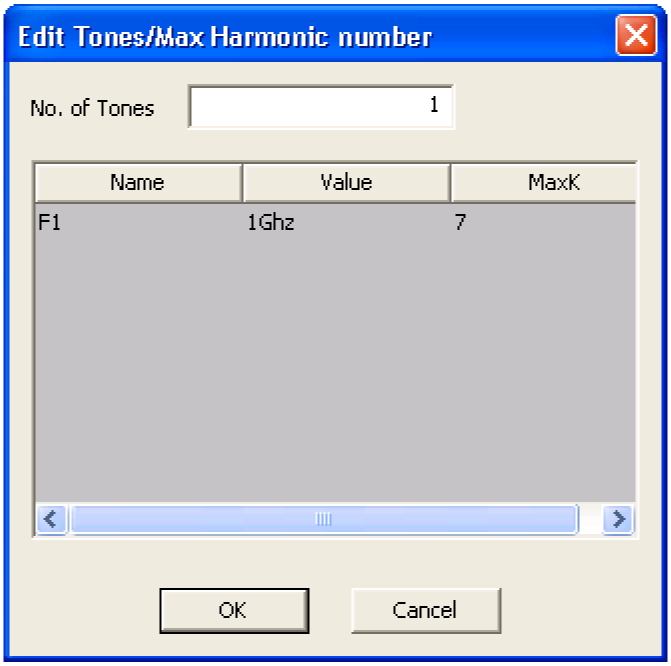
9. Expand the Analysis icon and double-click the Envelope analysis:



The setup implements the following parameters:

- The Method is **HB** (harmonic balance)
- The **Auto Refine Solution** option is **No**.
- The stop time is **0.001** seconds. This is less than the maximum time point in the IQ data file, **iq_qam_data.txt**.)
- The maximum timestep is set to **1e-007** seconds. This timestep is small enough to capture the envelope waveforms.

10. Click **Edit Tones/Maxk**:



- The number of tones is **1**, and the **Frequency** is **1GHz**.
- The maximum harmonic number **MaxK** is **7** harmonics

11. Click **OK** to close the **Edit Tones** window, then click **OK** again to close the **Envelope Analysis** window.

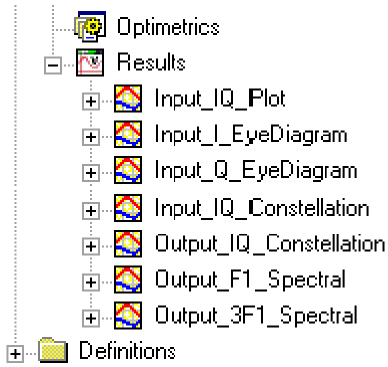
Perform an Envelope Analysis

Right-click the **Envelope** analysis and select **Analyze** on the menu. The Progress window shows the progress of the dual simulations: transient analysis for the Time/Slow Mode and harmonic balance for the Frequency/Fast Mode.

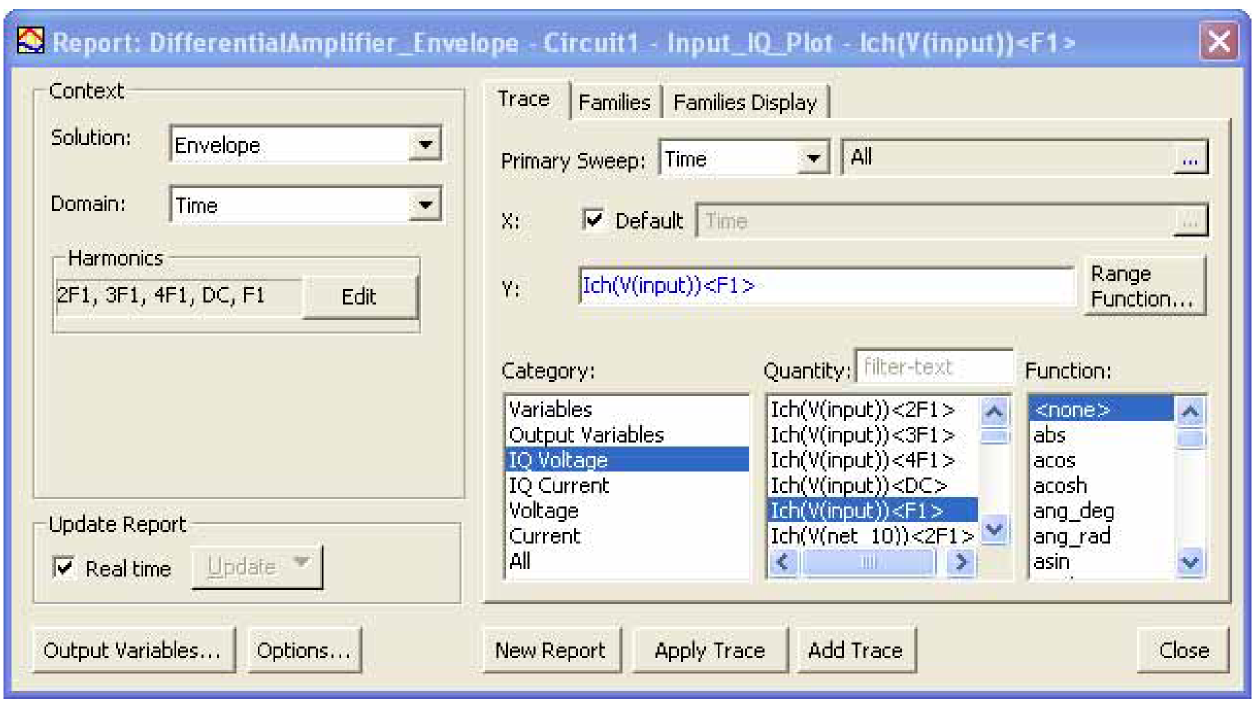
View the Results of the Analyses

When the analyses have completed, the reports that have been set up become valid.

1. From the **Project Manager** window, expand the **Results** icon:

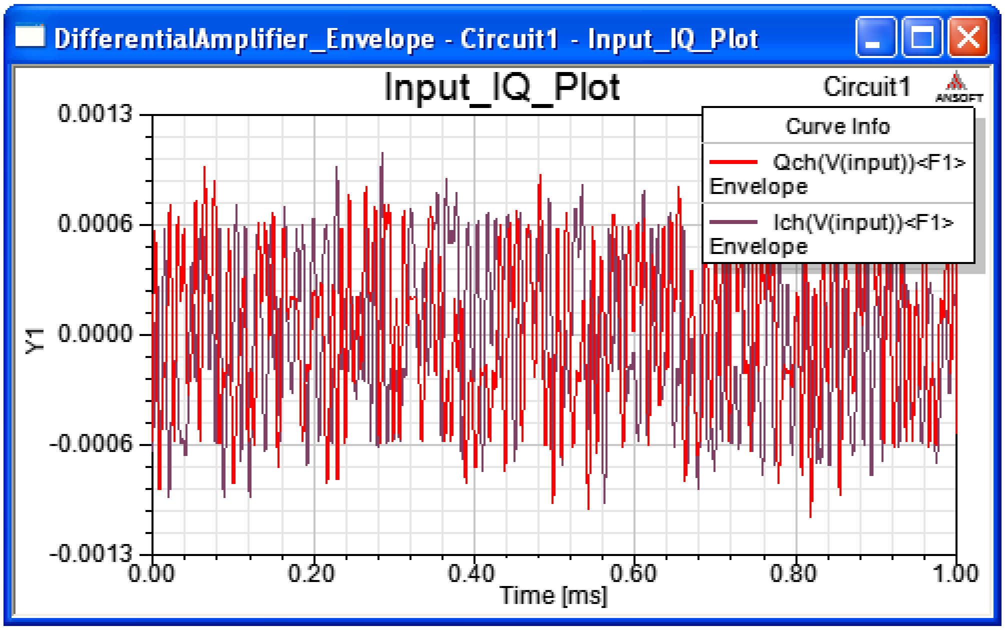


2. Right-click the **Input_IQ_Plot** report and select **Modify Report** to open the setup for the Rectangular Plot report:

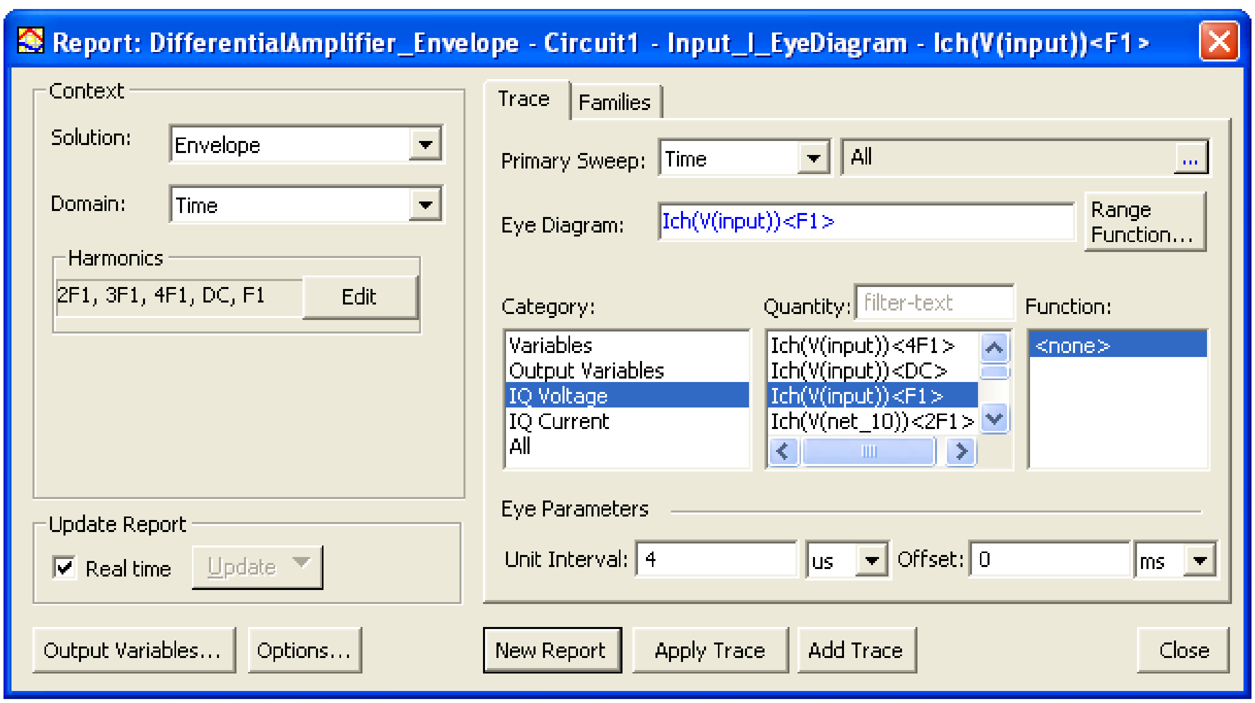


The IQ source drives input node **input** in our circuit. This window shows the setup for the In-quadrature (I) trace. The Q trace has a similar setup.

3. Double-click the **Input_IQ_Plot** report to view the input I-Q data against time:

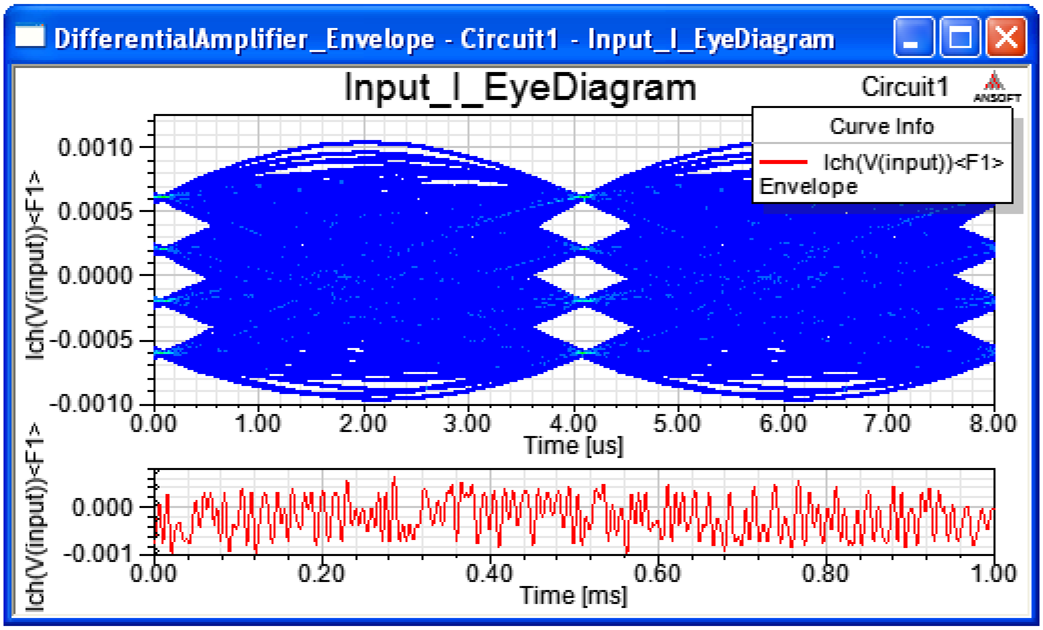


4. Right-click the **Input_I_EyeDiagram** report and select Modify report:



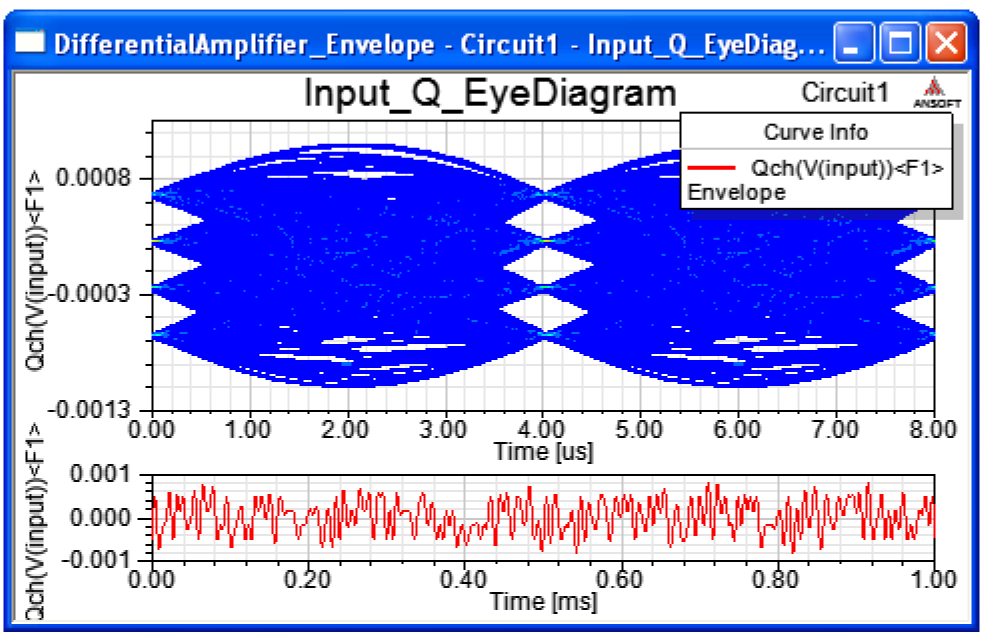
- The Unit Interval is equal to the symbol rate of 4µs.

5. Double-click the **Input_I_EyeDiagram** report to view the I (in-quadrature) data as an eye diagram:



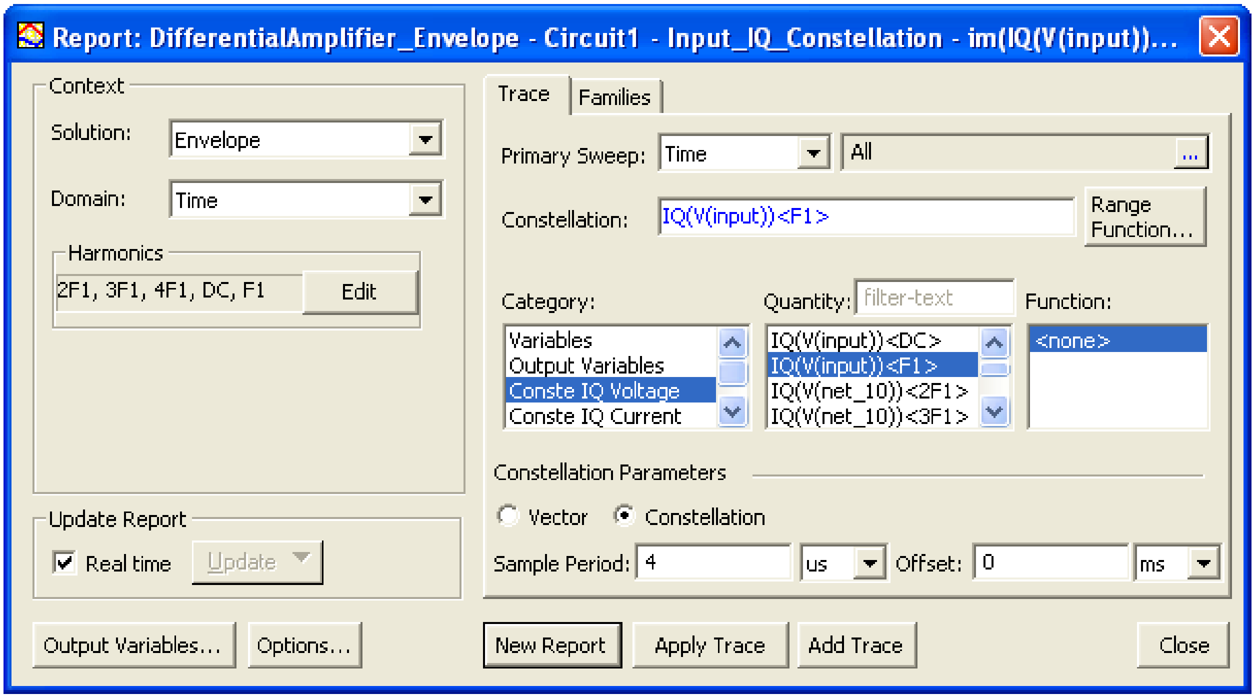
- The diagram illustrates the levels of amplitude (0.0006V, 0.0002V, -0.0002V,and -0.0006V) in the I-channel.

6. Right-click the **Input_Q_EyeDiagram** report to view the Q (quadrature) data as an eye diagram:



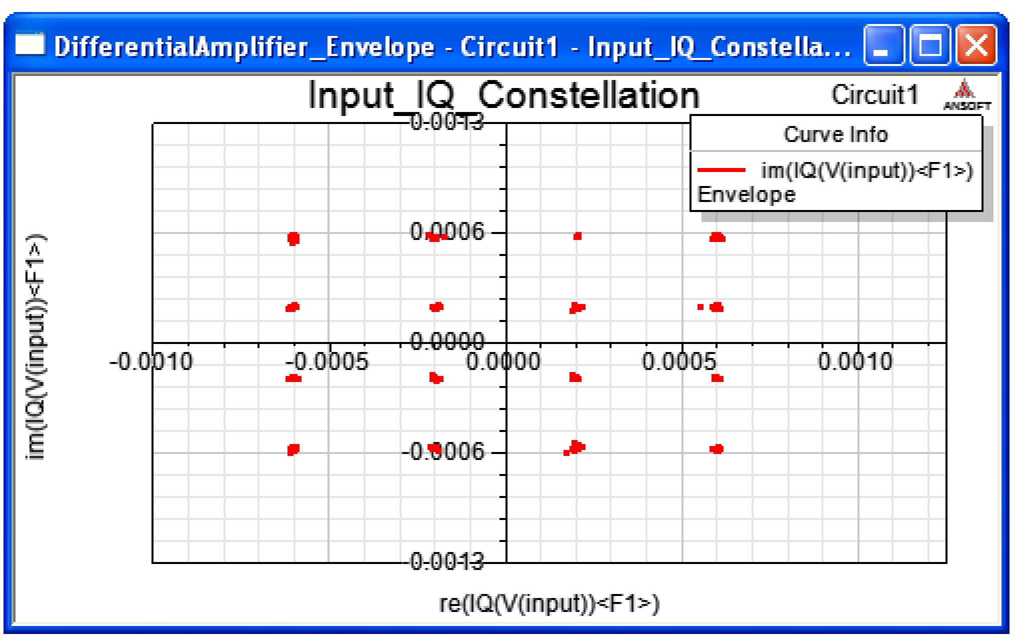
- This eye diagram was also created using a Unit Interval of 4µs.
- The diagram illustrates the levels of amplitude in the Q-channel, identical to those for the I-channel (0.0006V, 0.0002V, -0.0002V,and -0.0006V).

7. Right-click the **Input IQ Constellation** report:



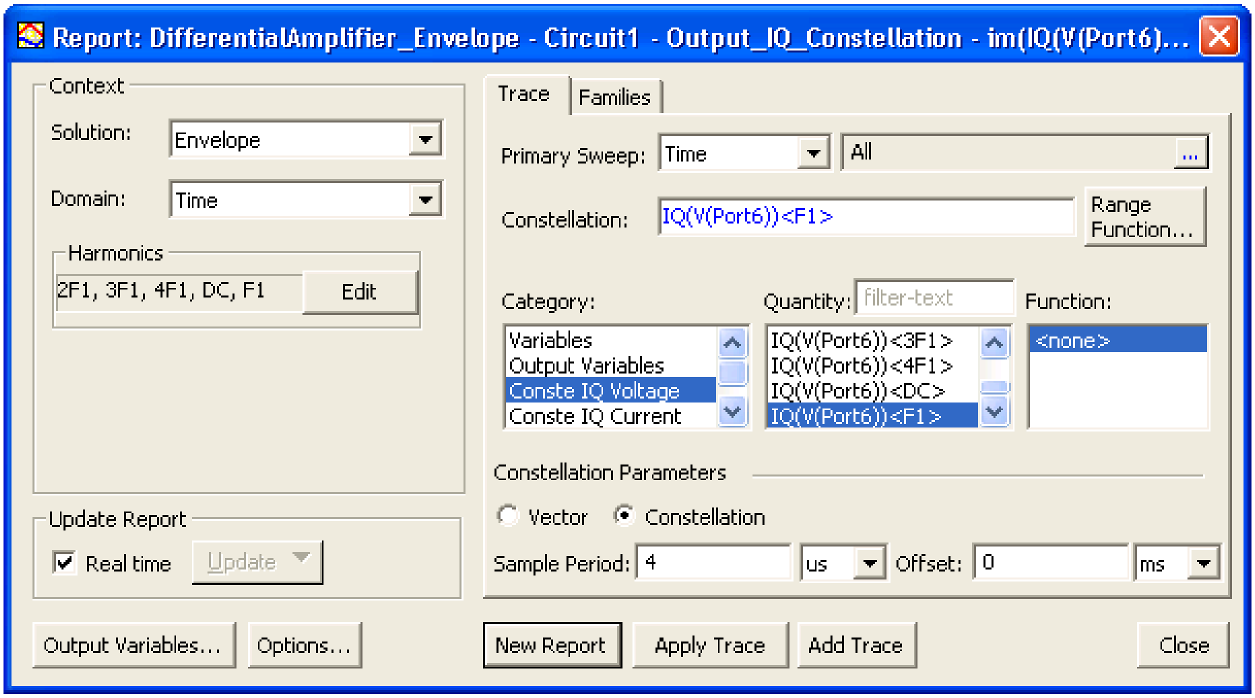
- In the **Constellation Parameters** group box, **Constellation** has been selected rather than **Vector**. This choice displays just the constellation points and not the transition vectors. The **Sample Period** is the same as the unit interval (4µs)

8. Double-click the **Input_IQ_Constellation** report to view the input IQ data plotted as a constellation chart:



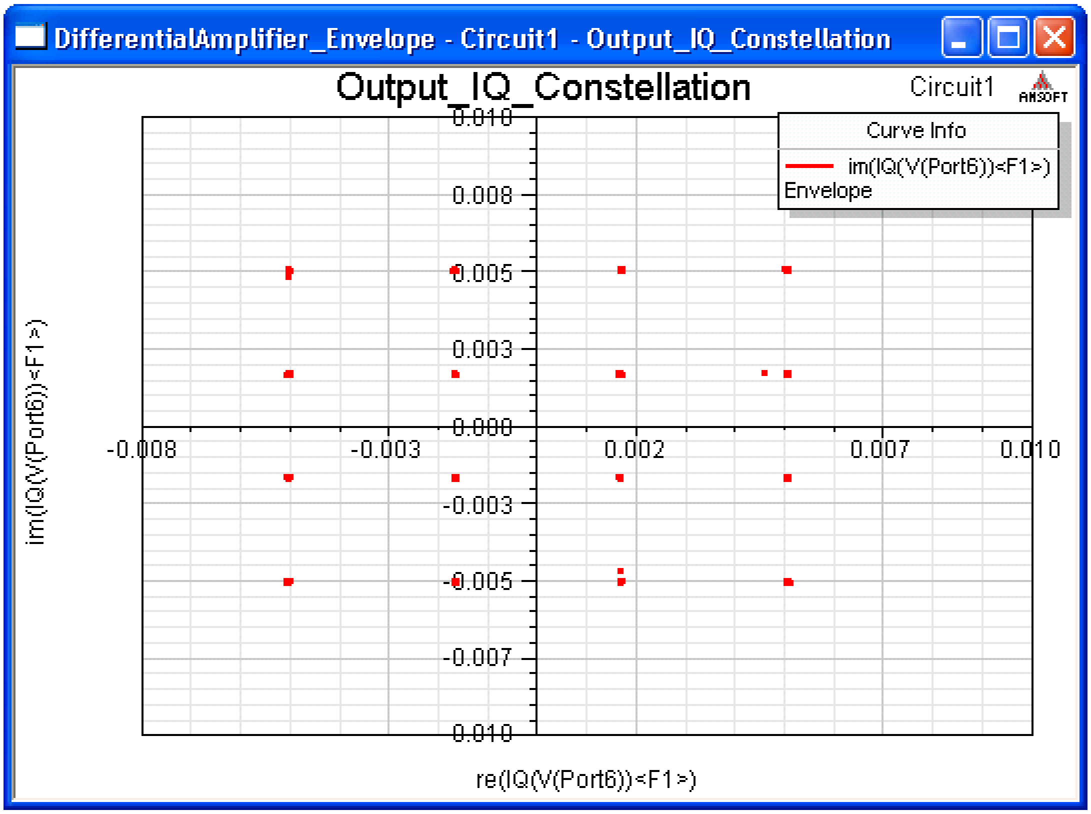
- The constellation diagram illustrates the clustering of the input I (real) and Q (imaginary) data points.

9. Right-click the **Output_Constellation** and select Modify Report to open the definition:



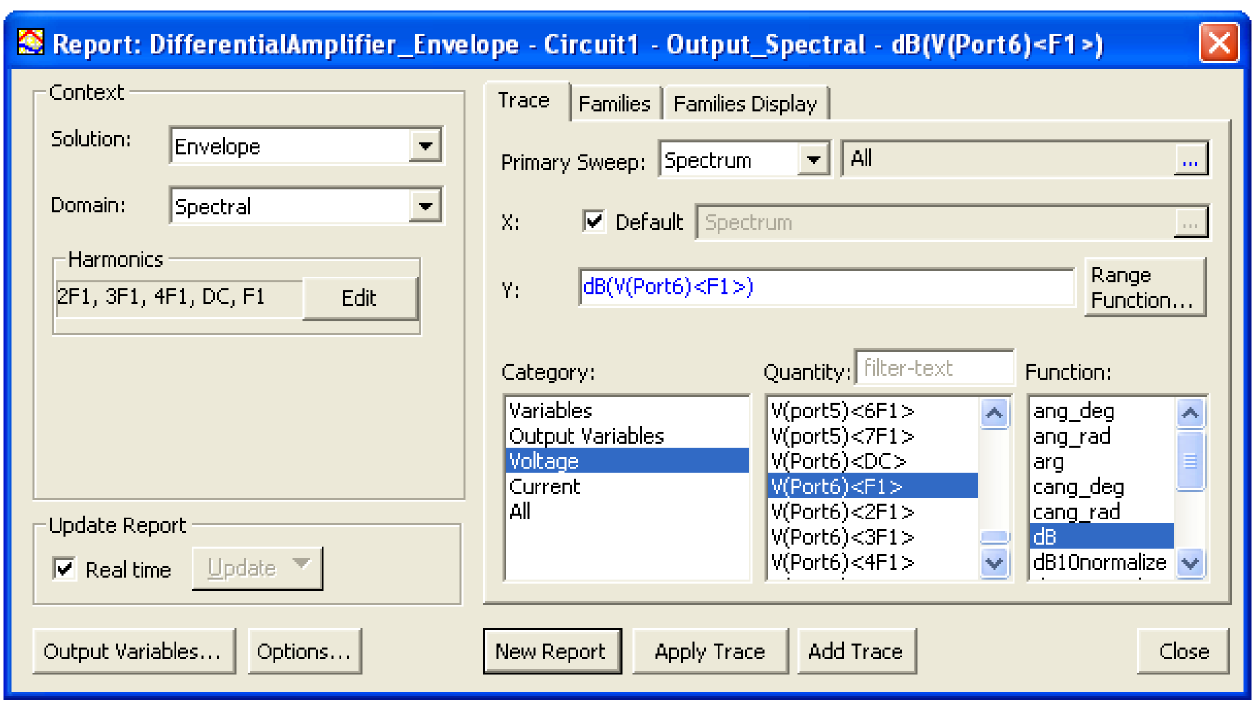
- The output node on our circuit is **Port6**. The sample period is 4μs as before.

10. Double-click the **Output_IQ_Constellation** report to view the output data as a constellation chart:



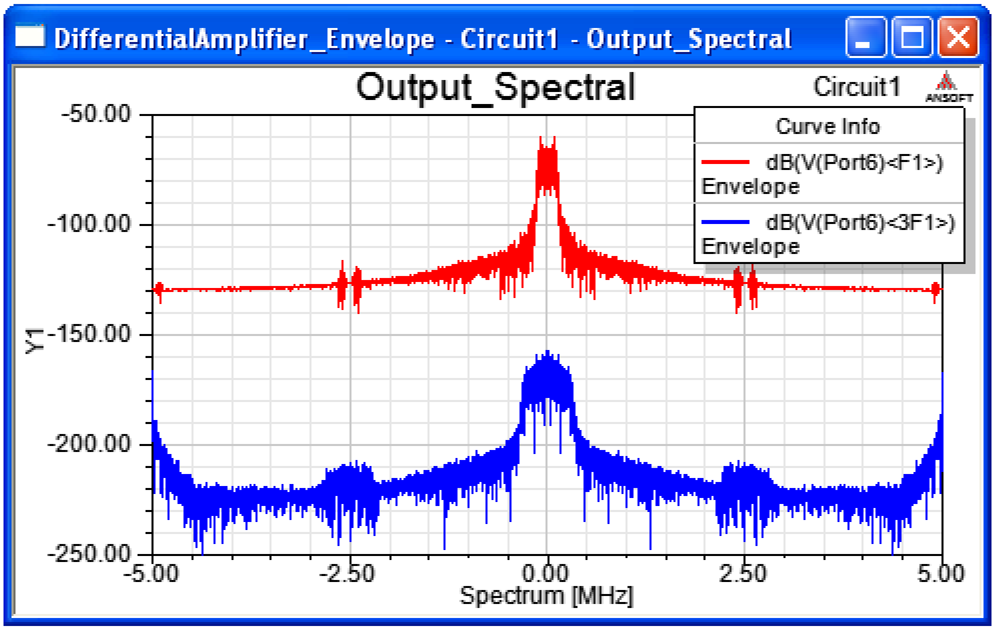
- The chart is not rotated, so no phase distortion is present.

11. Right-click the **Output_Spectral** report:



This window shows the setup for the F1 spectrum. The F3 setup is essentially the same.

12. Right-click the **Output_Spectral** report to view the spectral result for the output at the first harmonic (F1) and the third harmonic (F3):



- This complex baseband representation is centered at 0Hz (DC).
- The spectrum at 3GHz (3F1) is approximately 100dB lower than the spectrum for the fundamental 1Ghz (F1).

Nexsys Timestep Example

This topic presents two versions of a loop filter that combines Nexxim elements with components on the System simulator component library.

The point of this example is to illustrate that timesteps in Nexsys depend on whether the source that is driving the circuit is on the System library or on the Nexxim library.

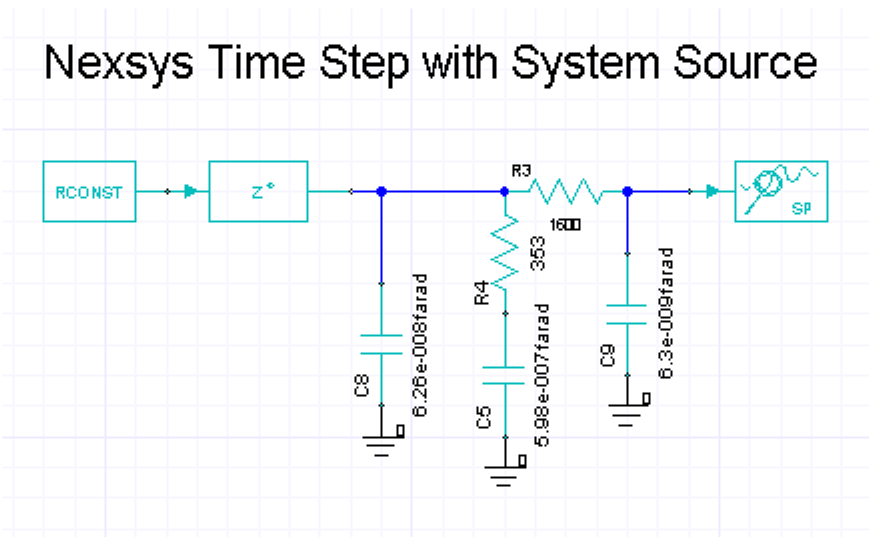
- When all sources are on the System library, the timestep is fixed and is taken from the sample rates in the System sources, and any timestep set in the transient analysis setup (Step and Stop entries) is ignored.
- When the source is an interface source, the timestep is variable, with a Step and Stop taken on the transient analysis setup.

First Loop Filter Schematic Analysis

Open the First Loop Filter Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. rom the **Examples** directory, select **Circuit** > **RF Microwave** > **System**. Then double-click **Loopfilter_1_NX.aedt**, or select **Loopfilter_1_NX.aedt** and click **Open**.

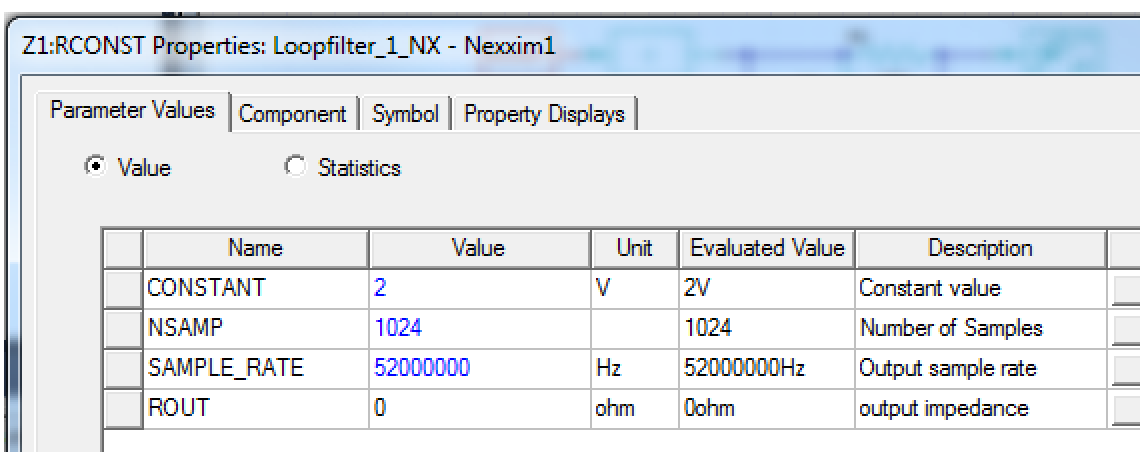
The filter schematic appears in the design window:



3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

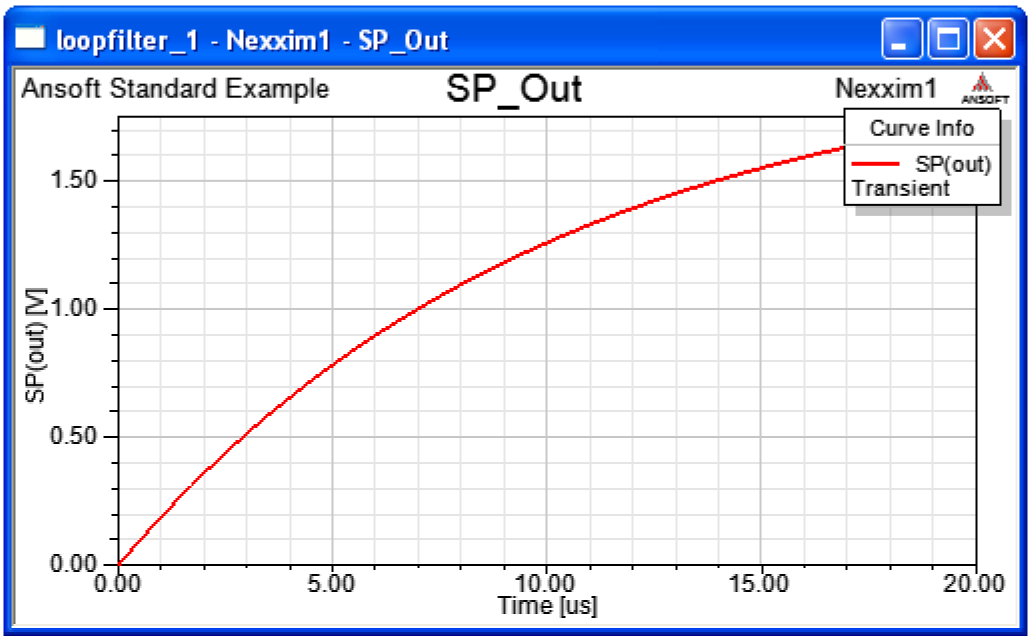
The circuit contains a source (RCONST) and a delay element (Z^{-D}) on the System simulator component library and discrete resistors and capacitors on the Nexxim component library. The output probe (SP) is also on the System library.

4. Click the RCONST element to display its property settings:



These parameters control the time-domain simulation in Nexxim, overriding the times specified for the Transient analysis.

5. Expand the **loopfilter_1** project tree, and double-click the Transient analysis icon. The times are Step=0.1ns and Stop=10ns, but *these are overridden by the System source*
6. Right-click the Transient analysis icon and select **Analyze** on the menu. The Progress window displays a red line to show the analysis as it runs.
7. Click the Results icon and double-click on **SP_Out** to see the results:



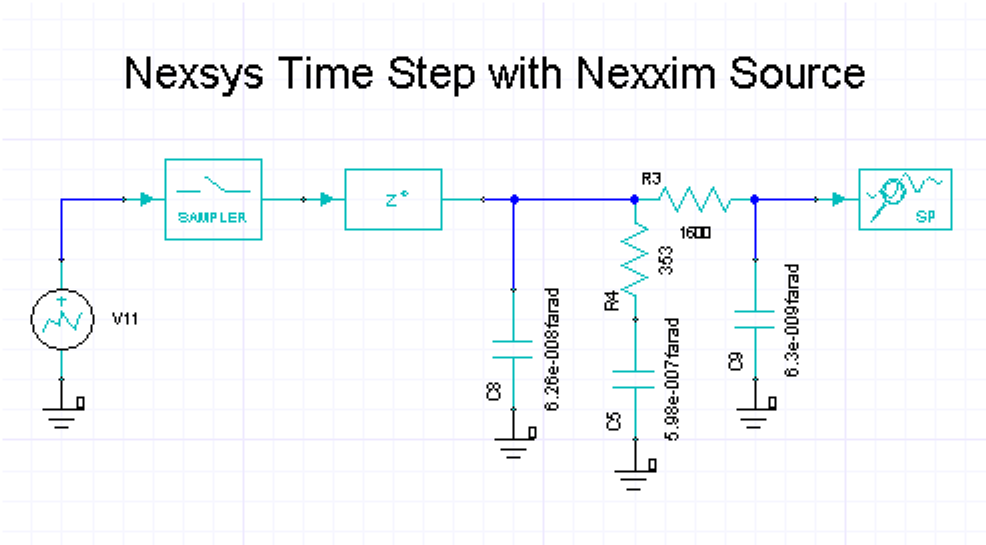
This graph shows the filtered samples over a time period of 20 microseconds (1024 samples at a sampling rate of 52e6 Hz).

Second Loop Filter Schematic Analysis

Open the second loop filter project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open **Circuit**, then **RF_Microwave**, then **System**, then select the file **Loopfilter_2_ NX.aedt**. Click **Open**.

The filter schematic appears in the design window:



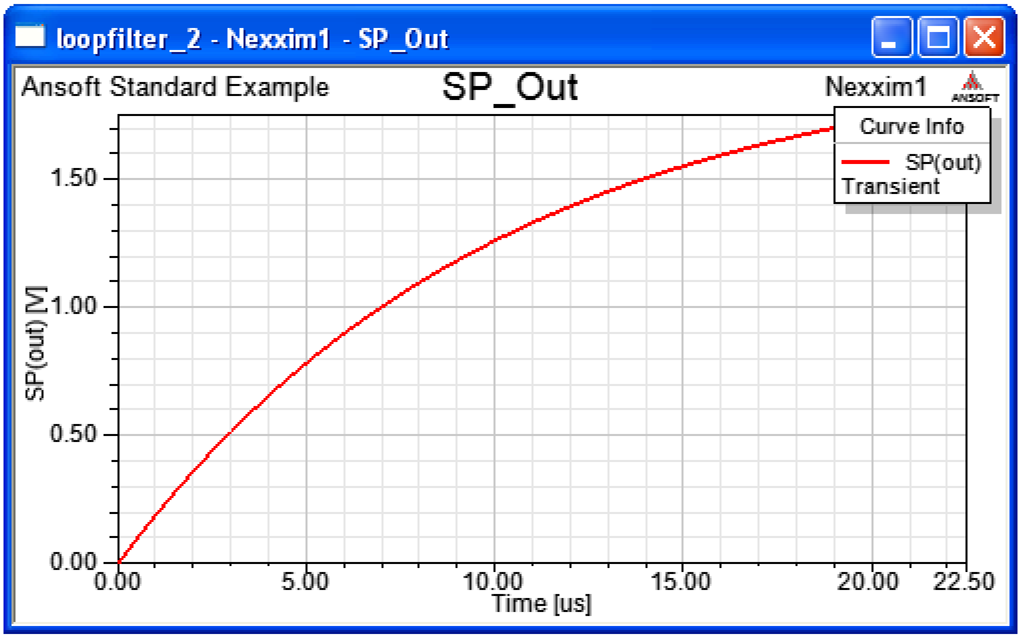
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

The System **RCONST** sampled source has been replaced with a Nexxim PWL source (V11) and a System Sampler element (SAMPLER). The interface source has been set to provide a constant 2V waveform, and the SAMPLER element samples at the 52MHz sample rate.

> **Note:**
>
> The SAMPLER element is required for a successful Nexxim simulation of this circuit. Without the sampler, Nexxim reports an error.

4. Expand the **loopfilter_2** project tree, and double-click the Transient analysis icon. Now the times are Step=0.1ns and Stop=20µs, and *these values now affect the simulation*. The Nexxim timesteps vary, with an initial step size of 0.1ns.
5. Right-click the Transient analysis icon and select **Analyze** on the menu. The Progress window displays a red line to show the analysis as it runs.

6. Click the Results icon and double-click on **SP_Out** to see the results:



Now the time of 20 microseconds is derived from the Transient analysis setup. The result is the same as the previous one.

Nexsys MPSK Example

This example displays the ability of Nexsys to model the flow in a complete communications design. The circuit consists of baseband and RF transmitters, a noisy channel, and RF and baseband receivers. Probes added to the circuit allow us to view the signal as it proceeds.

Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Navigate to **Electronics Desktop** > **Circuit** > **RF_Microwave** > **System** and select **MPSK_System_NX.aedt**. Then click **Open** to open the schematic in the design window.
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

The following topics review the operation and function of this design and show some of the results of analysis.

MPSK Netlist Parameters

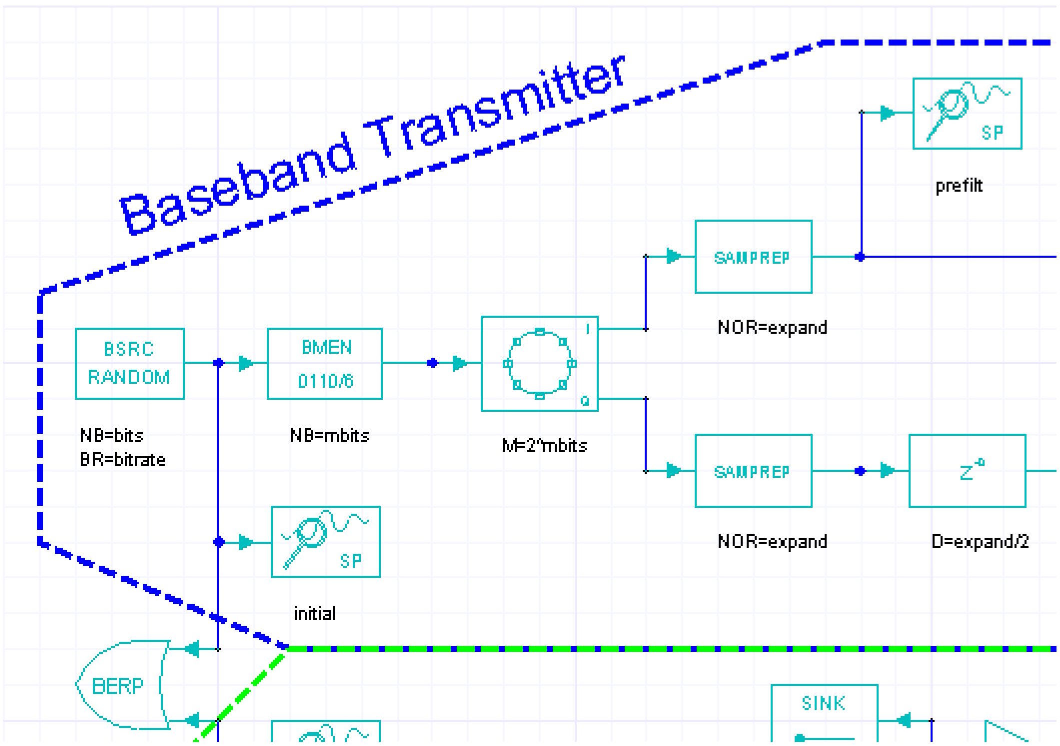
In the menu, select **Circuit > Browse netlist**. Note the netlist parameters defined near the top of the netlist.

```
* begin toplevel circuit

.param bits=1024
.param bitrate=2000000
.param mbits=2
.param expand=8
.param fchannel='bitrate/mbits/2'
.param flength='expand*8'
.param fcutoff='bitrate/mbits*4'
.param fbeta=0.35
.param channel_snr=60
```

The property settings of several components in the design are defined with these netlist parameters, to make it easy to experiment with different values.

Baseband Transmitter Schematic

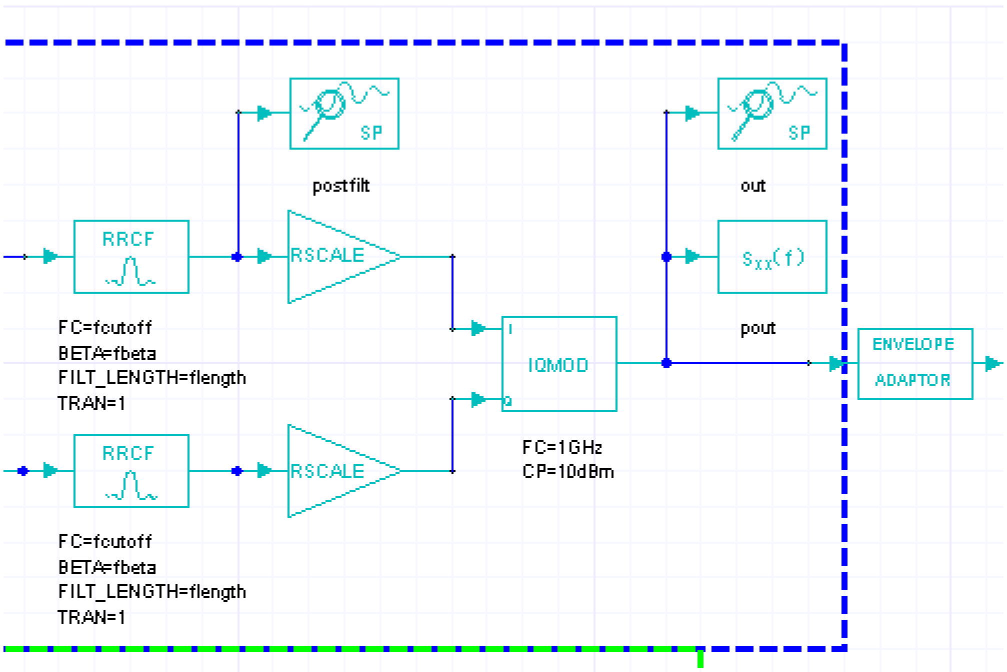
The baseband transmitter is at the upper-left of the schematic. The baseband transmitter stage takes an input bit stream, encodes it into symbols, converts the symbols using QPSK and I/Q modulation, and outputs the modulated signal upshifted in frequency. Here is the left side of the Baseband transmitter schematic:



This part of the Baseband transmitter contains the following elements on the System simulator component library:

- A pseudorandom source. The pseudorandom dataset is a good approximation to a compressed set of transmission data. The number of bits to be generated (NB) is set to the parameter **bits** =1024. The bitrate (BR) is set to the parameter **bitrate** =2000000 (2MHz).
- A Binary-to-M-ary encoder. The number of bits to group into one symbol (NB) is set to the parameter **nbits** =2. The bitstream is encoded into four symbols: 0, 1, 2, and 3.
- A phase-shift keying modulator. The order M is set to 2^**nbits** =4, or QPSK. This element divides the signal into its real and imaginary parts, so now there are two parallel signal paths.
- Symbol repeaters on both paths. The number of repeats is set to parameter **expand** =8. The sampling rate is likewise multiplied by 8. Oversampling makes the waveform appear more continuous to the time domain simulator.
- A delay element on the lower path, adding a delay of **expand** /2=4 symbols to the imaginary data.
- A system probe, **prefilt**, is inserted to enable you to view the real part of the signal at this point.

Here is the right side of the Baseband Transmitter schematic:



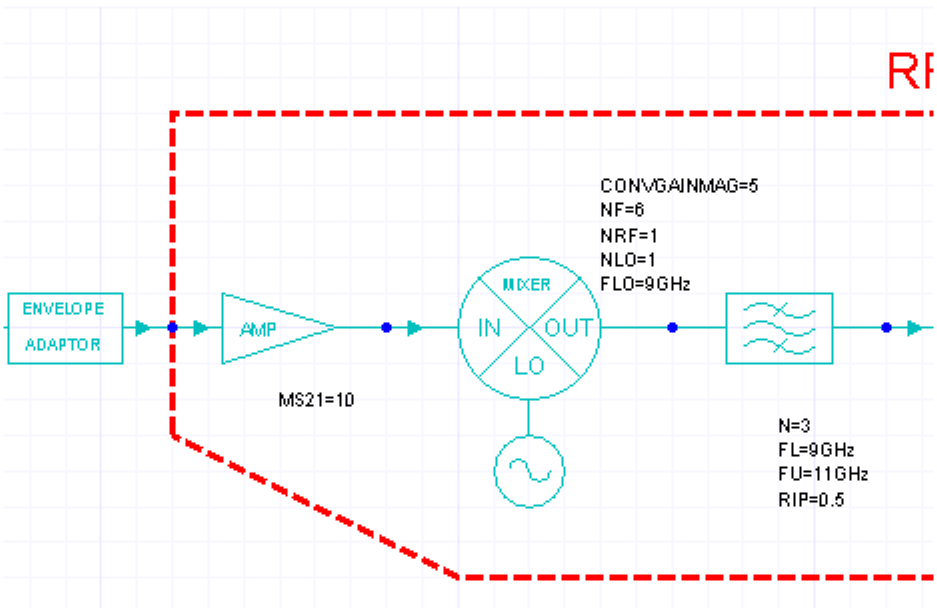
This part of the Baseband transmitter contains the following elements on the System simulator component library:

- Root raised cosine filters on both real and imaginary data paths. The cutoff frequency for both filters is set to parameter **fcutoff** =4MHz. The rolloff factor BETA is set to parameter **fbeta** =0.35. The number of filter coefficients FILT_LENGTH is set to parameter **flength** =64.
- A system probe, **postfilt**, is inserted to enable you to view the real part of the signal at this point.
- Real signal scaling elements on both paths. These are provided to enable you to experiment with real and imaginary signals with different gains. The gain on both elements is set to 1.
- An IQ modulator that receives the real part of the signal on input I (In-phase) and the imaginary part on input Q (Quadrature phase). The IQ modulator combines the signals and upshifts the output. The carrier frequency FC is set to 1GHz and the carrier power CP is set to 10dBm.
- A system probe, **out**, is inserted to enable you to view the signal as it leaves the Baseband transmitter.
- A power spectral density probe, **pout**, is inserted to enable you to view the power output of the Baseband transmitter.
- An Envelope adaptor. The adaptor is required to connect the Baseband transmitter stage, with only behavioral components, to the RF transmitter stage which includes electrical components. Behavioral components are unidirectional and time-based; the output is a

function only of the input. Electrical components are bidirectional and frequency-based; reflections at both input and output can be characterized with S-parameters. The Envelope adaptor identifies the waveform as an envelope to the Nexxim portion of the simulation. The Nexxim simulation then knows to send on an envelope waveform.

RF Transmitter Schematic

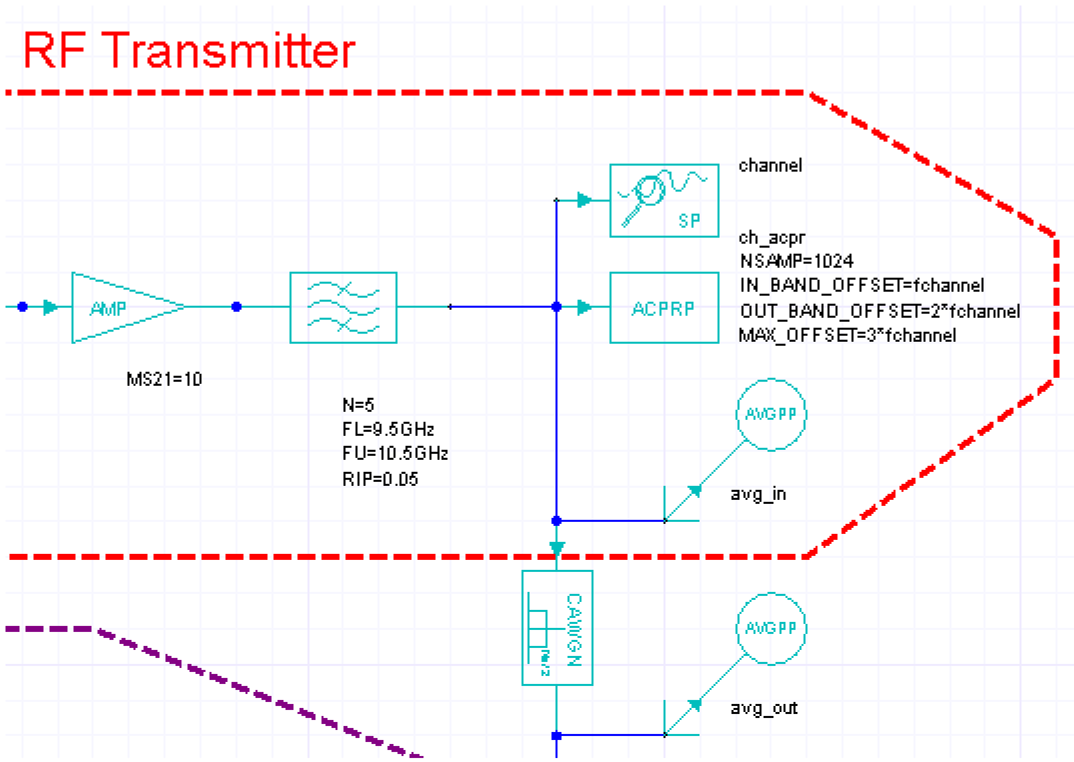
The RF transmitter stage takes the I/Q modulated signal as input and outputs the signal mixed with the carrier wave, amplified and filtered. Here is the left side of the RF Transmitter schematic:



This part of the RF transmitter contains the following elements, two on the System nonlinear RF library and one on the Nexxim component library:

- An amplifier on the System nonlinear RF library provides a 10dB gain.
- A mixer on the System nonlinear RF library mixes the LO frequency (9GHz) with the up-shifted data around 1GHz.The mixer adds a conversion gain from RF (in) to IF(out) of 5dB and a noise figure of 6dB.
- A third-order Chebyshev bandpass filter on the Nexxim component library. The filter is set to pass frequencies between 9GHz and 11GHZ with a maximum inband ripple of 0.5dB.

Here is the right side of the RF Transmitter schematic:

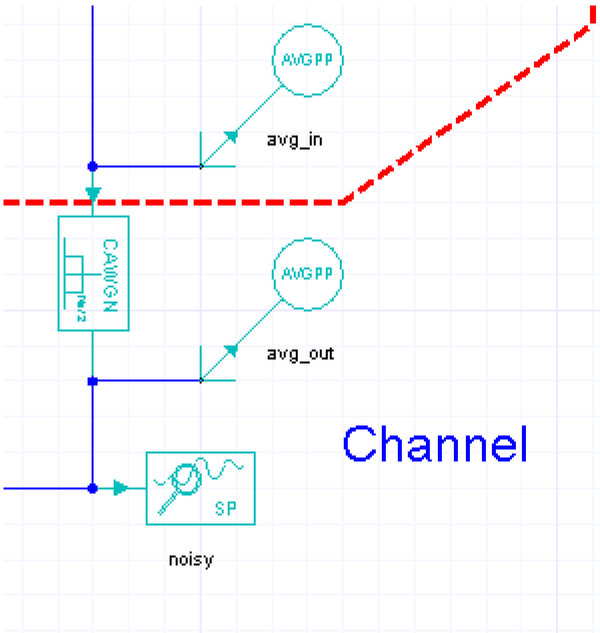


This part of the RF transmitter contains the following elements, one on the System nonlinear RF library and one on the Nexxim component library:

- Another amplifier on the System nonlinear RF library provides a 10dB gain post-filter.
- A fifth-order Chebyshev bandpass filter on the Nexxim component library. The filter is set to pass frequencies between 9.5GHz and 10.5GHZ, this time with a maximum in-band ripple of 0.05dB.
- A system probe, **channel**, is inserted to enable you to view the output of the RF transmitter.
- An adjacent channel power ratio probe, **ch_acpr**, is inserted to enable you to view the in-band and out-of-band power outputs on the RF transmitter.
- An average power probe, **avg_in**, is inserted to enable you to view the average output power on the RF transmitter.

MPSK Channel Schematic

Here is the schematic for the communication channel:

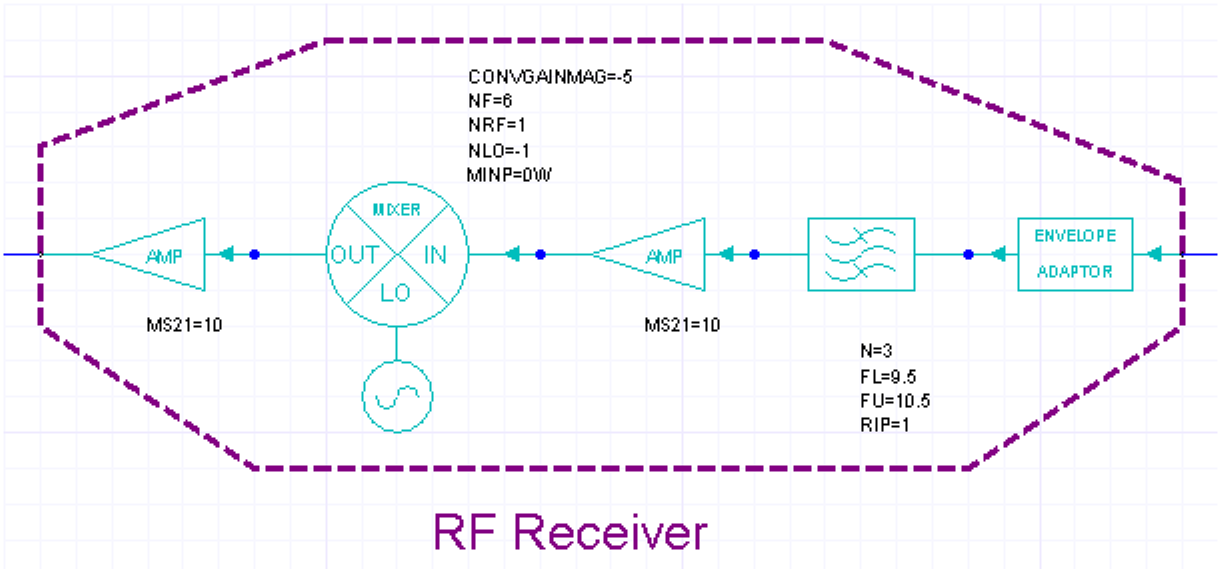


The channel models a noisy transmission line. The channel consists of one element and two probes on the System library.

- A channel element with added complex Gaussian white noise.
- An average power probe, **avg_out**, is inserted to enable you to view the average power after the channel.
- A system probe, **noisy**, is inserted to enable you to view the signal and noise at the receiving end of the channel.

RF Receiver Schematic

The RF transmitter essentially reverses the sequence of operations performed by the RF Transmitter. The Envelope adaptor interfaces the behavioral channel element with the electrical elements in the RF Receiver. The RF Receiver stage takes the signal mixed with the carrier wave as input and outputs the unmixed signal , amplified and filtered. Here is the RF Receiver schematic:

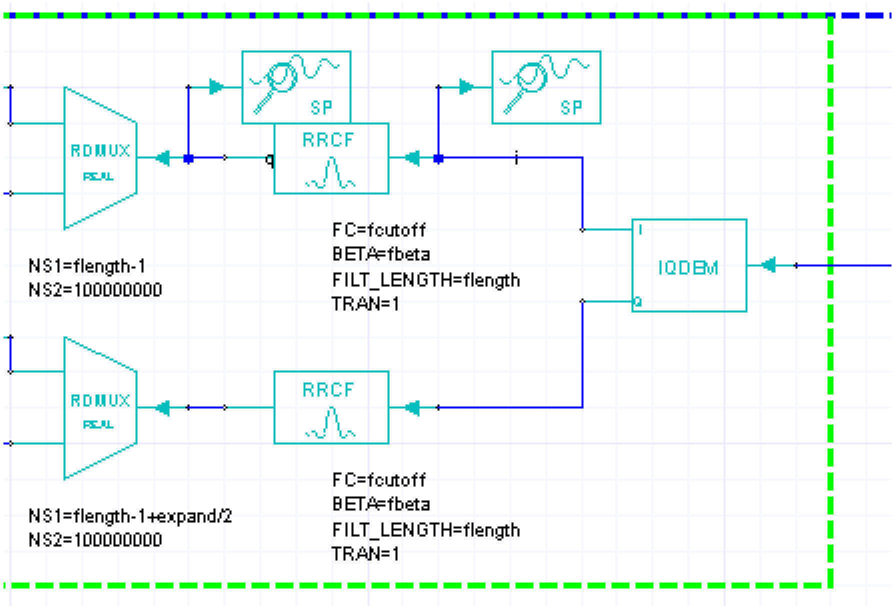


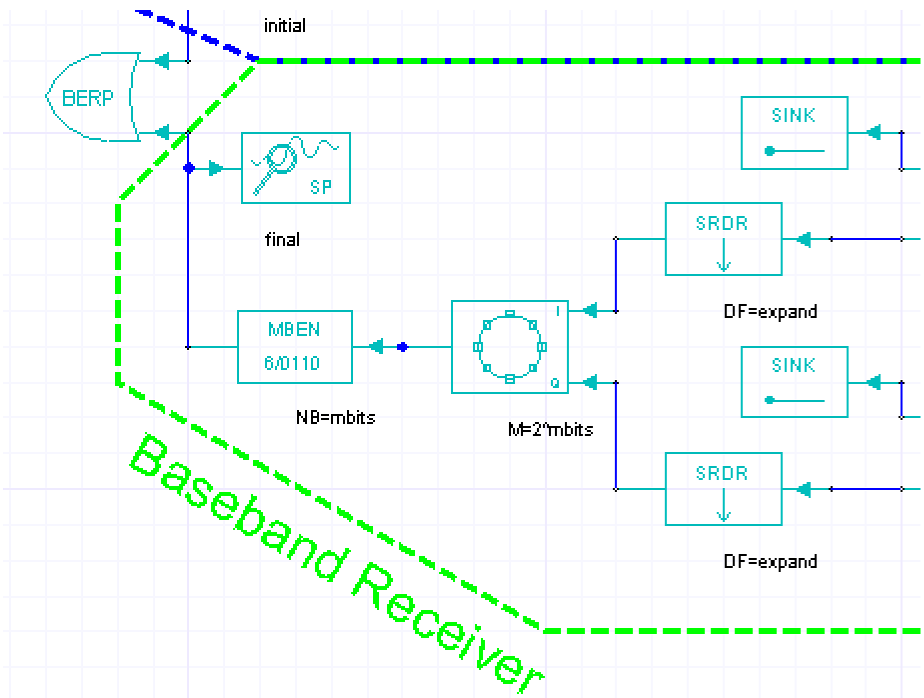
Baseband Receiver Schematic

The Baseband receiver essentially reverses the sequence of operations performed by the Baseband transmitter. The receiver inputs a modulated signal, then downshifts, demodulates, and decodes it, and finally outputs the received bit stream.

The output on the pseudorandom bit generator and the bit stream output on the Baseband receiver are both sent to a bit error probe (BERP) for direct comparison and calculation of the bit error rate (BER).

Here are the right and left sides of the Baseband receiver schematic:

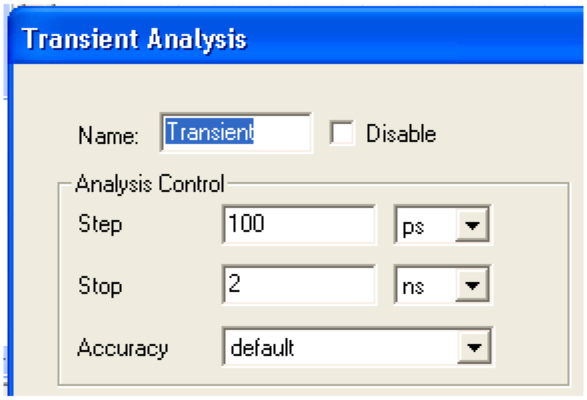




Run All Analyses and View Reports

The main project **MPSK_Nexxim_System** is organized in three sub-projects, **MPSK_ Baseband**, **MPSK_Mixed**, and **MPSK_SweptBER**. All of these sub-projects have the same

Transient analysis setup:

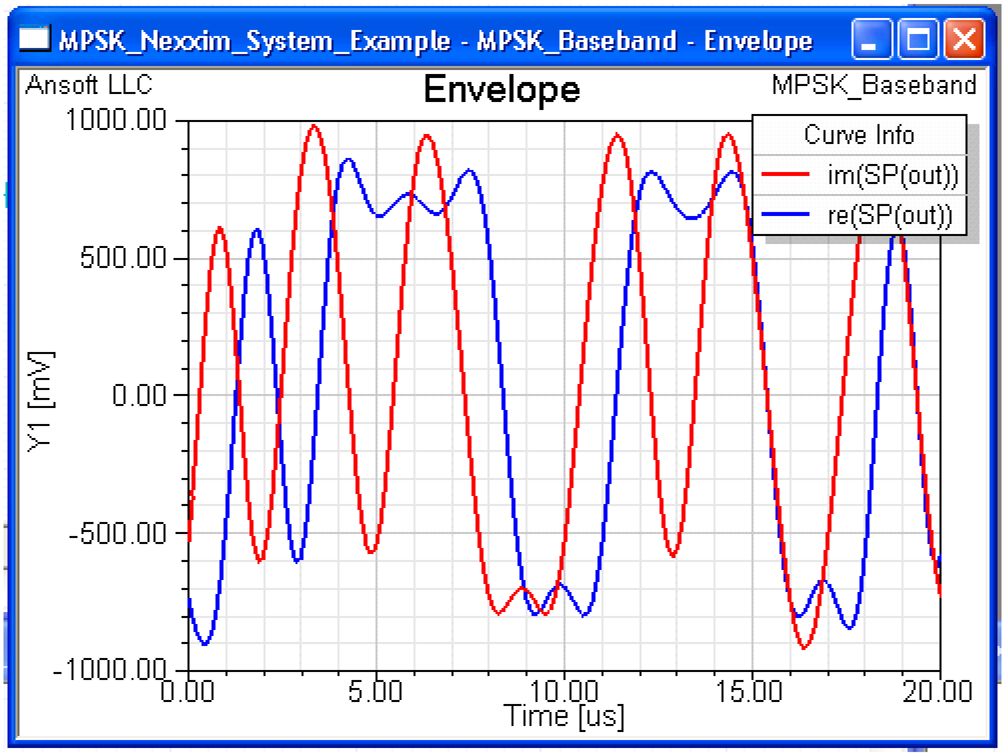


This setup controls the Nexxim library elements. For the System elements, the discrete time domain analysis uses a fixed timestep controlled by the bit rate in the System source.

Each sub-project includes a few typical reports.

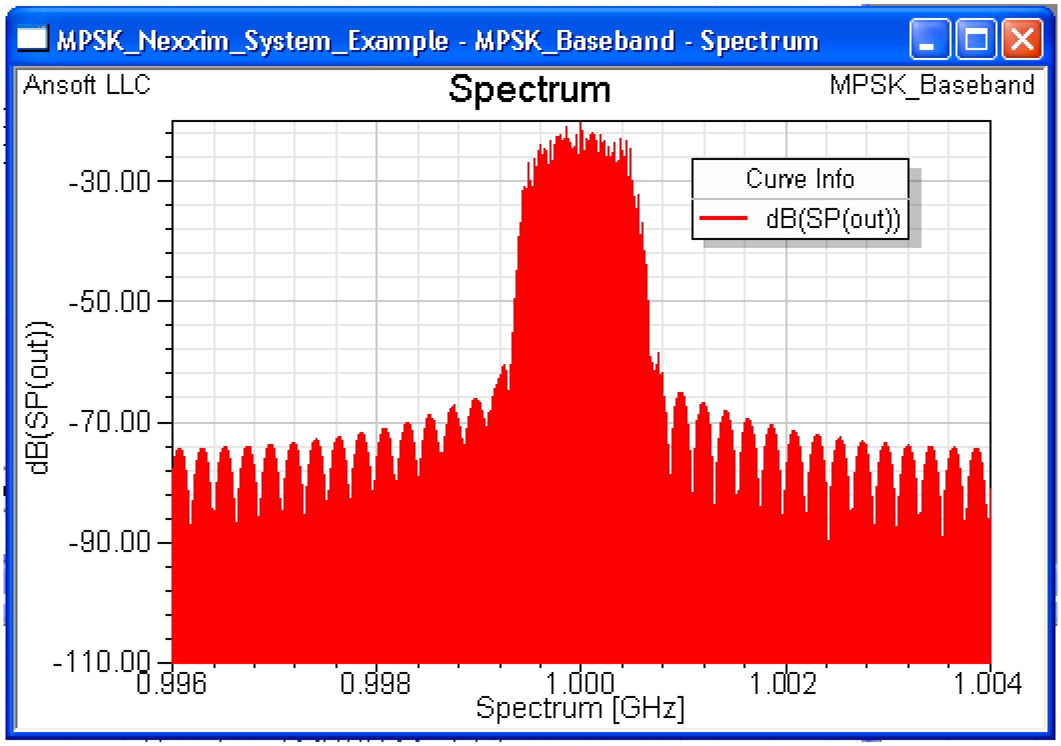
1. Click the **MPSK_Nexxim_System** project and select **Analyze All** on the menu. The Progress window displays a red line to show each analysis as it runs. It may take a few moments for all analyses to run to completion.
2. Expand the Results icon under the **MPSK_Baseband** project. The four reports, **Envelope**, **Spectrum**, **Constellation**, and **Eye Diagram** show the output on the Baseband transmitter (system probe **out**) in different ways. Double-click each of the four reports to open it.

Here is the **Envelope** report.



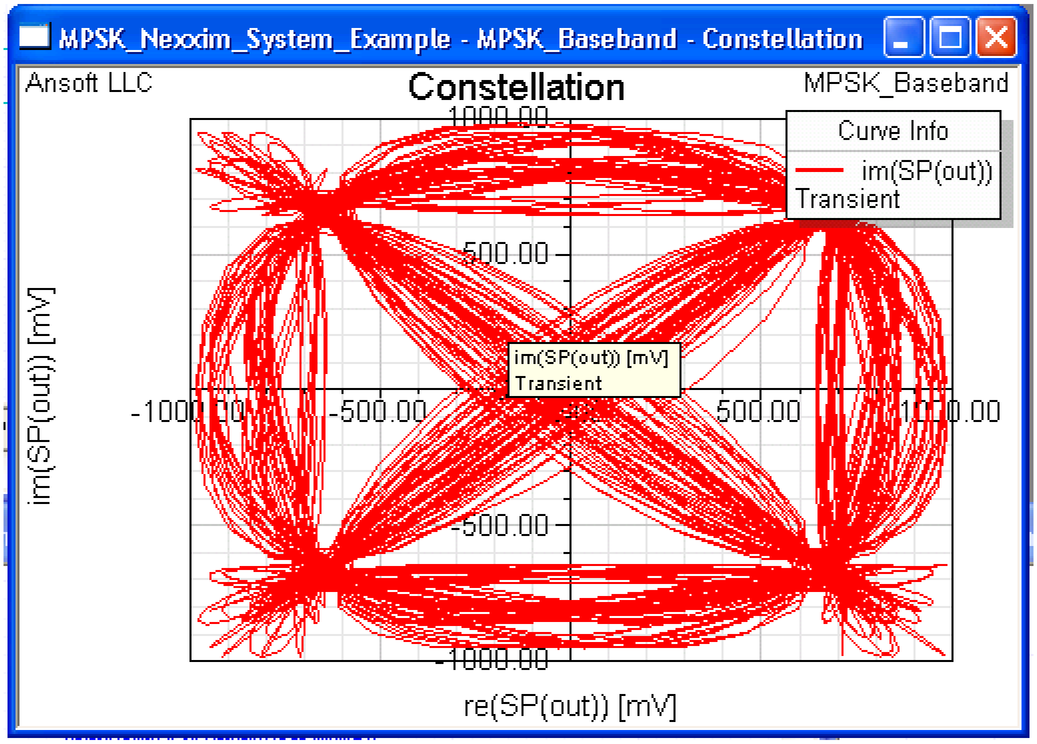
The System discrete time analysis automatically performs envelope analysis on an I/Q modulated signal. This graph shows the output on the Baseband transmitter [SP(out)] resolved into its In-band (re) and Quadrature-band (im) components.

Here is the **Spectrum** report.



This graph shows the output from the Baseband transmitter [SP(out)] analyzed in the spectral domain. The energy is centered around 1GHz, the upshifted signal frequency.

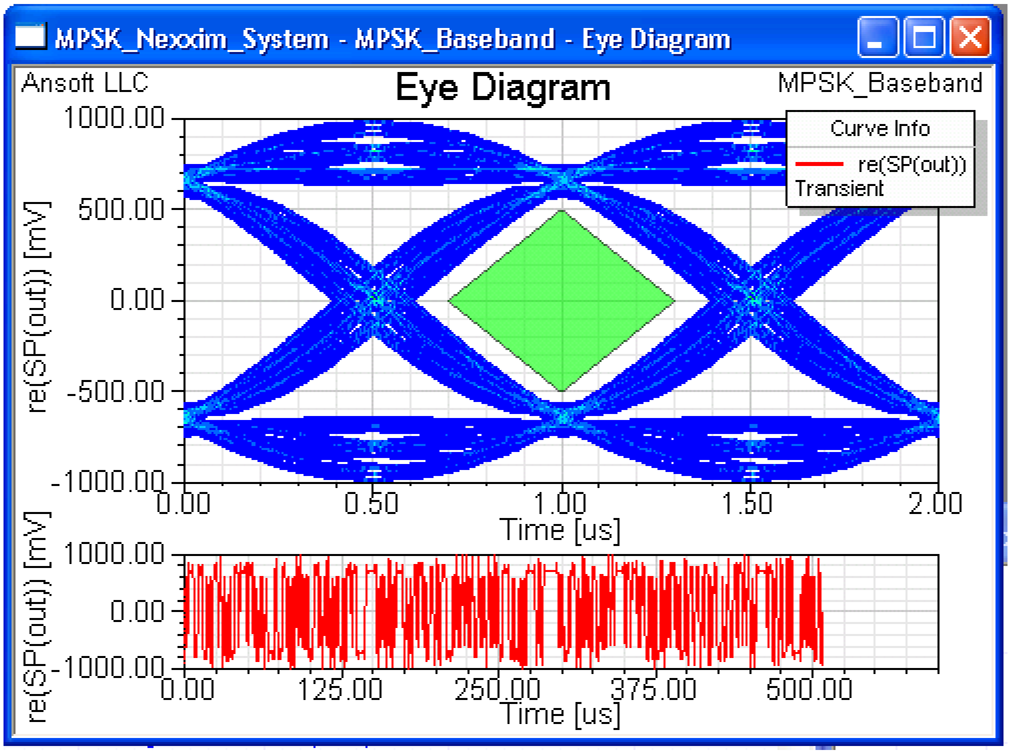
Here is the **Constellation** report.



This graph shows the Quadrature-band (im) component of the output on the Baseband transmitter [SP(out)] graphed against the In-band (re) component.

The four nodes about the points (+/-700, +/-700) correspond to the four symbols in the transmission. The distinctness of the constellation indicates a good signal.

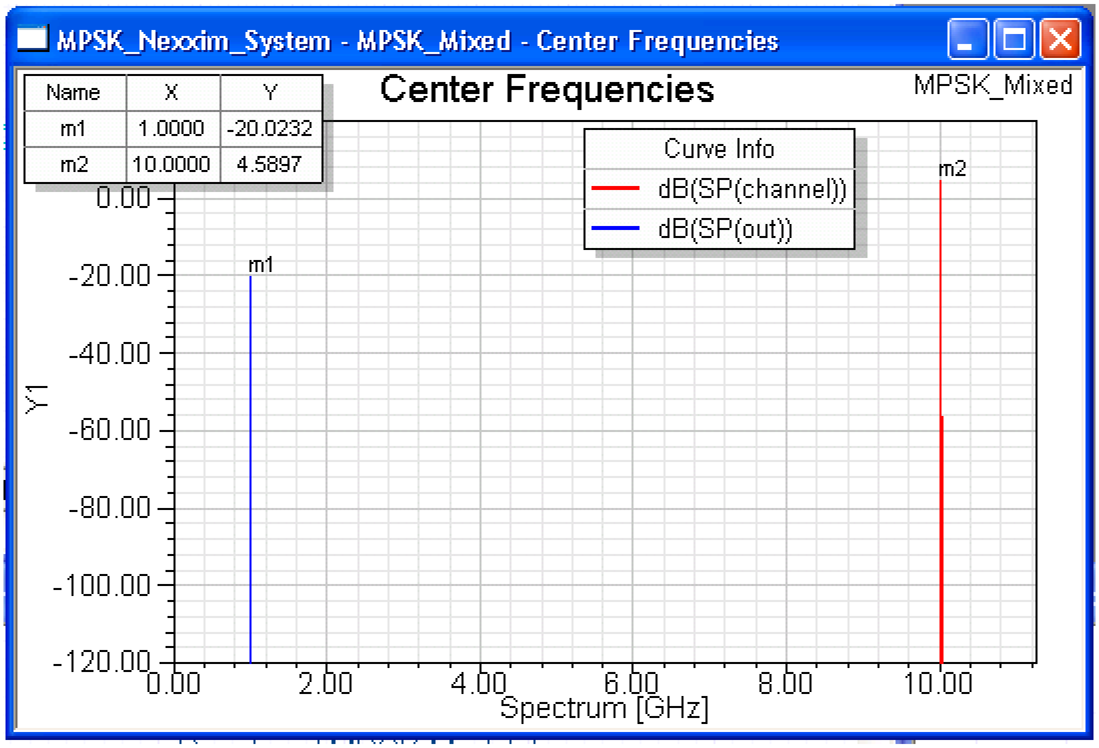
Here is the **Eye Diagram** report.



The Eye diagram report confirms the quality of the Baseband transmitter output. The green diamond indicates the portion of the unit interval (UI) that must be open to guarantee the appropriate bit error rate (BER).

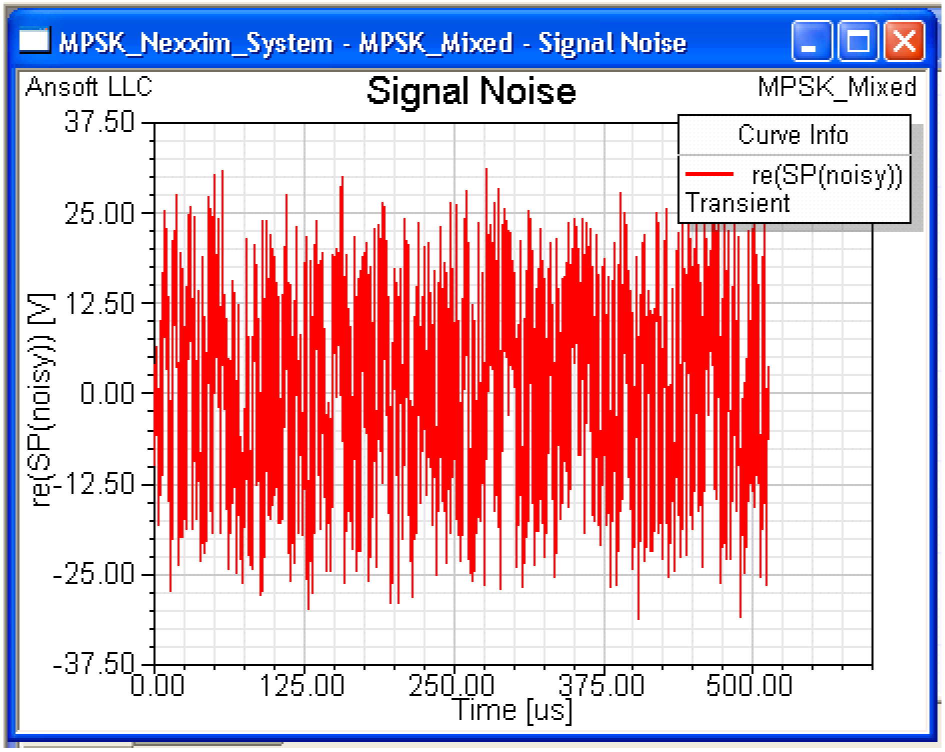
3. Expand the Results icon under the **MPSK_Mixed** project. The three reports, **Center Frequencies**, **Signal Noise**, and **Noisy Channel** show the output on the RF transmitter (system probe **channel**) plus the noise contribution on the channel (system probe **noisy**). Double-click each of the three reports to open it.

Here is the **Center Frequencies** report.



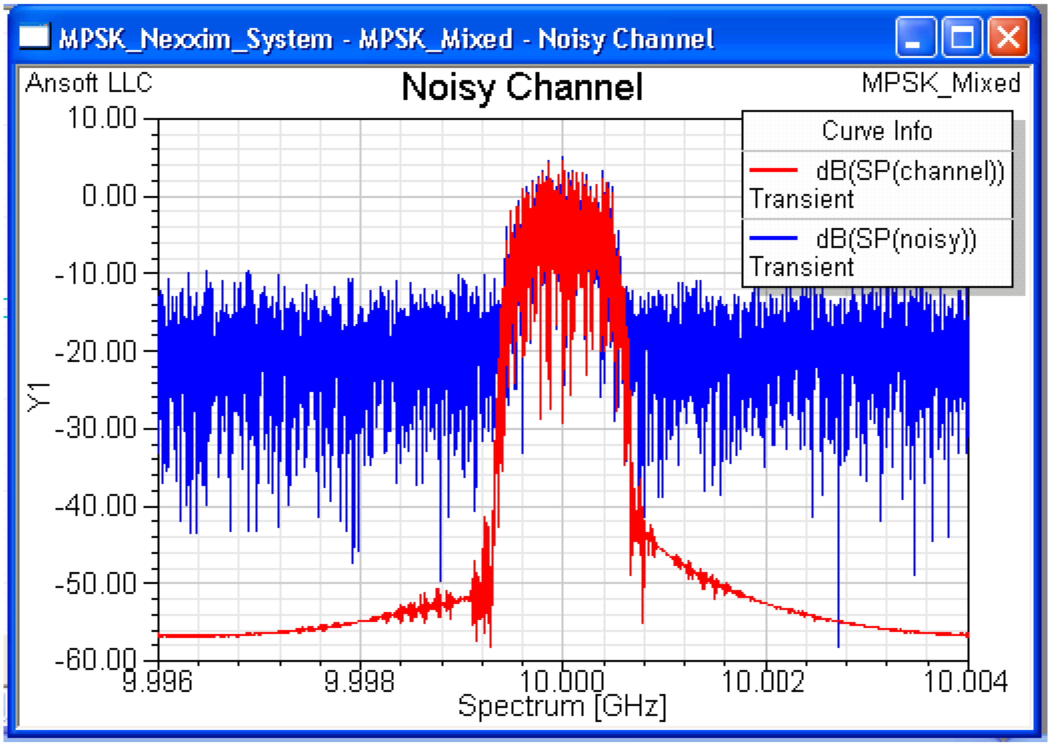
This graph compares the upshifted spectral output on the Baseband transmitter [SP(out)] at around 1GHz with the combined spectrum of the signal and carrier coming on the RF transmitter [SP(channel)] at around 10GHz.

Here is the **Signal Noise** report.



This graph shows the combined signal and noise output on the channel [SP(noisy)] analyzed in the time domain.

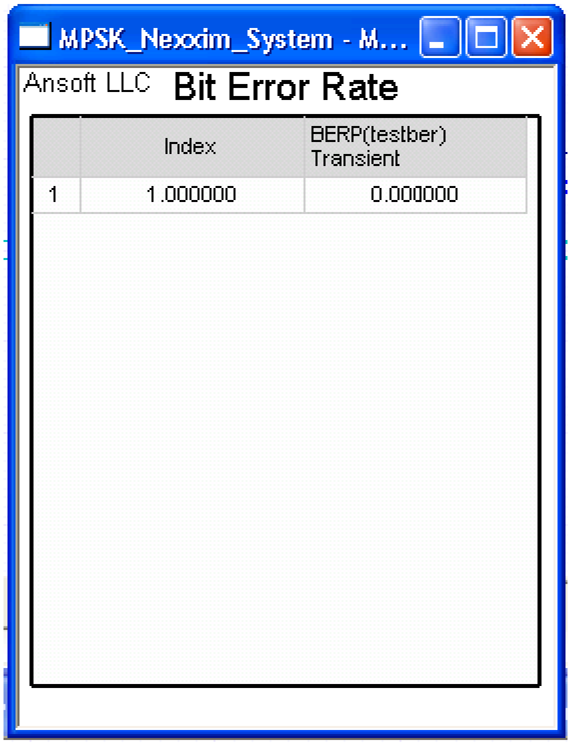
Here is the **Noisy Channel** report.



This graph shows the superimposition of the signal [SP(channel) and the Gaussian white noise added by the channel [SP(noisy)].

4. Expand the Results icon under the **MPSK_SweptBER** project. The single report, **Bit Error Rate**, shows the BER calculated by comparing the initial bit stream with the end result.

Here is the **Bit Error Rate** report.



This data table shows that no bit errors are detected.

These examples are just a few of the many possible reports. You can also vary or sweep any of the netlist parameters to investigate the effects.

For additional information, see Nexsys Discrete Time Domain Analysis.

Nonlinear Loadpull Amplifier Example

The nonlinear loadpull amplifier component (NLAMP) uses data-driven loadpull values to generate a behavioral amplifier model. The user supplies the loadpull data in LPD format (from Focus Microwaves, Inc.) The data gives the output power for a range of output impedances at constant input power and at a given frequency. To predict the non-linear behavior of the amplifier, the current version also provides the third-order inter-modulation products. The second-order IMD products are not provided.

The NLAMP component supports one-tone and two-tone harmonic balance simulation, including a sweep of input power. You can use Nexxim loadpull analysis to open the behavioral amplifier model such as a Smith chart.

For more information, see *Nonlinear Loadpull Amplifier*.

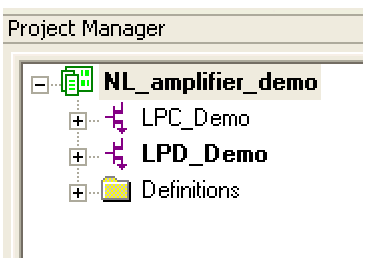
Open the NLAMP Design

The schematic for the demo and an example loadpull data file are available in the Examples directory.

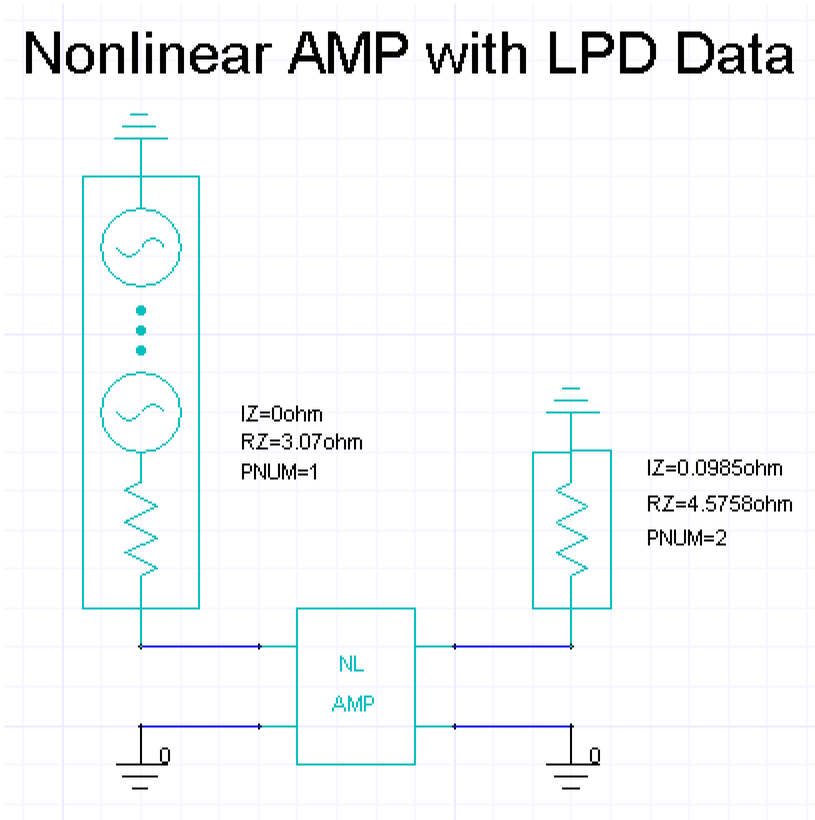
Open the **NL_Amplifer_Demo** Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then t**Circuit**, then **RF_Microwave**, the **Amplifiers** directory, then select the file **NL_Amplifier_Demo.aedt**. Click **Open**.
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

The project contains two circuits:



4. Click the **LPD_Demo** circuit. The **LPD_Demo** schematic appears in the design window:



- NLAMP is a four terminal device. In this example, two terminals (one input and one output) are connected to ports while the remaining terminals are grounded. The value of the characteristic source impedance specified in the loadpull data file (3.07) is added as the impedance of port 1 (input port). A load impedance of (4.5758 + j0.0985) is specified as the impedance of port 2 (output port).

5. Click the **NLAMP** element to display its property settings:

Name	Value	Unit	Evalu...	Description
loadpull_file	NLAMP_demo.lpd			Load pull data file
source_frequency1	freq1		1.96G...	Frequency of input source
source_frequency2	freq2		2.06G...	Frequency of second input source for 2-tone a
load_resistance	0	ohm	0ohm	Load resistance 0=calculated
load_reactance	0	ohm	0ohm	Load reactance 0=calculated
model_switch	0			Select implementation 0=freq domain 1=time c
CosimDefinition	Edit			
CoSimulator	DefaultNetlist			
Status	Active			

6.

The loadpull model data file is **NLAMP_demo.lpd**. The source frequency is given by the project variable *freq1*. The frequency value is 1.96 GHz, the frequency for the loadpull data in the file. A second, slightly higher frequency given by project variable *freq2* is used in two-tone harmonic balance analysis.

The model_switch is set to select the frequency domain implementation.

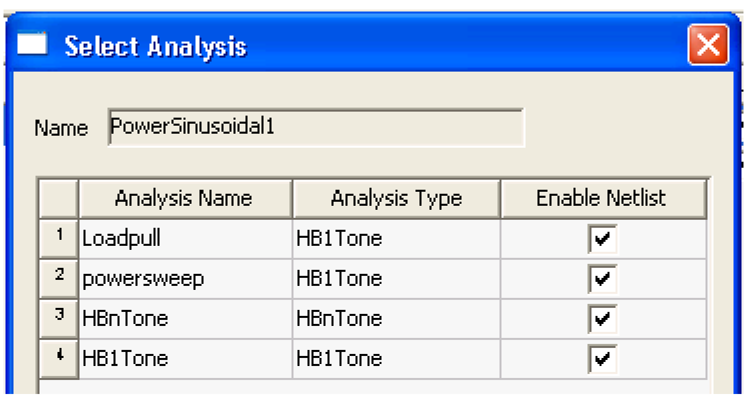
7. Click Port 1 (PNUM=1) and select **Edit Port**. Near the bottom of the Edit Port window, select **Edit Sources**.
8. Select the **PowerSinusoidal1** source and click **Edit Properties**.

Name	Value	Unit	Evaluate...	Descri
Name	PowerSinusoidal1			
ACMAG	nan		nan	AC magnitude for small-signal an
ACPHASE	0		0	AC phase for small-signal analysi
DC	0		0	DC voltage (Volts)
VO	0		0	Voltage offset from zero (Volts)
POWER	pin		20dBm	Available power of the source
FREQ	freq1		1.96GHz	Frequency (Hz)
TD	0		0	Delay to start of sine wave (secc

9.

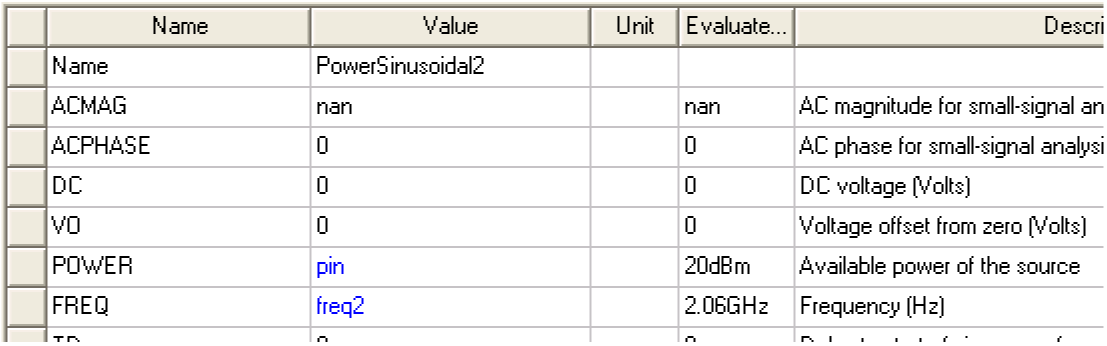
The **PowerSinusoidal1** source runs at a frequency given by project variable *freq1* at an input power given by project variable *pin*.

With **PowerSinusoidal1** still selected, click **Select** under **Analysis list**:



The **PowerSinusoidal1** source runs on all analyses in the demo. Click **OK** to close the **Select Analysis** window.

10. Select the **PowerSinusoidal2** source and click **Edit Properties**.

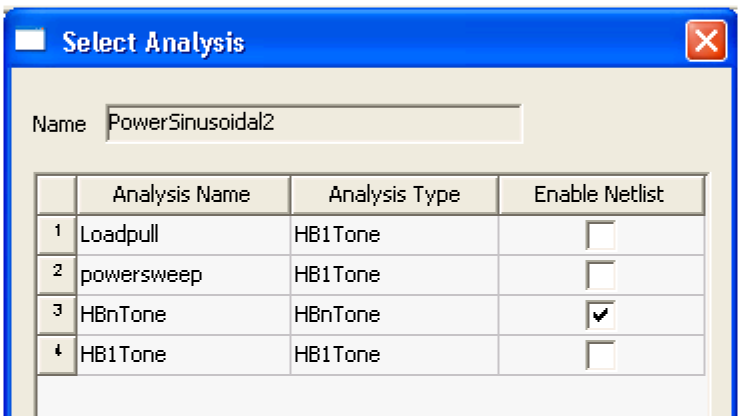


Name	Value	Unit	Evaluate...	Descri
Name	PowerSinusoidal2			
ACMAG	nan		nan	AC magnitude for small-signal an
ACPHASE	0		0	AC phase for small-signal analysi
DC	0		0	DC voltage (Volts)
VO	0		0	Voltage offset from zero (Volts)
POWER	pin		20dBm	Available power of the source
FREQ	freq2		2.06GHz	Frequency (Hz)

11.

The **PowerSinusoidal2** source runs at a frequency given by project variable *freq2* at an input power given by project variable *pin*.

With **PowerSinusoidal2** still selected, click **Select** under **Analysis list**:



Select Analysis

Name PowerSinusoidal2

	Analysis Name	Analysis Type	Enable Netlist
1	Loadpull	HB1Tone	☐
2	powersweep	HB1Tone	☐
3	HBnTone	HBnTone	☑
4	HB1Tone	HB1Tone	☐

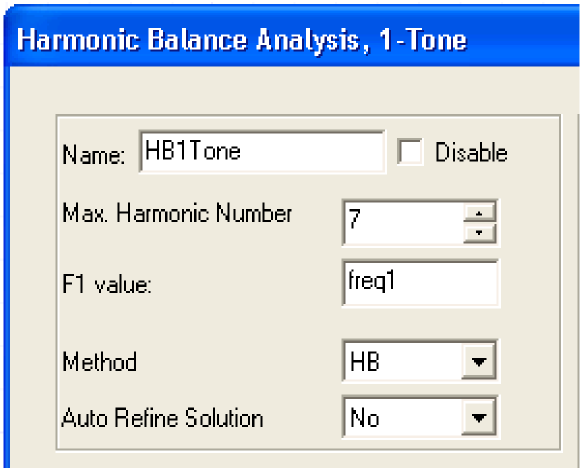
The **PowerSinusoidal2** source runs only on the **HBnTone** analysis in the demo.

12. Click **OK** to close the **Edit Sources** window, and **OK** again to close the **Edit Ports** window.

One-Tone HB with NLAMP

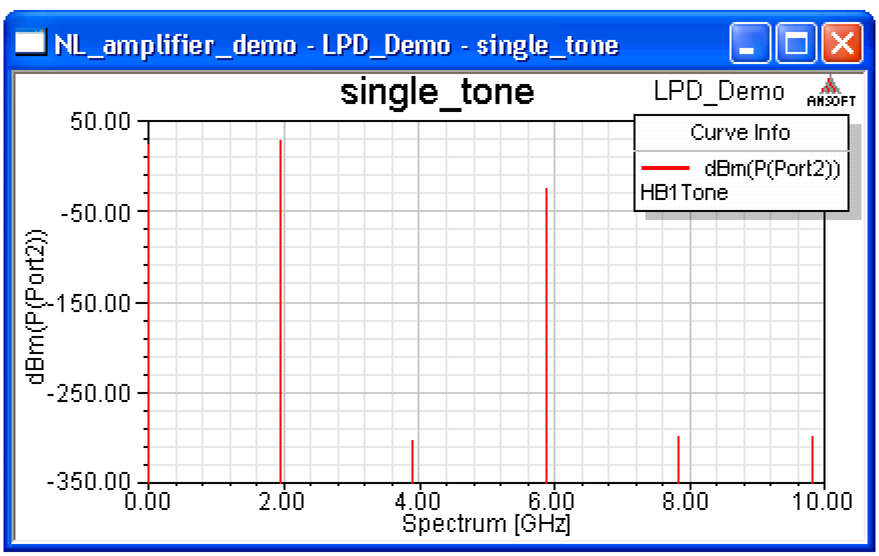
For the single-tone harmonic balance analysis, a power source **PowerSinusoidal1** is connected to the input port supplying an input power *pin* (initially 20dB) at frequency *freq1* (1.96GHz for this set of loadpull data).

1. Expand the **Analyses** icon and double-click on **HB1Tone**.



The **HB1Tone** analysis is set to analyze the first seven harmonics of frequency *freq1*. Click **OK** to close the setup window.

2. With analysis **HB1Tone** still selected, right-click and select **Analyze**. The analysis runs to completion.
3. Click the **Results** icon and double-click **single_tone** to see the results:

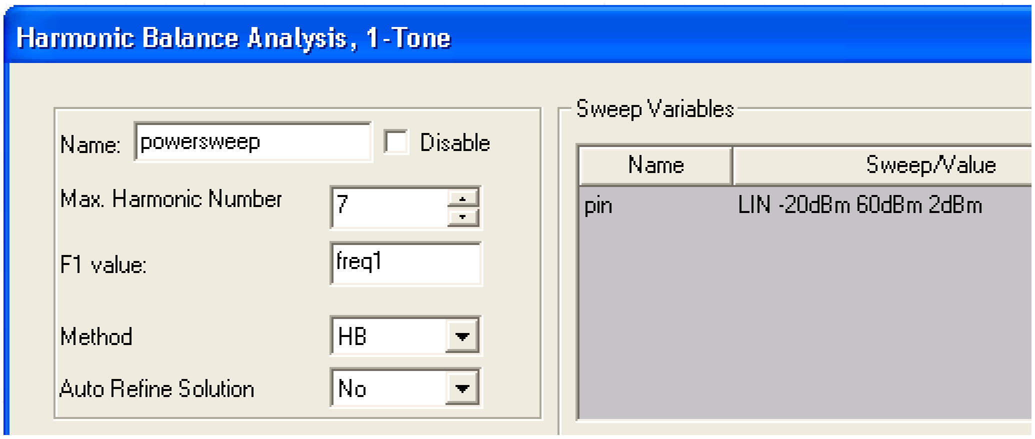


The **single_tone** plot shows the output power (dBm) for all the seven tones (DC and multiples of 1.96 GHz). As expected, the fundamental component is the most dominant followed by the 3rd harmonic. The subsequent odd order terms have lower values. Even-order terms (2nd, 4th etc.) should be considered to be zero, since the second order IMD is not provided.

Power Sweep with NLAMP

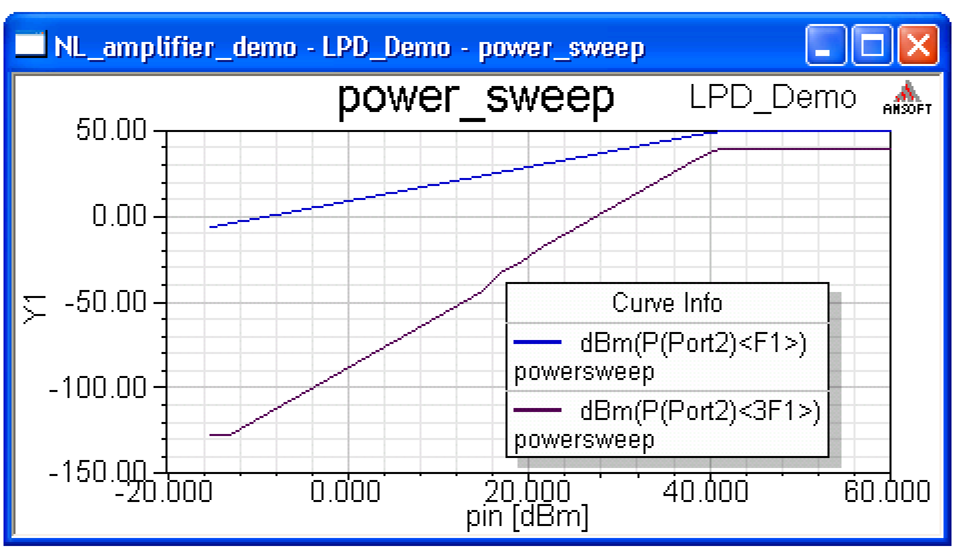
For the power sweep, the input power *pin* is swept over the range -20dBm to 60dBm in steps of 2dBm while performing one-tone harmonic balance.

1. Expand the **Analyses** icon and double-click on **powersweep**.



The **powersweep** analysis is the one-tone HB analysis with the added sweep of input power, given by project variable *pin*. Click **OK** to close the setup window.

2. With analysis **powersweep** still selected, right-click and select **Analyze**. The analysis runs to completion.
3. Click the **Results** icon and double-click **powersweep** to see the results:



The **powersweep** plot shows the fundamental (red) and third harmonic (blue) output power plotted vs. input power **pin**, and the saturation of the amplifier. In the linear region the fundamental output power has a slope of 1:1 while the third harmonic has a slope of 3:1 vs. the input power. Both axes are in log scale (dBm).

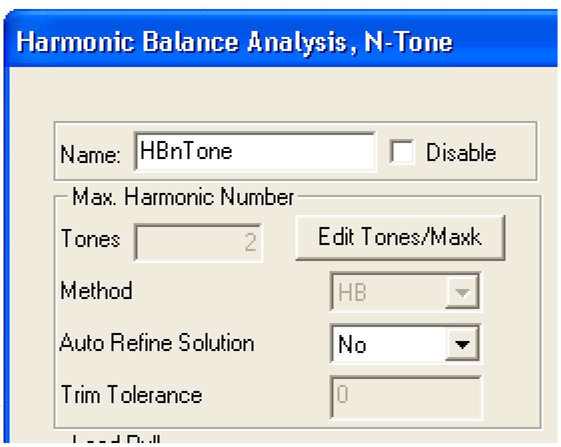
Two-Tone HB with NLAMP

Inter-modulation distortion (IMD) is typically measured with a spectrum analyzer by first injecting a two tone signal into the device , then measuring the resulting inter-modulation product signals. The two tones *freq1* and *freq2* are very close to each other. This ensures that the 3rd order products (2**freq1* – *freq2* and 2**freq2* – *freq1*) are generated very close to the fundamental. Thus their effects can be seen even when the amplifier is operating in a narrow band.

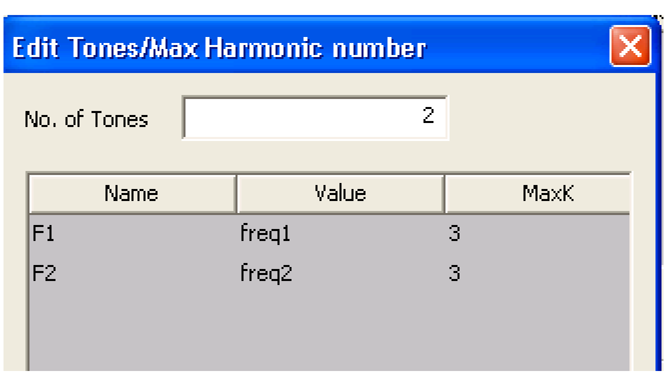
For the two-tone harmonic balance analysis, two simusoidal power sources connect to input Port 1:

- **PowerSinusoidal1** supplies power *pin* (20dB) at frequency *freq1* (1.96GHz).
- **PowerSinusoidal2** supplies power *pin* (20dB) at frequency *freq2* (2.06GHz).

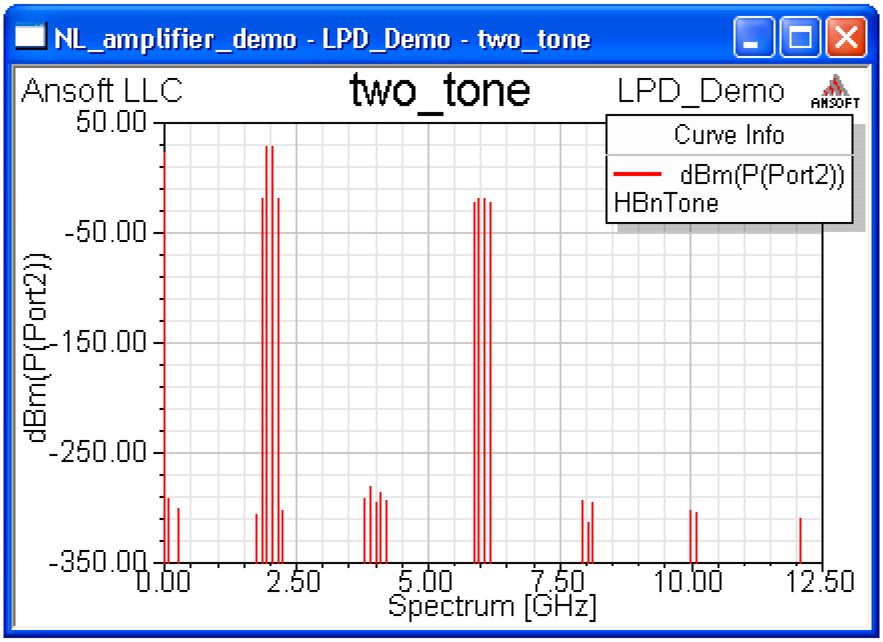
1. Expand the **Analyses** icon and double-click on **HBnTone**.



Click **Edit Tones/Maxk** to view the frequency setup, three harmonics each of *freq1* and *freq2*.



2. With analysis **HBnTone** still selected, right-click and select **Analyze**. The analysis runs to completion.
3. Click the **Results** icon and double-click **two_tone** to see the results:

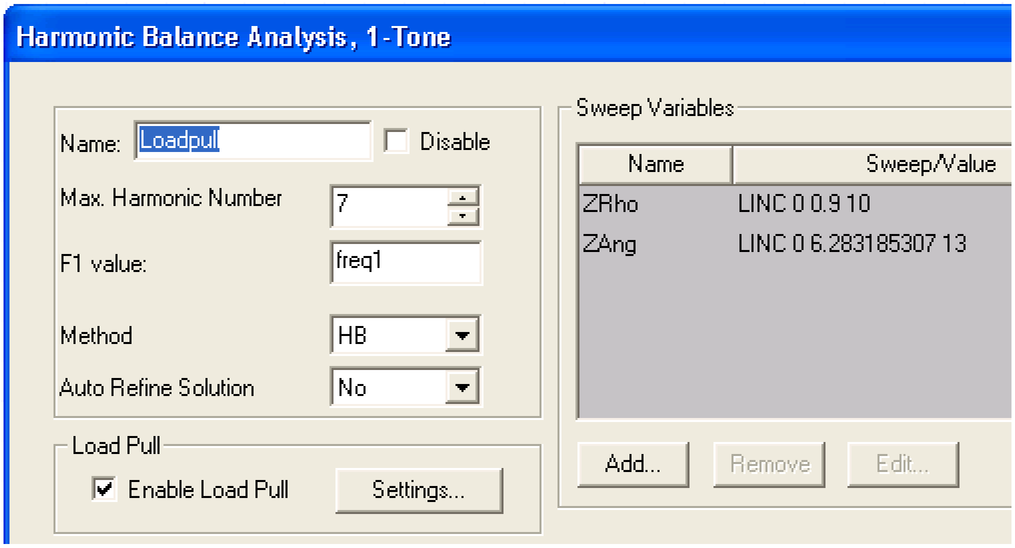


The **two_tone** report shows the two fundamental components corresponding to *freq1* (1.96 GHz) and *freq2* (2.06 GHz) to be dominant, followed by the third order terms 2**freq1 – freq2* (1.86 GHz) and 2**freq2 – freq1* (2.16 GHz). Since the input power for both the sources is the same, the amplitudes of the two fundamental and the two third harmonic components are identical.

Loadpull Analysis of NLAMP

For the loadpull analysis, the resistance and reactance of the load at Port 2 is varied to cover the entire Smith chart.

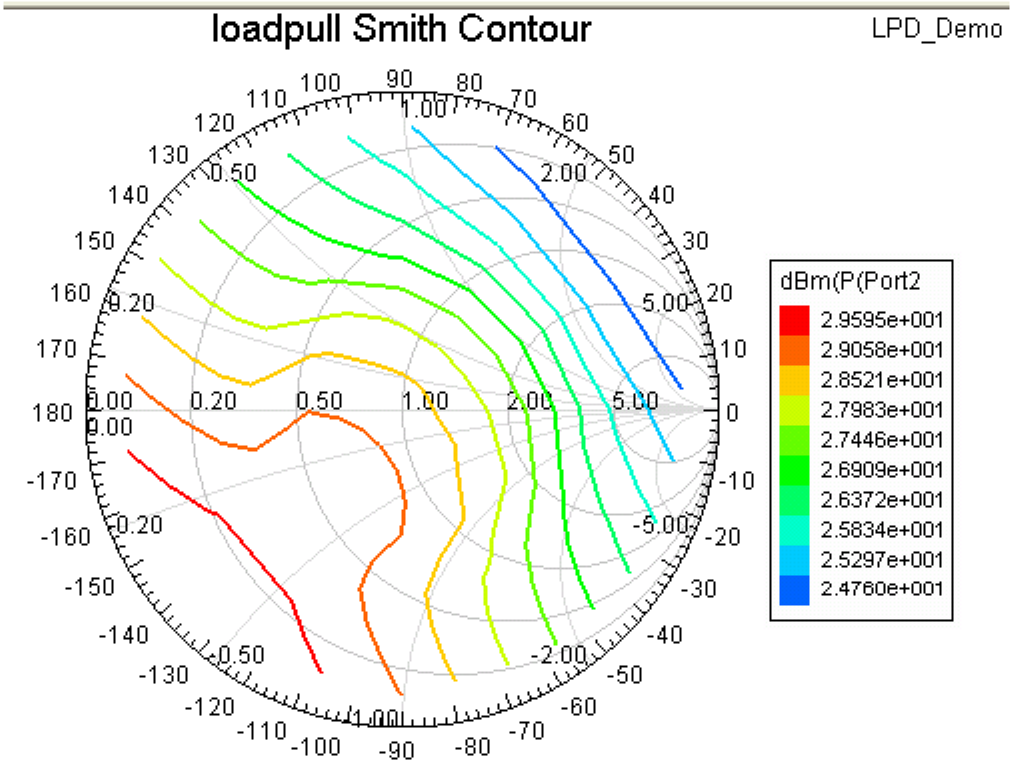
1. Expand the **Analyses** icon and double-click on **Loadpull**.



The **Loadpull** analysis is set to analyze the first seven harmonics of frequency *freq1*, with added sweeps of **ZRho** and **ZAng**, the resistance and reactance variables. Click **OK** to close the setup window.

2. With analysis **Loadpull** still selected, right-click and select **Analyze**. The analysis runs to completion.

3. Click the **Results** icon and double-click **loadpull Smith Contour** to see the results:



The **loadpull Smith Contour** plot shows the output power in dBm over the swept loadings as a Smith Chart contour. The plot shows the Maximum output power generated by the amplifier for a given load impedance.

Power Sweep using Time Domain Implementation

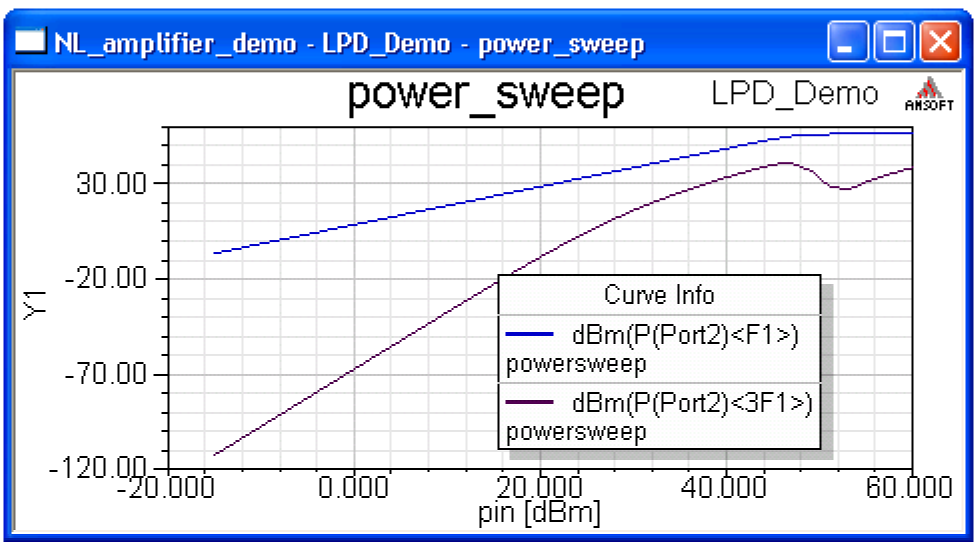
For most analyses, the newer frequency domain implementation produces much the same result as the earlier time domain implementation. In the case of the power sweep, there is a noticeable difference.

1. Select the **NLAMP** element and open its **Properties**.

Name	Value	Unit	Evalu...	Description
loadpull_file	NLAMP_d...			Load pull data file
source_frequency1	freq1		1.96G...	Frequency of input source
source_frequency2	freq2		2.06G...	Frequency of second input source for 2-tone analysis
load_resistance	0	ohm	0ohm	Load resistance 0=calculated
load_reactance	0	ohm	0ohm	Load reactance 0=calculated
model_switch	1			Select implementation 0=freq domain 1=time domain
CosimDefinition	Edit			
CoSimulator	DefaultNet...			
Status	Active			

Set the **model_switch** property to 1 to select the time domain implementation. Click **OK** to close the Properties window.

2. With analysis **powersweep** still selected, right-click and select **Analyze**. The analysis runs to completion.
3. Click the **Results** icon and double-click **powersweep** to see the results:



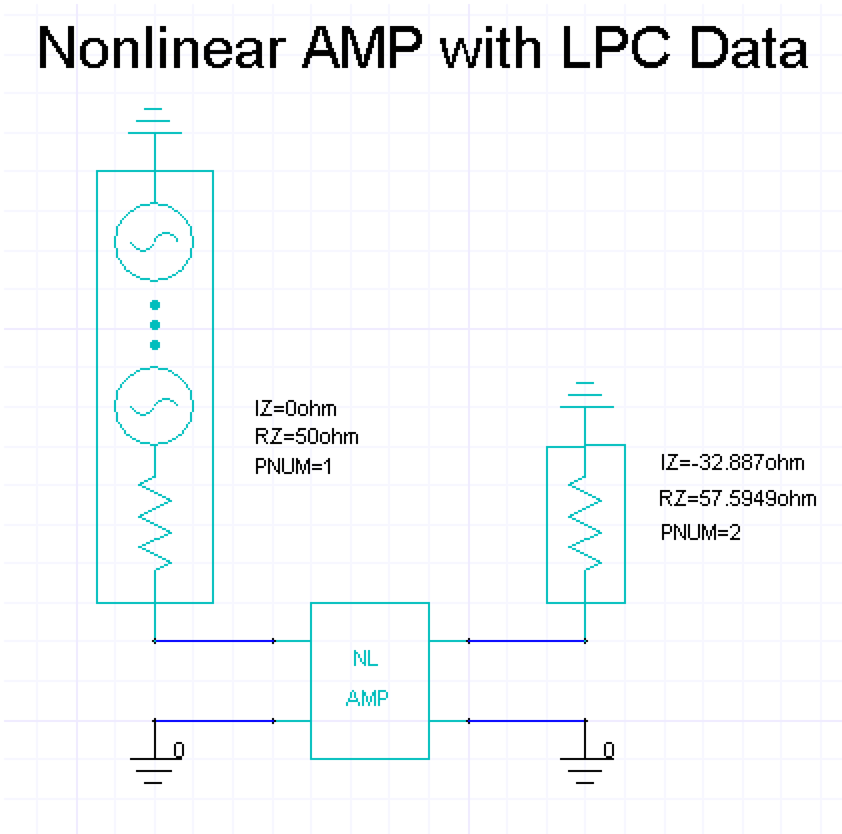
The **powersweep** plot shows the fundamental (red) and third harmonic (blue) output power plotted vs. input power **pin**, and the saturation of the amplifier. In the linear region the fundamental output power has a slope of 1:1 while the third harmonic has a slope of 3:1 vs. the input power. Both scales \are in logarithmic units (dBm). Notice the notch at the upper end of the 3F1 trace. This feature is missing on the frequency domain plot.

When you have finished viewing the reports, you may save the project.

NLAMP with the LPC Demo File

The LPC format is an alternative to the LPD format.

1. Select the LPC_Demo circuit. The schematic appears:



The source has a characteristic impedance of 50 ohms, and the output port impedance has been set at (57.5949 - j32.887), matching one of the values in the LPC file.

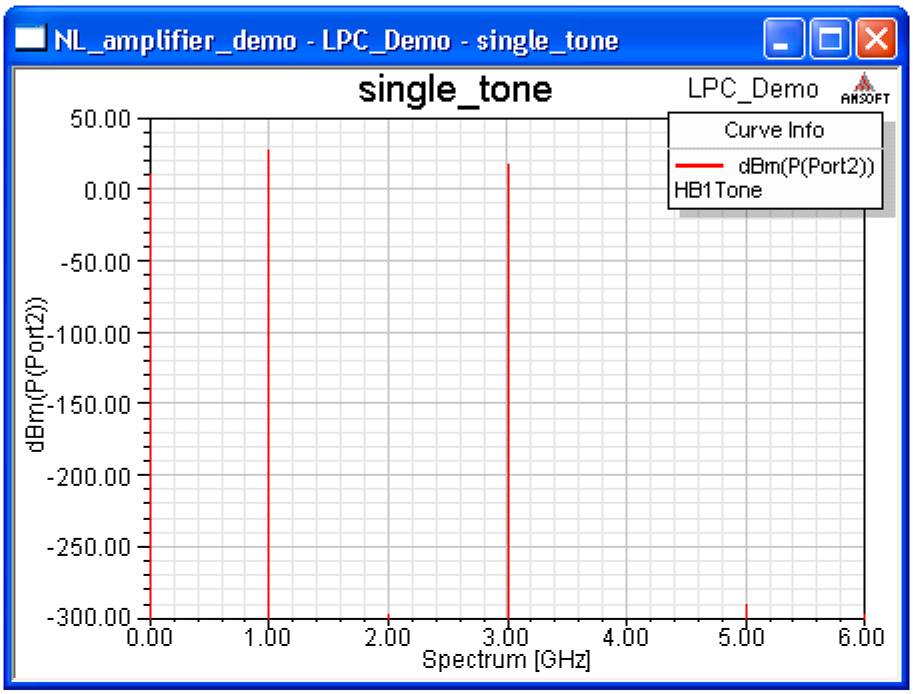
2. Click the NL AMP element and open its Properties:

Name	Value	Unit	Evalu...	Description
loadpull_file	NLAMP_demo.lpc			Load pull data file
source_frequency1	freq3		1GHz	Frequency of input source
source_frequency2	freq4		1.03G...	Frequency of second input source for 2-tone .
load_resistance	0	ohm	0ohm	Load resistance 0=calculated
load_reactance	0	ohm	0ohm	Load reactance 0=calculated
model_switch	0			Select implementation 0=freq domain 1=time c
CosimDefinition	Edit			
CoSimulator	DefaultNetlist			
Status	Active			

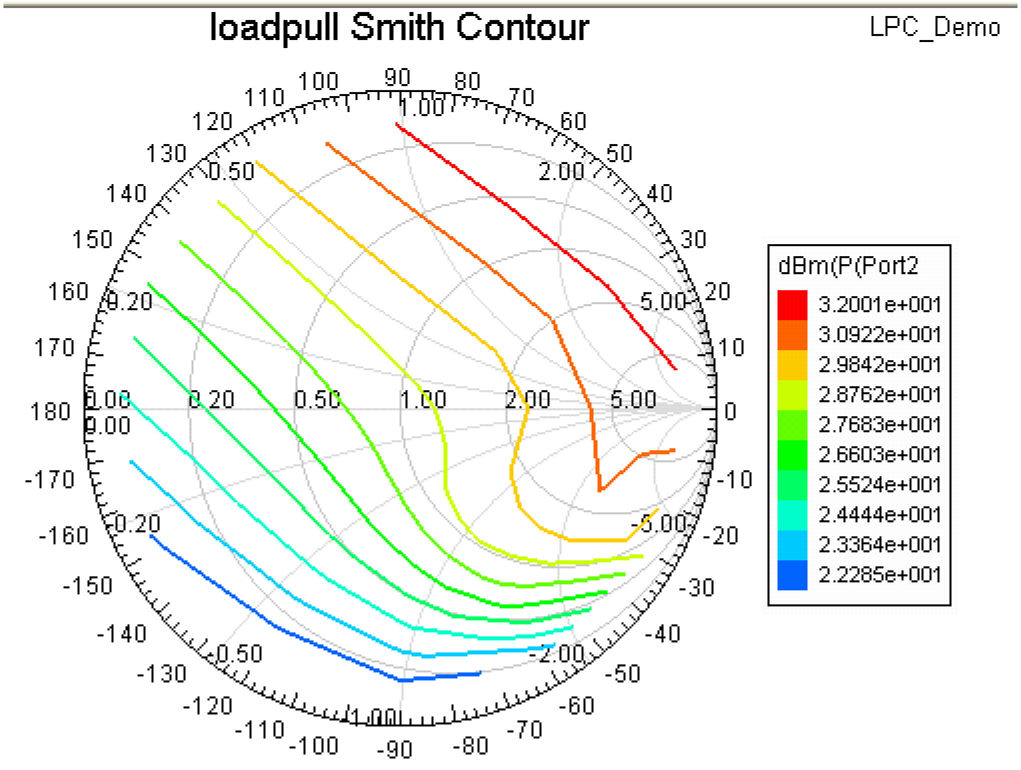
The new loadpull_file is **NLAMP_demo.lpc**. The source frequency given by project variable *freq3* is 1GHz, the frequency at which the file data is derived. A second, nearby frequency given by project variable *freq4* is used in two-tone harmonic balance analysis. The sinusoidal sources and the analysis setups have been modified to use *freq3* and *freq4*.

The **model_switch** is set for the frequency domain implementation. The time domain implementation gives the same results for this LPC file.

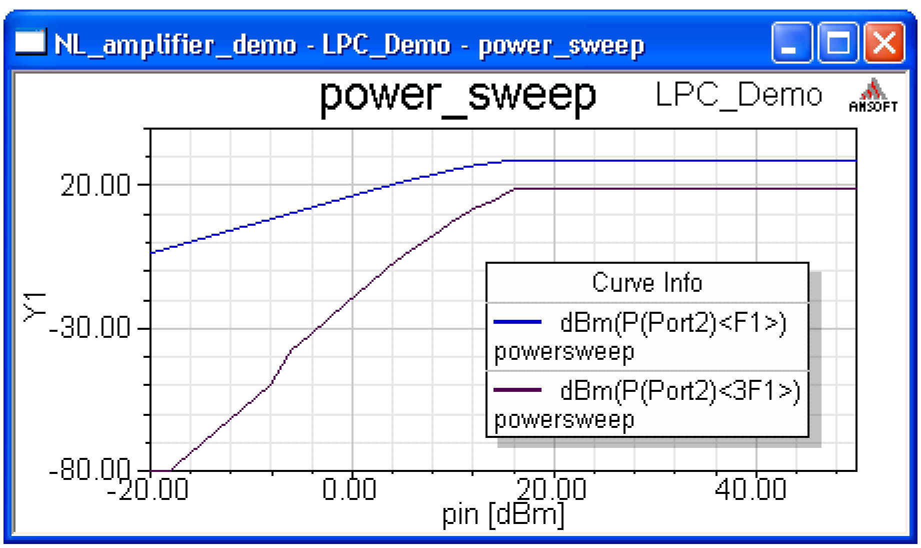
3. Right-click **Analysis** in the Project window and select **Analyze** on the menu. This runs all four analyses.
4. When all four analyses have completed, click the **one tone** Report:



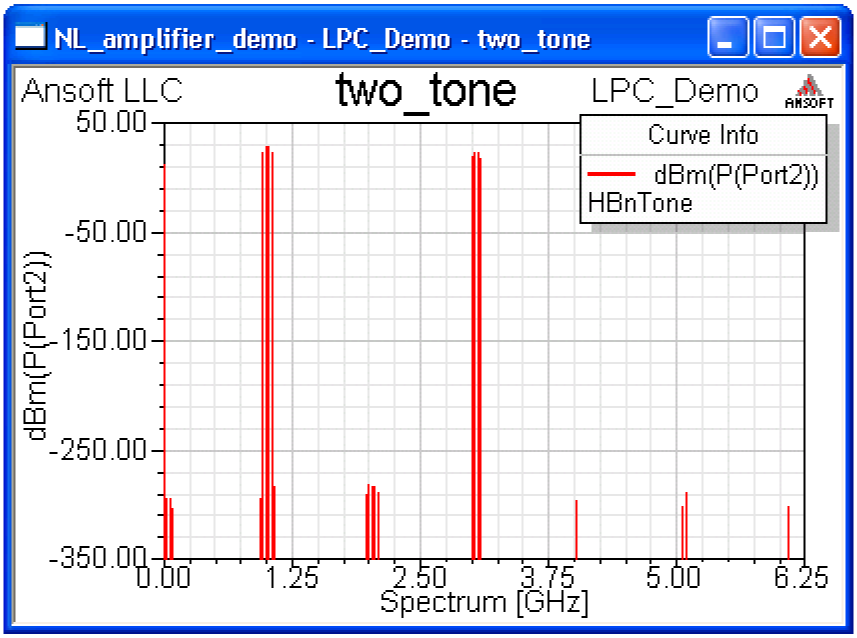
5. Click the **loadpull Smith Contour** Report:



6. Click the **power_sweep** Report:



7. Click the **two tone** Report:



8. When you are finished with the demo, you may save and close the project.

X-Parameter Example

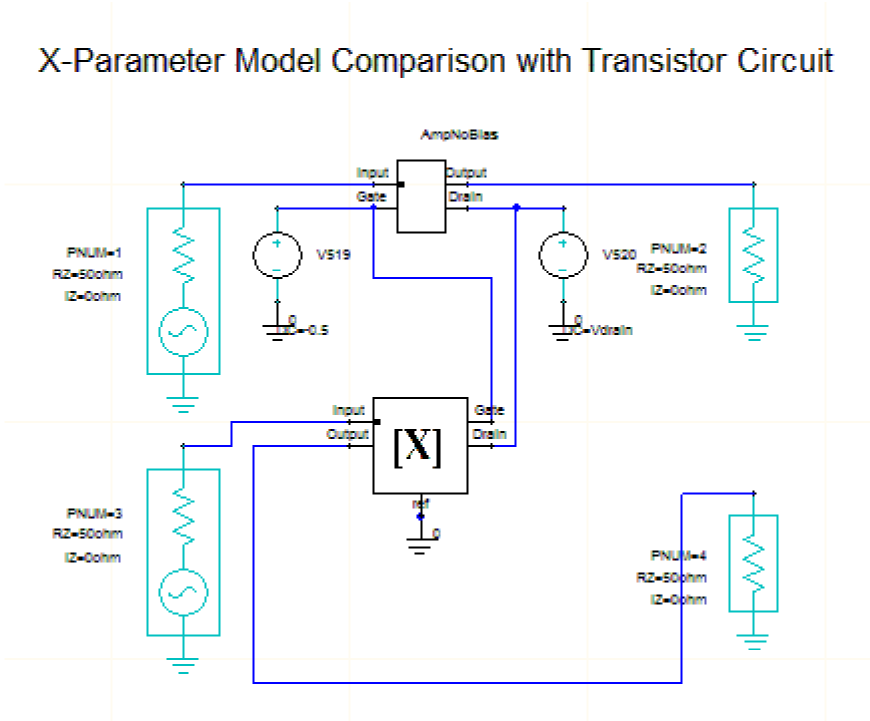
X-parameters describe the frequency-dependent behavior of non-linear systems. In this example, compare the outputs of an amplifier, modeled using both a conventional transistor model and an X-parameter model.

The schematic and data file for the example are available in the Examples directory.

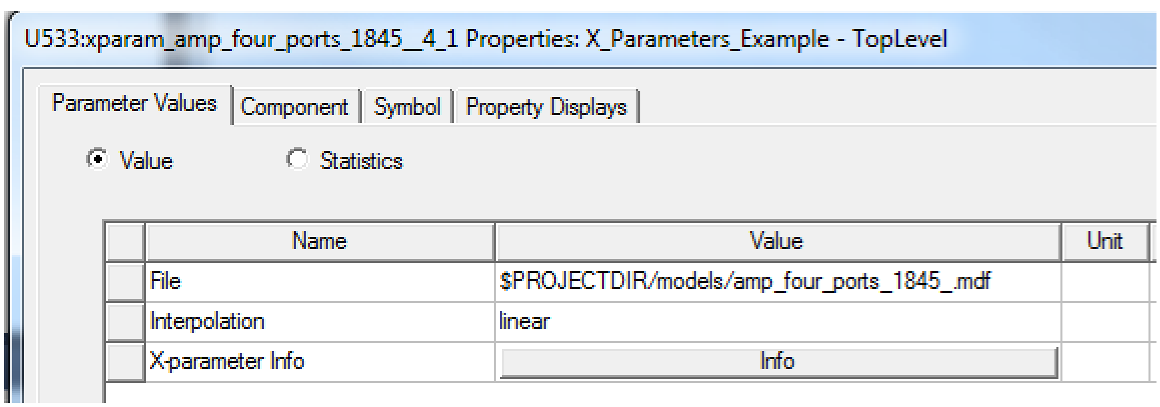
Open the **X_Parameter_Example** Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **Amplifiers** directory, then select the file **X_Parameter_Example.aedt**. Click **Open**.
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

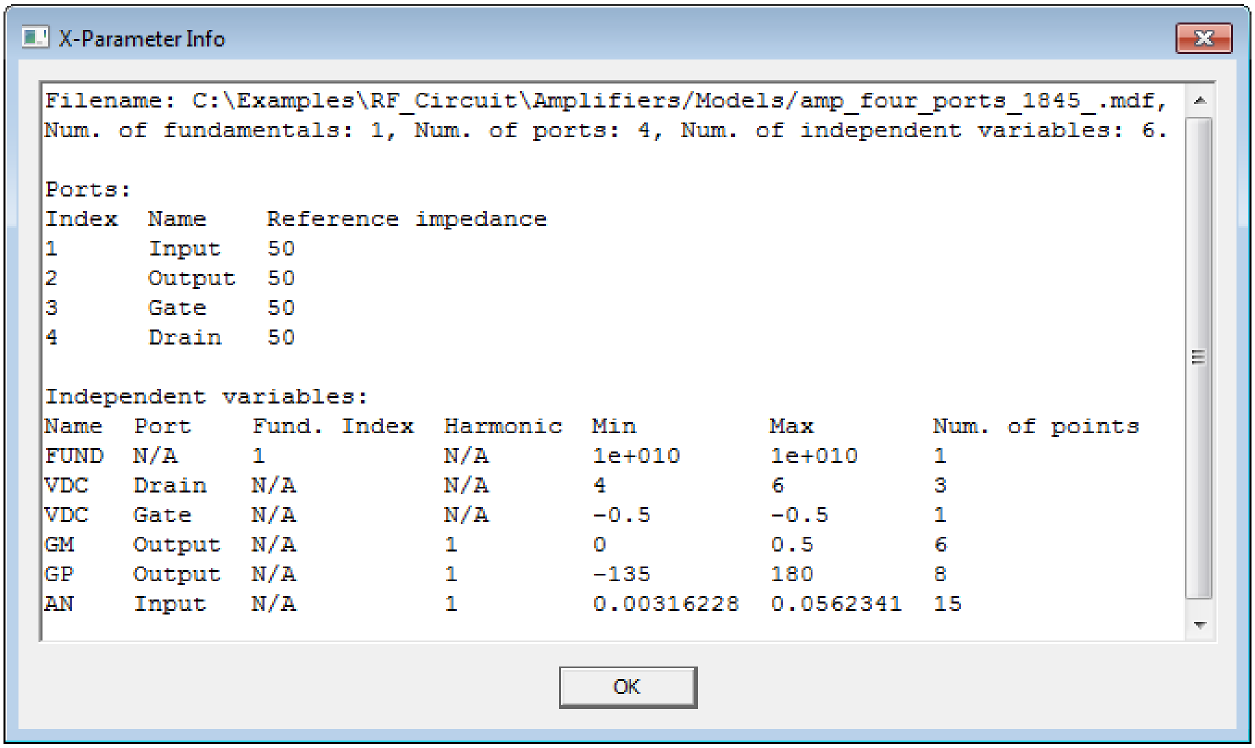
The Project consists of two designs (circuits), **AmpNoBias** with an amplifier circuit design, and **TopLevel** to instantiate the amplifier circuit and compare it to an X-parameter model. Here is the TopLevel schematic:



4. Select the X-parameter component and display its Properties:



5. The **File** property points to the file with the X-parameter data.
6. The **Interpolation** should be **linear**.
7. Click **Info** in the X-parameter Info **Value** field to display a summary of the variables in the data file.



The upper lines show the name of the model file and the numbers of fundamentals (1), ports (4), and independent variables (6).

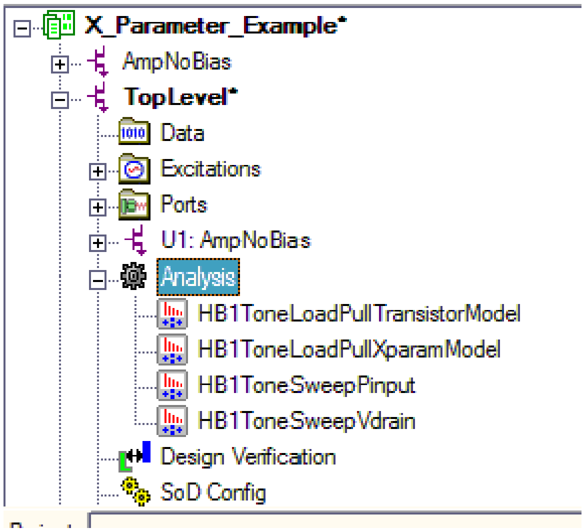
The Ports entries show the name and reference impedance associated with each of the numbered ports.

The X-parameter model was derived for the following variable ranges:

- Variable **FUND**: One fundamental frequency, 10GHz.
- Variable **VDC**: Drain voltage 4V to 6V, 3 sample points.
- Variable **VDC**: Gate voltage -0.5V, 1 sample point.
- Variable **GM**: Magnitude of load reflection coefficient 0.5, 6 sample points.
- Variable **GP**: Phase of load reflection coefficient -135 degrees to 180 degrees, 8 sample points.
- Variable **AN**: Input power ranging from -20dBm to 5dBm, 15 sample points.

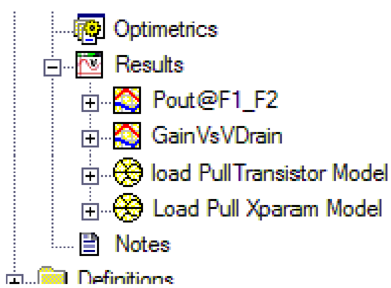
Click **OK** to close the Info window.

8. Expand the **Analysis** icon in the **TopLevel** project.

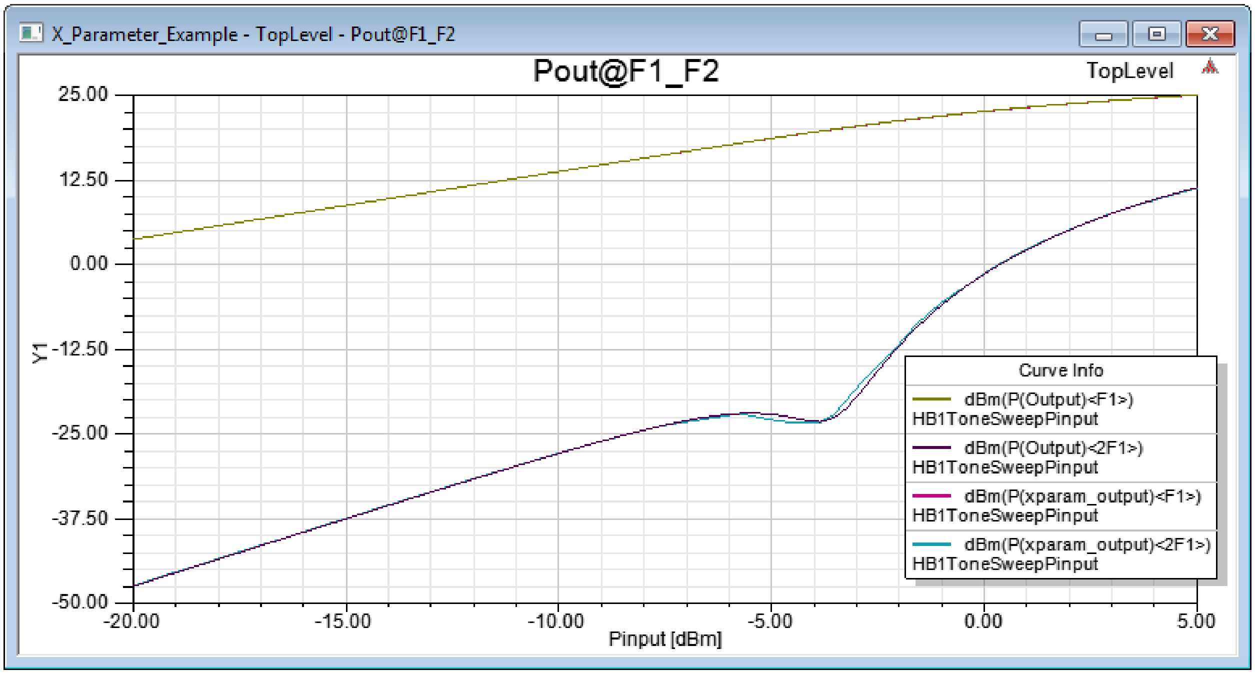


Right-click the Analysis icon and select **Analyze**. (You may need to select **Force Analysis** or change something to get the analysis to run.) All four HB1Tone setups run to completion.

9. Expand the Results icon:



10. Click the **Pout@F1_F2** report to open it.

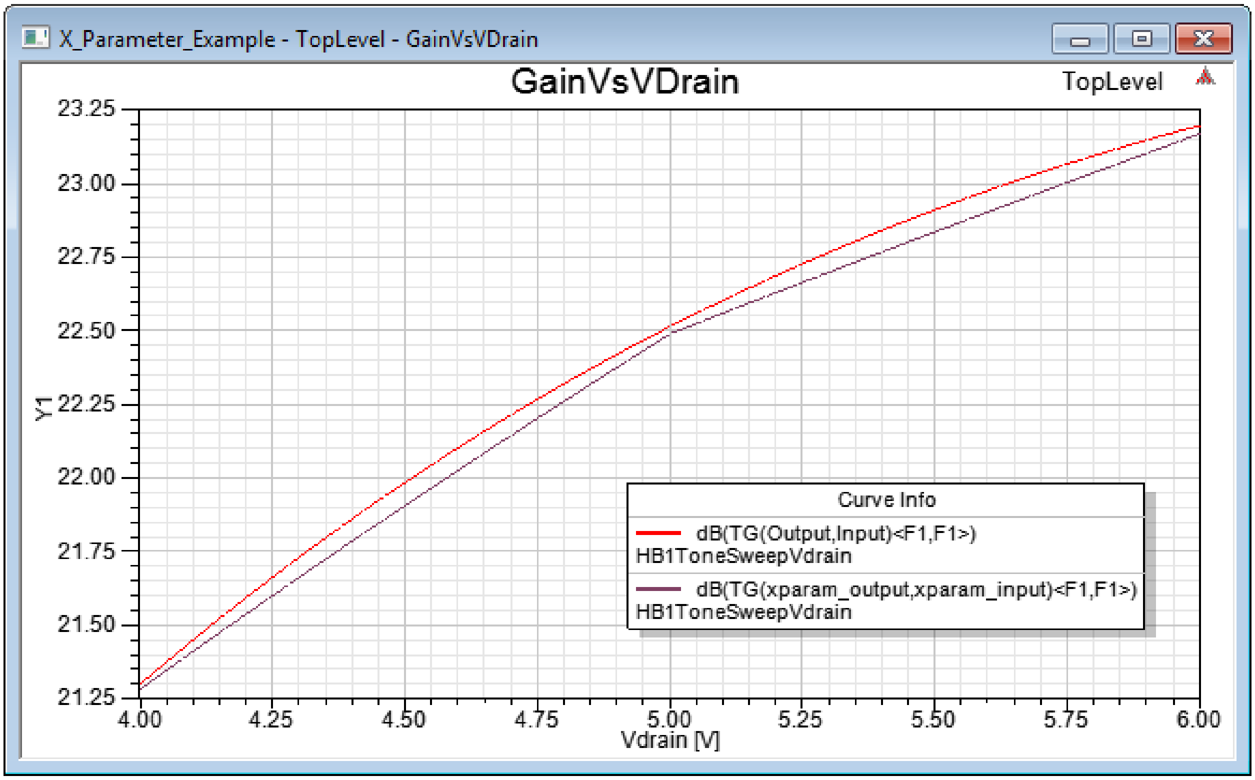


This plot shows the power out versus the power in for the F1 harmonic (the fundamental) and the F2 harmonic (2F1).

The upper line is actually two traces: At the F1 harmonic, the power out vs power in for the transistor model and for the X-parameter model are nearly identical everywhere in this range.

The lower set of traces shows that at the second harmonic, the X-parameter model is very close to the transistor model.

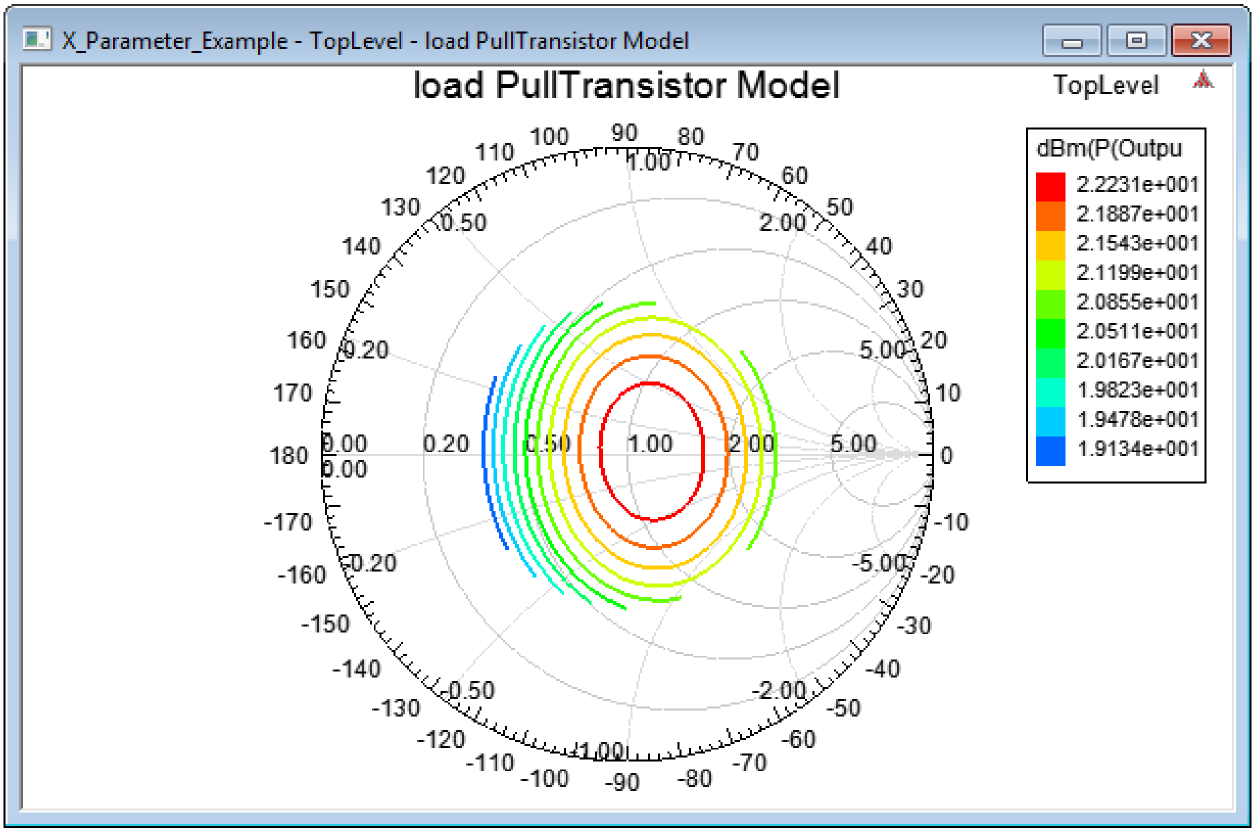
11. Click the **GainVsVDrain** report to open it.



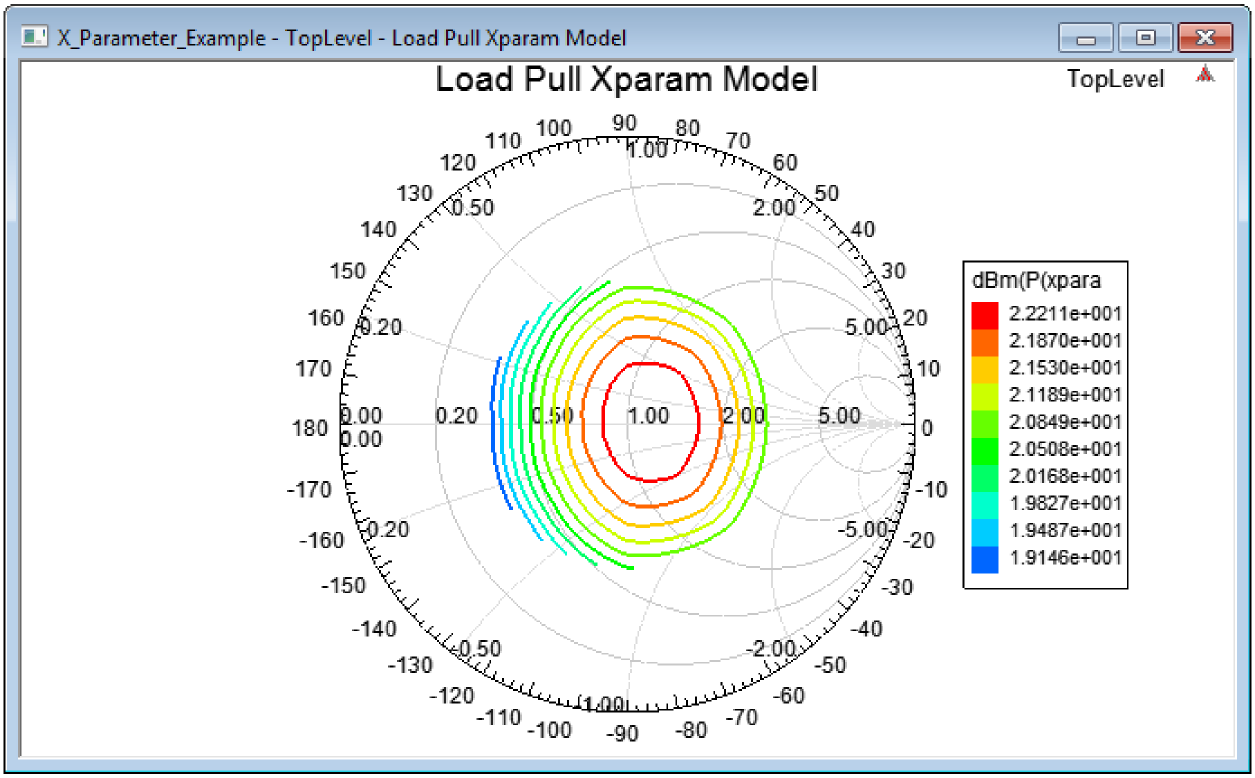
This plot shows the gain as the drain voltage is swept from 4V to 6V.

The set of traces shows that again the X-parameter model is very close to the transistor model.

12. Click the **Load Pull Transistor Model** report to open it.



13. Click the **Load Pull Xparam Model** report to open it.



A comparison of the two Smith Charts shows that the X-parameter model is very close to the transistor model.

14. Click **File>Save** to save the example.

System Frequency Domain Analysis: Receiver Circuit

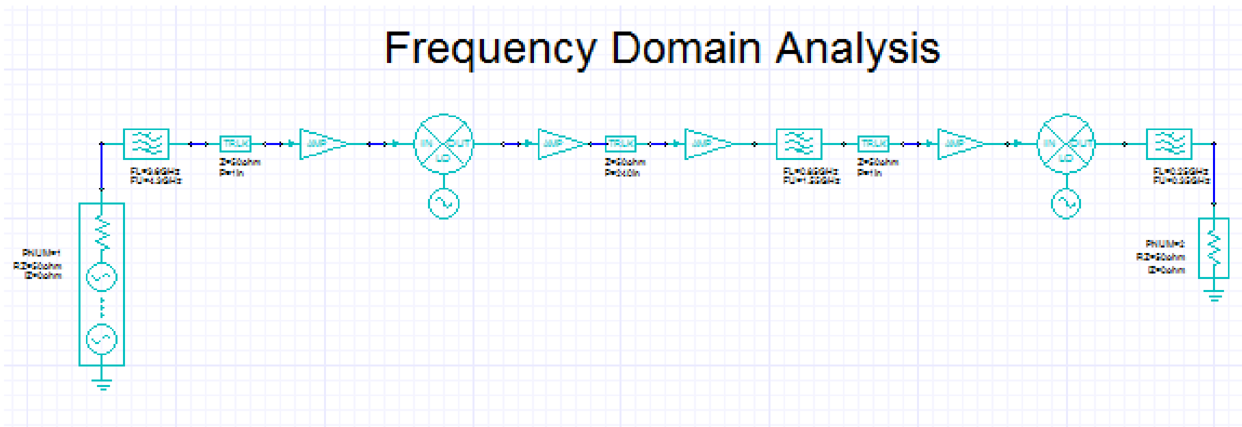
This example shows System Frequency Domain (FD) analysis of a receiver circuit consisting of Nexxim filters and transmission lines combined with Nexsys behavioral MIXER and AMP elements.

The schematic and data file for the example are available in the Examples directory.

Open the project on the Examples directory.

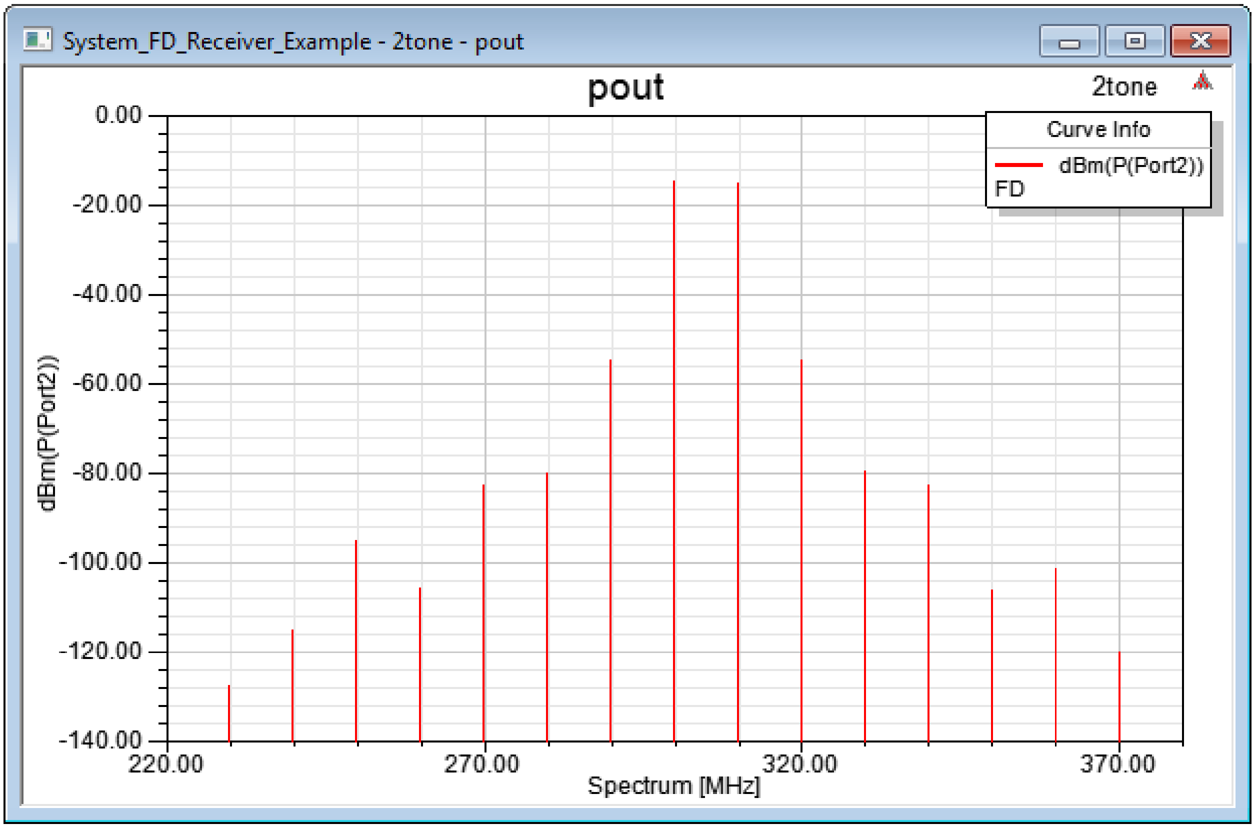
1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **System** directory, then select the file **System_FD_Receiver_Example.aedt**. Click **Open**.

3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.
4. The schematic opens in the design window.



The central transmission line TRLK component is 240 inches (20 feet) in length. The AMP elements on the transmitter and receiver ends of the long TRLK use NMF files for the compression data. The other AMP elements and the MIXER elements have S-parameters specified as component properties.

5. Expand the Analysis folder, right-click and select Analyze (you may need to use Force Analysis).
6. When the analysis is complete, expand the Results folder and click the **pout** report to see the spectral distribution of the output power.



7. Click **File>Save** to save the project.

For more information, see System Frequency Domain Analysis.

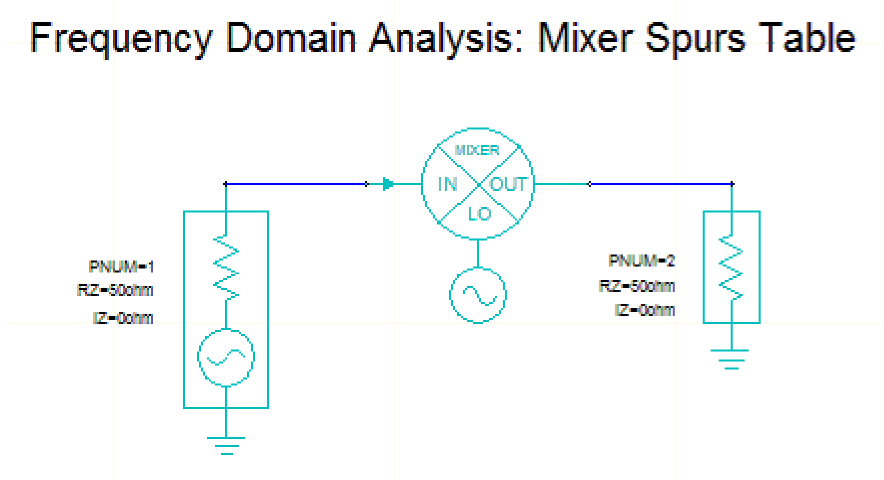
System Frequency Domain Analysis: Mixer Data

This example demonstrates System Frequency Domain analysis of a behavioral MIXER element with mixer spurs data from a NMF file. The schematic and data file for the example are available in the Examples directory.

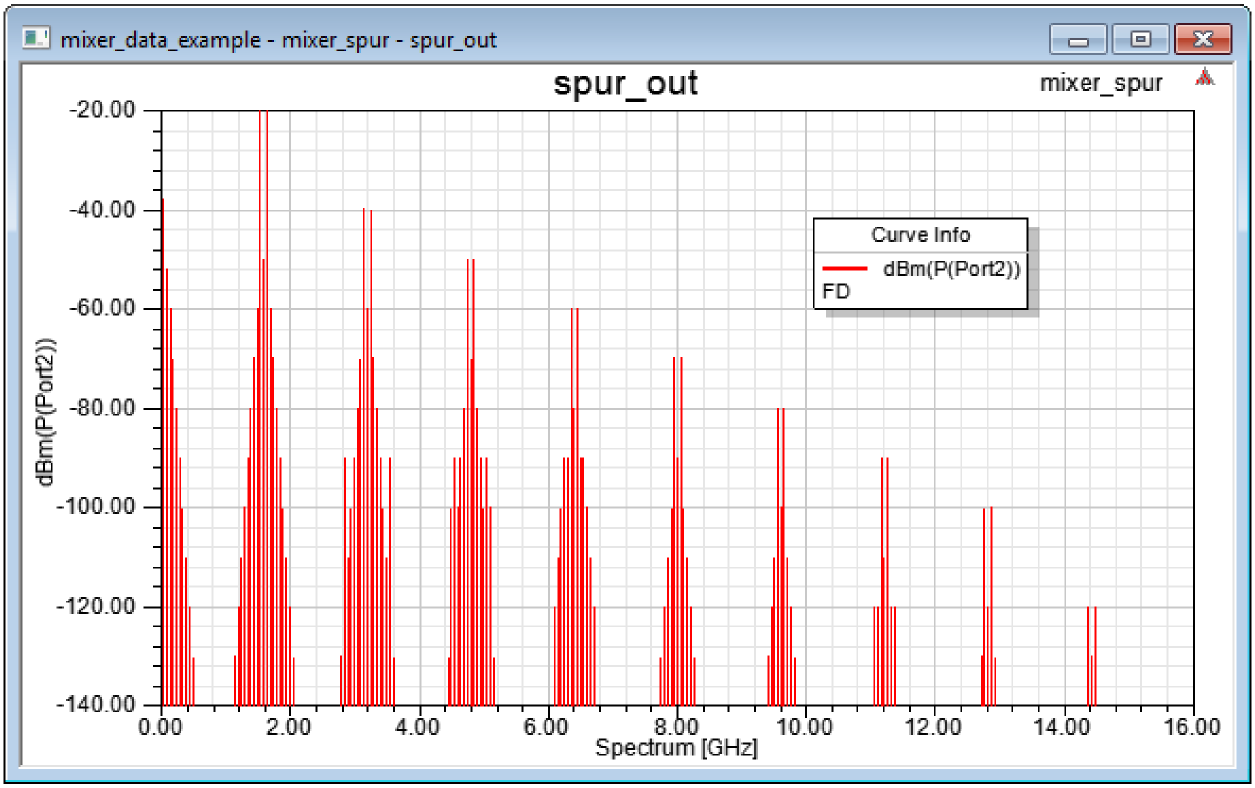
Open the project on the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **System** directory, then select the file **System_FD_Mixer_Data_Example.aedt**. Click **Open**.
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

4. The schematic opens in the design window:



5. Select the MIXER component to display its properties. The mixer reads the mixer spurs data from a data file, NMFSpurTable.nmf (**file** property).
6. Expand the Analysis folder, right-click and select Analyze (you may need to use Force Analysis).
7. When the analysis is complete, expand the Results folder and click the **spur_out** report to see the spectral distribution of the mixer spurs output.



8. Click **File>Save** to save the project.

For more information, see System Frequency Domain Analysis.

DC-IV Characteristics Transistor Example

This example demonstrates how to perform DC-IV characterization and S-parameter sweep for various transistors. The example project can be used as a template to customize the simulation and serve as a starting point for an amplifier design.

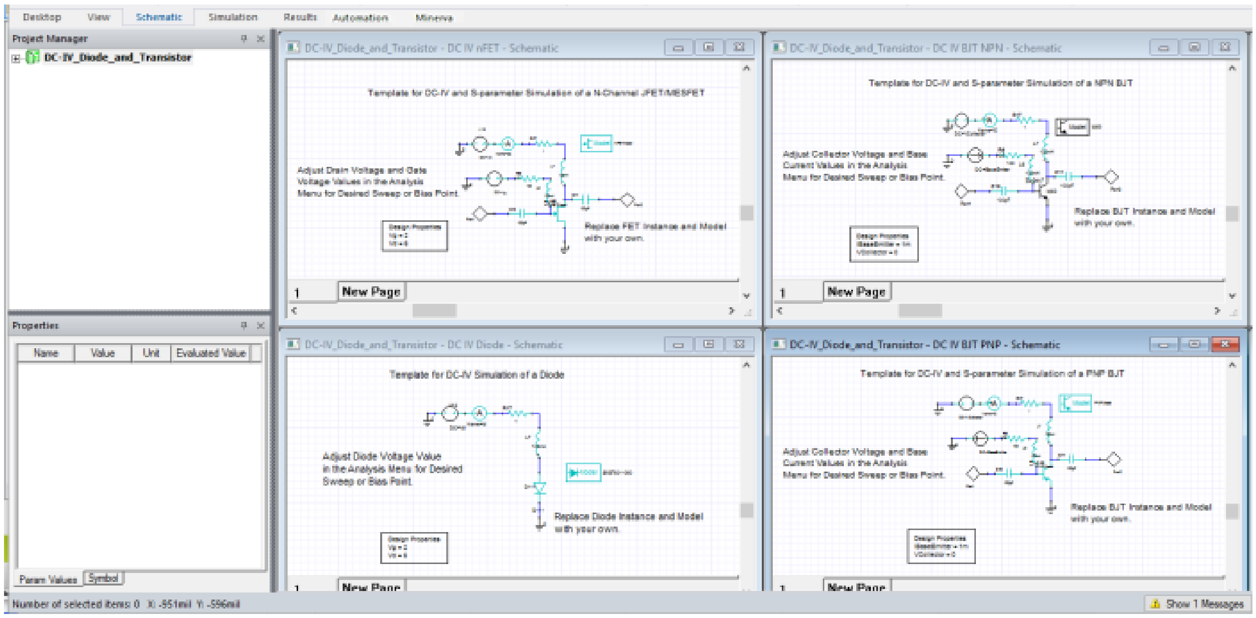
Open the DC-IV Example

This example demonstrates the I-V Characteristic Curves of an electrical device that show the relationship between the current flowing through the device and the applied voltage across its terminals.

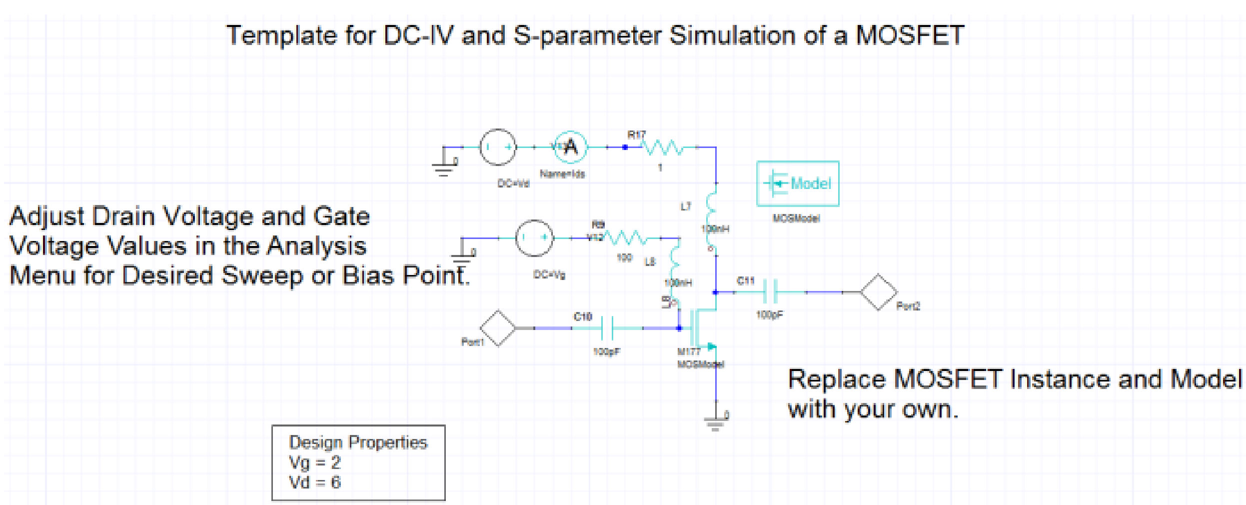
The schematic and data file for the example are available in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. In the **File Open** window, open the **Electronics Desktop** directory, then **Circuit**, then **RF_Microwave**, the **DC-IV** directory, then select the file **DC-IV_Diode_and_ Transistor.aedt** and click **Open**.

3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission, then click **OK**. The schematics for the **DC-IV_Diode_and_Transistor.aedt** project open in the design window.



4. In the **Project Tree**, open **DC-IV_Diode_and_Transistor** and select the **DC IV MOSFET** design, or any design that is listed. In the design window double-click the design to make it appear full screen.



5. Right-click in the design to open the **Design Properties** window and adjust any of the properties to meet your needs. The transistor model and instance can even be replaced with your own versions. For more information, see Plot I-V Curves of Active Components.

6. Double-click an I-V solution beneath **Analysis** to make adjustments to the voltage, frequency sweeps, or the lumped circuit element parameters. For more information, see Plot I-V Curves of Active Components.
7. Click **File>Save** to save the project.

Complete the DC-IV Analysis

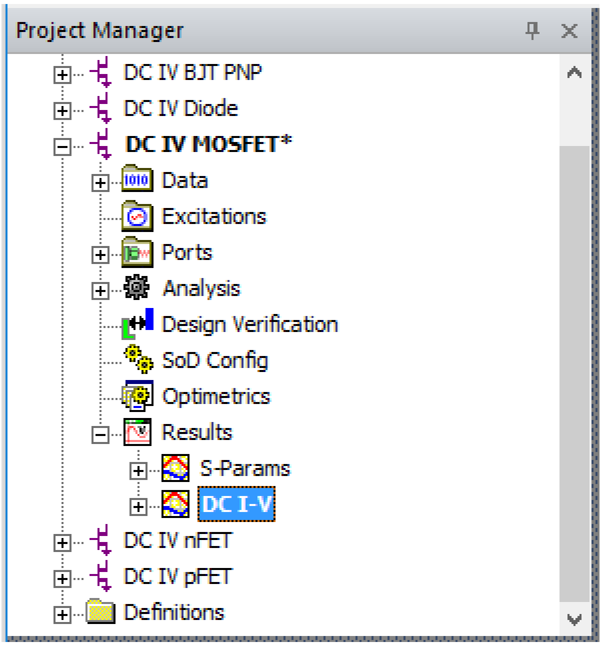
To perform the DC-IV analysis:

1. In the **Project Tree**, right-click a **DC-IV** analysis, such as **DC IV MOSFET**, then select **Analyze** on the menu.

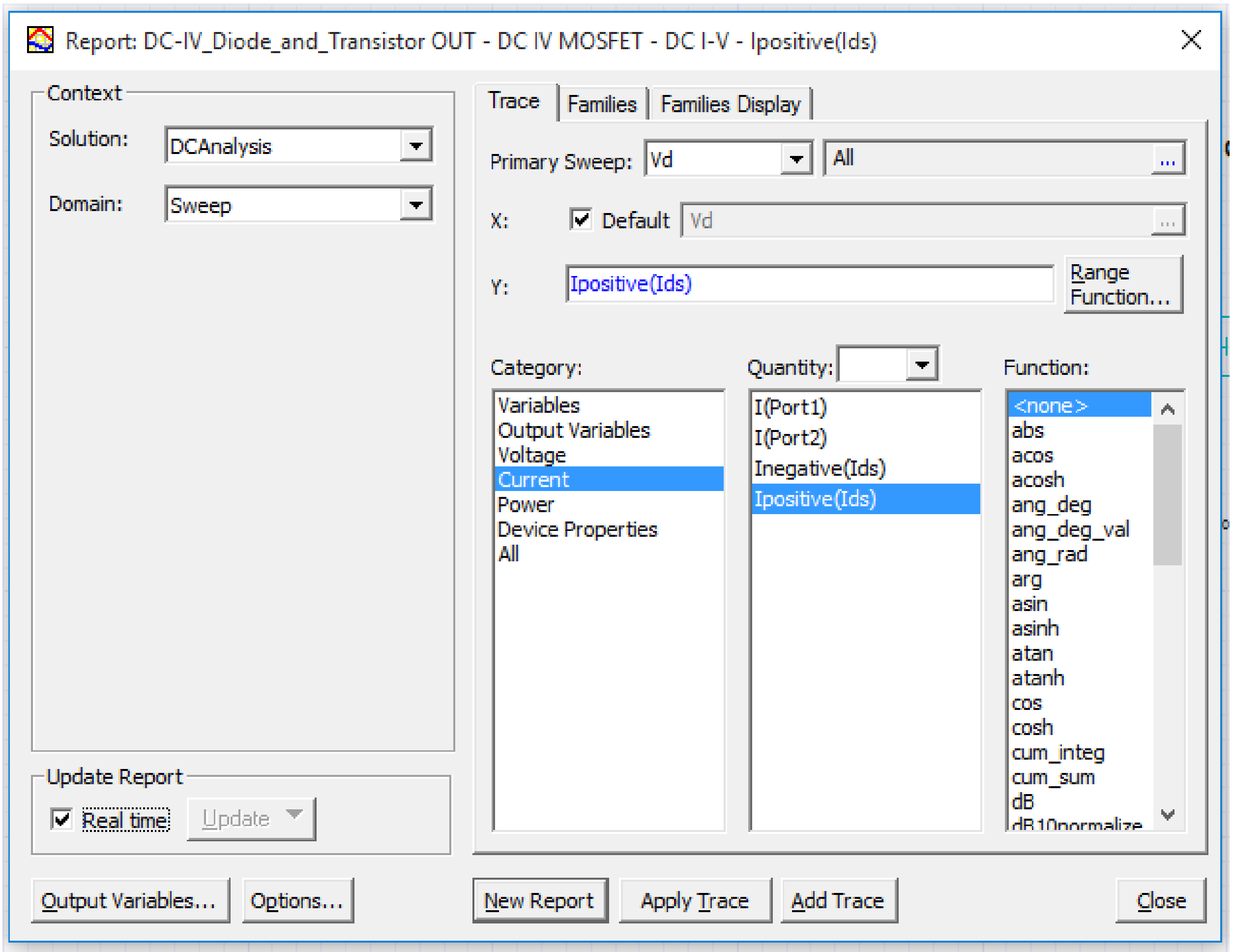
View the Results of the DC-IV Analyses

When the analyses have completed, the reports that have been set up become valid.

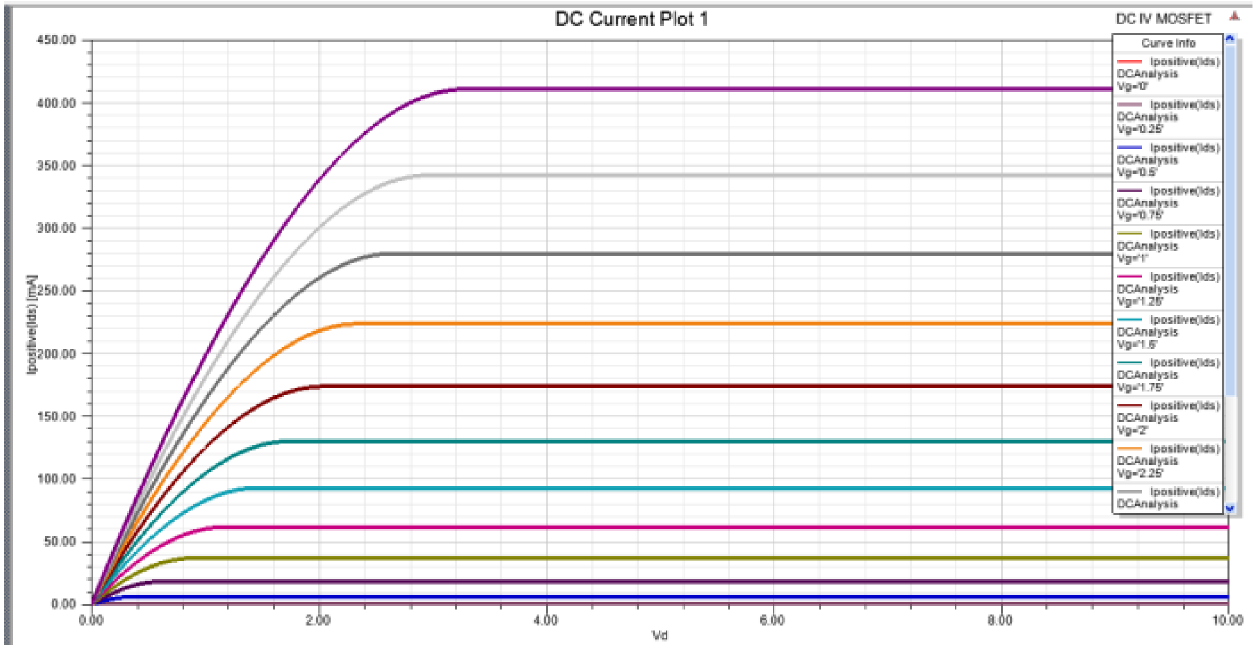
1. From the **Project Manager** window, expand the **Results** icon:



2. Right-click the **DC-IV** report and select **Modify Report** to open the setup for the Rectangular Plot report:



3. After making your modifications, click **New Report** ; a new report appears in the **Project Tree** within **Results**.



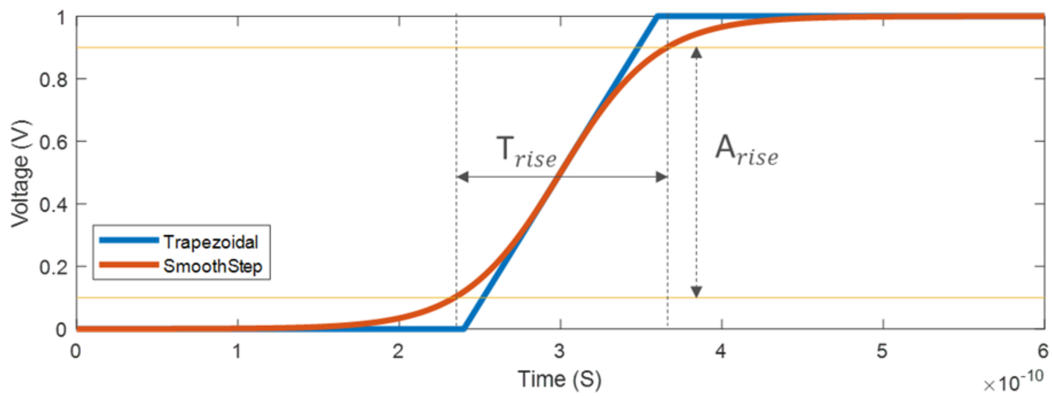
Time Domain Reflectometer Example

This topic presents differential and single-ended versions of a Time Domain Reflectometer (TDR). The setup demonstrates the process of analyzing both impedance and delay. The schematics are configured to substitute another design for the Device Under test, then run the analysis and view the results. The project includes an example showing how to obtain the equivalent reactance (capacitance or inductance) for a non-uniformity that is detected by the TDR.

Single-Ended TDR

The schematics for all three TDR demonstrations are in one project in the Examples directory.

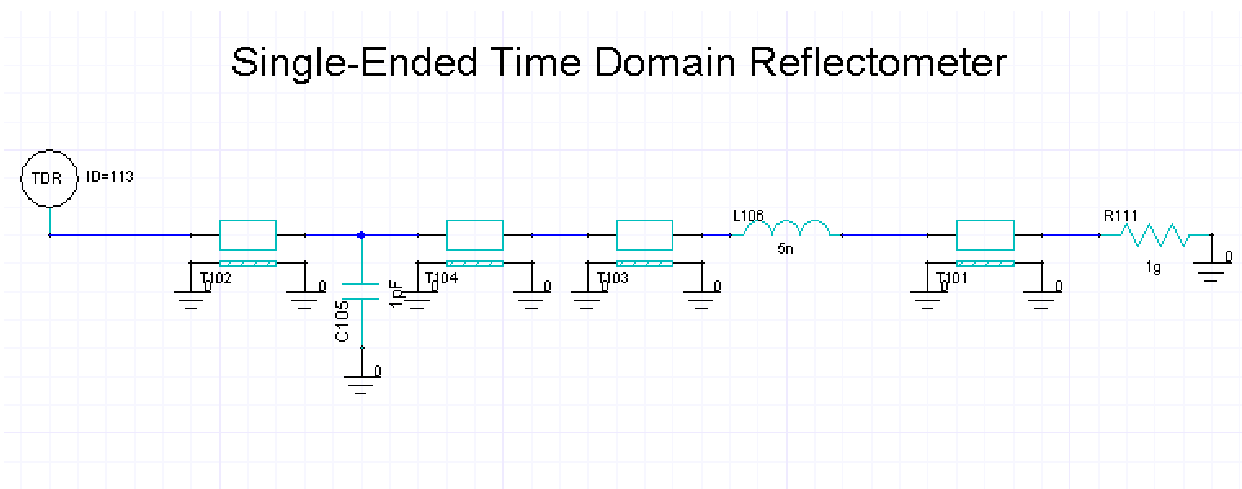
The following example is for a Single-Ended Time Domain Reflectometer using a Trapezoidal waveform. The TDR probe also provides the option for a Smooth Step waveform, which provides a more realistic step input and allows control of the rise interval. Step 4 is optional; it demonstrates how to choose between the Trapezoidal and SmoothStep waveform for analysis.



Open the TDR Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open a file explorer window.
2. Select **Circuit > Signal Integrity**, scroll to find and select **TDR_Example.aedt**, click **Open**.

   In the Project Tree, Expand the **TDR_Example** icon and the **1_Single Ended** icon. The single-ended TDR schematic appears in the design window:



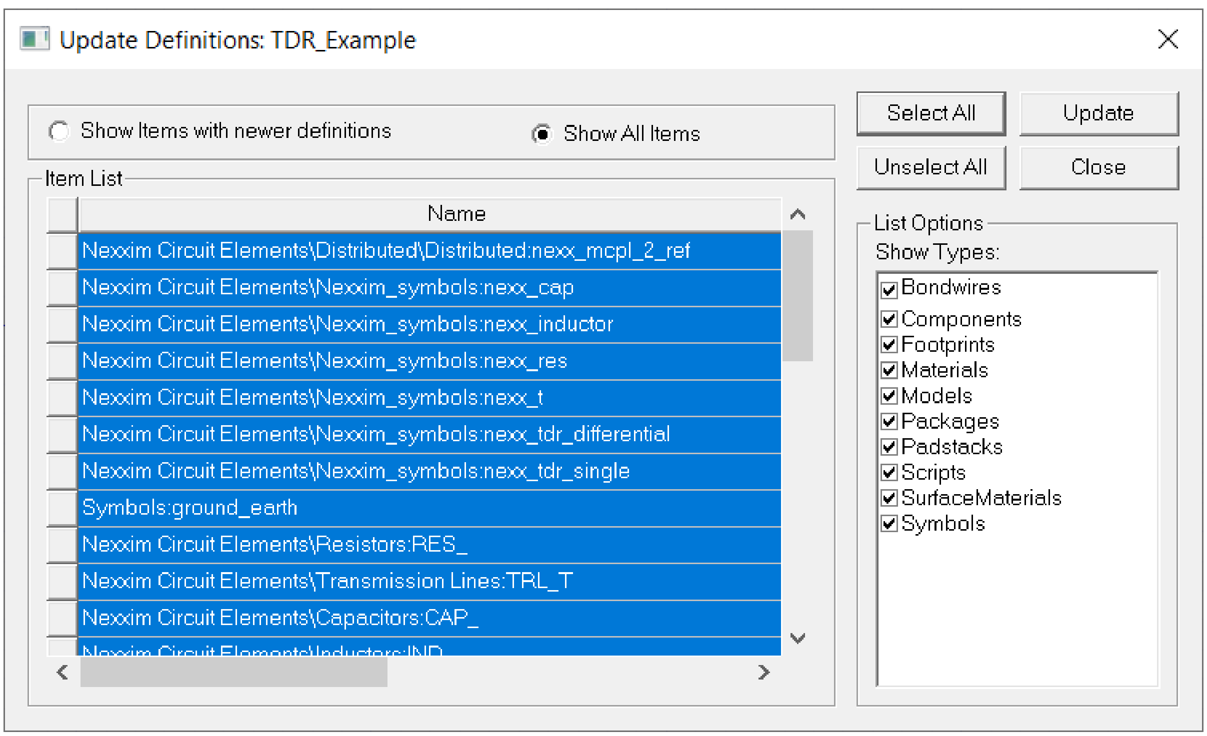
The single-ended TDR consists of a unit pulse source, an internal impedance of Z0 ohm, a reference length of transmission line (Z0-ohm impedance, 0.5ns delay), and two voltage probes, Vexcited_pos for the step voltage and Vdetected_pos for the reflected voltage. (Z0 is the TDR component parameter.)

The Device Under Test (DUT) is a series of transmission lines and discrete passives to represent the typical TDR problem of finding where impedance changes in a line.

In sequence, the DUT elements are:

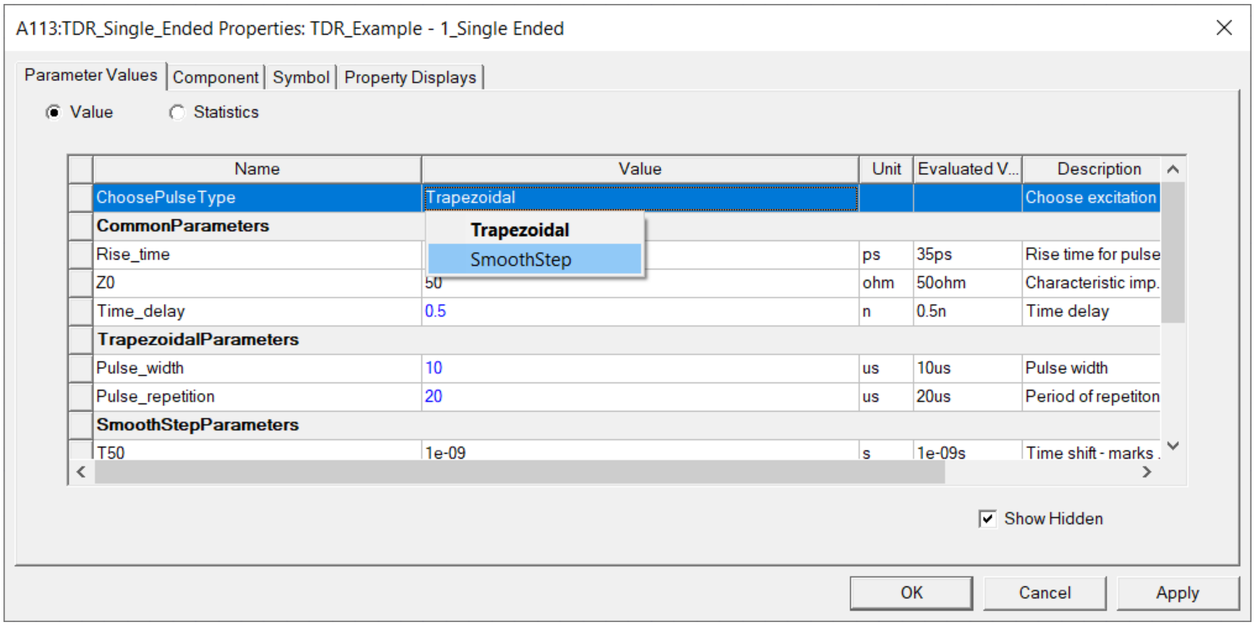
- Transmission line, 50-ohm impedance, 1 ns delay
- Shunt capacitor, 1 pf
- Transmission line, 50-ohm impedance, 3 ns delay
- Transmission line, 55-ohm impedance, 5 ns delay
- Series inductor, 5 nH
- Transmission line, 25-ohm impedance, 2 ns delay
- Series resistor, 1 gigohm
- Terminating ground

3. The Examples directory is write-protected. Use **File** > **Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.
4. (Optional) Choose between Trapezoidal and SmoothStep.
   i. Click **Tools** > **Project Tools** > **Update Definitions**.
   ii. In the **Update Definitions** dialog, select **Show All Items**, **Select All**, then click **Update** . Close the dialog once updates are successful.

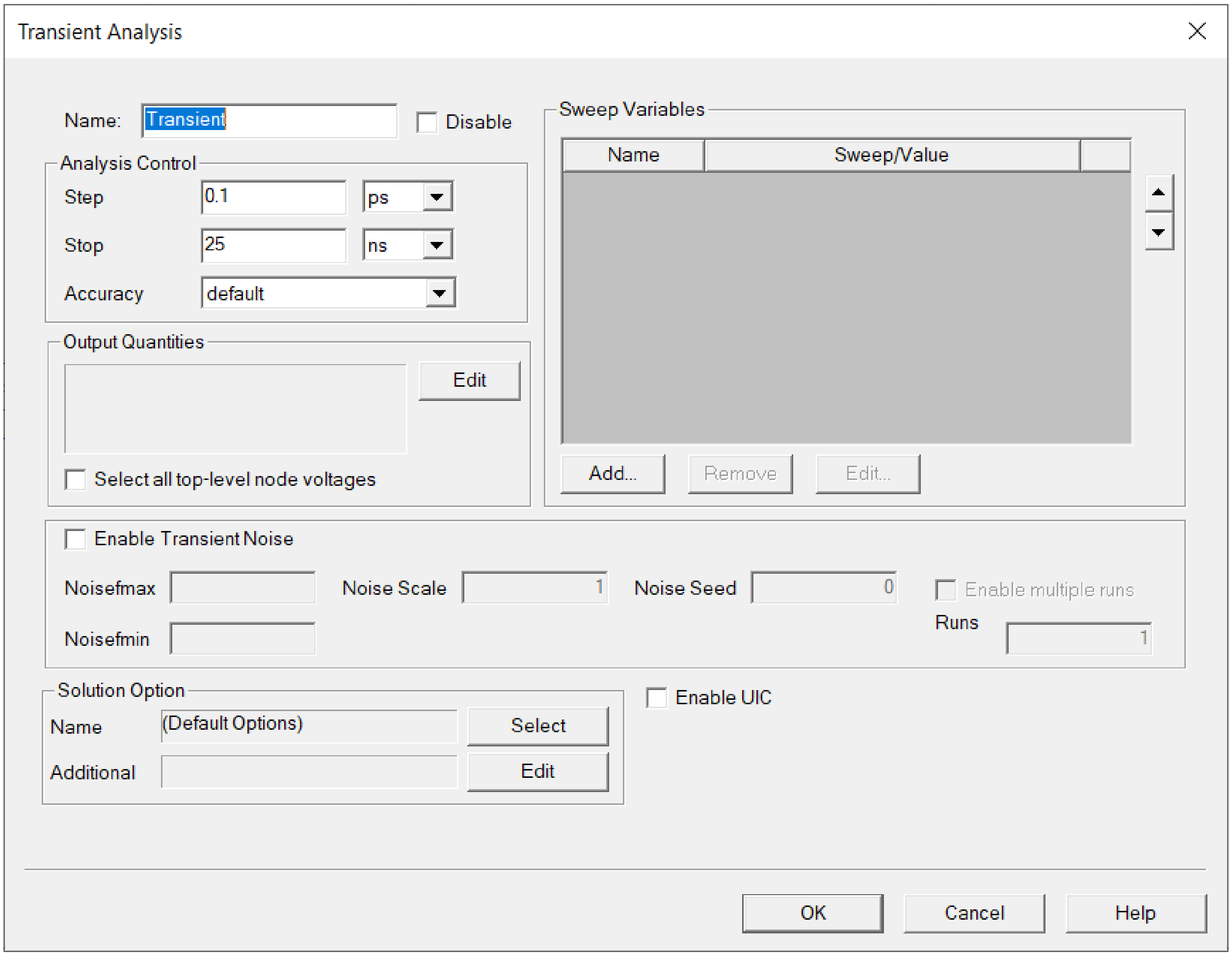


   iii. Right-click the TDR model symbol and click **Properties** to open the properties dialog.

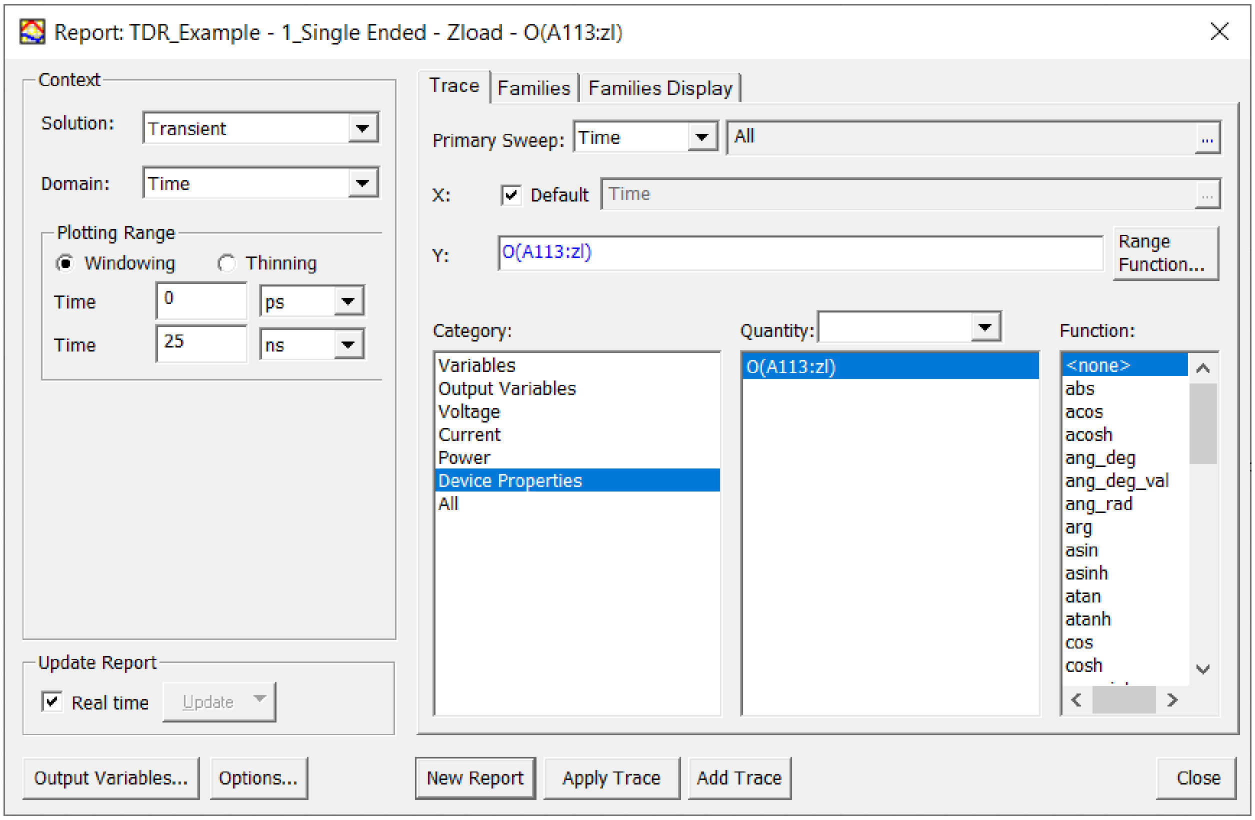
iv. Click the **Value** field for **ChoosePulseType** to view the two options. Select **SmoothStep**, click **Apply**, then **OK** to close the dialog.



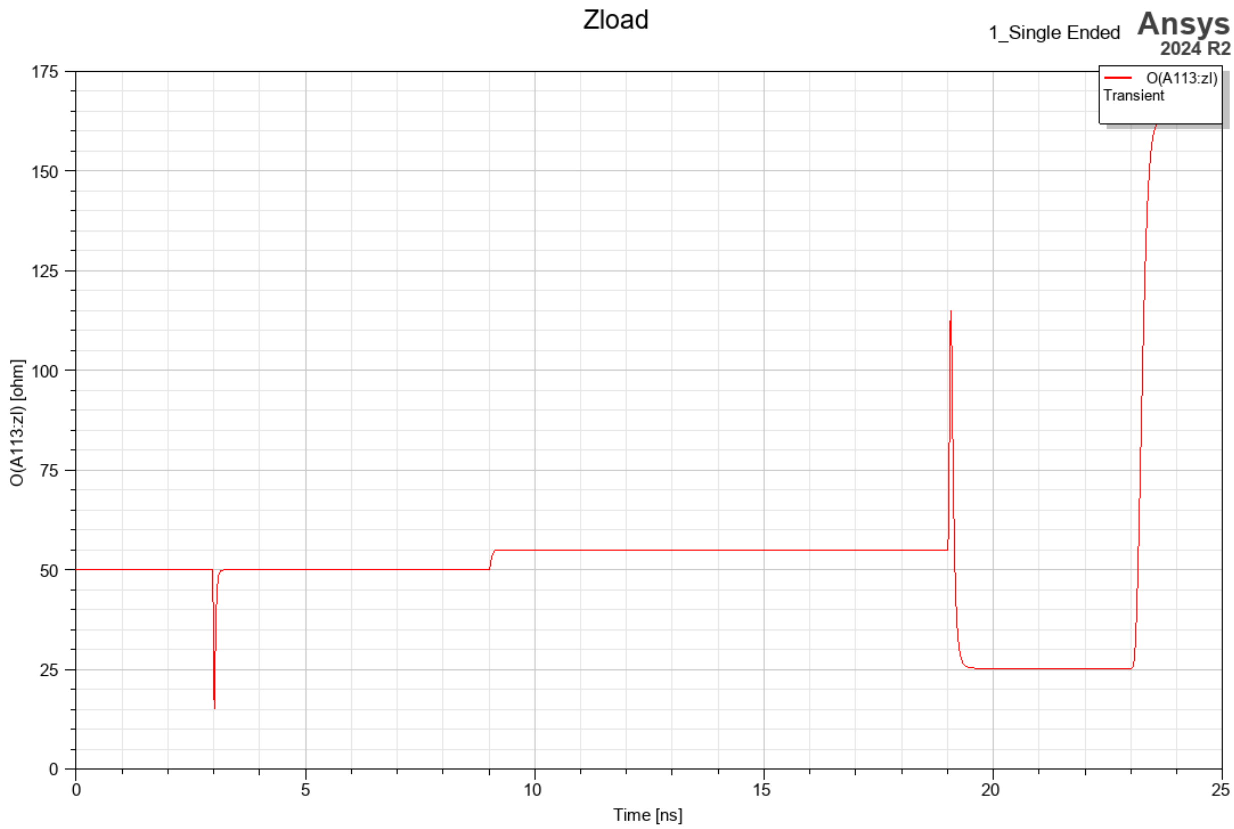
5. In the **Project Tree**, expand the Analysis icon and double-click on **Transient** to view the Transient Analysis setup. Click **OK**.



6. In the **Project Tree**, right- click **Transient** and select **Analyze**. The analysis runs to completion.
7. Expand the Reports icon and right-click on the **Zload** report setup. Select **Modify Report** to open the Report Setup.


Report: TDR_Example - 1_Single Ended - Zload - O(A113:zl)
Context
Solution: Transient
Domain: Time
Plotting Range
Windowing
Thinning
Time 0 ps
Time 25 ns
Update Report
Real time
Update
Trace
Families
Families Display
Primary Sweep: Time
All
X:
Default
Time
Y:
O(A113:zl)
Range Function...
Category:
Variables
Output Variables
Voltage
Current
Power
Device Properties
All
Quantity:
O(A113:zl)
Function:
<none>
abs
acos
acosh
ang_deg
ang_deg_val
ang_rad
arg
asin
asinh
atan
atanh
cos
cosh
Output Variables...
Options...
New Report
Apply Trace
Add Trace
Close

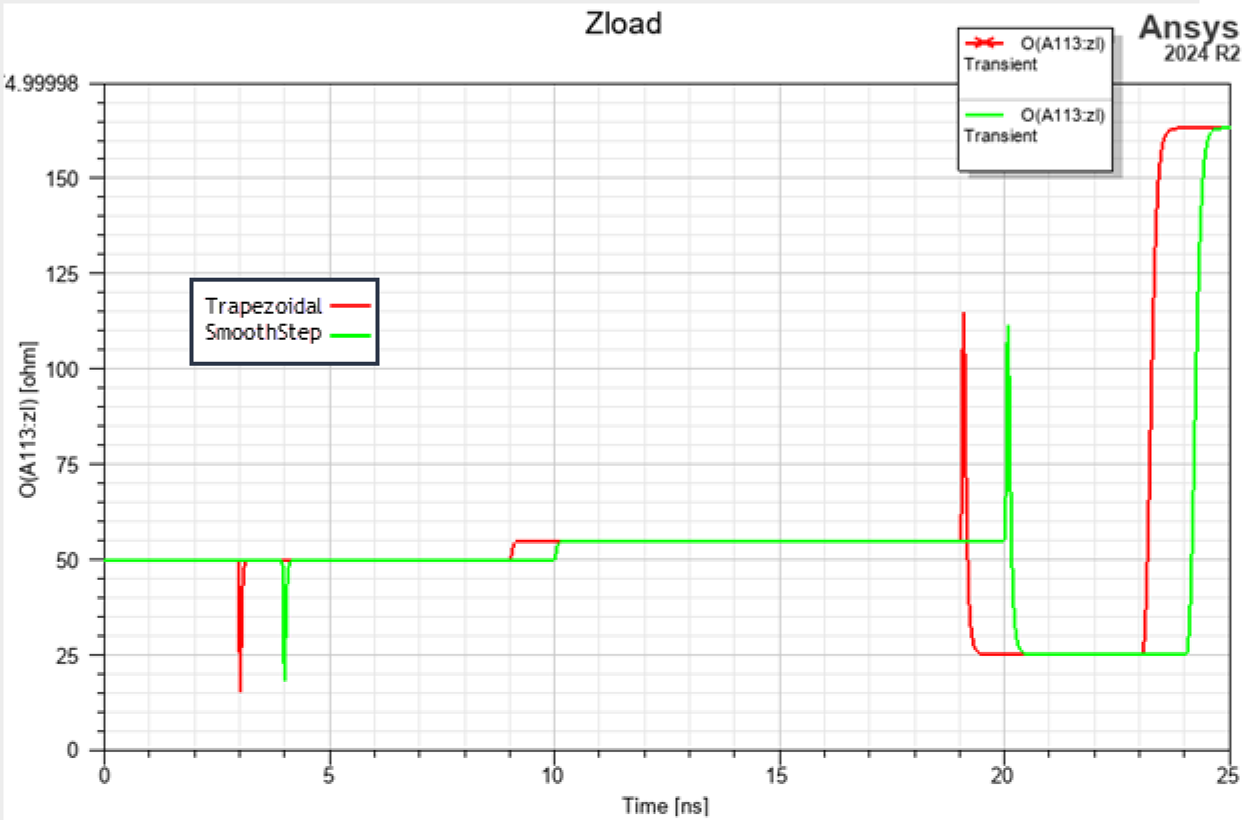
8. Click **New Report** to generate the report:



The TDR analysis allows us to identify each of the components of the DUT by their timing and behavior. Using the Trapezoidal waveform, the reflection timings are expected at 2*(Total Delay):

- At 3ns, the impedance drops to indicate the presence of the capacitor. The 3ns delay is the TDR internal delay (0.5ns) plus the first transmission line delay (1ns), times two for the reflection. Both transmission lines are at 50 ohms.
- At 9ns, the impedance increases to 55 ohm at the beginning of the 55-ohm line segment. 9ns=2(0.5+1.0+3.0).
- At 19ns, the impedance jumps to indicate the presence of the inductor, then immediately drops to25 ohm for the resistor. 19ns=2((0.5+1.0+3.0+5.0).
- At 23ns, the impedance goes to infinity at the high final resistor. 23ns=2(0.5+1.0+3.0+5.0+2.0).

Note: If SmoothStep waveform is selected, the parameter T50 will have an effect on the time-shift of the plot. See **Circuit > Nexxim Components > Probes > Time Domain Reflectometer** for more information about TDR properties, including the T50 SmoothStep parameter.



For a SmoothStep waveform, the reflection timings are expected at T50 + 2*(Total Delay):

- At 4ns, the impedance drops to indicate the presence of the capacitor. The 4ns delay is the TDR internal delay (0.5ns) plus the first transmission line delay (1ns), times two for the reflection, plus 1 ns for the T50 parameter. Both transmission lines are at 50 ohms.
- At 10ns, the impedance increases to 55 ohm at the beginning of the 55-ohm line segment. 10ns= 1 + 2(0.5+1.0+3.0).
- At 20ns, the impedance jumps to indicate the presence of the inductor, then immediately drops to 25 ohm for the resistor. 20ns= 1 + 2((0.5+1.0+3.0+5.0).

- At 24ns, the impedance goes to infinity at the high final resistor.
  24ns= 1 + 2(0.5+1.0+3.0+5.0+2.0).

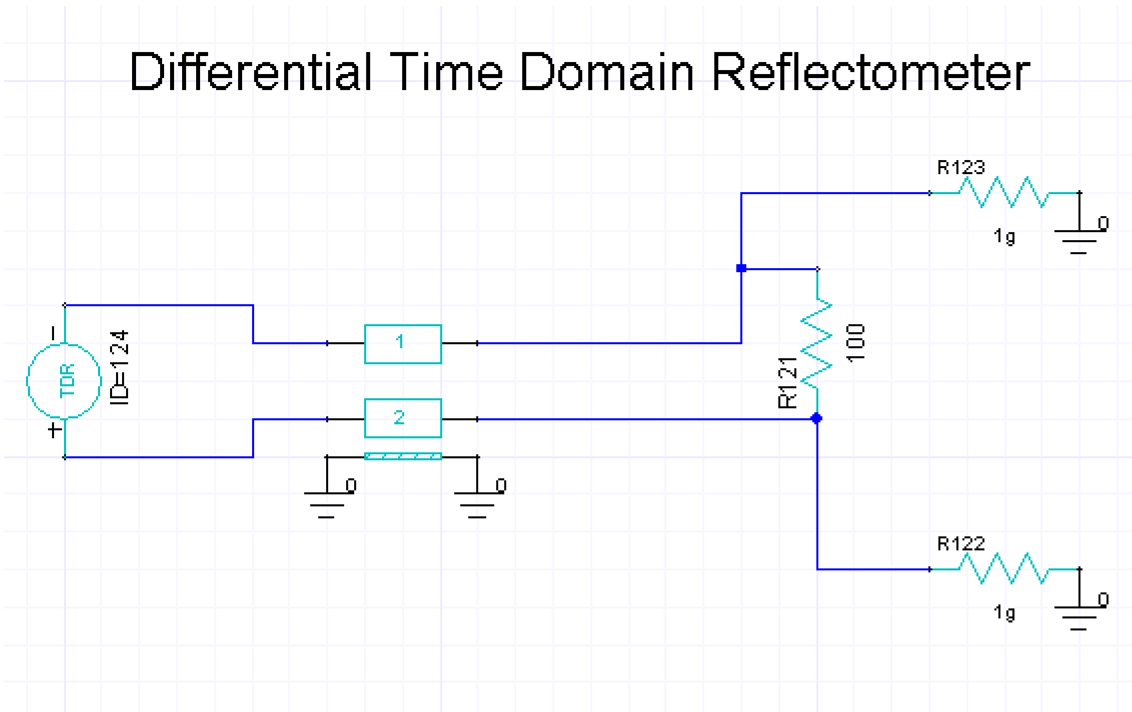
.

Differential TDR

Open the TDR Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open **Circuit**, then **Signal Integrity**, and select the file **TDR_Example.aedt**. Click **Open**.

   Expand the **TDR_Example** icon and the **2_Differential** icon. The differential TDR schematic appears in the design window:



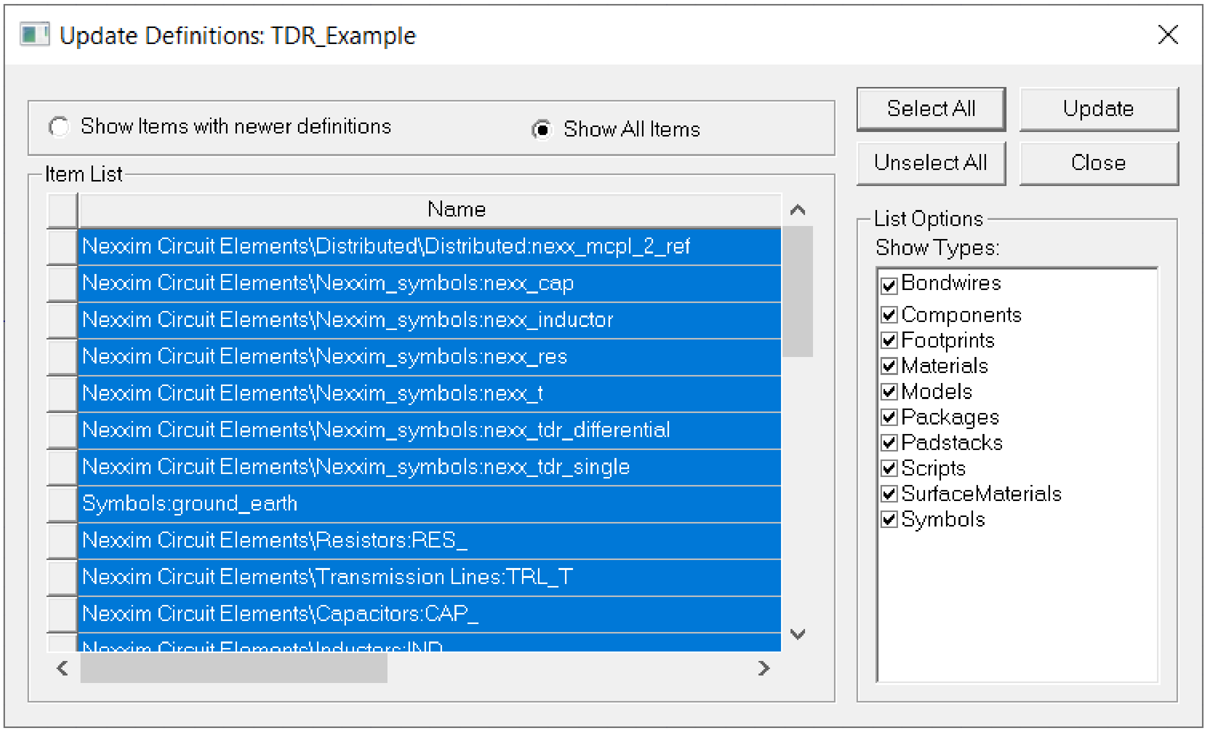
The differential TDR consists of two channels, each with a pulse source, an internal impedance equal to 0.5Z0 ohm, a transmission line with 0.5Z0-ohm impedance and with 0.5-ns delay, and two voltage probes, Vexcited_pos or Vexcited_neg for the step voltage and Vdetected_pos or Vdetected_neg for the reflected voltage. (Z0 is the TDR component parameter.)

The TDR component provides two options for the pulse type: Trapezoidal and SmoothStep. SmoothStep provides a more realistic step input and allows control of the rise interval.

(Optional) Choose between Trapezoidal and SmoothStep.

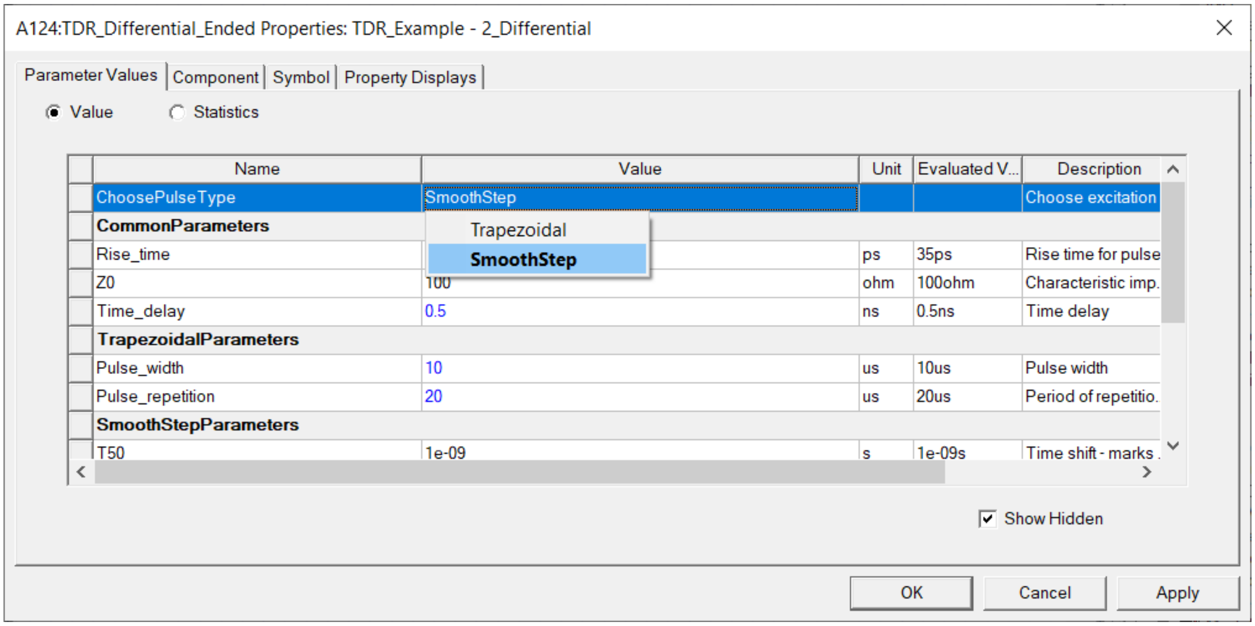
i. Click **Tools** > **Project Tools** > **Update Definitions**.

ii. In the **Update Definitions** dialog, select **Show All Items**, **Select All**, then click **Update** . Close the dialog once updates are successful.



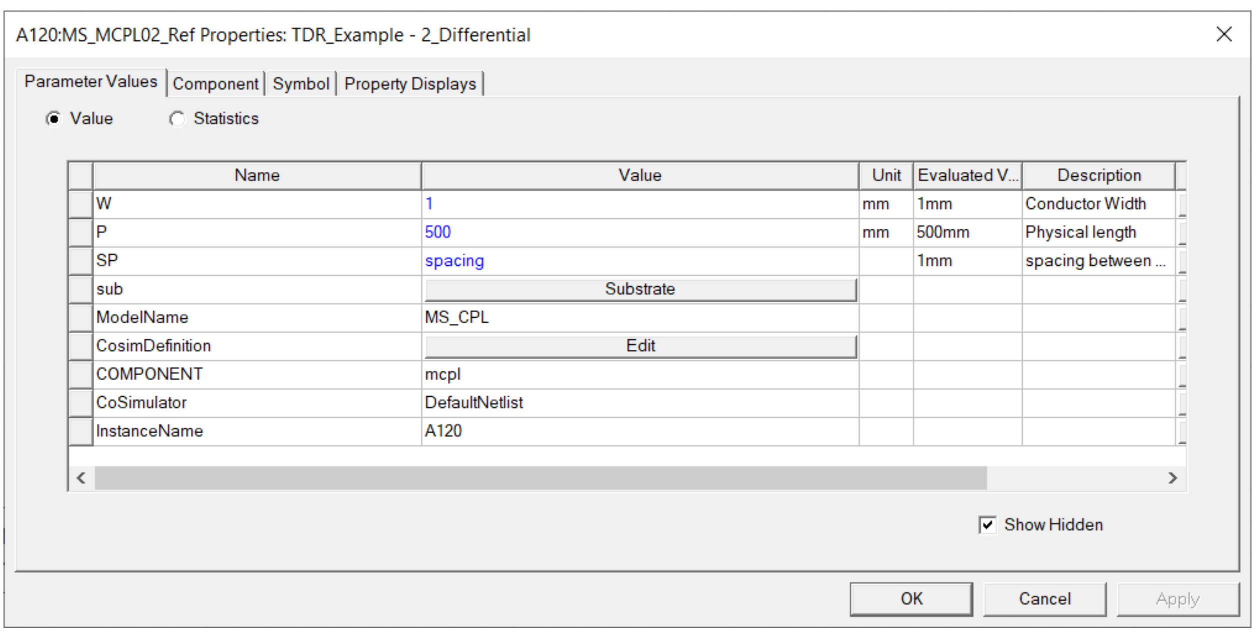
iii. Right-click the TDR model symbol and click **Properties** to open the properties dialog.

iv. Click the **Value** field for **ChoosePulseType** to view the two options. Select **SmoothStep**, click **Apply**, then **OK** to close the dialog.

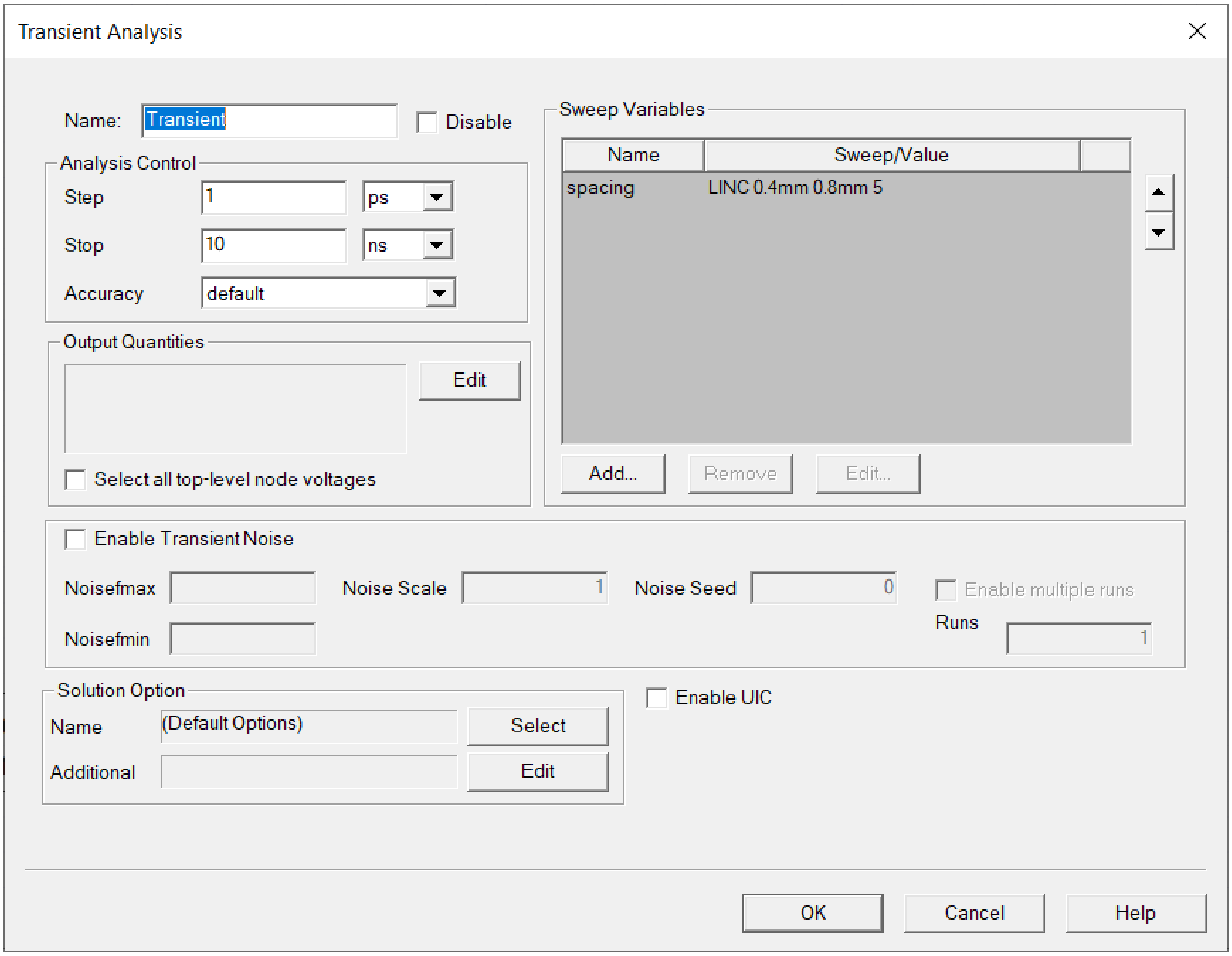


The Device Under Test (DUT) is microstrip coupled line pair and a termination network. The transient analysis sweeps the spacing between the coupled lines, which shows how the spacing affects the differential impedance, Zdiff.

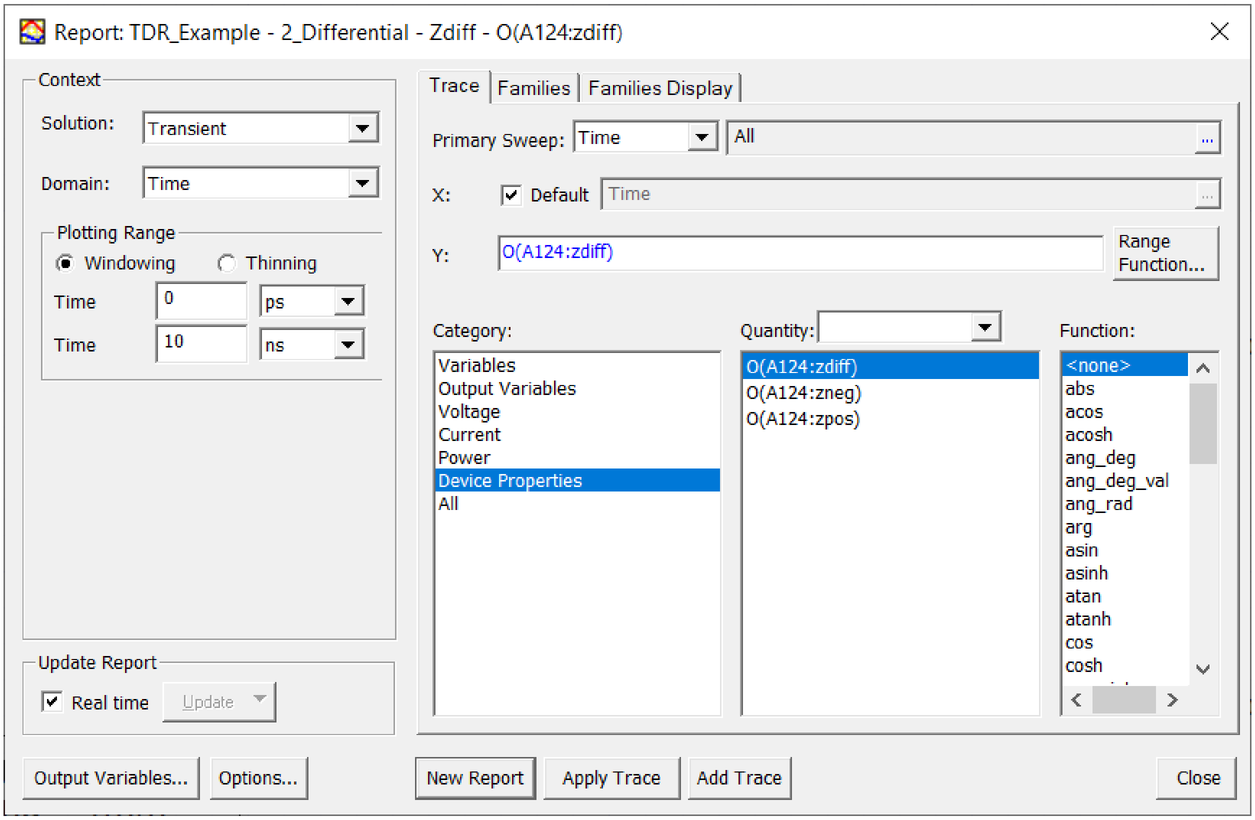
1. Select the coupled lines. The Property window shows that the spacing parameter SP has been defined with a local variable.



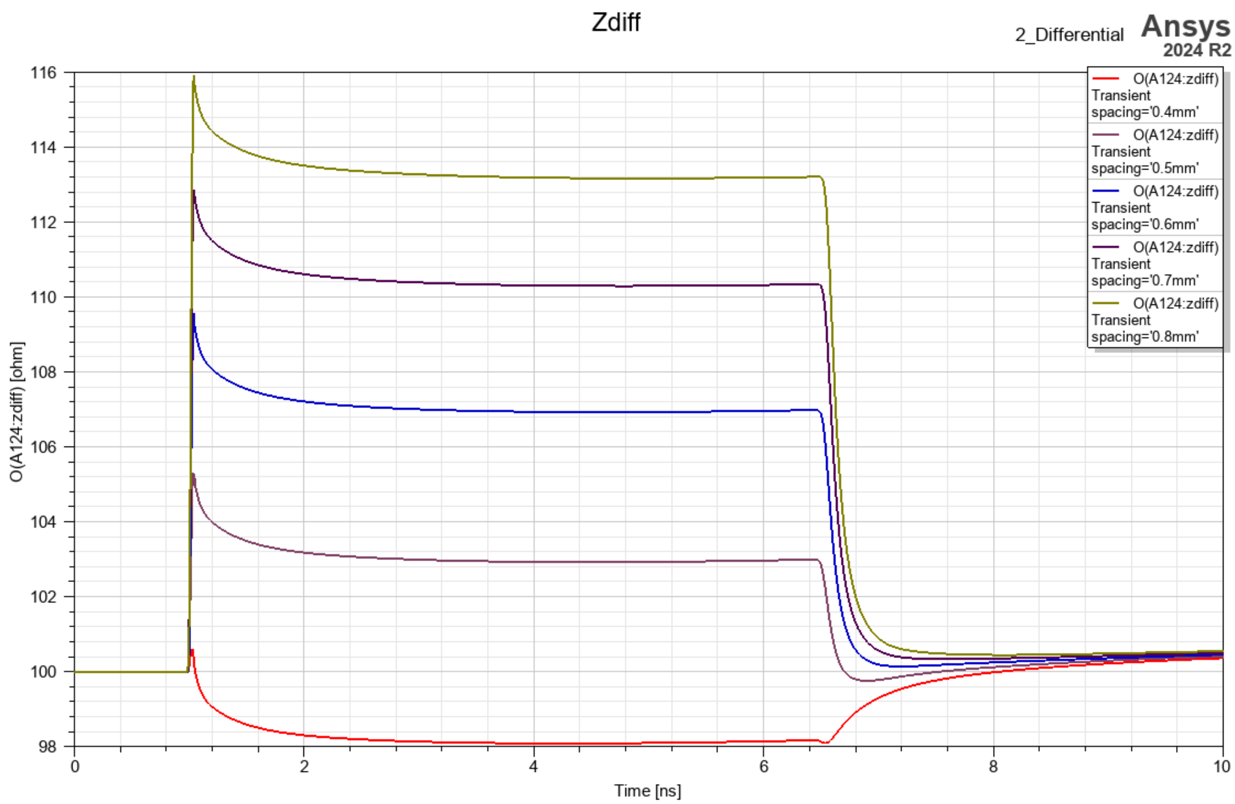
2. Expand the Analysis icon, double-click on the Transient analysis setup, and verify the sweep of the spacing variable.



3. In the Project Tree, right-click Transient and select Analyze. The analysis runs to completion.
4. Expand the Reports icon and right-click on the **Zdiff** report setup. Select **Modify Report** to open the Report Setup.

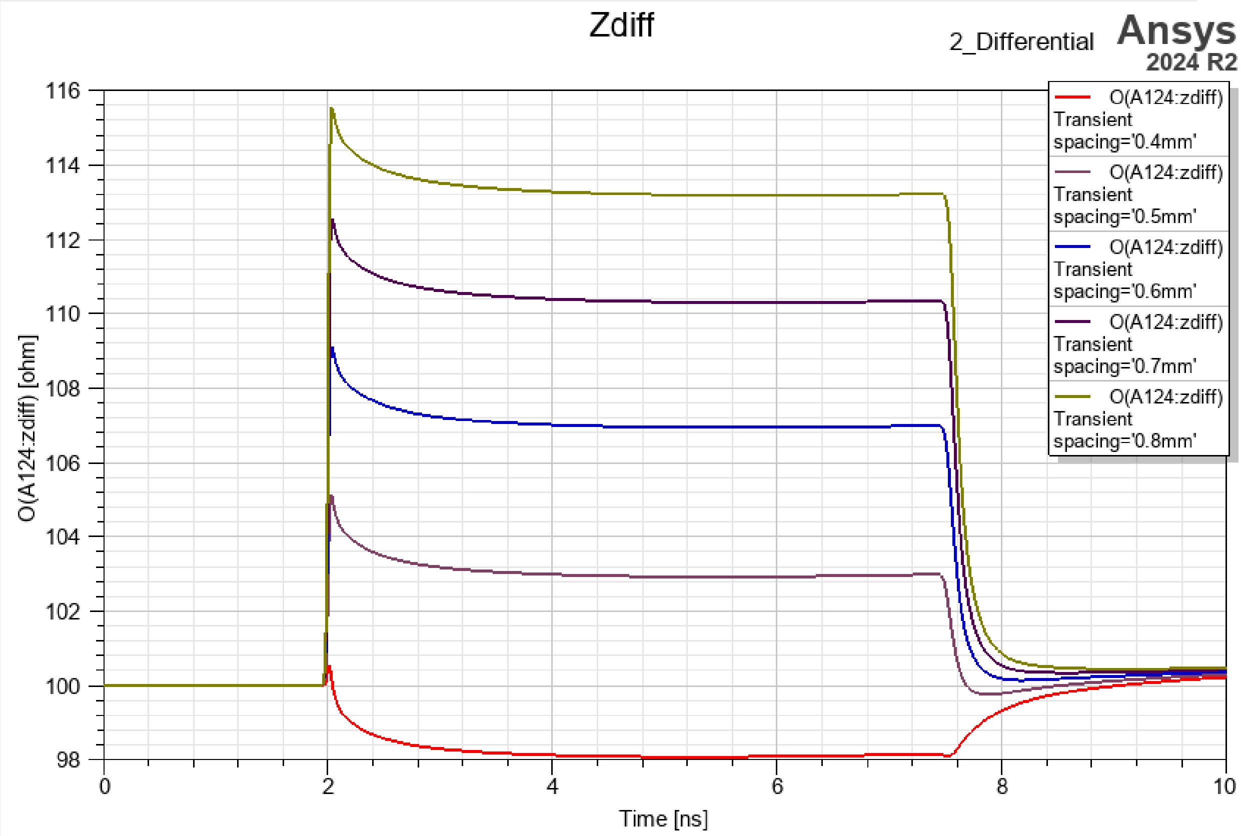

Report: TDR_Example - 2_Differential - Zdiff - O(A124:zdiff)
Context
Solution: Transient
Domain: Time
Plotting Range
Windowing
Thinning
Time 0 ps
Time 10 ns
Update Report
Real time
Update
Trace
Families
Families Display
Primary Sweep: Time
All
X: Default Time
Y: O(A124:zdiff)
Range Function...
Category:
Variables
Output Variables
Voltage
Current
Power
Device Properties
All
Quantity:
O(A124:zdiff)
O(A124:zneg)
O(A124:zpos)
Function:
<none>
abs
acos
acosh
ang_deg
ang_deg_val
ang_rad
arg
asin
asinh
atan
atanh
cos
cosh
Output Variables...
Options...
New Report
Apply Trace
Add Trace
Close

5. Click New Report to generate the report:



The TDR analysis shows how the spacing affects the differential impedance. Using the Trapezoidal waveform, the reflection timings are expected at 2*(Total Delay).

Note: If SmoothStep pulse type is selected, the reflection timings are expected at T50 + 2*(Total Delay). See **Circuit > Nexxim Components > Probes > Time Domain Reflectometer** for more information about TDR properties, including the T50 SmoothStep parameter.

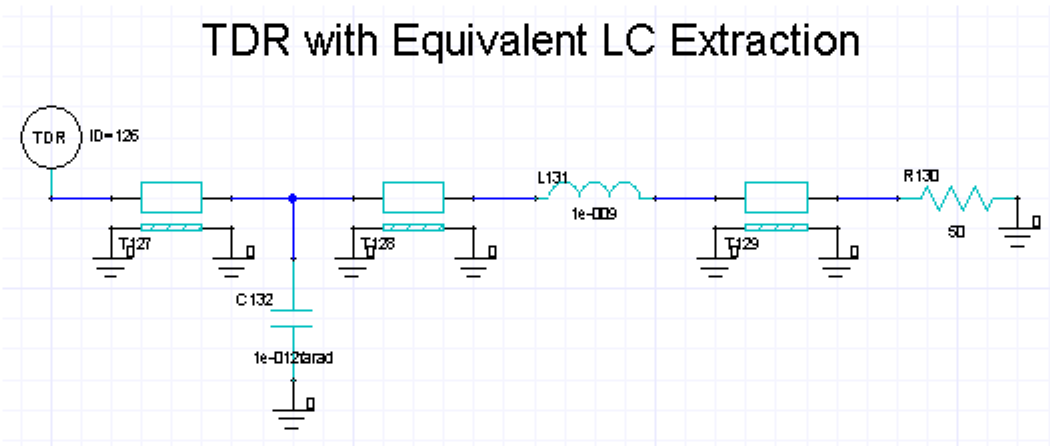


LC Extraction

Open the TDR Schematic project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open **Circuit**, then **Signal Integrity**, and select the file **TDR_Example.aedt**. Click **Open**.

Expand the TDR_Example icon and the **3_LC_Extraction** icon. The LC_Extraction schematic appears in the design window:



This circuit consists of:

- A TDR component with pulse_width = 1sec and pulse_repetition = 1sec, and Time_delay = 0.
- Three time delay elements (TRL_T), each one set for a delay of 1ns (1e-9 sec).
- A shunt capacitance of 1pF (1e-12 farad)
- a series inductance of 1nH (1e-9 henry).
- A terminating impedance. The impedance of all lines and the termination is 50 ohm.

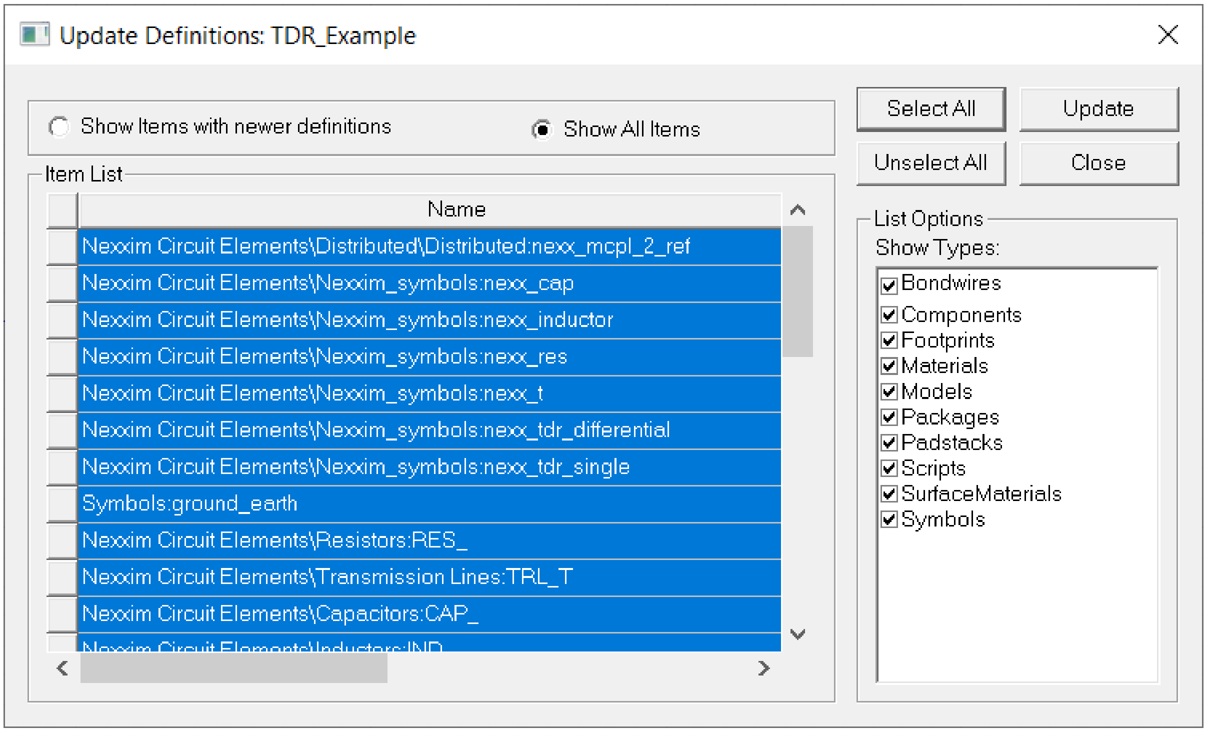
The TDR element can analyze the changes in impedance and calculate the equivalent capacitance for a drop in impedance or inductance for a spike in impedance. For more information see *Time Domain Reflectometer* in the Circuit Components help.

The TDR component provides two options for the pulse type: Trapezoidal and SmoothStep. SmoothStep provides a more realistic step input and allows control of the rise interval.

(Optional) Choose between Trapezoidal and SmoothStep.

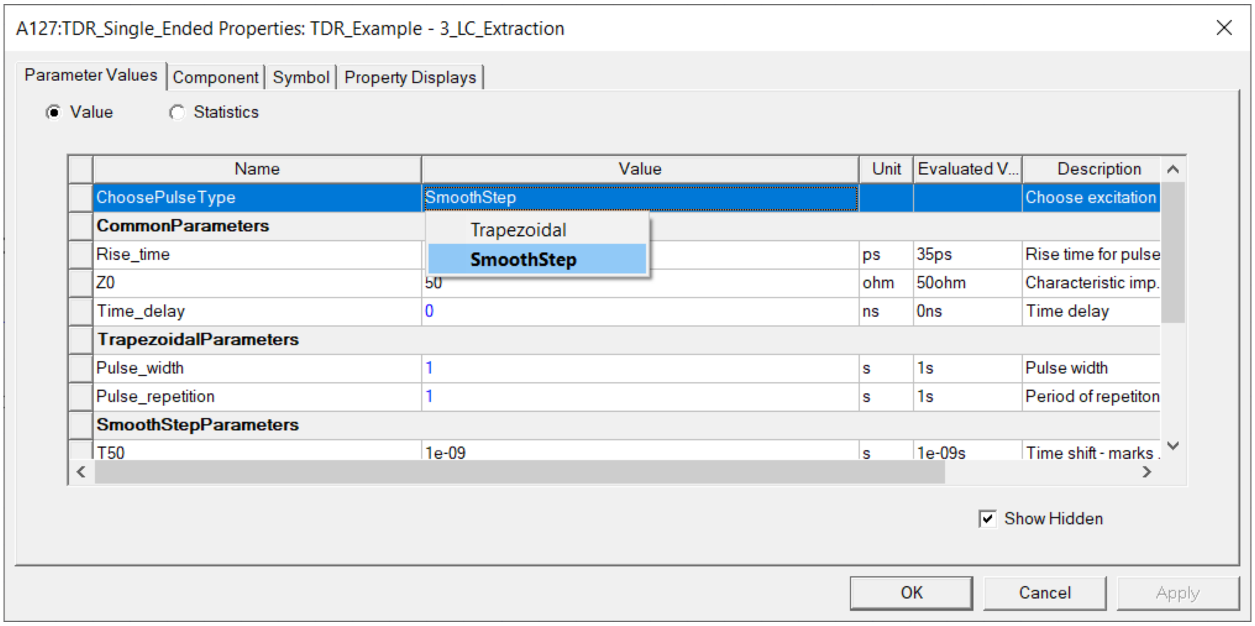
i. Click **Tools** > **Project Tools** > **Update Definitions**.

ii. In the **Update Definitions** dialog, select **Show All Items**, **Select All**, then click **Update** . Close the dialog once updates are successful.



iii. Right-click the TDR model symbol and click **Properties** to open the properties dialog.

iv. Click the **Value** field for **ChoosePulseType** to view the two options. Select **SmoothStep**, click **Apply**, then **OK** to close the dialog.



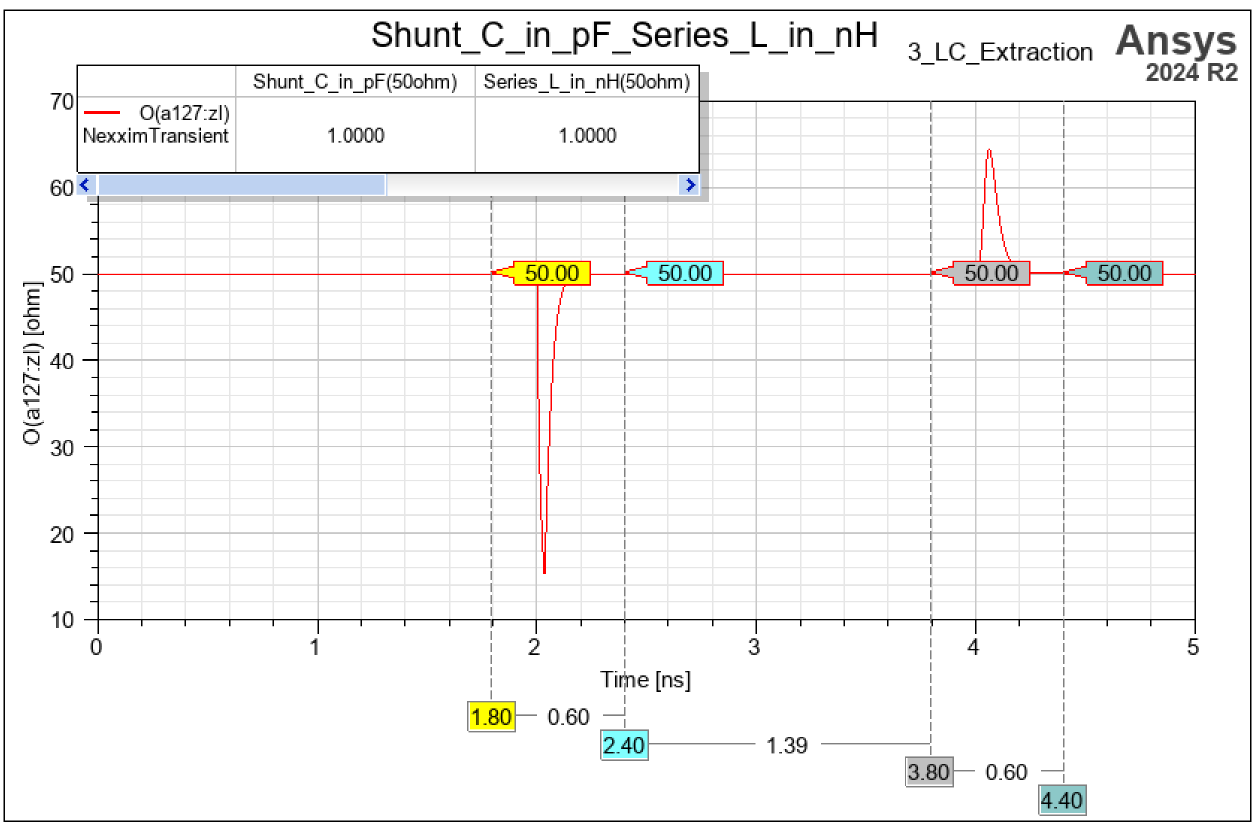
This demonstration includes the Transient setup and an example of a Report with the equivalent reactance values. A routine report setup follows.

1. Click the three transmission line components and verify the delay (**D** property) is 1e-9 sec.
2. Expand the Analysis folder and double-click on the **NexximTransient** setup. Verify that the Time is set to run from 0 to 5ns, with a Step of 0.1ps. This time period allows us to see the reflected impedance on the capacitor (at 2ns) and the inductor (at 4ns) and omit secondary reflections.
3. Click the **NexximTransient** setup and select **Analyze** (or **Force Analysis** if necessary). The analysis runs to completion.
4. Expand the Results folder, right-click on the **Shunt_C_in_pF_Series_L_in_nH** report, and select **Open Report**.

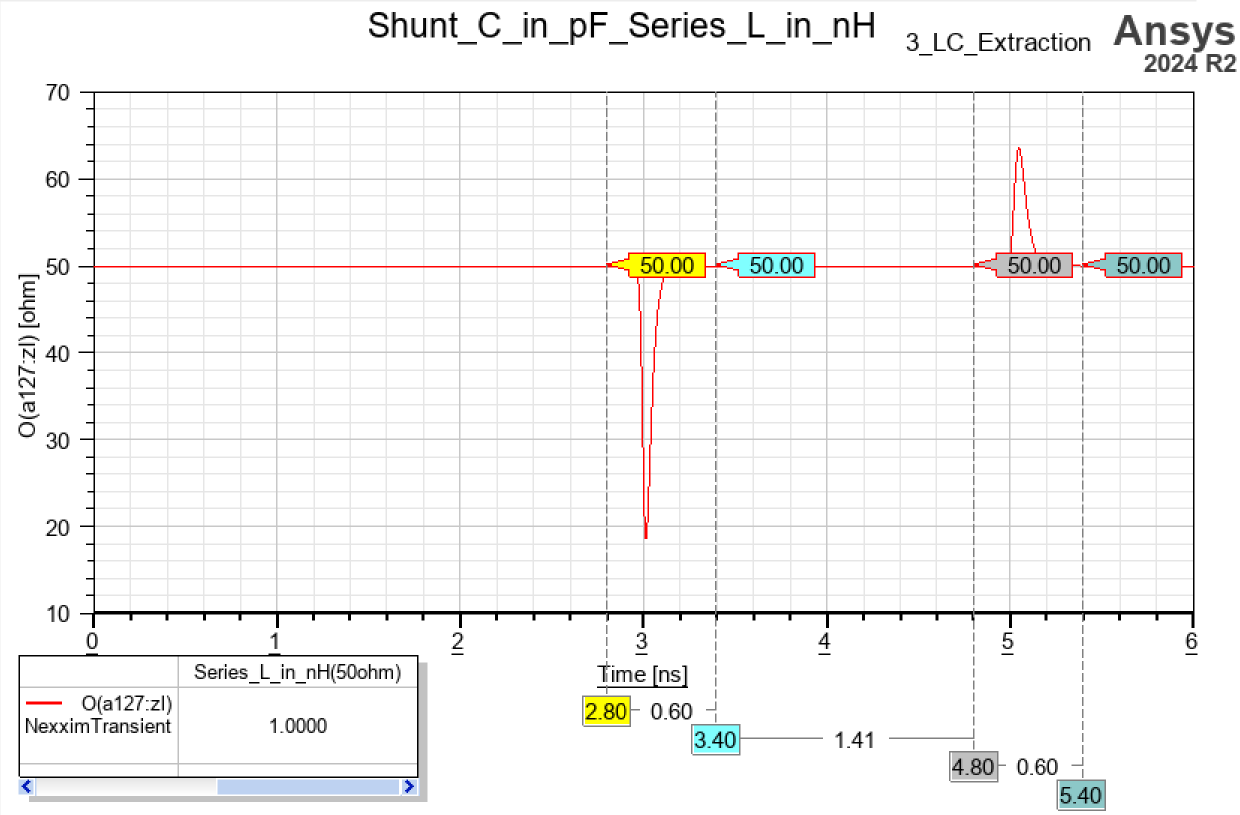
A display of the report is shown later. Here are the steps used to create it after the analysis runs to completion.

1. Right-click the **Results** folder, then select **Create Standard Report>Create Rectangular Report**.
2. From the **Report** window, set the **Category** to **Device Properties**.
3. Verify that the (only) **Quantity** is **O(a126:z)**.
4. Click **New Report**, then **Close**. The **XY Plot1** report opens with the trace of impedance versus time. The graph has a dip at 2ns and a peak at 4ns.
5. Under **Report 2D** at the top, select **Add X Marker**. The X marker appears at time 0 on the plot.

6. Click the tag at the bottom of the X marker and position it just to the left of the dip on the graph, around 1.8ns. The location need not be exact.
7. Repeat the previous step three times more, to locate X markers on both sides of the dip and on both sides of the peak, at approximately 2.4ns, 3.8ns, and 4.4ns, respectively.
8. Under **Report 2D** at the top, select **Trace Characteristics >All** to open the **AddTrace Characteristics** window.
9. For the **Category**, select **TDR** at the very end of the menu.
10. For the **Function**, click the check box to select **Shunt_C_in_pF**.
11. In the table, Row 2 is the **Range**. Click the button in the **Range Value** field, and select **X Markers** on the menu. In the **AddTrace Characteristics** window, table rows 3 and 4 are populated with the first two X-Markers, designated **MX1** and **MX2**.
12. Click **Add**.
13. Uncheck **Shunt_C_in_pF** in the **Function** group box, then check **Series_L_in_nH**.
14. In the **Trace Characteristics** window, table rows 3 and 4 are still populated with the first two X-Markers, designated **MX1** and **MX2**.
15. Click the XMarker1 **Value** field and select **MX3** on the menu.
16. Click the XMarker2 **Value** field and select **MX4** on the menu.
17. Click **Add**, then **Done**.
18. The **Curve Info** group box displays the calculated values, 1.0pF for **Shunt_C_in_pF** and 1.0nh henry for **Series_L_in_nH**.



Note: If SmoothStep pulse type is selected, the reflection timings are expected at T50 + 2*(Total Delay). See **Circuit > Nexxim Components > Probes > Time Domain Reflectometer** for more information about TDR properties, including the T50 SmoothStep parameter.



Eye Analysis Example

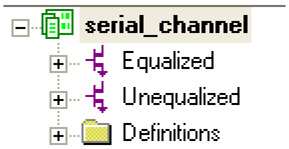
This example of a high-speed serial communications channel allows you to experiment with Nexxim Transient, QuickEye, and VerifEye analyses. It focuses on modeling the effect of equalization on the performance of the channel. The schematic for the channel with complete analysis and report setups is available in the Examples directory.

Open and examine a High-Speed Channel Schematic project.

1. From the **File** menu, select **Open Examples**, or click **Open Examples** on the **Desktop** ribbon, to open an explorer window to the **Examples** folder.
2. Select the file **Electronics Desktop** / **Circuit** / **Signal Integrity** / **Serial_Channel.aedt** and click **Open**.

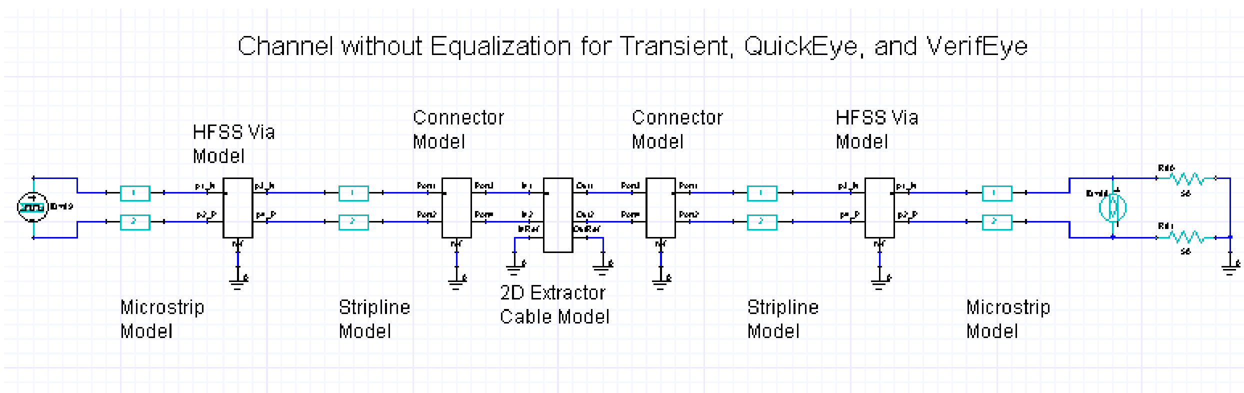
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

   There are two circuits in the project:



4. Examine the **Unequalized** schematic.

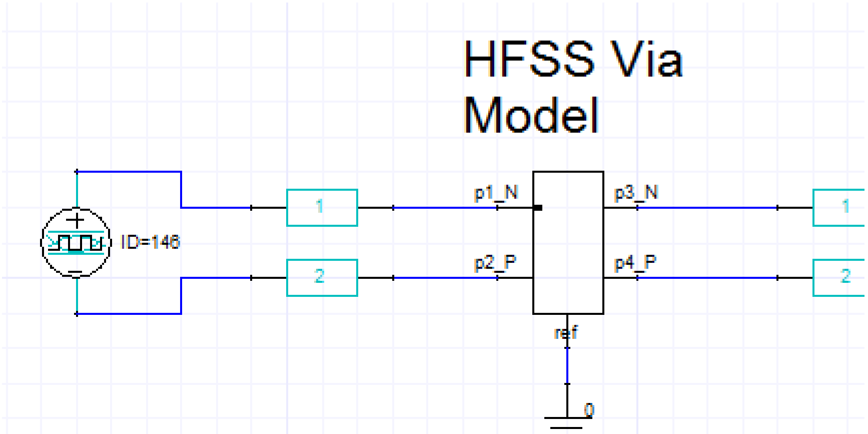
   a. From the **Project Manager** window, double-click **Unequalized**. The **Schematic Editor** displays the schematic:



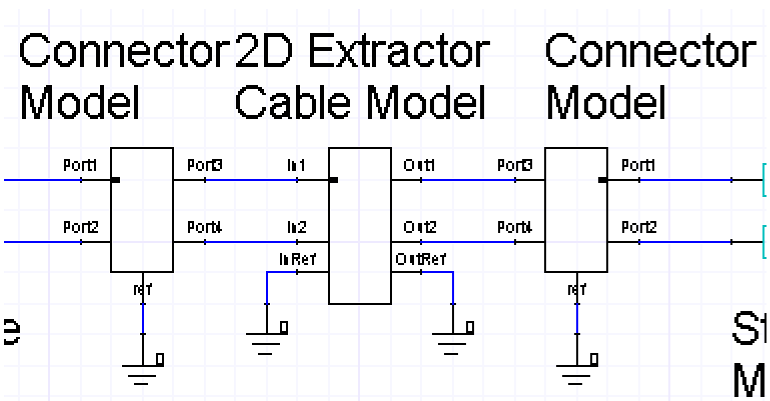
   This differential communication channel transmits first over surface-level traces, then through via connectors to buried traces, then through a cable connector to a cable, then back through cable connector, buried traces, via connectors, surface traces, and termination on the receiver end.

   The Eye Source at the transmitter end provides a controllable bit stream for both Transient and Quick Eye analyses.

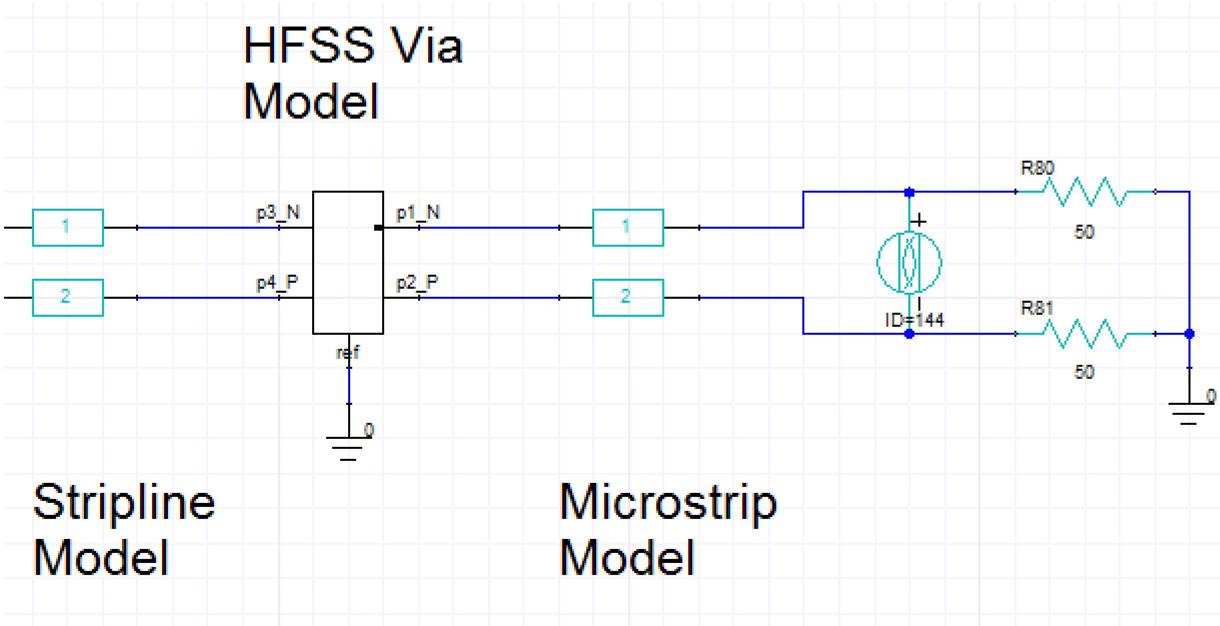
   The left side of the circuit consists of the Eye Source, microstrip differential line pair, via connector, and stripline differential line pair.



The center components of the circuit includes the left cable connector, the cable modeled as a W-element, and the right cable connector.

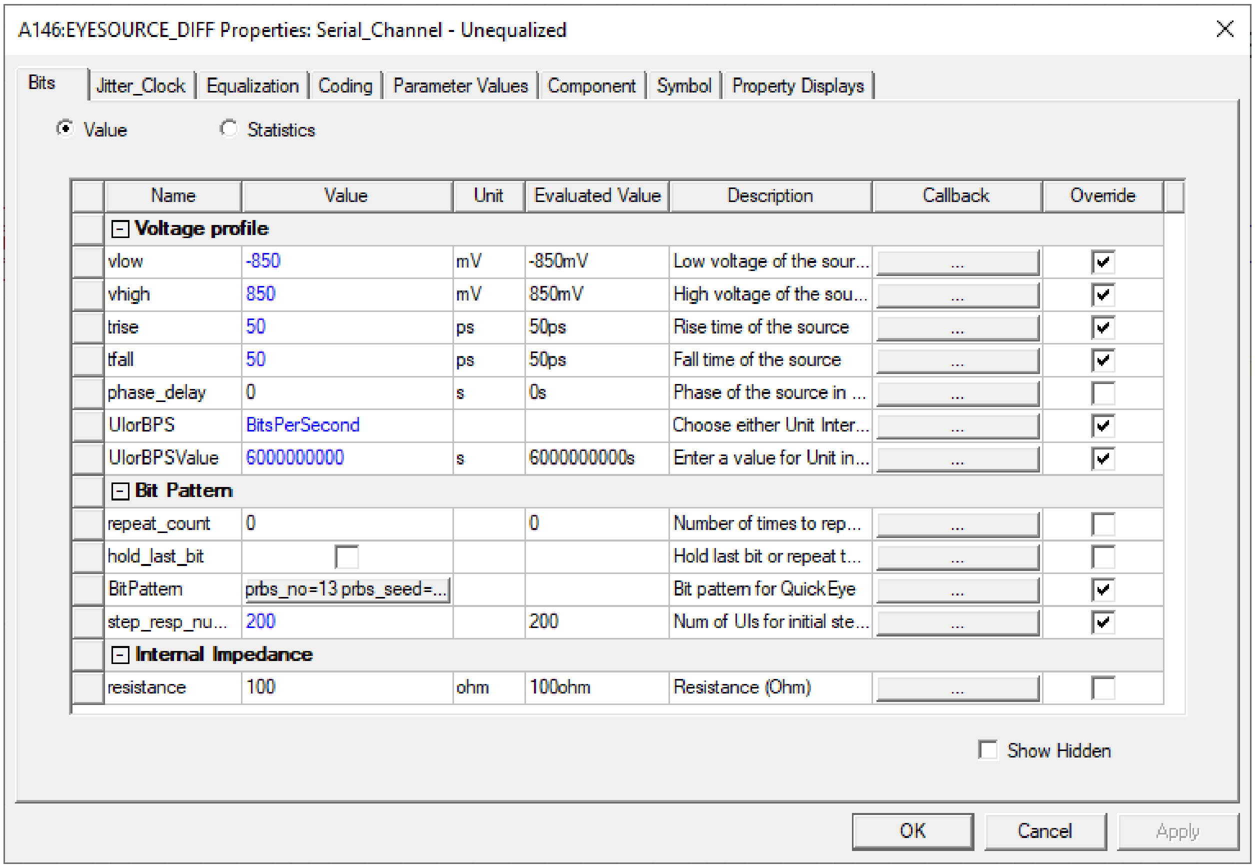


The right side of the circuit consists of the stripline differential line pair, via connector, microstrip differential line pair, Eye Probe, and line termination.



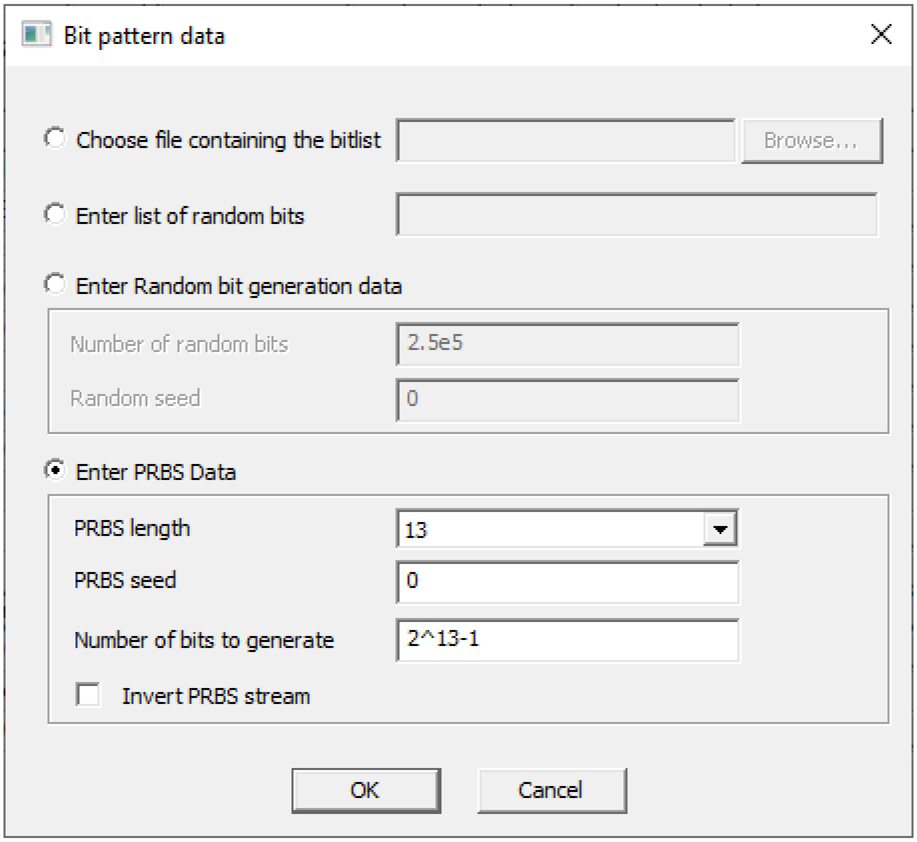
b. Double-click the Eye Source at the front (left) end of the channel to open its **Properties** window.

1. Open the **Bits** tab:



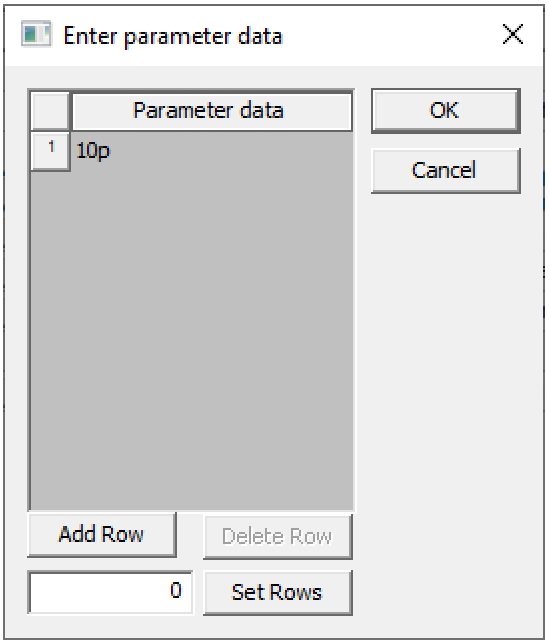
The source has a data rate of 6 GBits per second. Rise and fall times are 50 ps.

2. Click the button in the **BitPattern Value** field to open the **Bit Pattern Data** window:



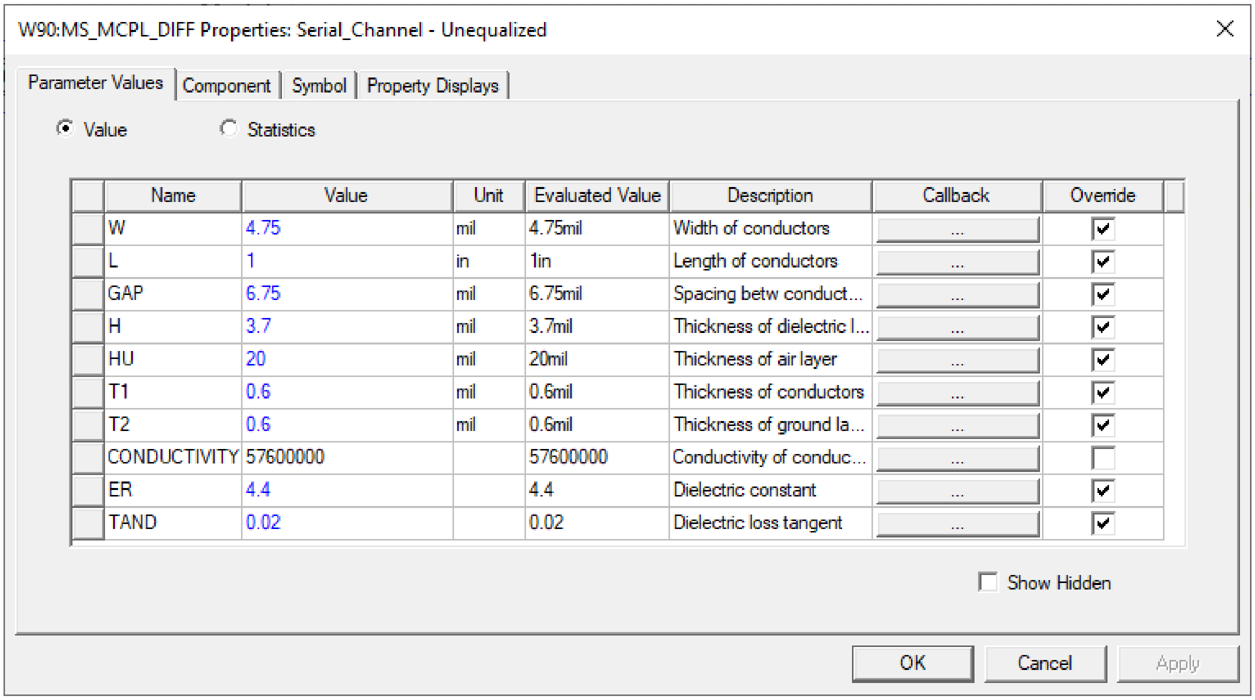
The Random bit generation data or pseudo random bit sequence (PRBS) data provides the bit stream for both transient and Quick Eye analyses. Click **OK** to close the **Bit Pattern Data** window.

3. Click the **Jitter_Clock** tab. Click **txrj** to open the **ParameterData** window.



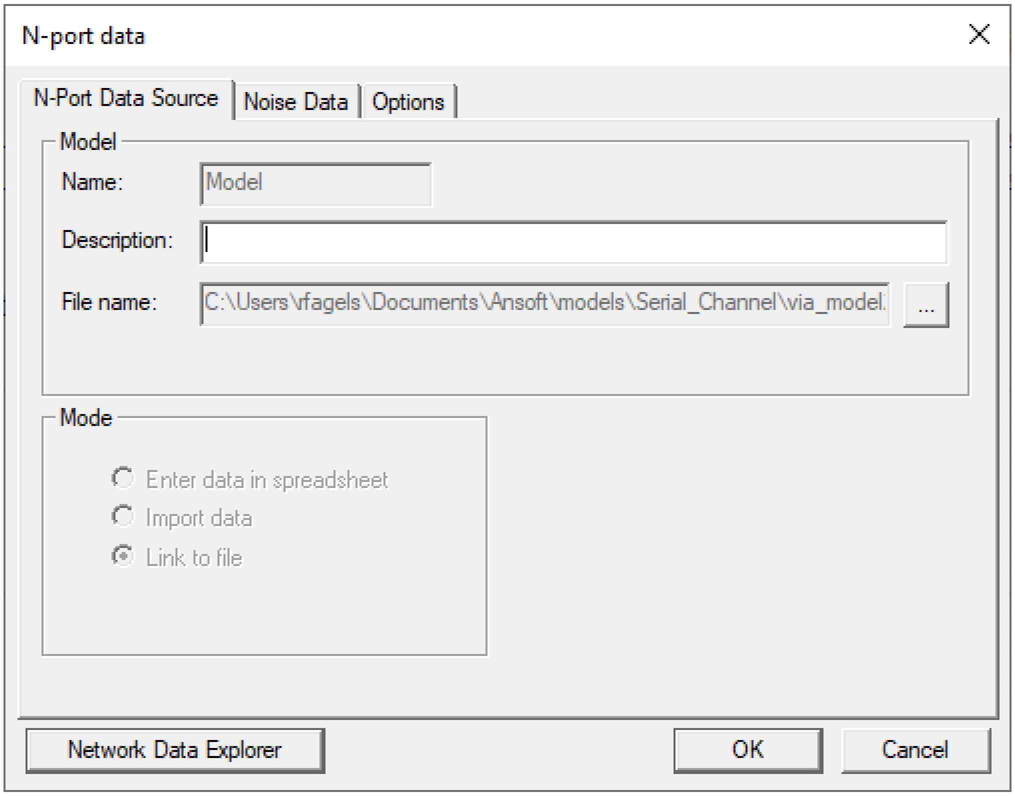
A random (Gaussian) transmit jitter distribution with a standard deviation of 10ps is applied to the transmitted signal. The source is set to transmit with a Gaussian jitter source using 10ps as the standard deviation. The jitter appears in all analyses. Click **OK** to close the **ParameterData** window, and **OK** again to close the **Properties** window.

c. Double-click the left Microstrip differential line pair to open its **Properties** window:



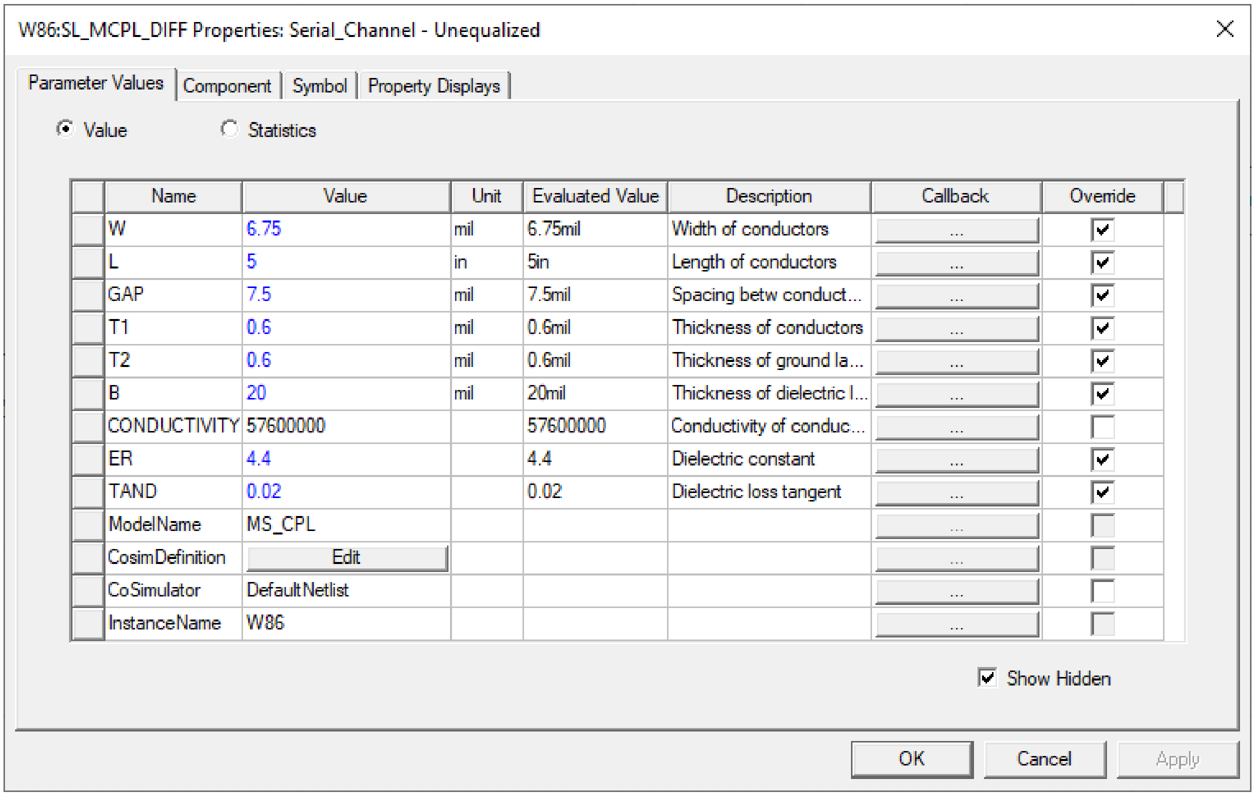
The two microstrip portions of the channel have the same parameters. The length of both segments is one inch. Click **Cancel** to close this window.

d. The via connector is modeled with S-parameters. Right-click the **HFSS Via Model** symbol and click **Edit Model** to open the N-port data window:



File **via_model2.s4p** contains the Touchstone data. Both via connectors use this model. Click **Cancel** to close the window.

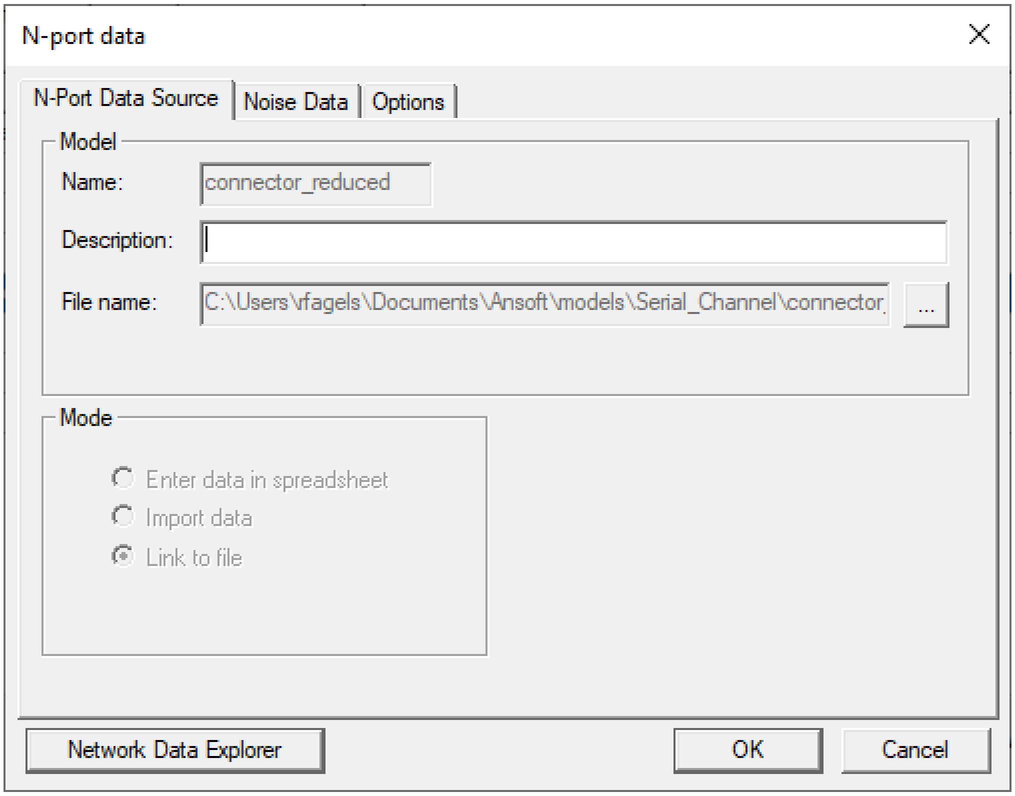
e. Double-click the left Stripline differential pair to open its **Properties** window:



Name	Value	Unit	Evaluated Value	Description	Callback	Override
W	6.75	mil	6.75mil	Width of conductors	...	✓
L	5	in	5in	Length of conductors	...	✓
GAP	7.5	mil	7.5mil	Spacing betw conduct...	...	✓
T1	0.6	mil	0.6mil	Thickness of conductors	...	✓
T2	0.6	mil	0.6mil	Thickness of ground la...	...	✓
B	20	mil	20mil	Thickness of dielectric l...	...	✓
CONDUCTIVITY	57600000		57600000	Conductivity of conduc...	...	
ER	4.4		4.4	Dielectric constant	...	✓
TAND	0.02		0.02	Dielectric loss tangent	...	✓
ModelName	MS_CPL				...	
CosimDefinition	Edit				...	
CoSimulator	DefaultNetlist				...	
InstanceName	W86				...	

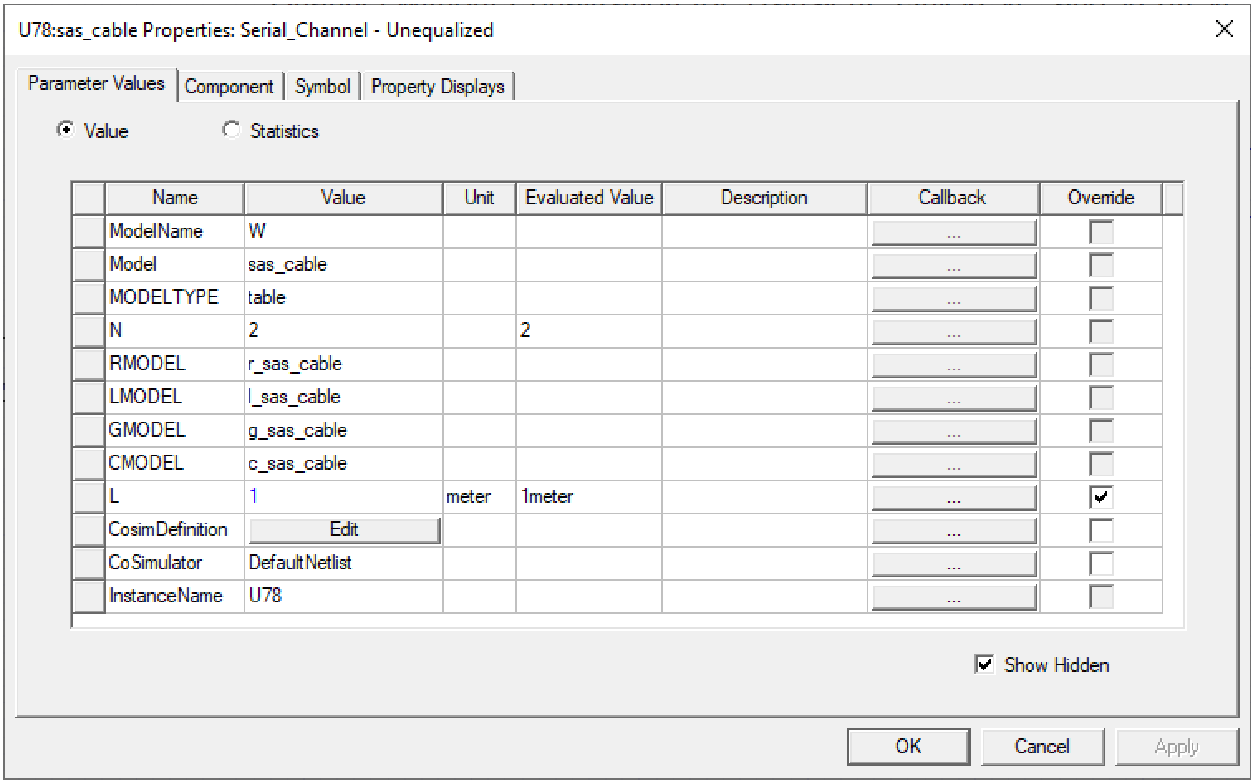
Both stripline segments use the same Parameter values. The length of both segments is five inches. Close this window.

f. The cable connector is modeled with S-parameters. Double-click the **Connector Model** symbol, select the **Components** tab, and click **Edit** for the **Cosim Definition**.



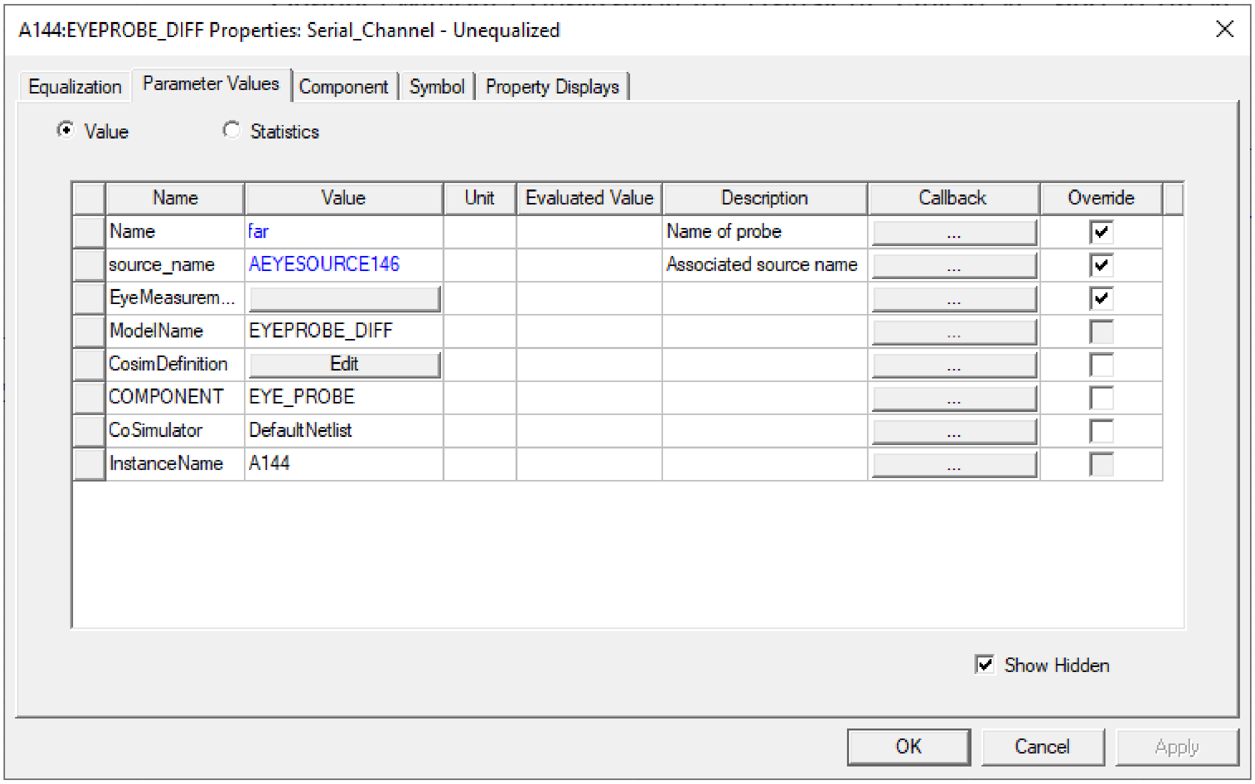
File **connector_mode2.s4p** contains the Touchstone data. Both cable connectors use this model. Click **OK** to close both windows.

g. Double-click the **2D Extractor Cable Model** symbol to open its **Properties** window. Open the **Parameter Values** tab.



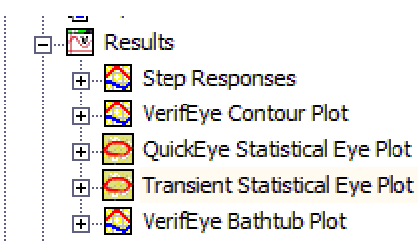
The cable is modeled with a W-element RLGC table model. Its length is one meter. Both ends of the channel uses the same elements. Click **Cancel** to close the window.

h. Double-click the Eye Probe on the right side to open its **Properties** window. Open the **Parameter Values** tab:

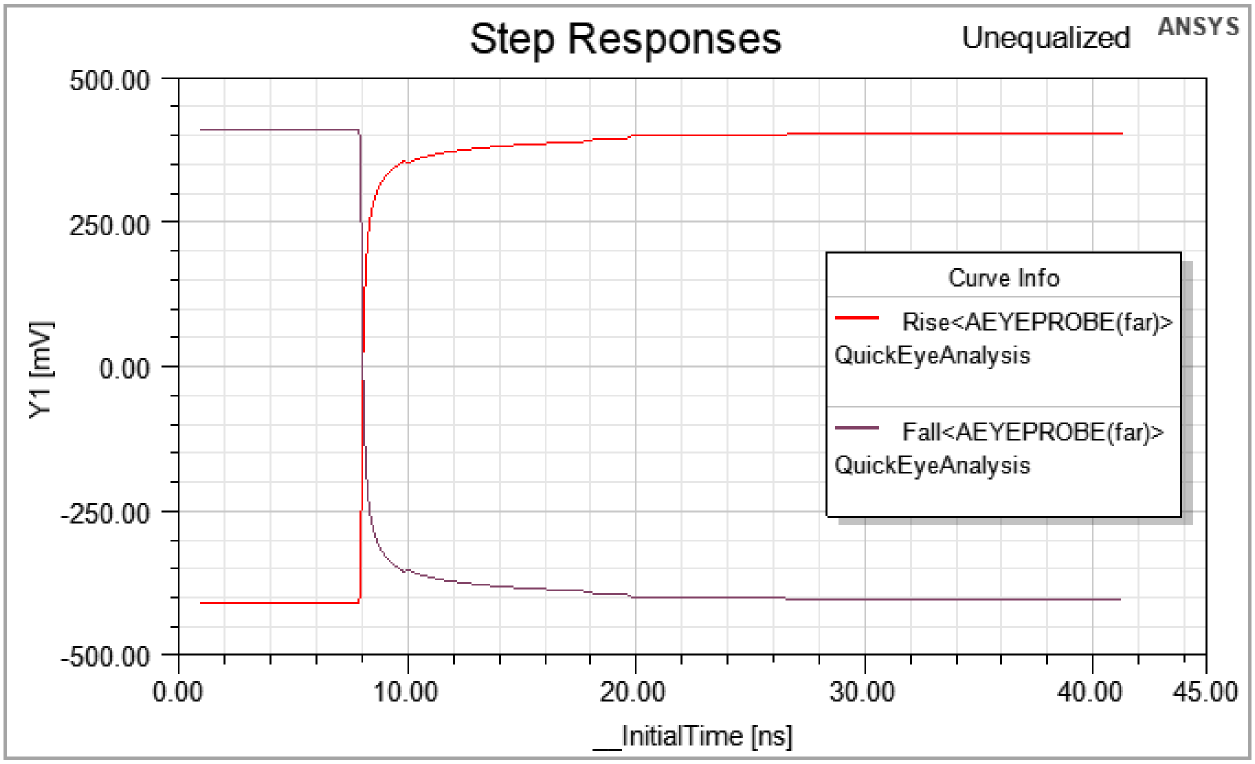


No receive equalization is set up in the probe. The next step is to simulate the circuit without equalization. Click **Cancel** to close the window.

5. Generate and review the project's unequalized Transient, QuickEye, and VerifEye analyses.
   a. From the **Project Manager** window under **Unequalized**, right-click **Analysis** and select **Analyze**. This starts all three analyses, which takes a few minutes to complete.
   b. Expand the **Results** folder. Several reports are set up.

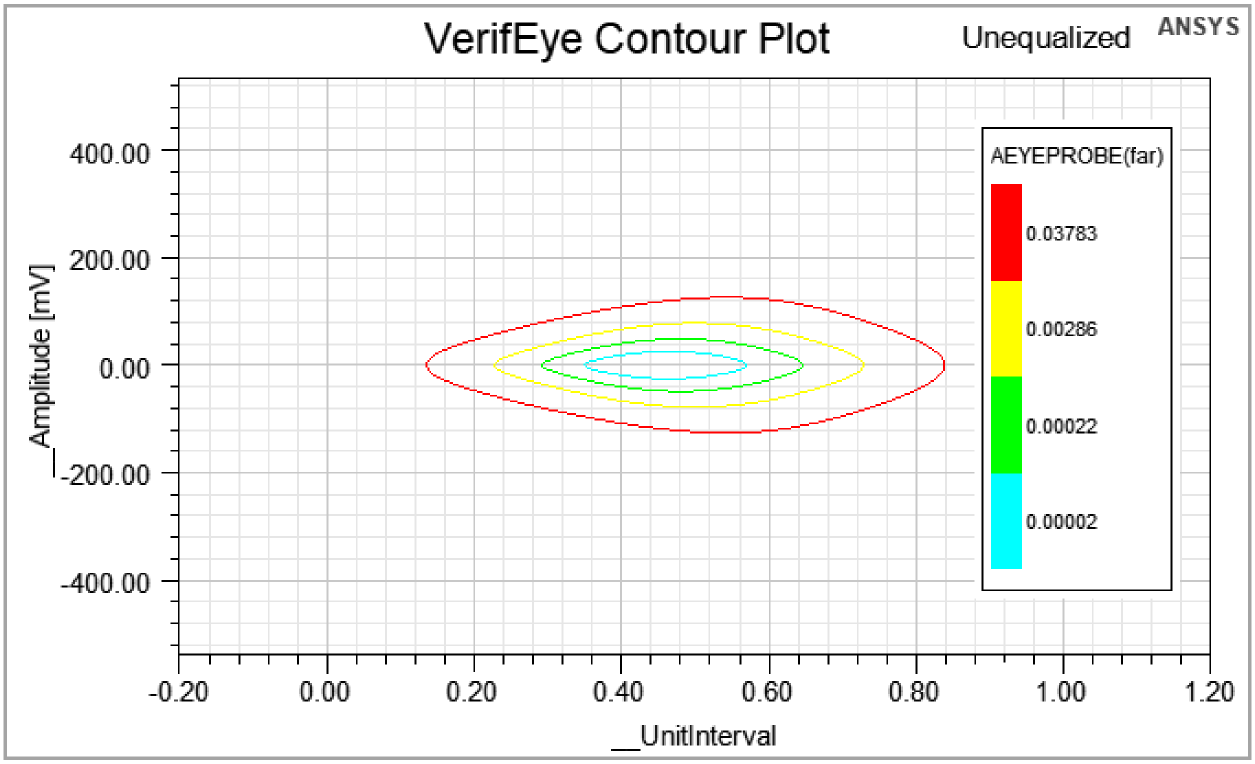


c. Double-click the **Step Responses** report to open it:

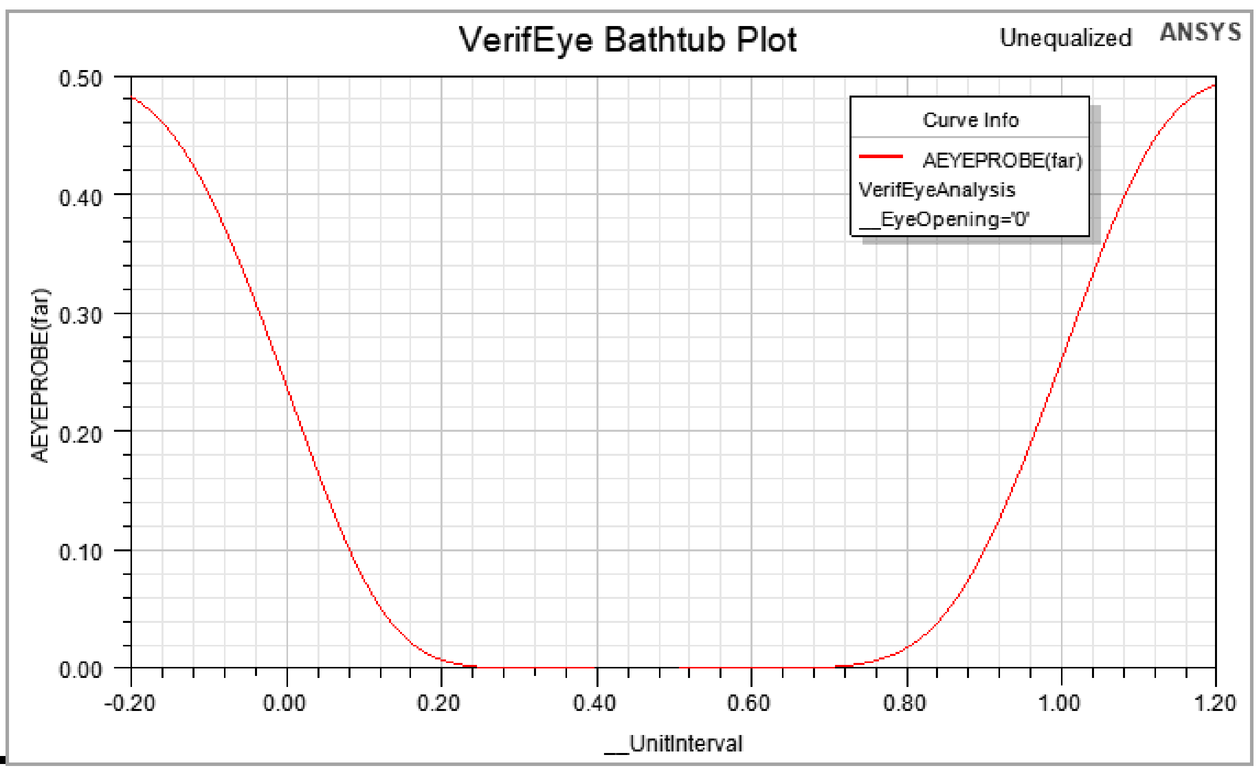


The step responses are symmetrical.

d. VerifEye calculates the channel bit error rate using the channel parameters. Double-click the **VerifEye Contour Plot** report:

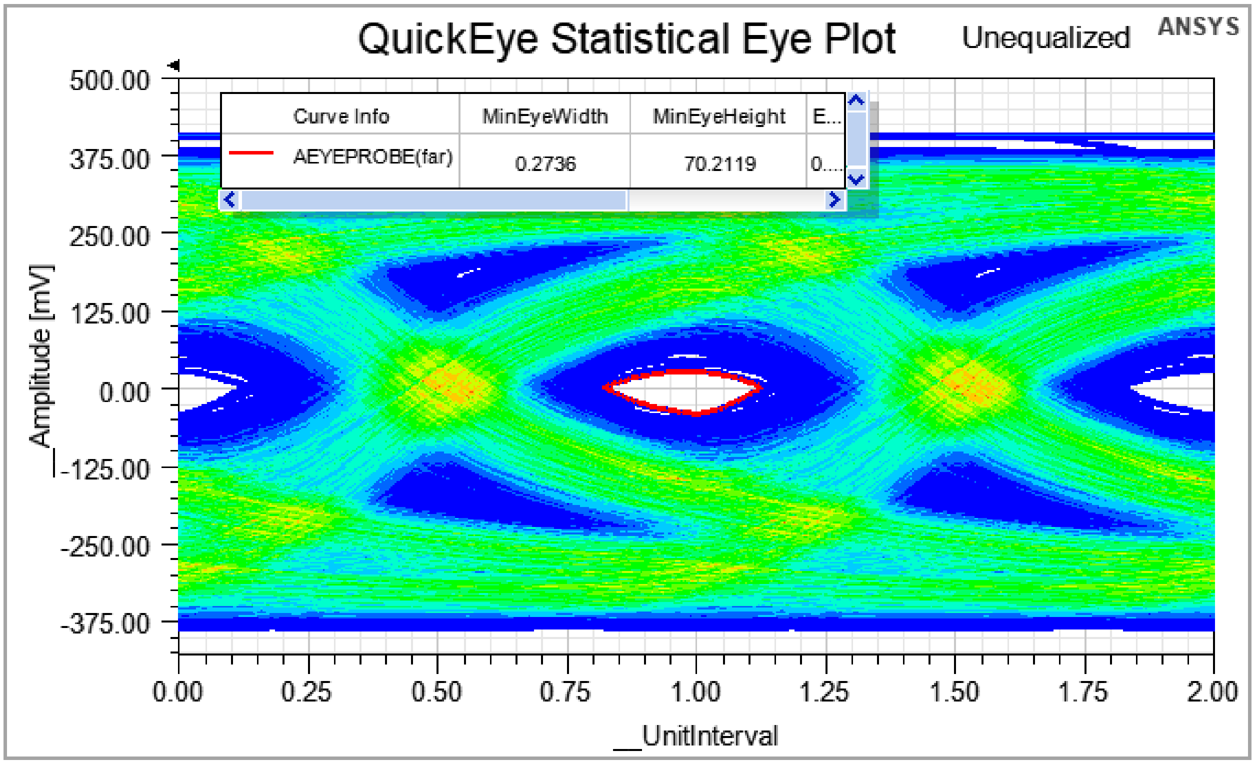


The **VE Bathtub** plot shows the BER curve.



All the Unequalized plots show the negative effect of jitter on the eye opening.

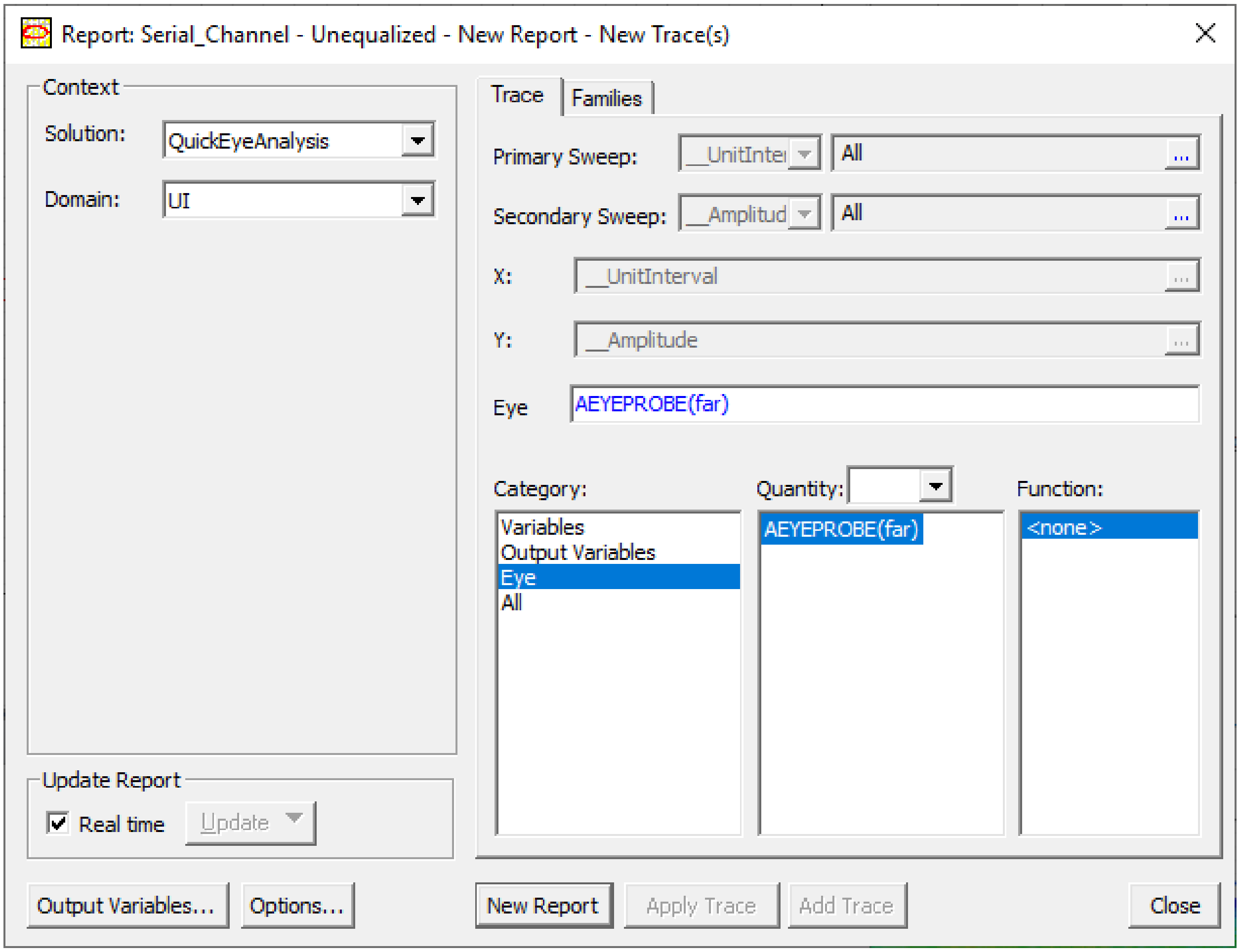
e. Quick Eye analysis involves the impulse response of the channel with the bit pattern generated by the Eye source. Double-click the **QuickEye Statistical Eye Plot** report:



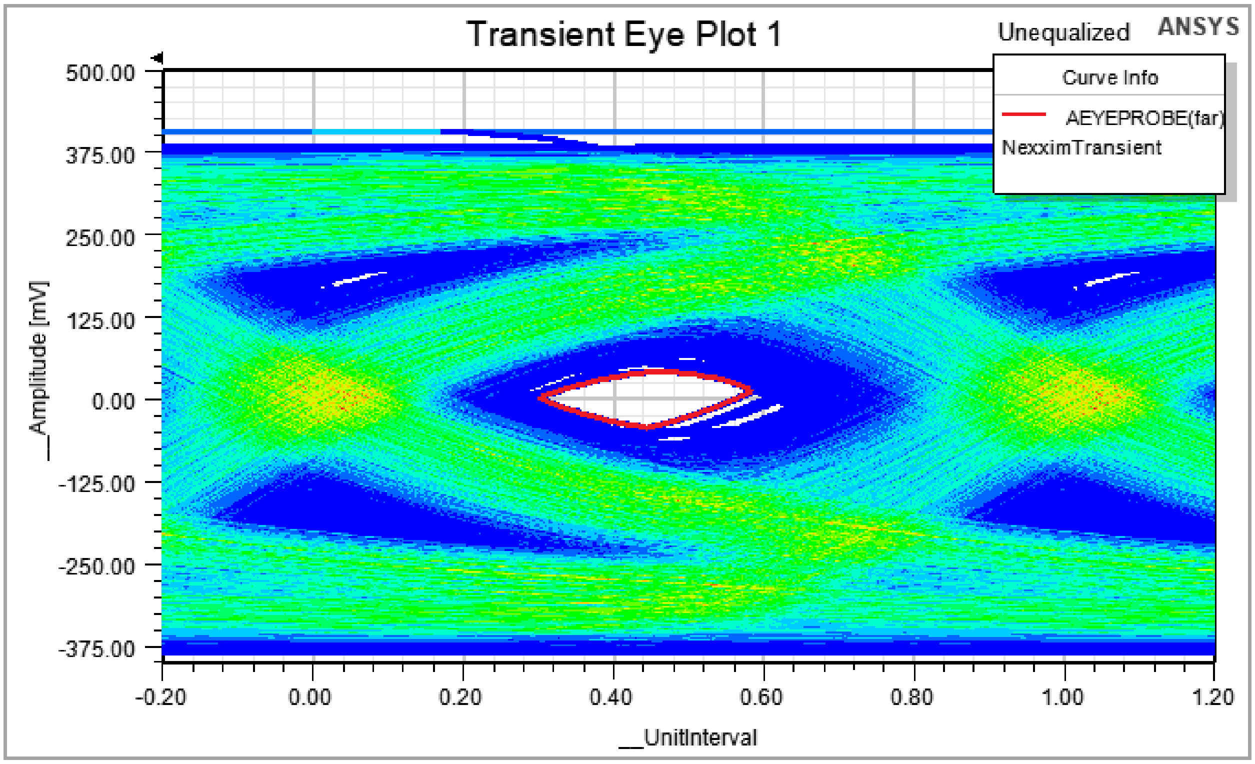
f. A Transient analysis runs the entire bit sequence:

1. From the **Project Manager** window, right-click **Results** and click **Create Statistical Eye Report** > **Statistical Eye Plot**.

2. In the **Report** window, set **Solution** to **NexximTransient**.



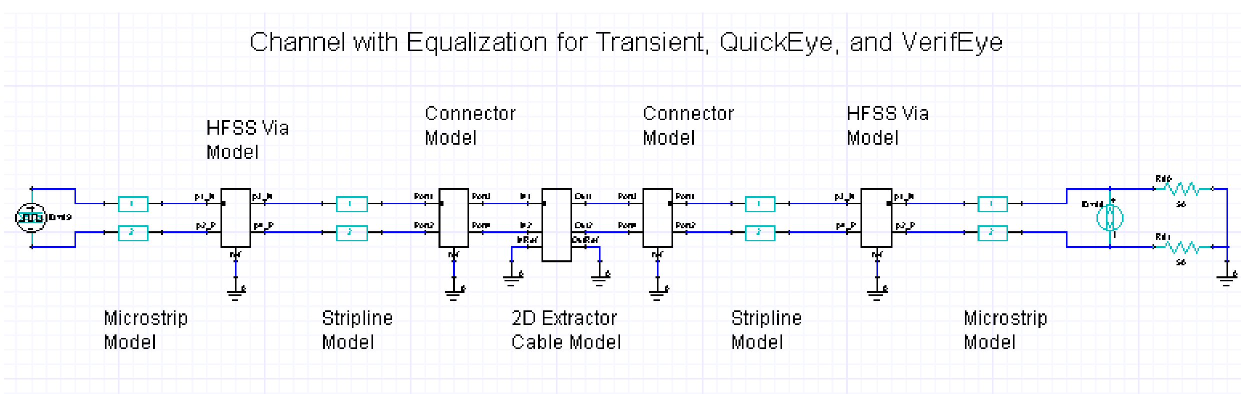
3. Click **New report** to generate the **Transient Statistical Eye Plot** report and **Close** to close the **Report** window.



g. From the menu bar, click **Window** > **Close All** to close all the schematic and report windows.

6. Examine the **Equalized** schematic.

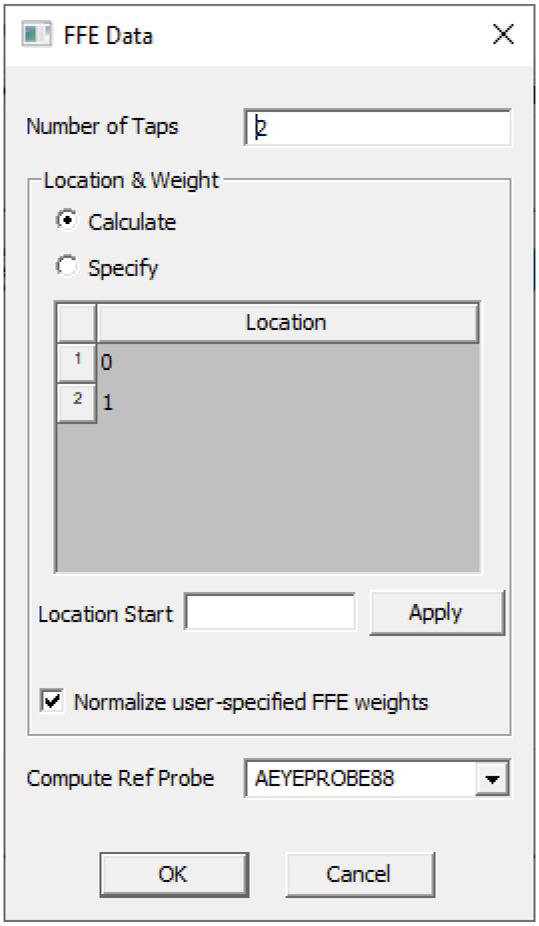
   a. From the **Project Manager** window, double-click **Equalized** to open its schematic.



   The circuit elements are the same, but the properties of the Eye Source and Eye Probe have been changed to add transmit and receive equalization.

   b. Double-click the left Eye Source to open the **Properties** window.

c. From the **Equalization** tab, click in the **FFE_data Value** field to open the **FFE Data** window. Two taps of Feed-Forward Equalization are set up for the source:

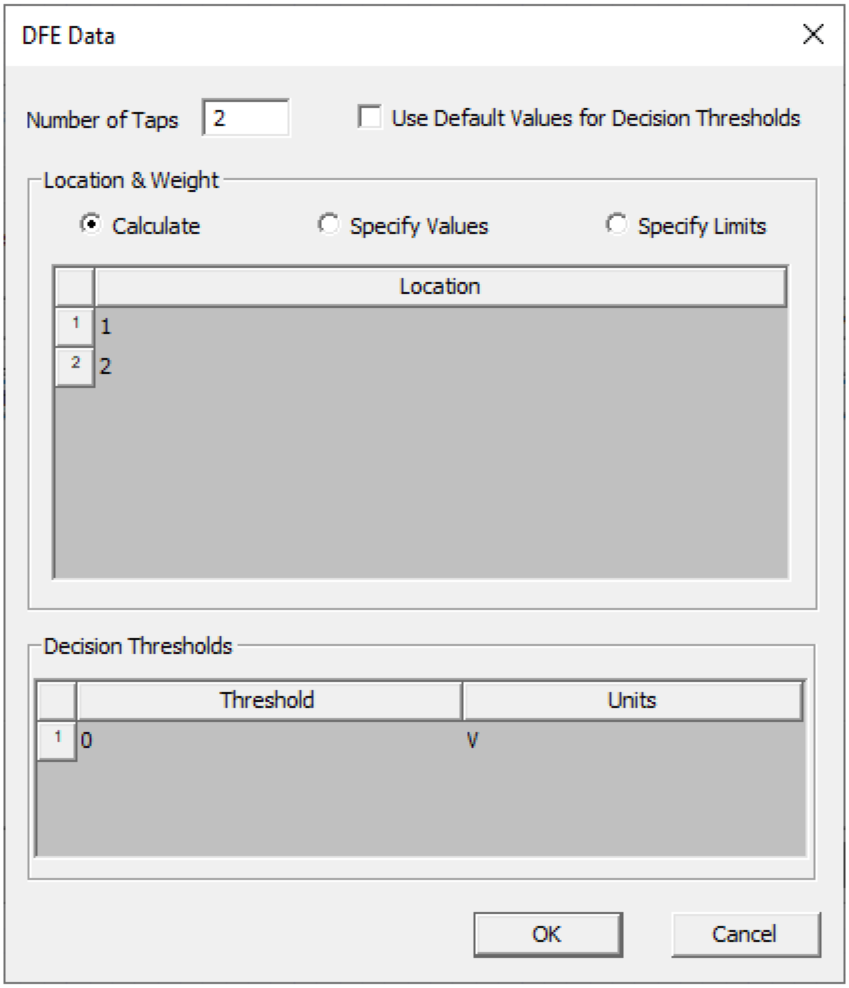


To disable Feed-Forward Equalization, set the number of taps to zero. To enable FFE, set the number of taps to an integer greater than 0.

d. Click **OK** to close the **FEE Data** window and **OK** again to close the **Properties** window.

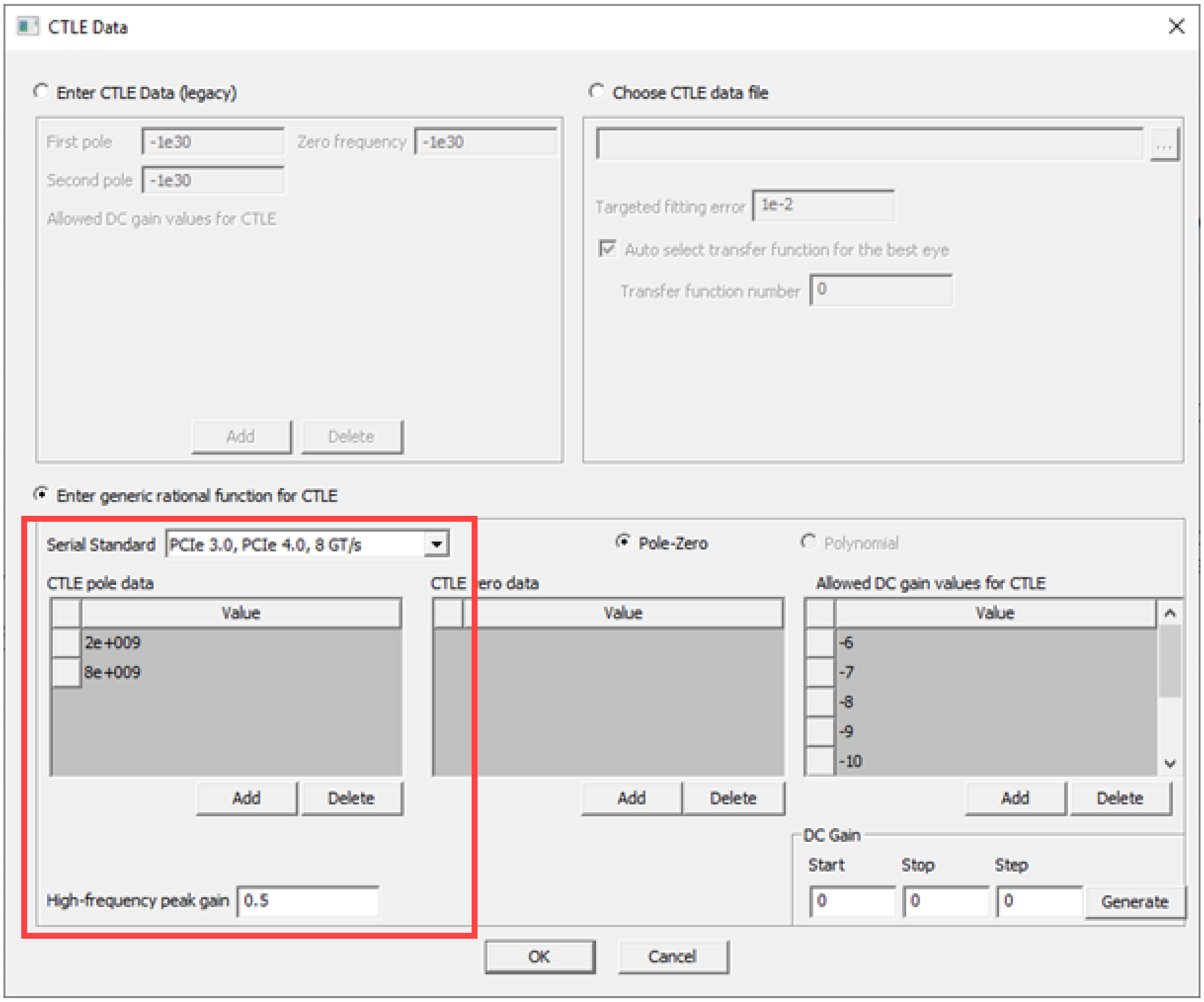
e. Double-click the right Eye Source to open its **Properties** window.

f. From the **Equalization** tab, click **DFE_Data** to open the **DFE Data** window. Two taps of Decision-Feedback Equalization are set up for the probe:



To disable Decision-Feedback Equalization, set the number of taps to zero. To enable DFE, set the number of taps to an integer greater than 0. Click **OK** to close the **DFE Data** window.

g. From the **Equalization** tab of the same **Properties** window, click **CTLE_Data** to open the **CTLE Data** window. Two pole frequencies for Continuous-Time Linear Equalization are set up for the Eye Probe, using the Serial Standard menu to select the PCIe3.0 pole settings, with the added specification of 0.5 for the High-frequency peak gain (maximum CTLE transfer function value.)

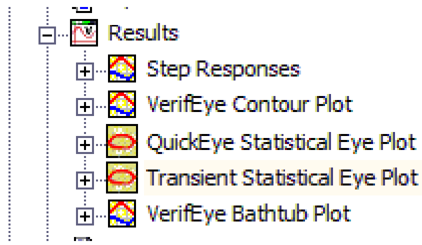


A range of Allowed DC gain values from -6 dB to -12 dB has been specified. Nexxim tries the values in the range and select the one that gives the best eye.

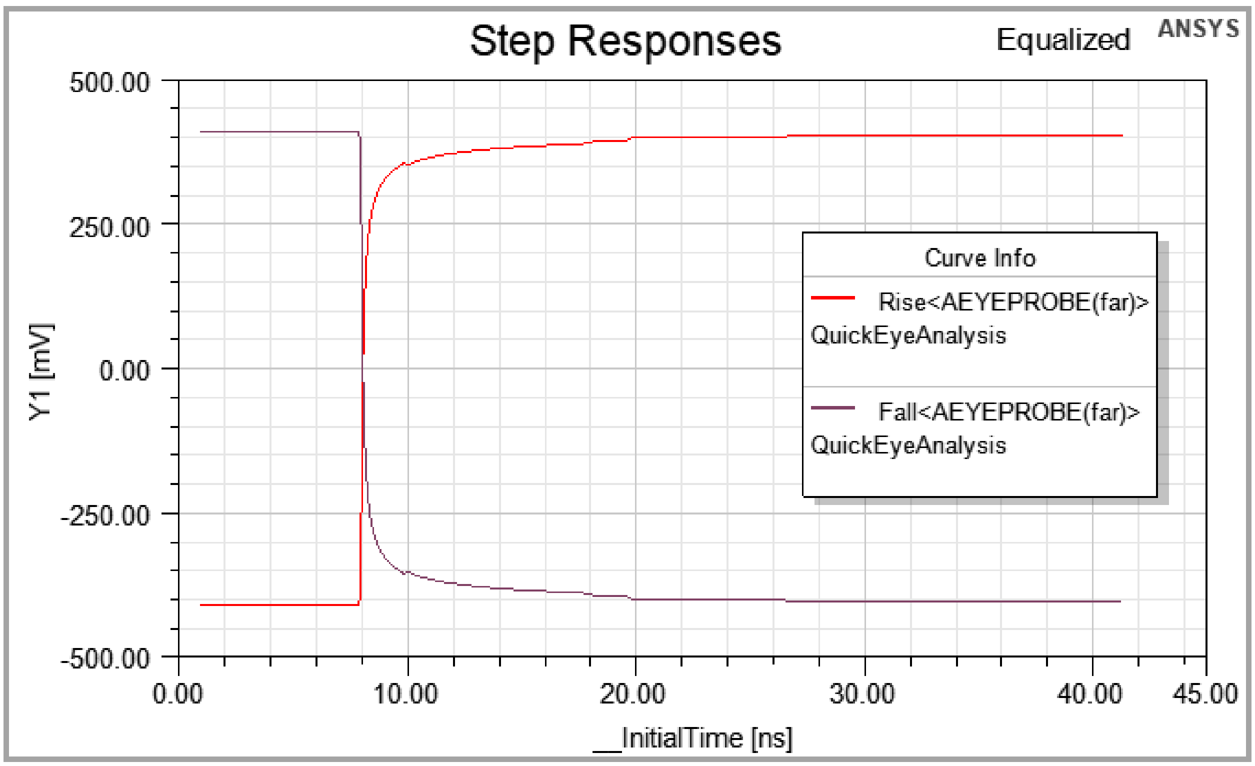
Click window and **OK** again to close the **Properties** window.

7. The **Equalized** circuit project has setups for Transient, QuickEye, and VerifEye analyses.
   a. From the **Project Manager** window, open **Equalized**, right-click **Analysis**, and select **Analyze**. This starts all three analyses. It takes a few minutes for all analyses to complete.

b. The same reports are set up.

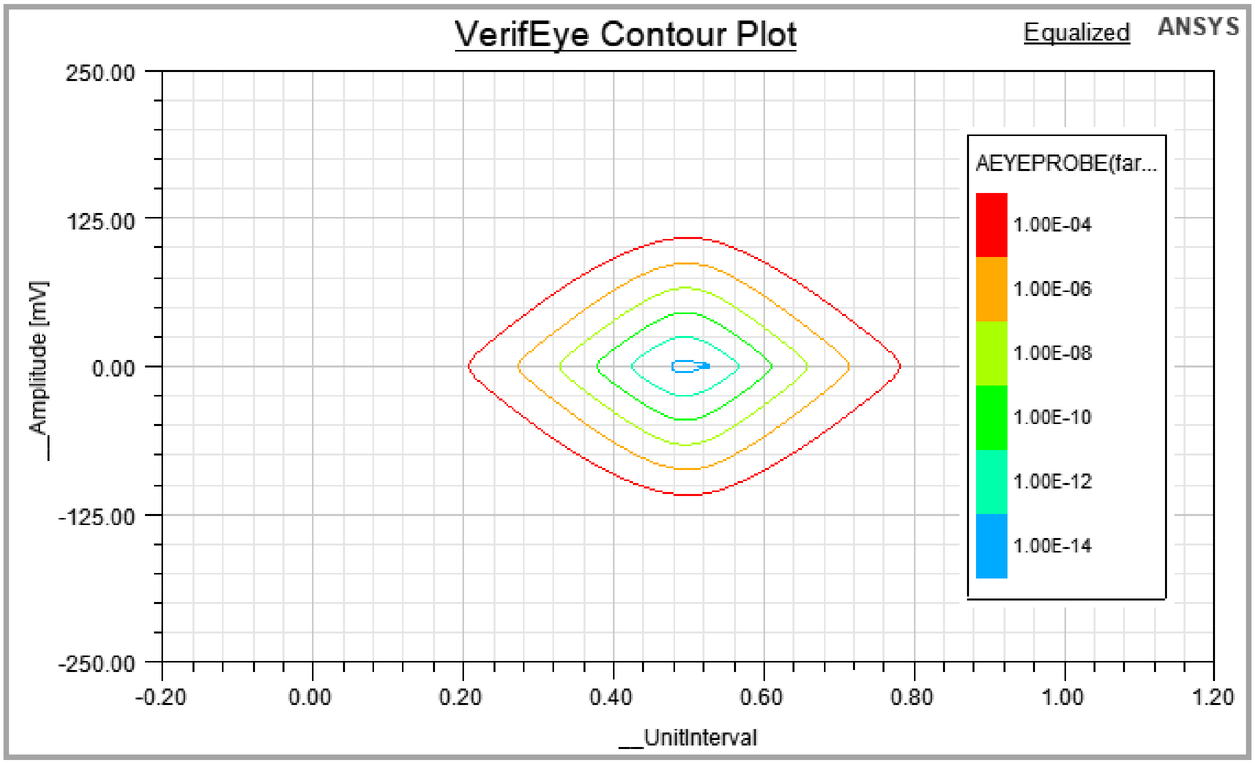


c. Double-click the **Step Responses** report to open it:

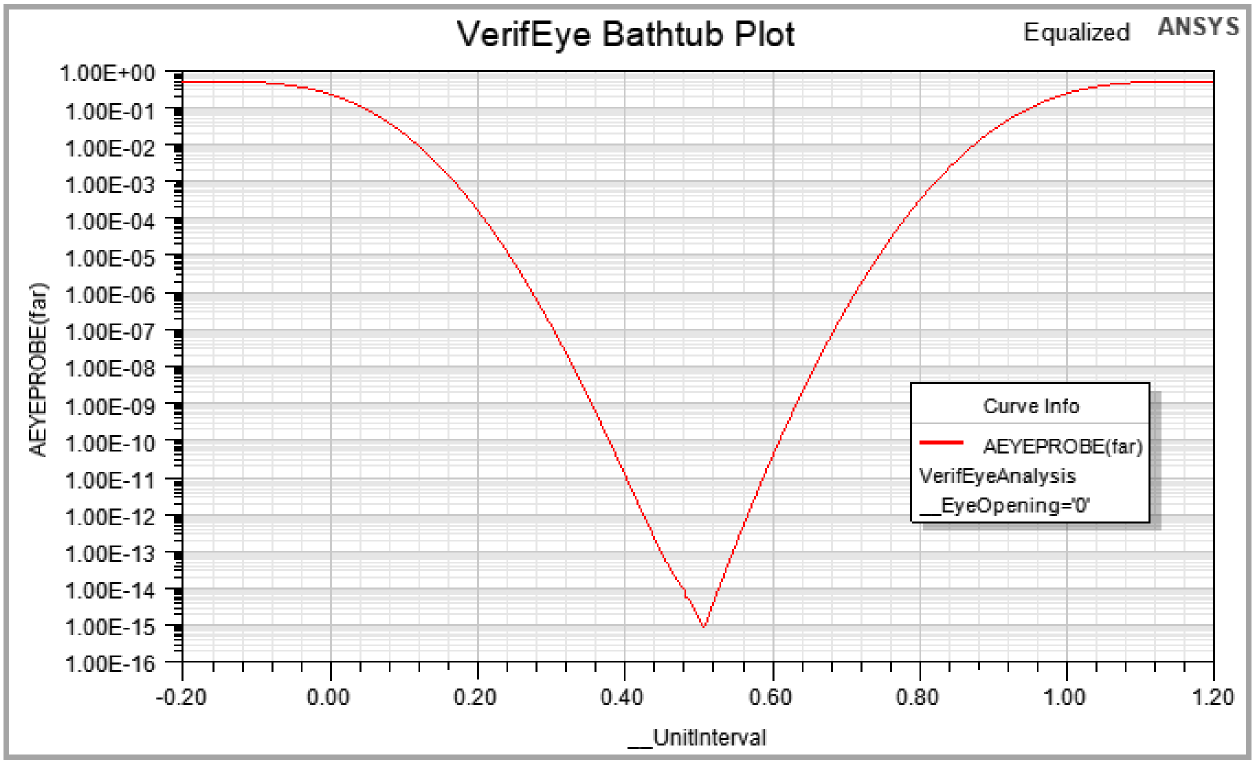


The step responses are unchanged by the equalization.

d. Double-click the **VerifEye Contour Plot** report. Equalization has significantly reduced the bit error rate in the center of the eye.

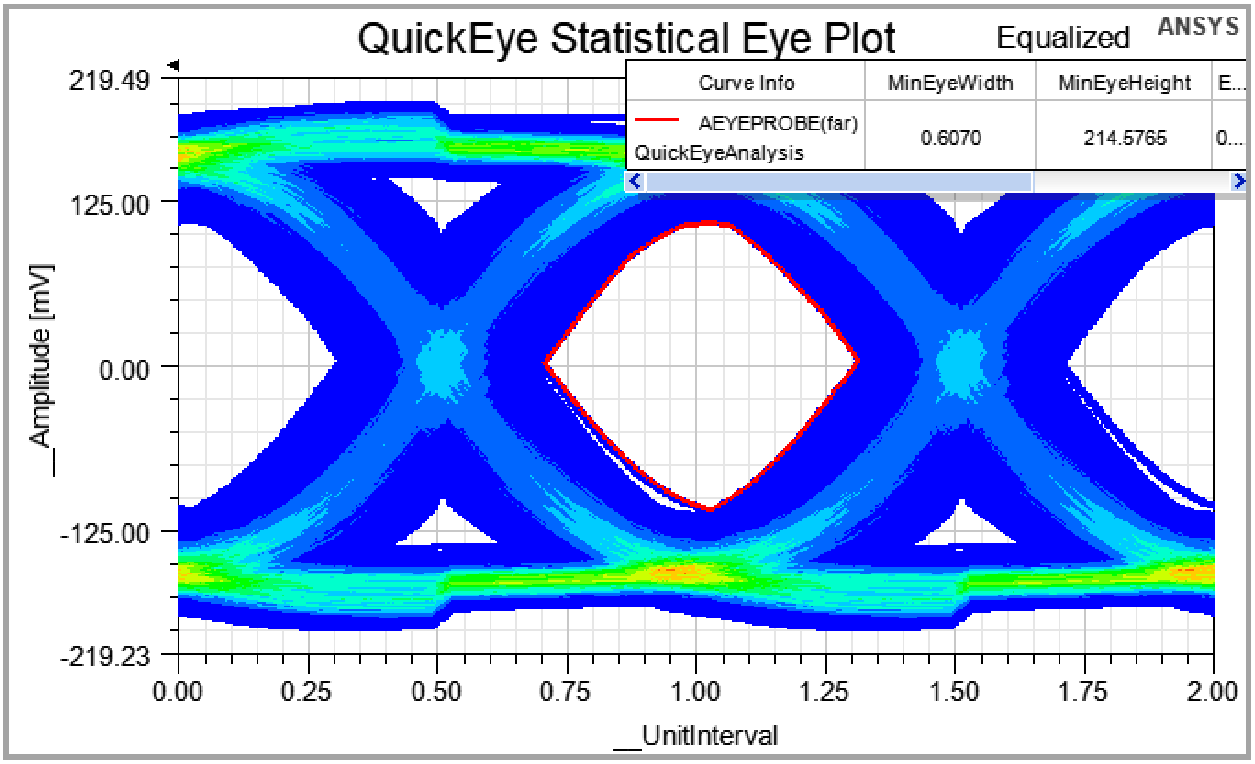


The VerifEye Bathtub Plot shows the corresponding BER curve.



VerifEye Bathtub Plot
Equalized
ANSYS
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
1.00E-05
1.00E-06
1.00E-07
1.00E-08
1.00E-09
1.00E-10
1.00E-11
1.00E-12
1.00E-13
1.00E-14
1.00E-15
1.00E-16
AEYEPROBE(far)
-0.20
0.00
0.20
0.40
0.60
0.80
1.00
1.20
__UnitInterval
Curve Info
AEYEPROBE(far)
VerifEyeAnalysis
__EyeOpening='0'

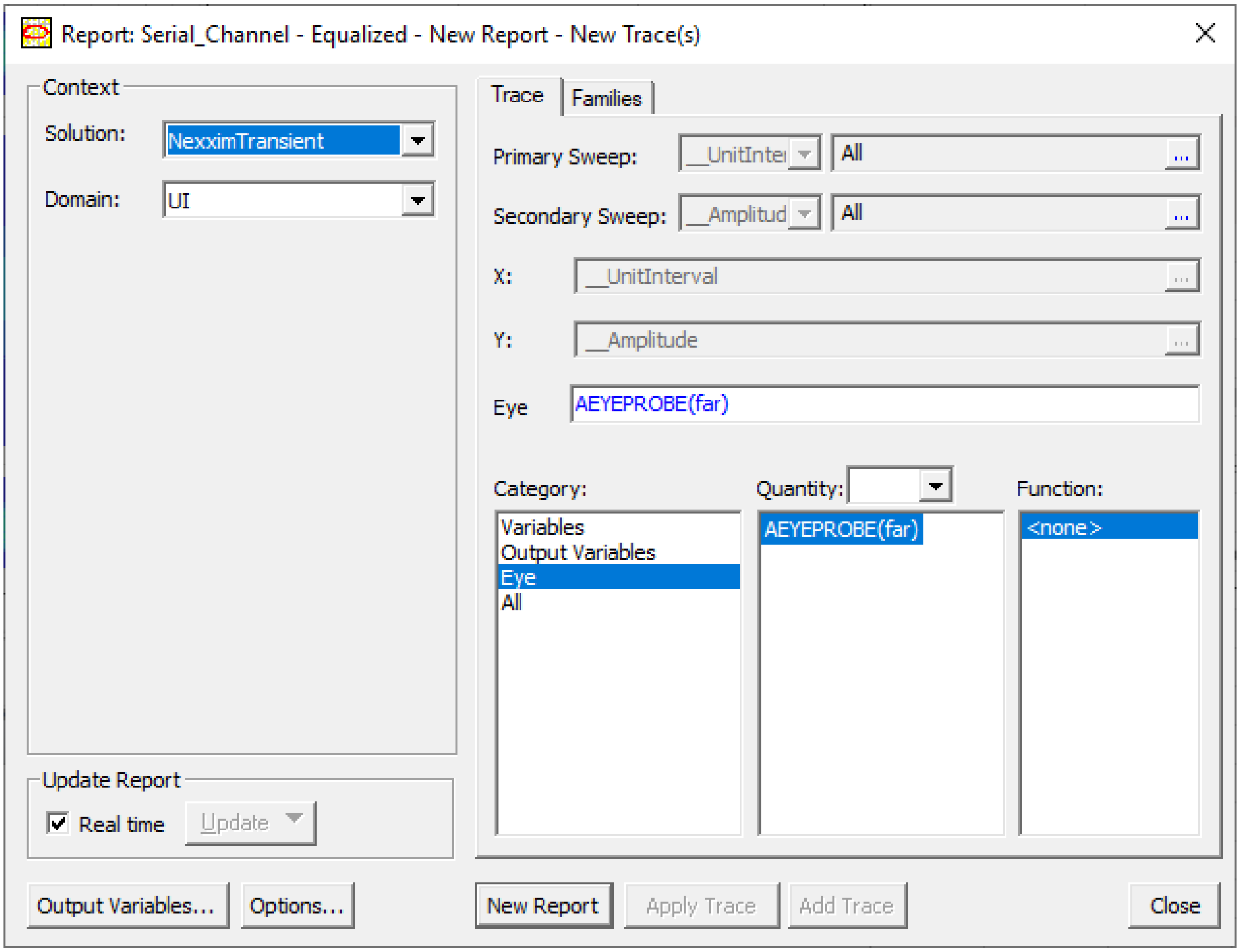
e. Double-click the **QuickEye Statistical Eye Plot** report:



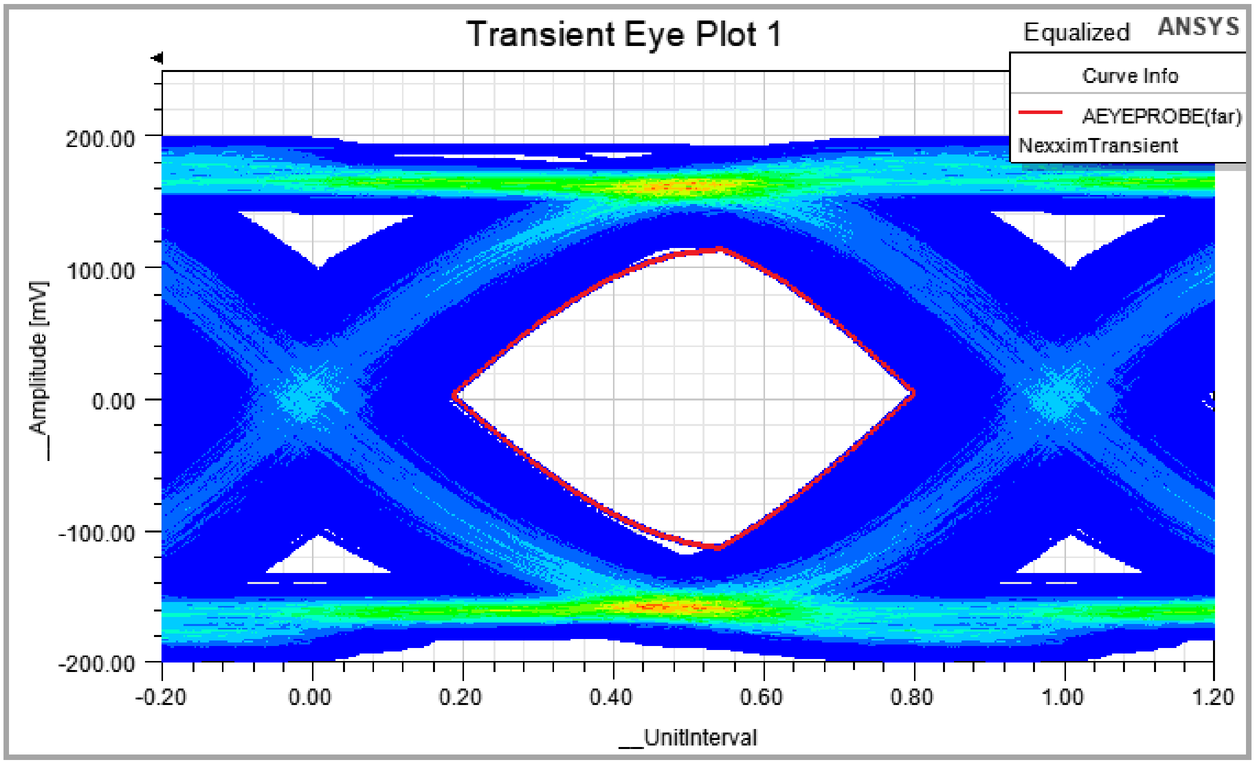
f. Create a Transient analysis report:

1. From the **Project Manager** window, right-click **Results** and click **Create Statistical Eye Report** > **Statistical Eye Plot**.

2. In the **Report** window, set **Solution** t o**NexximTransient**.



3. Click **New report** to generate the **Transient Statistical Eye Plot** report and **Close** to close the **Report** window.



g. When you are done, click **Window** > **Close All** to close all the schematic and report windows.

External Step Response Example

This example shows how to use the External Step Response (ESR) element in Circuit schematics to provide the impulse response data for QuickEye and VerifEye. The example demonstrates how to generate an impulse-response file using transient analysis. Once the impulse-response data has been generated and exported to a file, QuickEye and VerifEye can use the file data to calculate the channel response without re-running a transient analysis. This can save time when the channel step-response does not change between analyses.

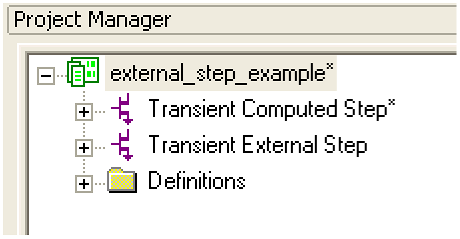
Generating Impulse Response Data

The example design is available in the Examples directory. The design contains both parts of the example as separate projects.

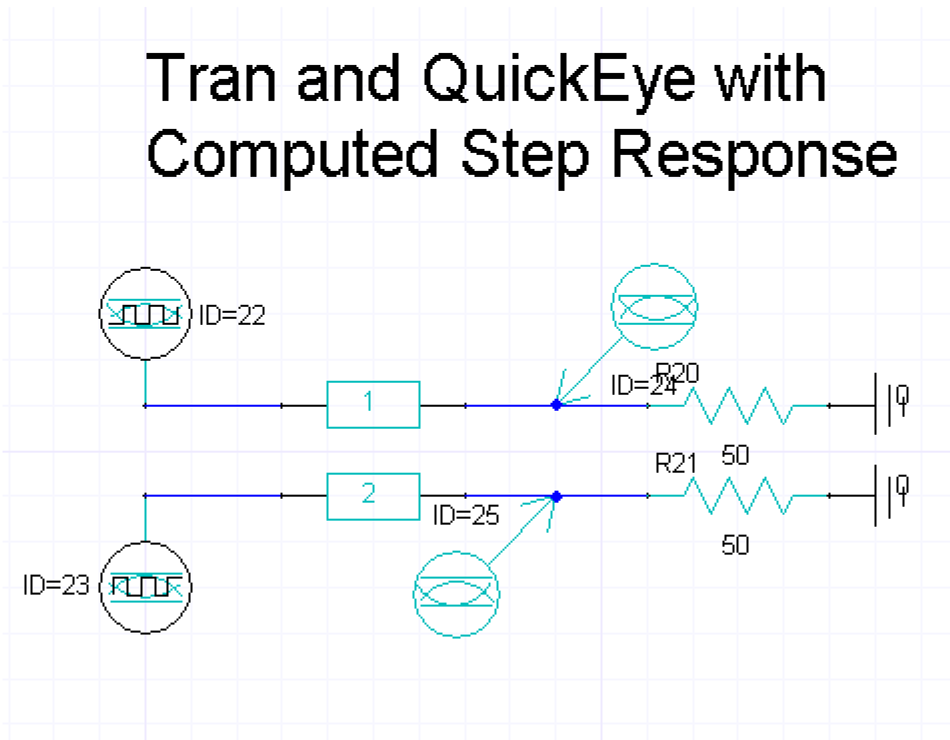
Open the design from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **SignalIntegrity**, then select the file **External_Step_Example.aedt**. Click **Open**.

The design opens with two projects:



3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.
4. Click the **Transient Computed Step** project to open the schematic.



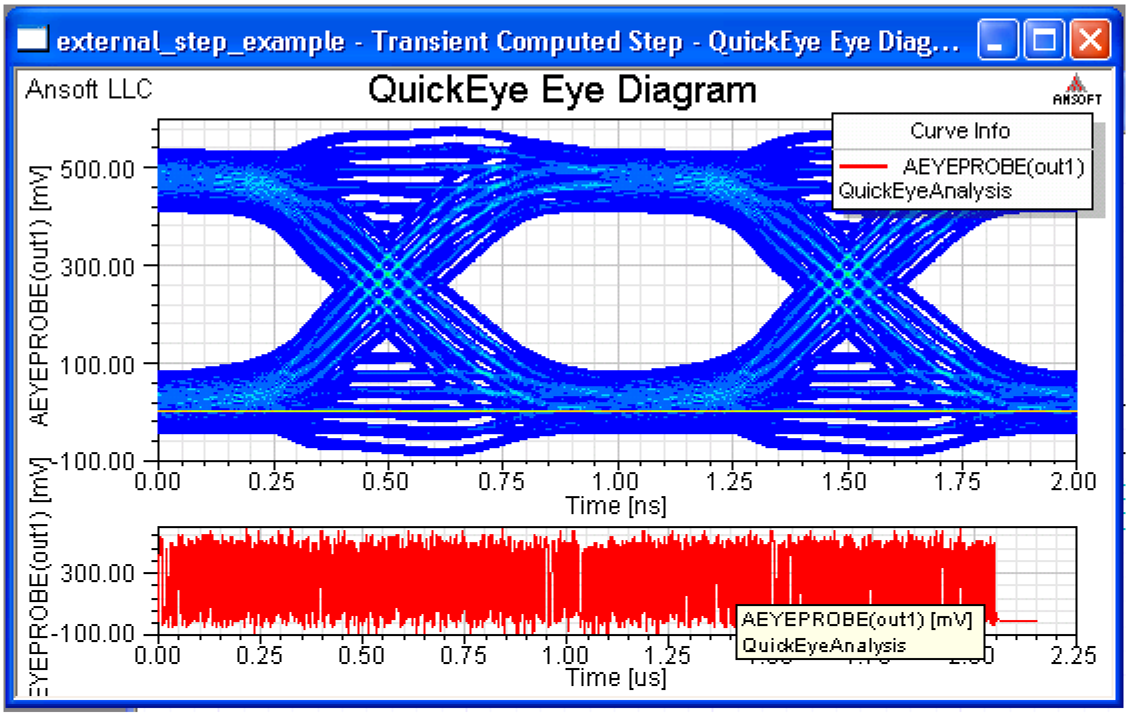
The circuit contains two Eye Sources and two Eye Probes.

The through path from Eye Source ID=22 to Eye Probe ID=24 (out1) is the main channel. The through path from Eye Source ID=23 to Eye Probe ID=25 (out2) is the secondary (competing) channel). The two sources produce PRBS bit streams with different seeds.

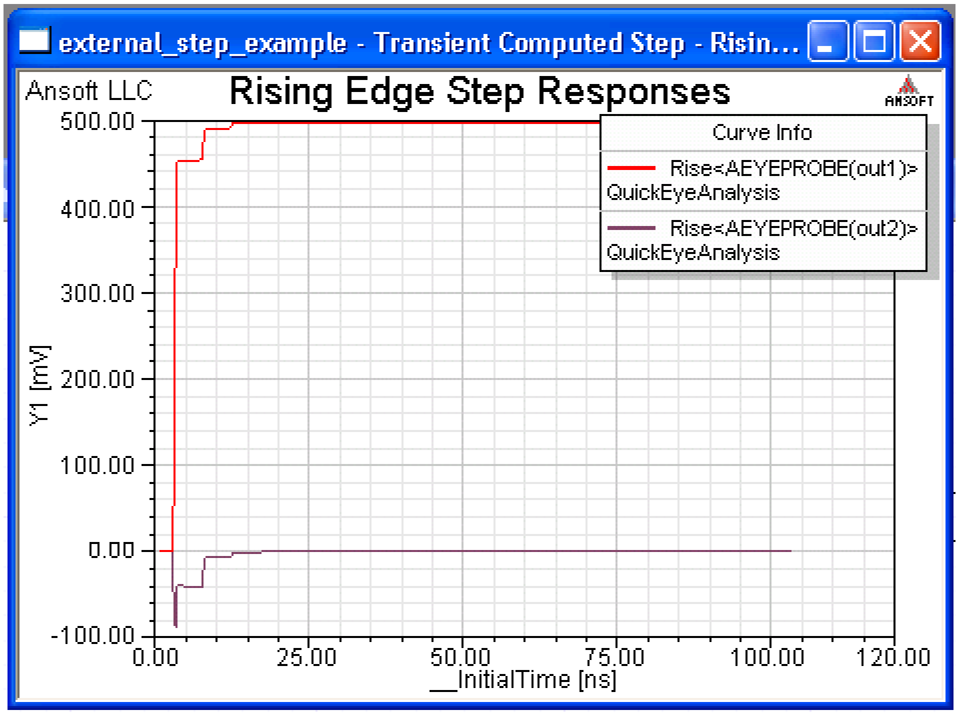
The paths from Eye Source ID=22 to Eye Probe ID=25 (out2) and from Eye Source 23 to Eye Probe ID=24 (out1) generate the crosstalk terms.

From the **Project Manager** window, expand the **Analysis** icon, right-click **QuickEyeAnalysis**, and select **Analyze**. The analysis runs to completion.

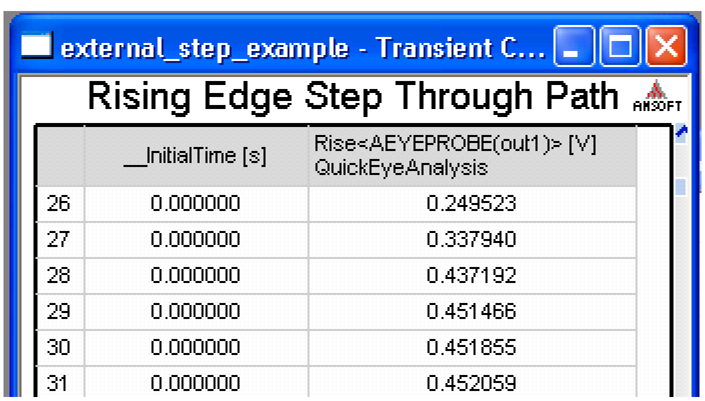
Expand the **Results** icon and double-click on the **QuickEye Eye Diagram** report:



To view the rising step response data, double-click on the **Rising Edge Step Responses** report:



The upper trace is the main channel response. The lower trace is the crosstalk response. To view the step response data in tabular form suitable for exporting, double-click the **Rising Edge Step Through Path** report:



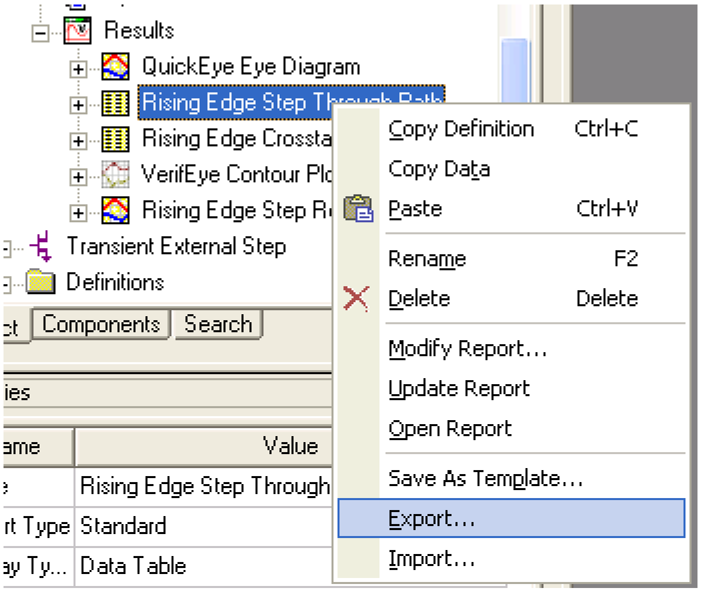
Rising Edge Step Through Path

	__InitialTime [s]	Rise<AEYEPROBE(out1)> [V] QuickEyeAnalysis
26	0.000000	0.249523
27	0.000000	0.337940
28	0.000000	0.437192
29	0.000000	0.451466
30	0.000000	0.451855
31	0.000000	0.452059

The data window has been scrolled down to a segment with data.

By default, the data is in units such as nanoseconds and millivolts. The ESR component requires units of **seconds** and **volts**. To change to the units shown above, click the column heading (e.g., **_InitialTime**) and set the **Units** property to the appropriate unit.

The data from this table have been exported to step response files for the second part of this example. To export the data on the data table report, right-click the report and select **Export** to open an explorer window.



Browse to the appropriate directory. Save the data using the **Post-processor format files (*.txt)** format.

The data file **rise_through_path.txt** in the Examples directory was created in this way.

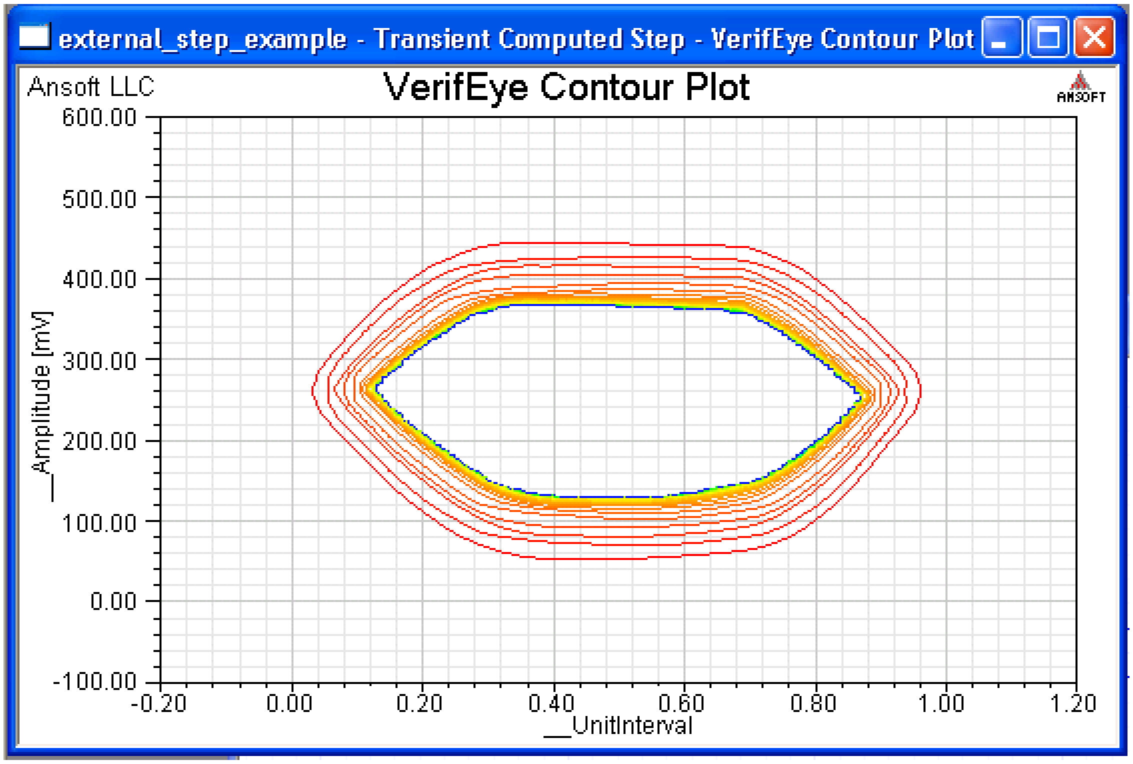
The data table report **Rising Edge Crosstalk Response** contains the response data for one crosstalk path, Eye Source ID=22 to EyeProbe ID=25 (out2). The exported data is in the data file **rise_xtalk_term.txt** in the Examples directory.

Similar reports are generated and exported to create the data files **fall_through_path.txt** and **fall_xtalk_term.txt** in the Examples directory.

In this simplified example, both through paths use the same rise and fall times, and the crosstalk terms are assumed to be the same in both crosstalk paths.

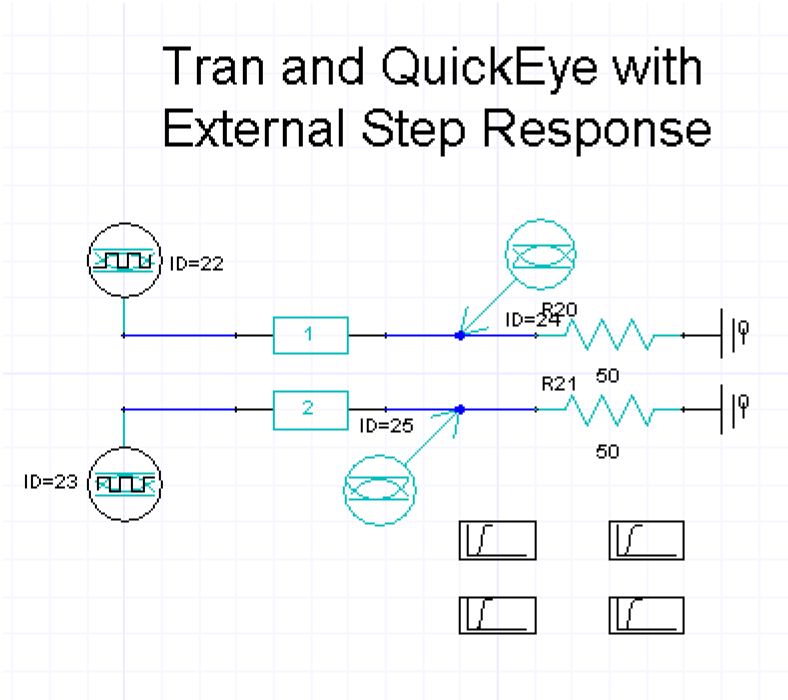
The same analysis may be run using VerifEye instead of QuickEye. Under the Analysis icon, right-click **VerifEye Analysis** and select **Analyze**. The analysis runs to completion.

To view the results: double-click the **VerifEye Contour Plot** report:

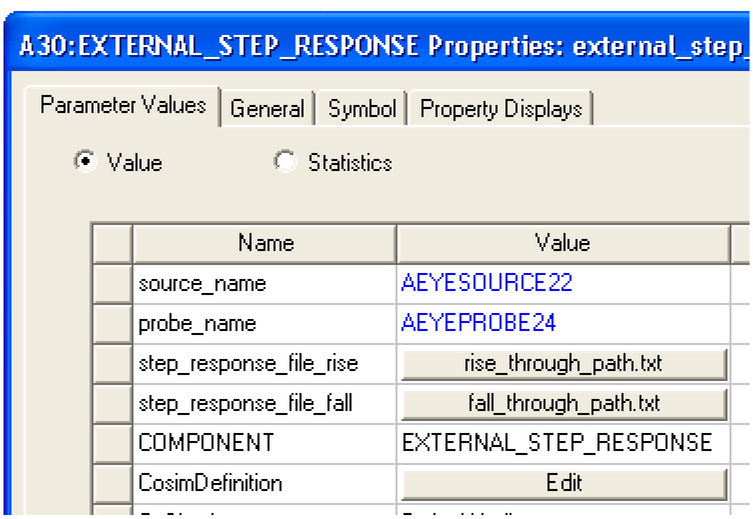


Using External Step Response Data

The second project uses the step response data instead of repeating the transient analysis. Under the Design icon, click the **Transient External Step** project:



The circuit is basically the same, only now the responses is controlled by the four External Step Response ESR) elements in the schematic. Click the upper-left ESR element and display its properties:



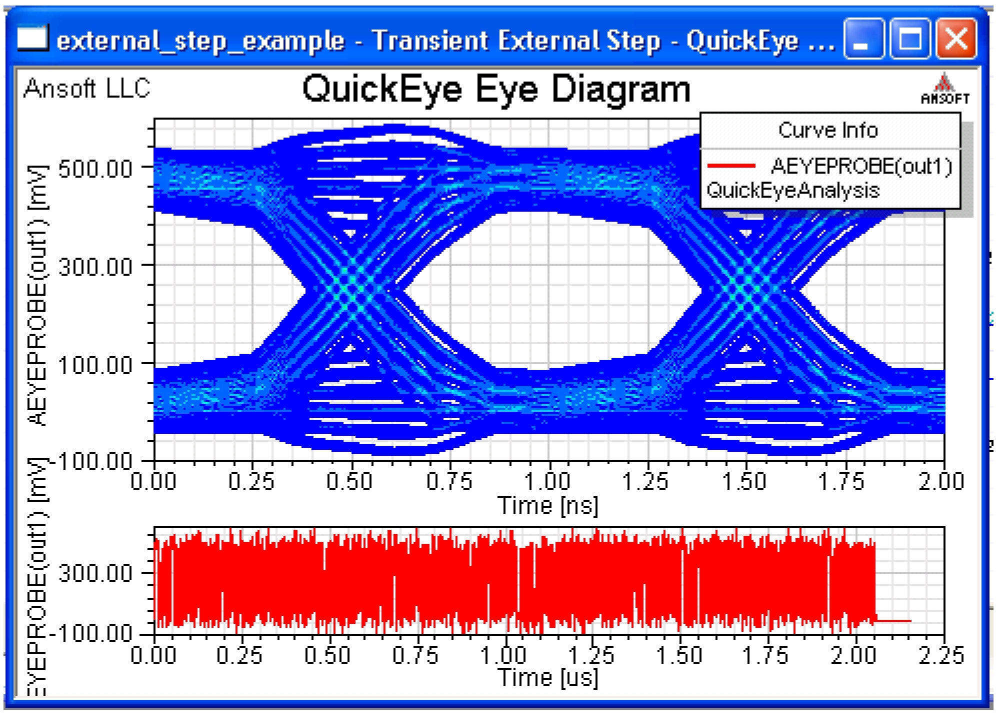
The ESR elements do not physically connect to the circuit. Each ESR element specifies a rise time and a fall time for one of the four paths:

- The main through path from Eye Source 22 to Eye Probe 24 (out1).
- The second through path from Eye Source 23 to Eye Probe 25 (out2).
- The crosstalk path from Eye Source 22 to Eye Probe 25 (out2).
- The crosstalk path from Eye Source 23 to Eye Probe 24 (out1).

The two through paths use the same rise and fall data files. The two crosstalk paths share a different set of rise and fall data files.

From the **Project Manager** window, expand the **Analysis** icon, right-click **QuickEyeAnalysis**, and select **Analyze** on the dropdown menu. The analysis runs to completion very quickly compared to the earlier transient analysis. Because the step data is provided, no transient analysis is run, saving significant time.

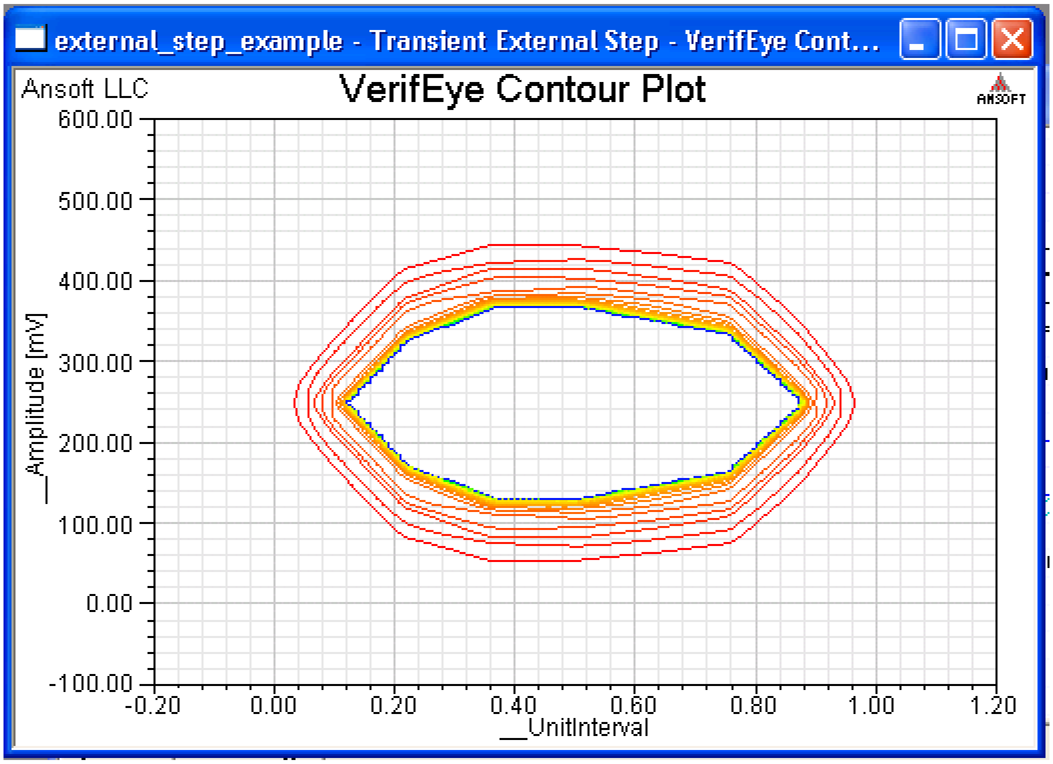
Expand the **Results** icon and double-click the **QuickEye Eye Diagram** report:



The result is nearly identical to the QuickEye Eye diagram produced by transient analysis.

An equivalent speedup is also available for VerifEye analysis. From the **Project Manager** window, expand the **Analysis** icon, right-click **VerifEyeAnalysis**, and select **Analyze**. This analysis also runs to completion quickly (much of the time is needed to generate the eye diagram). Providing the step data saves VerifEye some processing time.

Expand the **Results** icon and double-click the **VerifEye Contour Plot** report:



The result is identical to the VerifEye Contour diagram produced by transient analysis.

AMI Analysis Example

The Algorithmic Modeling Interface (AMI) allows time-domain simulation of a linear channel using customer-supplied models and parameters for the transmitter and receiver. Reports include eye diagrams and bit-error-rate (BER) contours. The Nexxim implementation allows you to sweep the value of a parameter in an AMI file. VerifEye analysis can then be run with the AMI channel definition in order to give an accurate picture of the channel response.

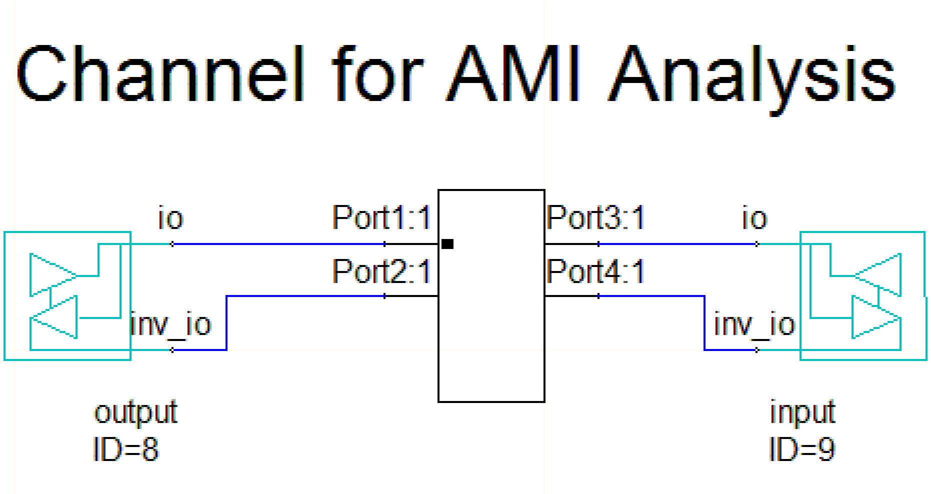
Channel Example with AMI

The schematic for the circuit is available in the Examples directory.

Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open the **Examples** directory in an **Open** explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **AIM_ Example.aedt**, or select **AIM_Example.aedt** and click **Open**.

The channel schematic appears in the design window:



3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.
4. Double-click the IBIS output buffer to display its properties:

Name	Value	Unit	Evaluated...	Description
file	ibis_ami_example_tx.ibs			Name of IBIS file (required)
typ	typ		"typ"	Parameter range (typ, min, max, fast, slow)
power	internal			Power mode (on=internal, off=external)
buffer	output			Buffer type (output)
Model1	example_model_tx		"example...	
Model2	example_model_tx		"example...	
polarity	Non-Inverting			Polarity (Non-Inverting, Inverting, No Polarity,
--Pin Properties				
logic_in	Internal Source			
--End Pin Properties				

The **file** parameter points to the IBIS transmitter file. The **logic_in** property is set to **Internal Source**, enabling the Eye Source properties.

Eye Source Parameters				
trise	300	ps	300ps	Rise time of the source
tfall	300	ps	300ps	Fall time of the source
phase_delay	0	s	0s	Phase of the source in seconds
UIorBPS	UnitInterval			Choose either Unit Interval or Bits per Second.
UIorBPSValue	1	ns	1ns	Enter a value for Unit interval or Bits per second
DCDFractionorTime	Fraction			Fraction=DCD is a fraction of the UI, 0 <= DCD <= 1. ...
dcd	0		0	Duty cycle distortion
txrj	[1e-11]			Gaussian (random) jitter std deviation
txpj				Periodic (random) jitter amplitude
txuj				Uniform (random) jitter amplitude
txcj				Name of user-defined transmit jitter file
repeat_count	0		0	Number of times to repeat the bit pattern.
step_resp_num_ui	100		100	Num of UIs for initial step response
do_encoding	None			
hold_last_bit				Hold last bit or repeat the sequence.
BitPattern	prbs_no=1...			Bit pattern for QuickEye
EyeMeasurementFun...				
End Eye Source Parameters				

The Eye Source parameters call for rise and fall times of 300ps, with a bit pattern generated via the internal PRBS. A small amount of random jitter has been added at the transmitter.

LIBRARY_W32	ibis_ami_example_w32.dll
LIBRARY_W64	ibis_ami_example_w64.dll
LIBRARY_L32	ibis_ami_example_l32.so
LIBRARY_L64	ibis_ami_example_l64.so
PARAMETERS_FILE	$PROJECTDIR/models/AMI_Example/ibis_ami_example_TX.ami
COMPONENT	AMI_SOURCE

The **library** files contain the AMI code for both transmitter and receiver.

The **parameter_file** contains FFE data for the transmitter:

```
| method = 1 applies the tap weights inside AMI_init to the channel
impulse
| method = 2 applies the tap weights inside AMI_Getwave to the wave
| results of both methods should be nearly identical, but method 1 is
faster.
(method 1) | 1 or 2
(ffe 1 -0.2 0.1 -0.05) | arbitrary number of tap weights
```

> **Note:** Method 1 is required for VerifEye analysis to produce the same result as AMI. VE uses just the impulse data from AMI_Init, and does not use the AMI_Getwave data.

The AMI File Properties display the data on the parameter file:

Properties from AMI file			
Ignore_Bits	4		4
Max_Init_Aggressors	10		10
Init_Returns_Impulse	☑		
GetWave_Exists	☑		
Use_Init_Output	☑		
method	2		2
ffe_weight_1	-0.2		-0.2
ffe_weight_2	0.1		0.1
ffe_weight_3	-0.05		-0.05
End Properties from AMI file			

5. Double-click the AMI Receiver to display its properties:

Name	Value	Unit	Evaluated...	Description
file	ibis_ami_example_rx.ibs			Name of IBIS file (required)
typ	typ		"typ"	Parameter range (typ, min, max, fast, slow)
power	internal			Power mode (on=internal, off=external)
buffer	input			Buffer type (input)
Model1	example_model_rx		"example...	
Model2	example_model_rx		"example...	
polarity	Non-Inverting			Polarity (Non-Inverting, Inverting, No Polarit
--Pin Properties				
Out	Out_pin			
--End Pin Properties				

The **file** parameter points to the IBIS transmitter file.

LIBRARY_W32	ibis_ami_example_w32.dll
LIBRARY_W64	ibis_ami_example_w64.dll
LIBRARY_L32	ibis_ami_example_l32.so
LIBRARY_L64	ibis_ami_example_l64.so
PARAMETERS_FILE	$PROJECTDIR/models/AMI_Example/ibis_ami_example_RX.ami

The **library** files contain the AMI code for both transmitter and receiver.

The parameters file contains FFE and clock data for the receiver:

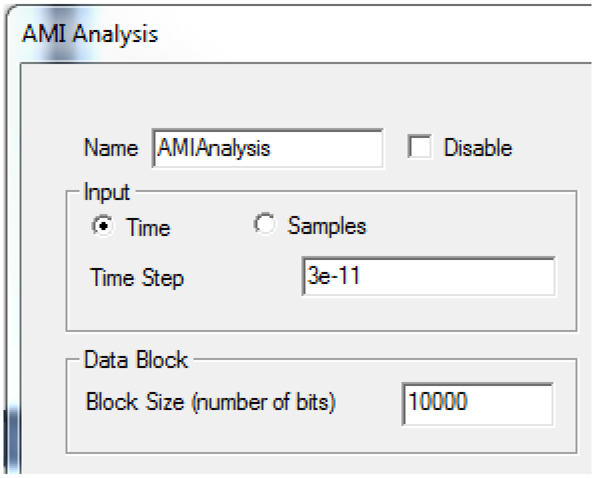
```
| method = 1 applies the tap weights inside AMI_init to the channel impulse
| method = 2 applies the tap weights inside AMI_Getwave to the wave
| results of both methods should be nearly identical, but method 1 is faster
(method 1) | 1 or 2
(ffe 1 -0.1 -0.05) | arbitrary number of tap weights
(clock_threshold 0.4) | threshold for clock tics
```

> **Note:** Method 1 is required for VerifEye analysis to produce the same result as AMI. VE uses just the impulse data from AMI_Init, and does not use the AMI_Getwave data.

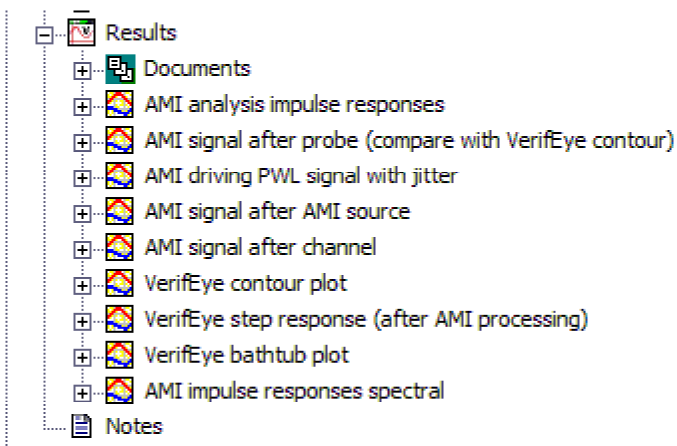
The AMI File Properties display the data on the parameter file:

--Properties from AMI file				
Ignore_Bits	4		4	
Max_Init_Aggressors	10		10	
Init_Returns_Impulse	☑			
GetWave_Exists	☑			
Use_Init_Output	☑			
method	1		1	
clock_threshold	0.4		0.4	
ffe_weight_1	rx_ffe1		1.2	
ffe_weight_2	rx_ffe2		-0.1	
ffe_weight_3	rx_ffe3		0.05	
--End Properties from AMI file				

6. Expand the **Project Manager** window to show the Analysis setups, and double-click the **AMI Analysis** setup. The setup specifies the input timestep and data block size:

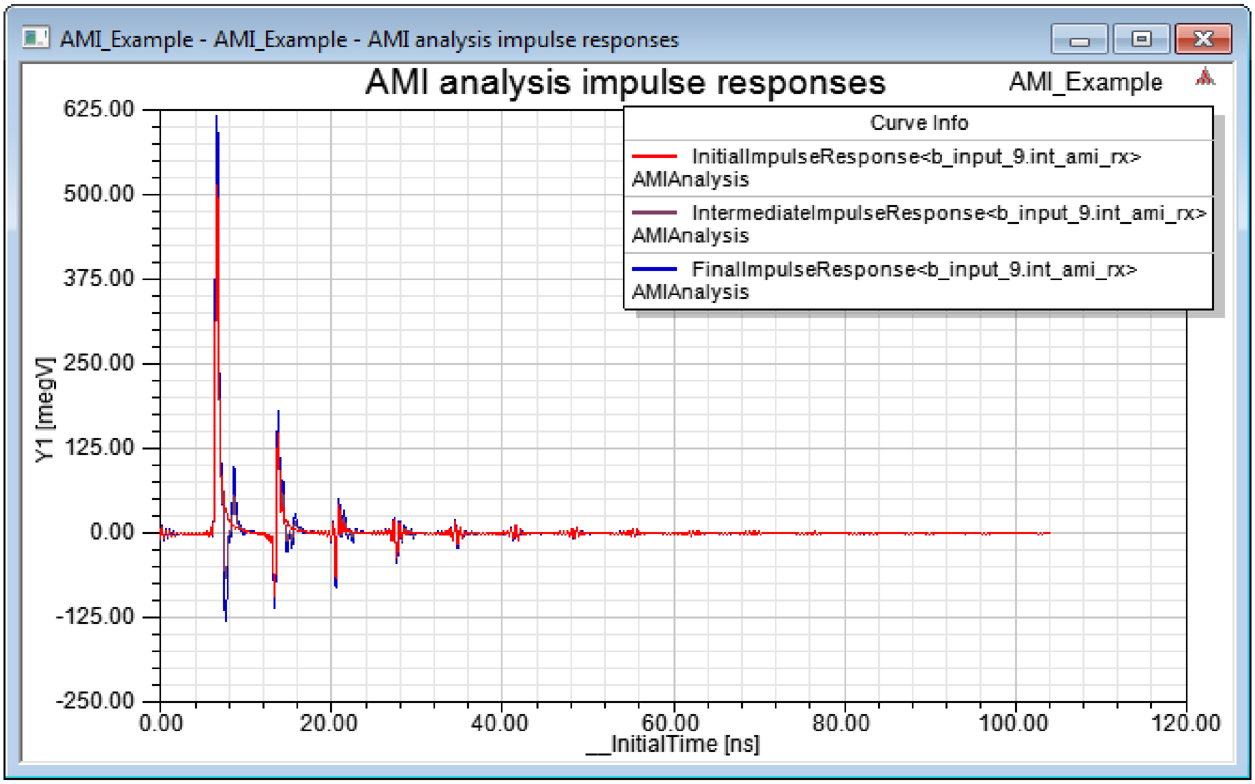


7. Right-click the **AMI Analysis** setup and select **Analyze**. The AMI analysis runs to completion.
8. Expand the **Project Manager** window to show the Reports:



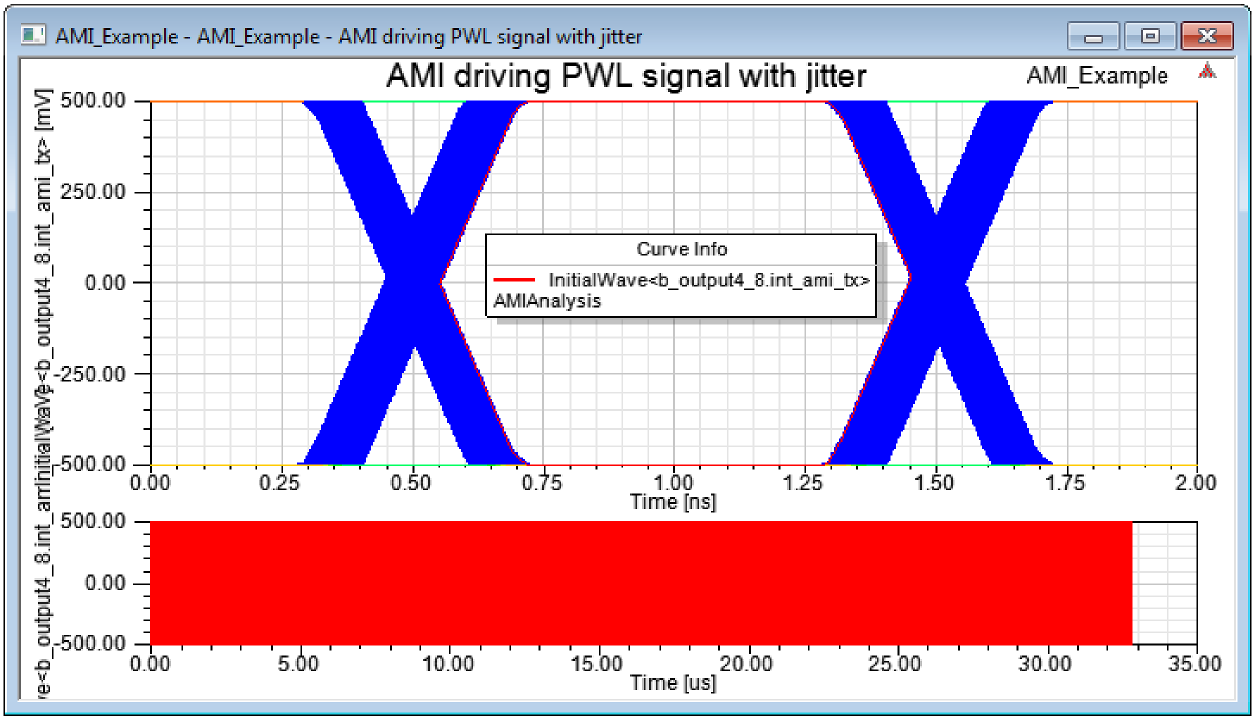
9. Double-click **AMI analysis impulse responses** to open the report showing the impulse responses as initially calculated and as modified by the FFE tap weights in the Source or Probe by FFE method 1 or 2. With method 1 (shown), the FFE taps are applied to the impulse response and are reflected on this graph. With method 2, the equalization is

applied internally to the model and has no affect the impulse response.

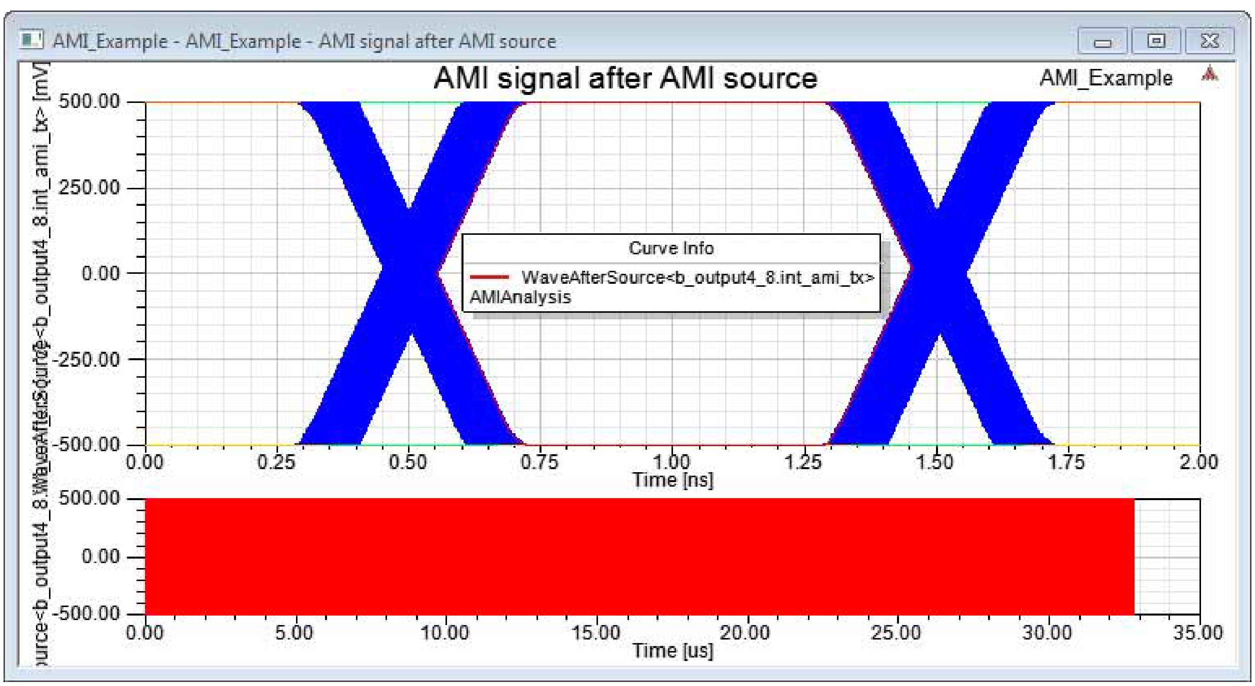


Edit the parameters files to select the method and change the FFE tap numbers and weights.

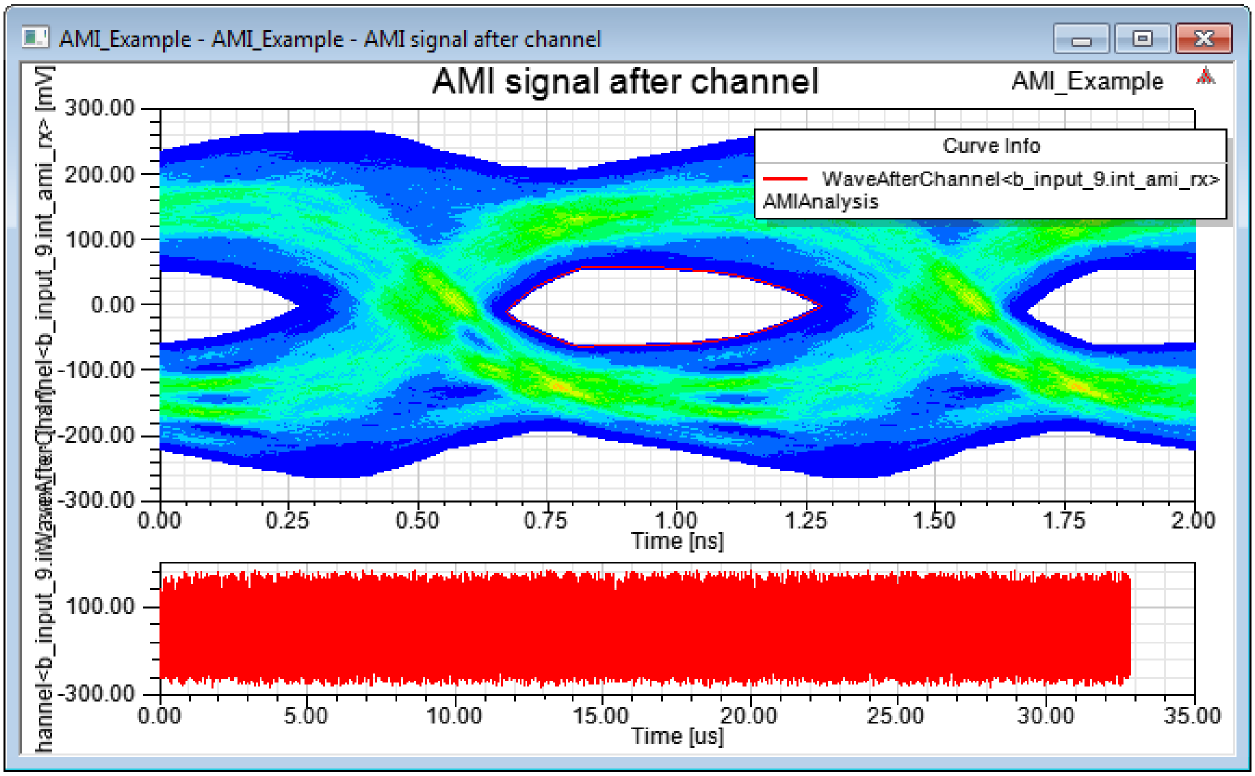
10. Double-click **AMI driving PWL signal with jitter** to open the report showing the eye diagram of the input signal.



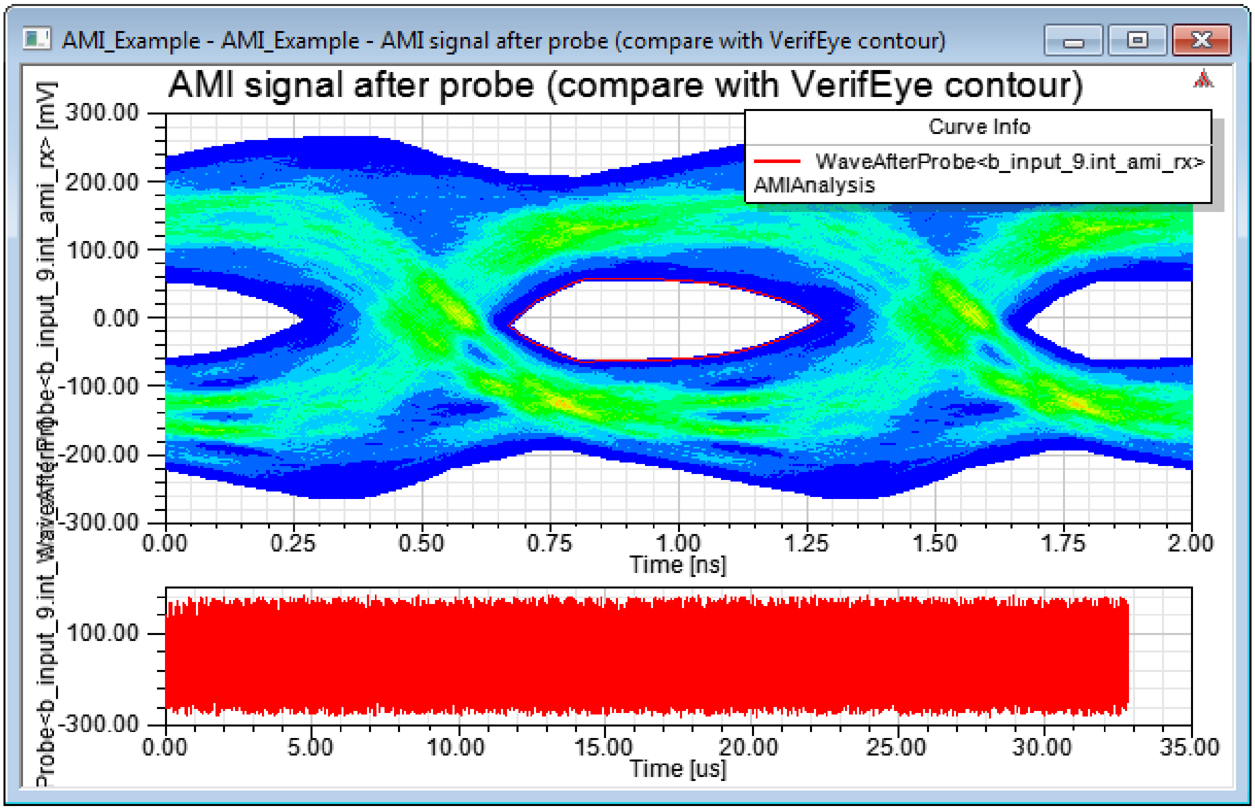
11. Double-click **AMI signal after AMI source** to open the report showing the eye diagram of the output signal on the AMI Source.



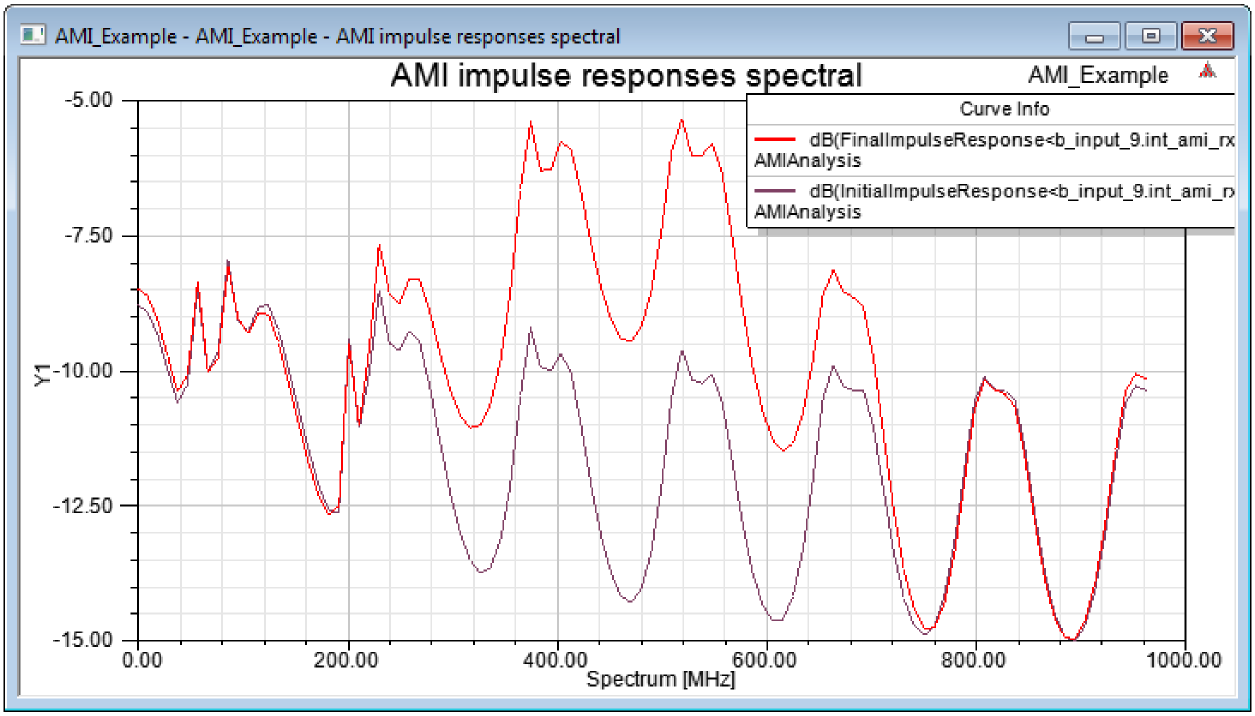
12. Double-click **AMI signal after channel** to open the report showing the eye diagram of the output signal from the channel.



13. Double-click **AMI signal after probe** to open the report showing the eye diagram of the output signal from the AMI Probe.



1. Double-click **AMI impulse responses spectral** to open the report showing the initial and final impulse responses in the spectral domain.

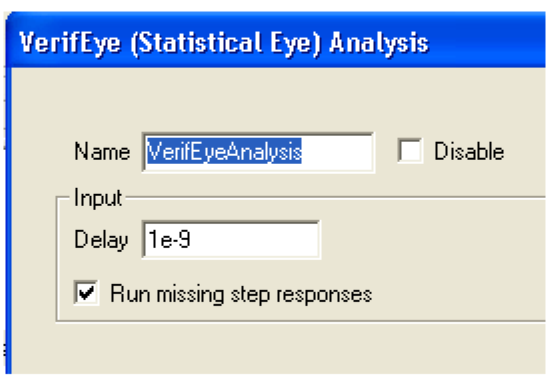


See Plotting Spectral Domain Data in the Reports topic. Although that topic mentions only transient solutions, the Context controls also apply to AMI analyses.

Channel Example with VerifEye

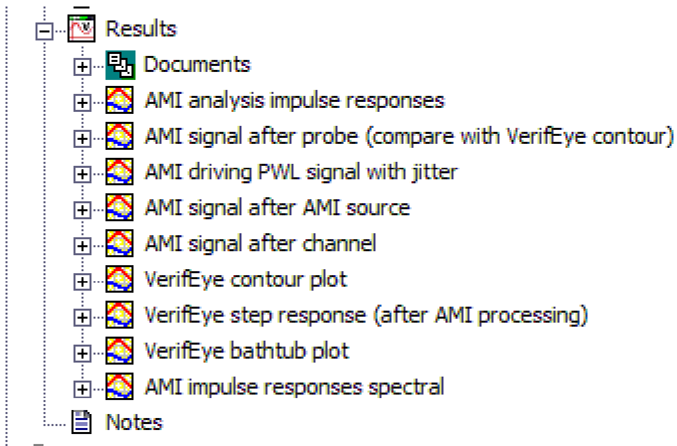
Use the same project, **AMI_Example.aedt**, to compare the AMI analysis with VerifEye analysis using the AMI impulse responses definitions.

1. Double-click the **VerifEye Analysis** setup:

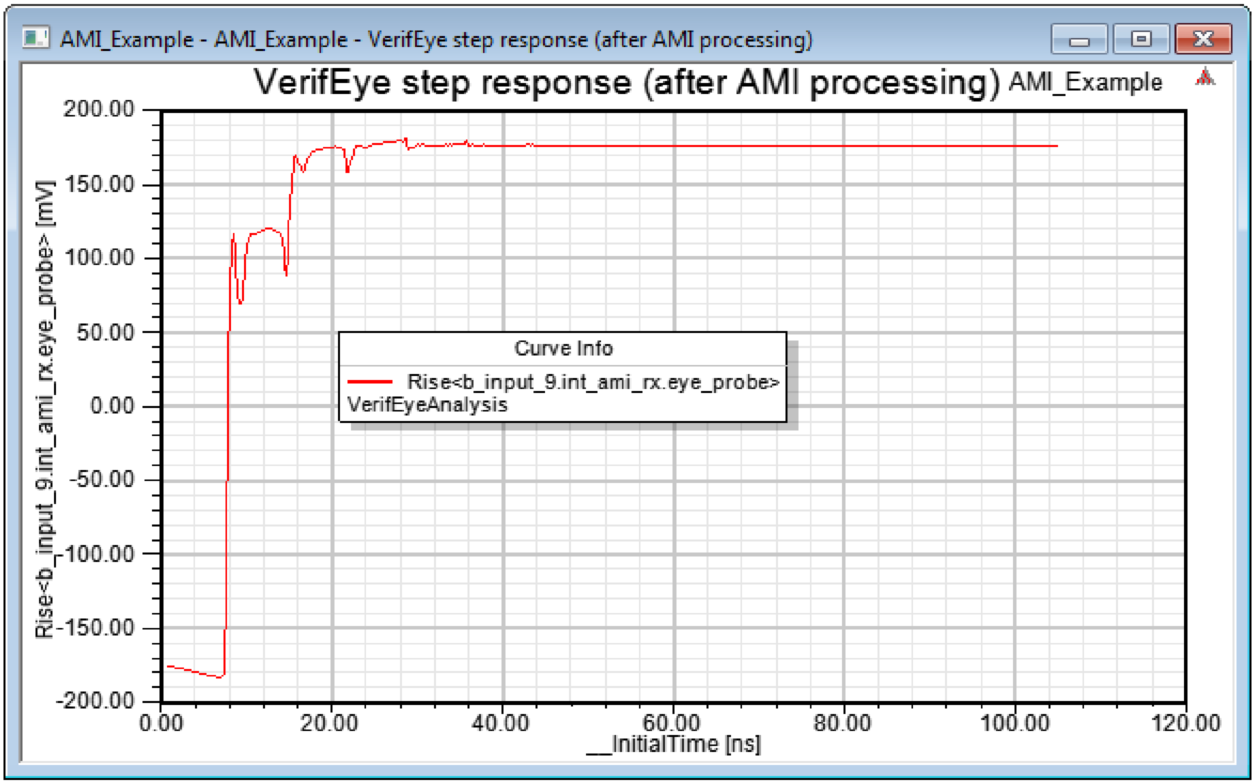


This is the default setup for VerifEye.

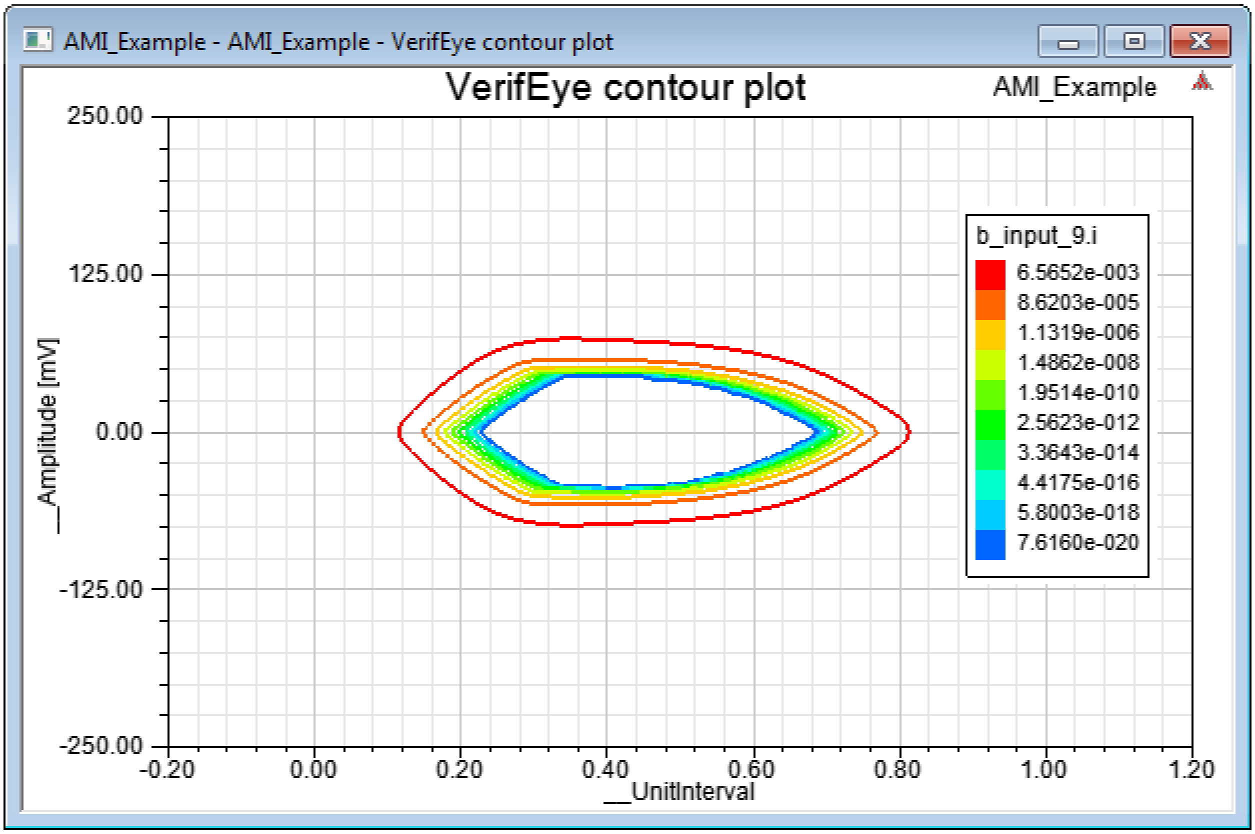
2. Right-click the **VerifEye Analysis** setup and select **Analyze**. The VerifEye analysis runs to completion.
3. Expand the **Project Manager** window to show the Reports:



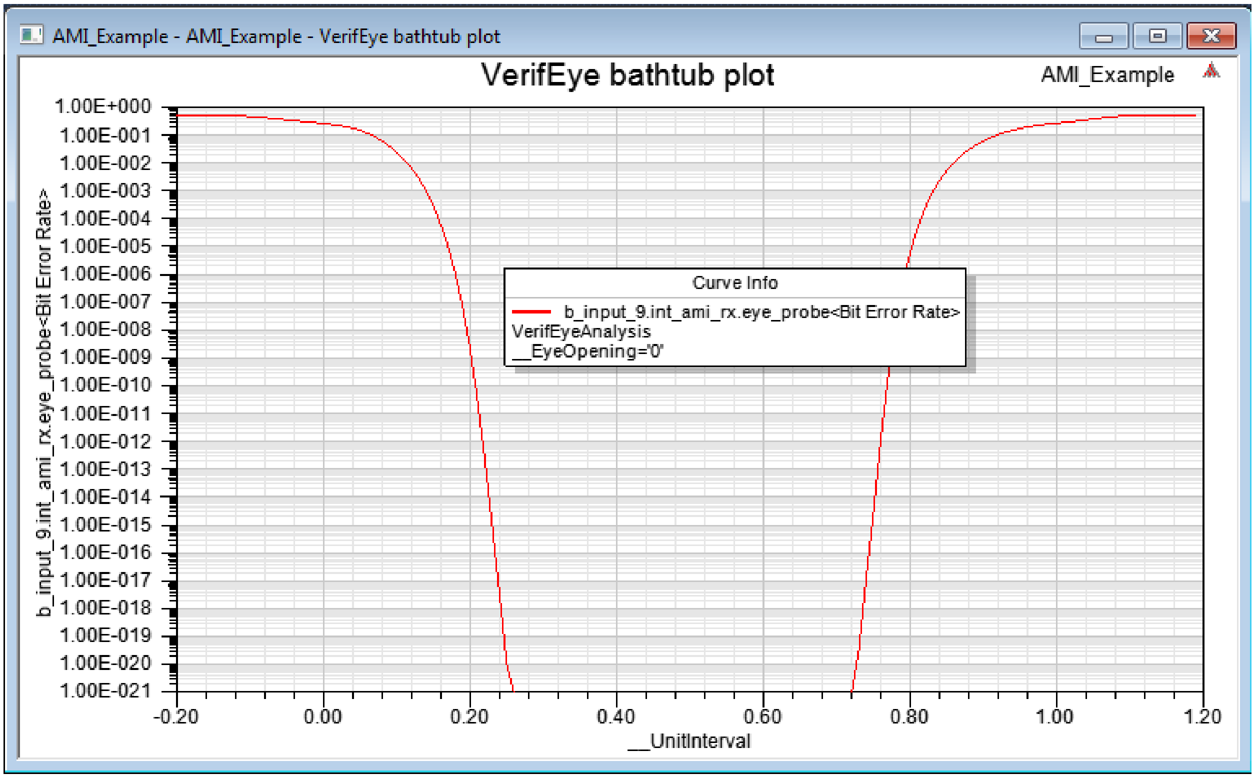
4. Double-click **VerifEye step responses** to open the report showing the rising step response as calculated by VerifEye using the FFE tap weights in the Source or Probe by FFE method 1. With method 1, the FFE taps are applied to the impulse response and are reflected on this graph. Because VerifEye uses only the **AMI_Init** function, only Method 1 guarantees that the results with VE is the same as the results for AMI analysis.



5. Double-click **VerifEye contour plot** to open the report showing the VerifEye contour plot of the of the output signal.



Double-click **VerifEye bathtub plot** to open the report showing the VE data as a bathtub plot.



Transmit Jitter Demonstration

Transmit jitter is variation in the timing of edges. Both QuickEye and VerifEye can include jitter distributions when calculating the eye measurements and Bit Error Rate (BER). Several common jitter types—uniform, periodic, random or Gaussian, and duty cycle distortion—can be added in any combination including multiple distributions of the same type. A custom jitter distribution can be read from a file provided by the user. See "Transmit Jitter in QuickEye and VerifEye " on page 25-304 for further information. Following are the examples for this Transmit Jitter Demonstration.

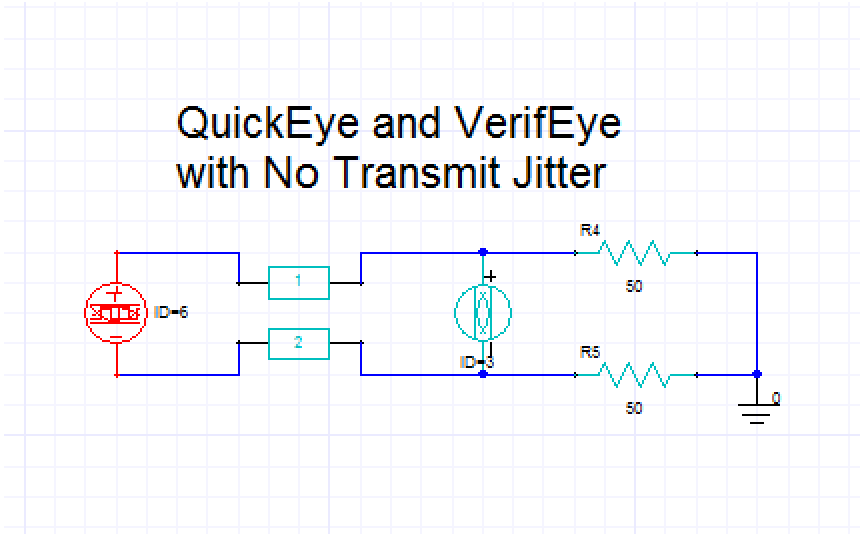
Channel Example with No Transmit Jitter

The schematic for the circuit is available in the Examples directory.

Open the project from its file in the Examples directory.

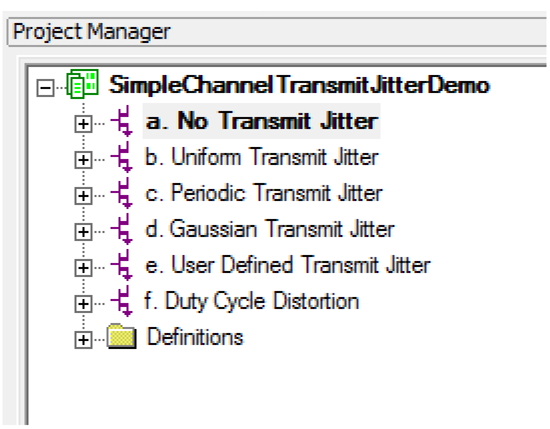
1. From the **File** menu, select **Open Examples** to open an explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **SimpleChannelTransmitJitterDemo.aedt**, or select **SimpleChannelTransmitJitterDemo.aedt** and click **Open**.

The channel schematic appears in the design window:



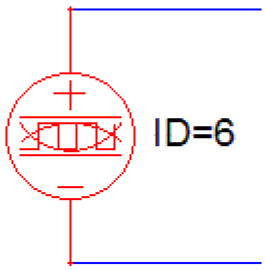
3. The Examples directory is write-protected. Use **File>Save As** to save the example project to a directory where you have write permission. A **Handle Project Directory Files** window opens showing you the supporting model files that is copied along with the project. Click **OK**.

The project contains six designs:

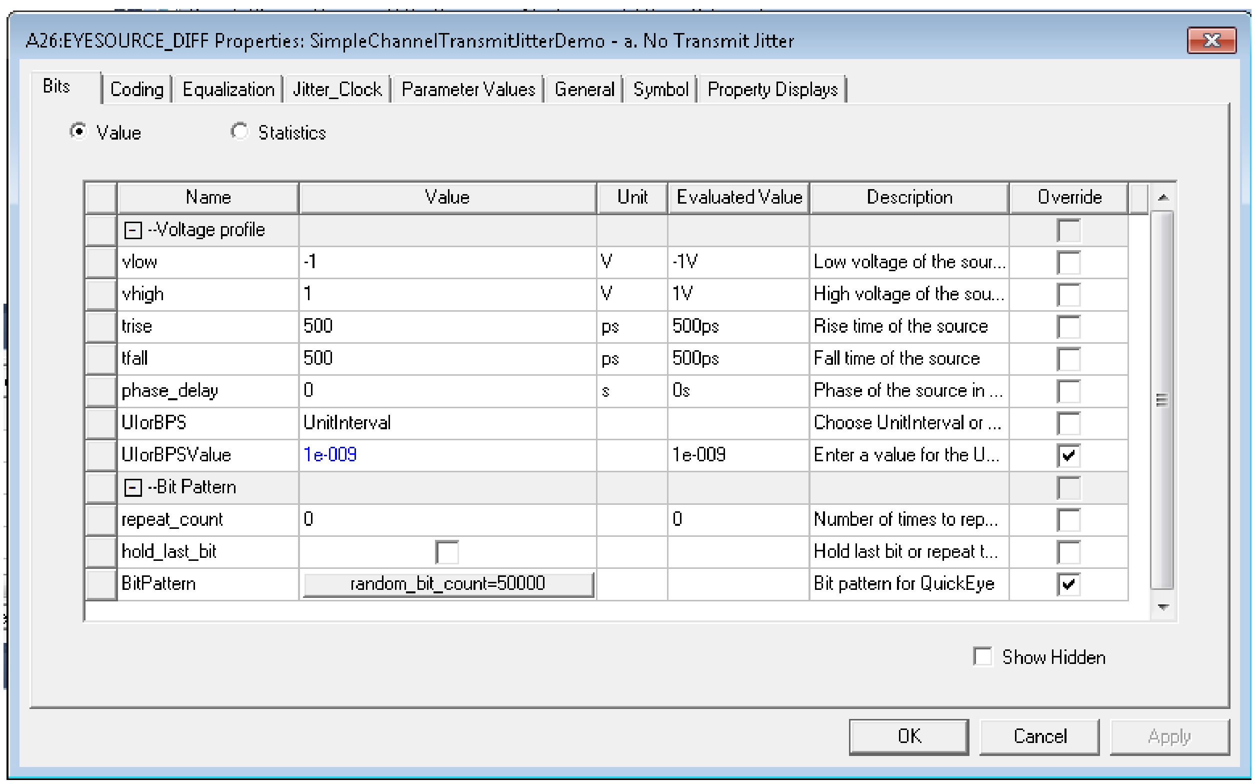


Initially, the **No Transmit Jitter** design is displayed.

Each of these designs uses the same schematic elements, with the same analysis setups and reports. The differences are in the Eye Source:

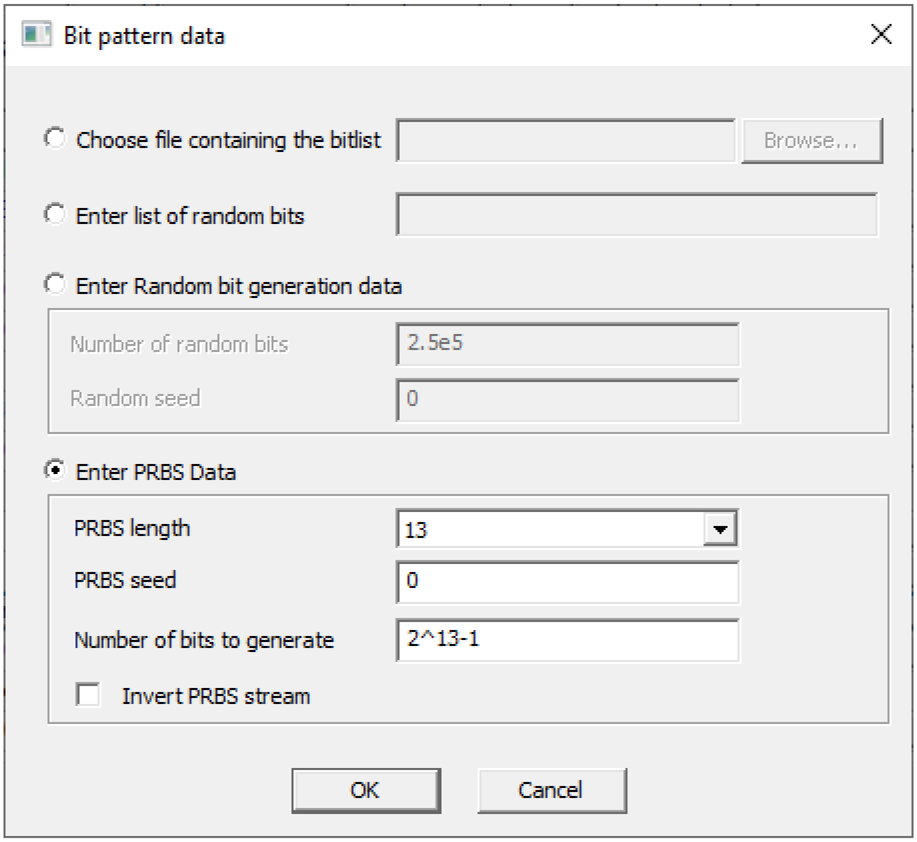


4. Click the Eye Source to open the **Properties** window, **Bits** tab:



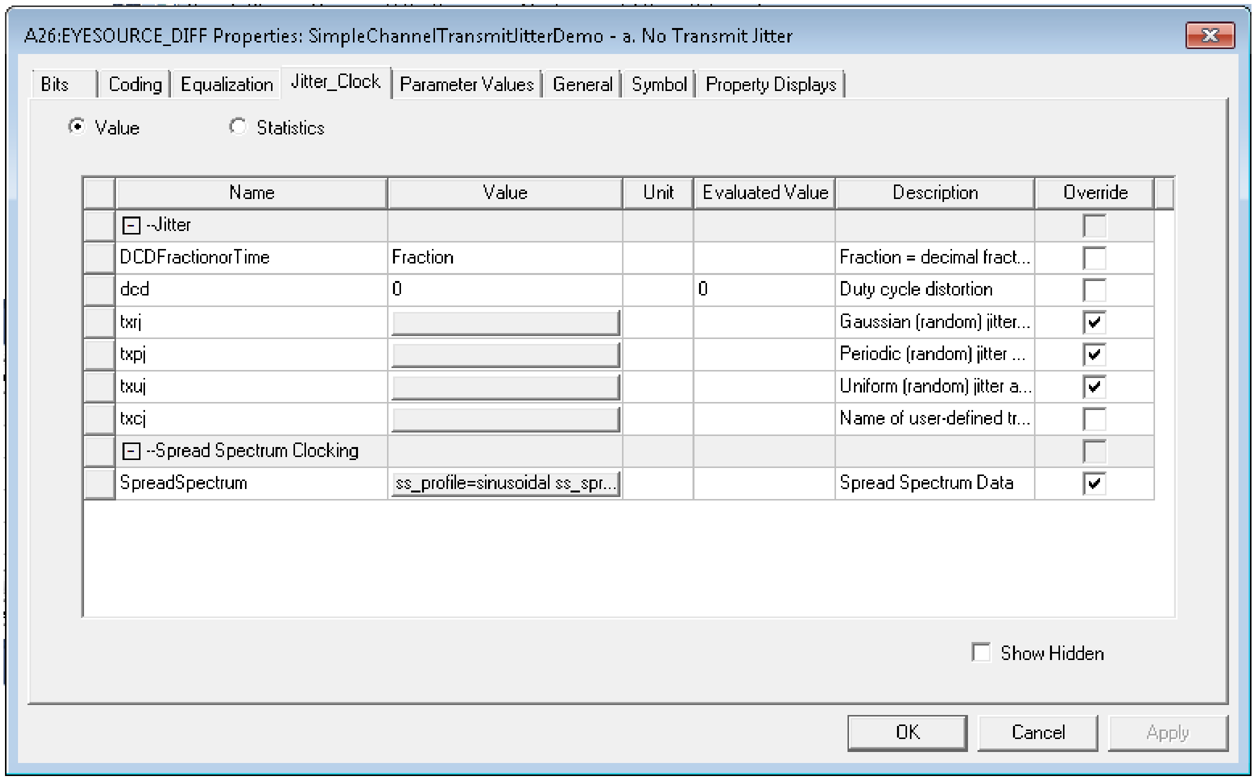
All the designs use the same basic transmitter parameters. The voltage swing is from -1V (low) to +1V (high). The Unit Interval (UI) is 1ns.

5. Click the Bit Pattern group box to open the **Bit pattern data** window:



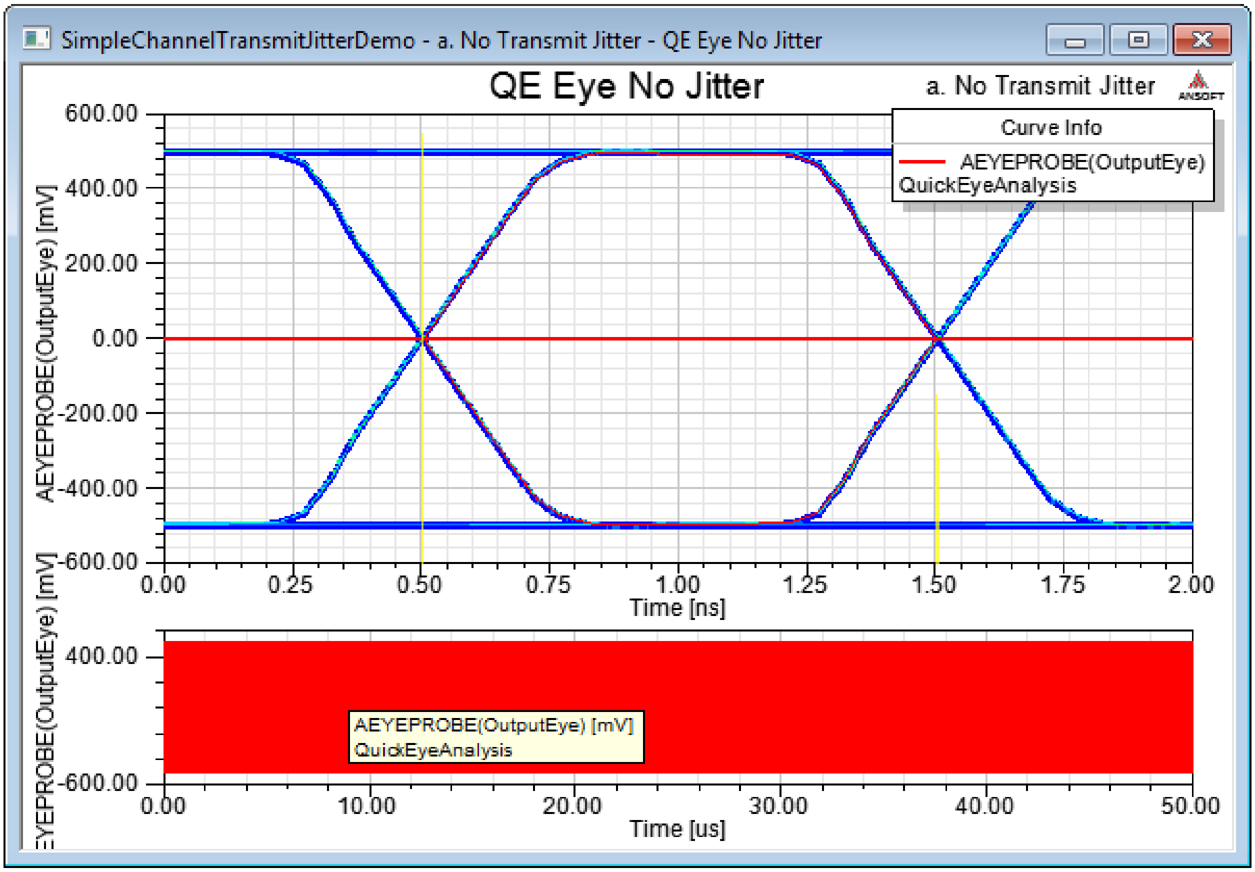
All the designs use the same random bit pattern for the Quick Eye analysis.

Click the **Jitter_Clock** tab to verify that no transmit jitter has been specified.

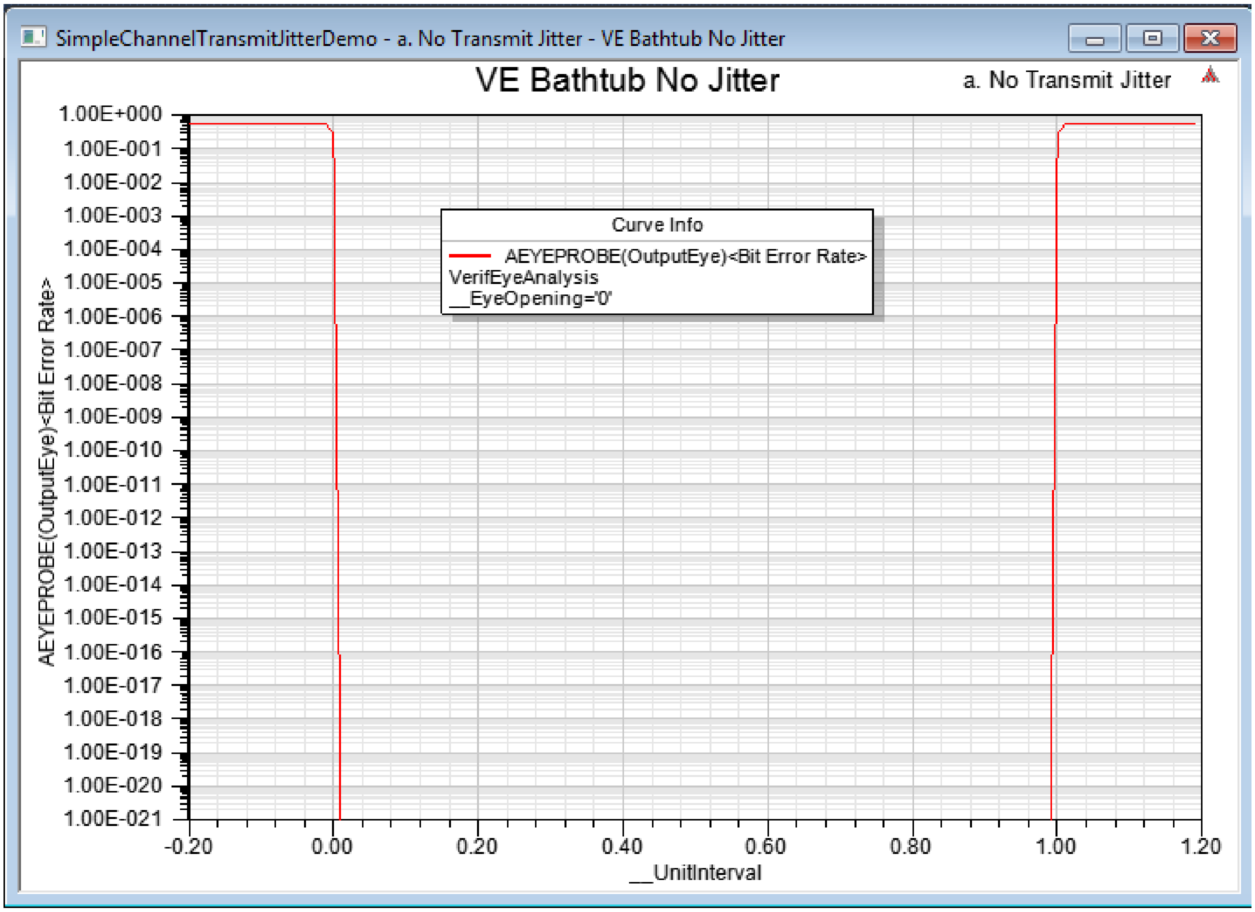


6. Click the Analysis icon in the Project window and select Analyze to run the QuickEye and VerifEye analyses.
7. When the analyses have run to completion, expand the Results icon and open the two reports.

The QuickEye Eye diagram illustrates the channel response with no transmit jitter:



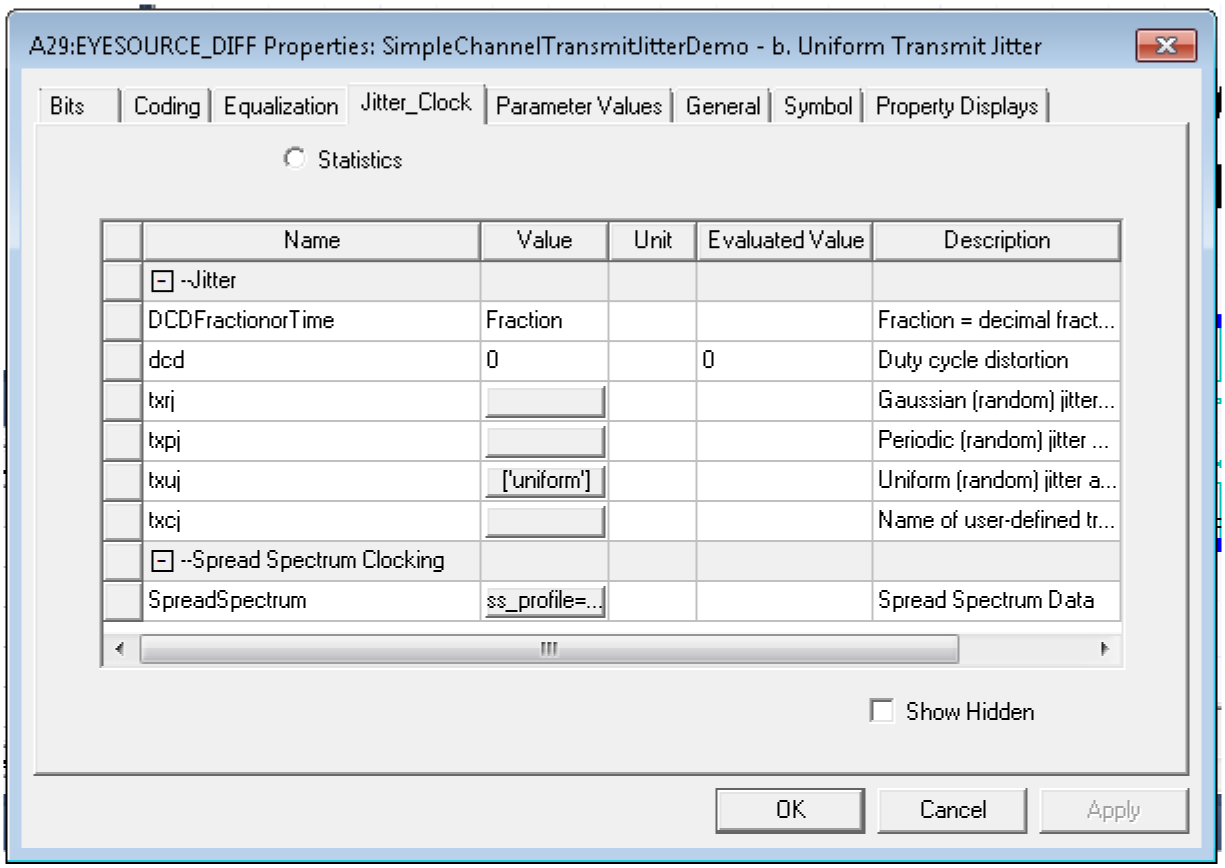
The VerifEye Bathtub diagram illustrates the channel Bit Error Rate (BER) with no transmit jitter:



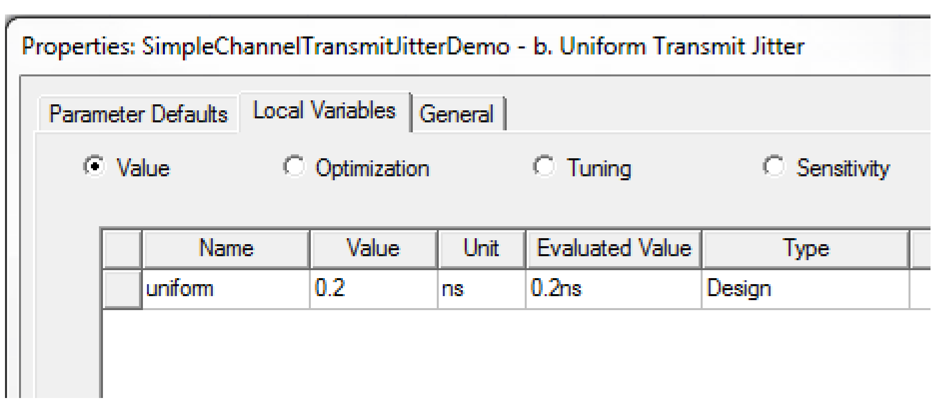
Channel Example with Uniform Transmit Jitter

Next, the effect of jitter, starting with uniform jitter.

1. Click the design **Uniform Transmit Jitter** to open the schematic.
2. Click the Eye Source to open the Properties. Click the **Jitter_Clock** tab:



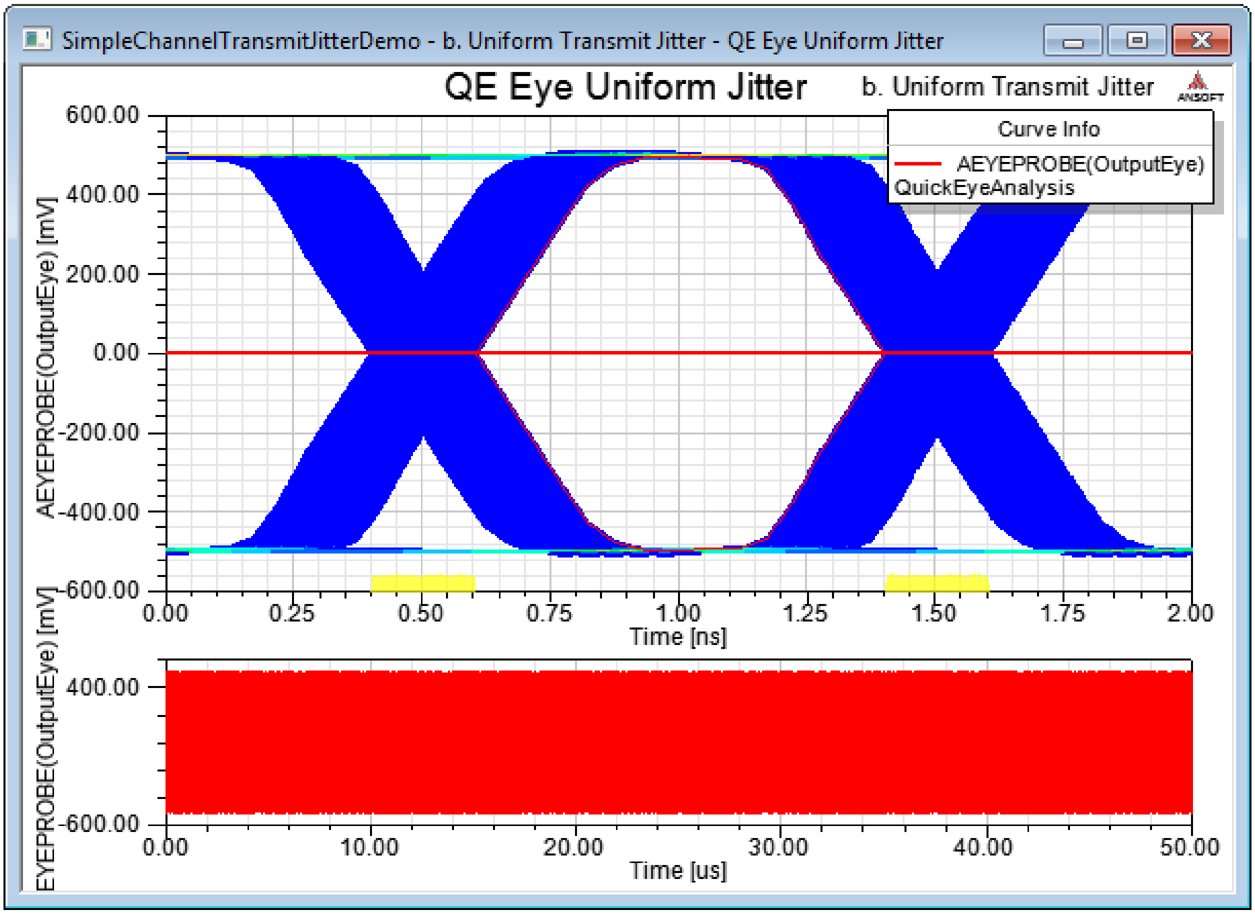
3. The **txuj** parameter sets the bounds of the uniform jitter distribution. A local variable, 'uniform,' has been set up to allow us to change or sweep the parameter:



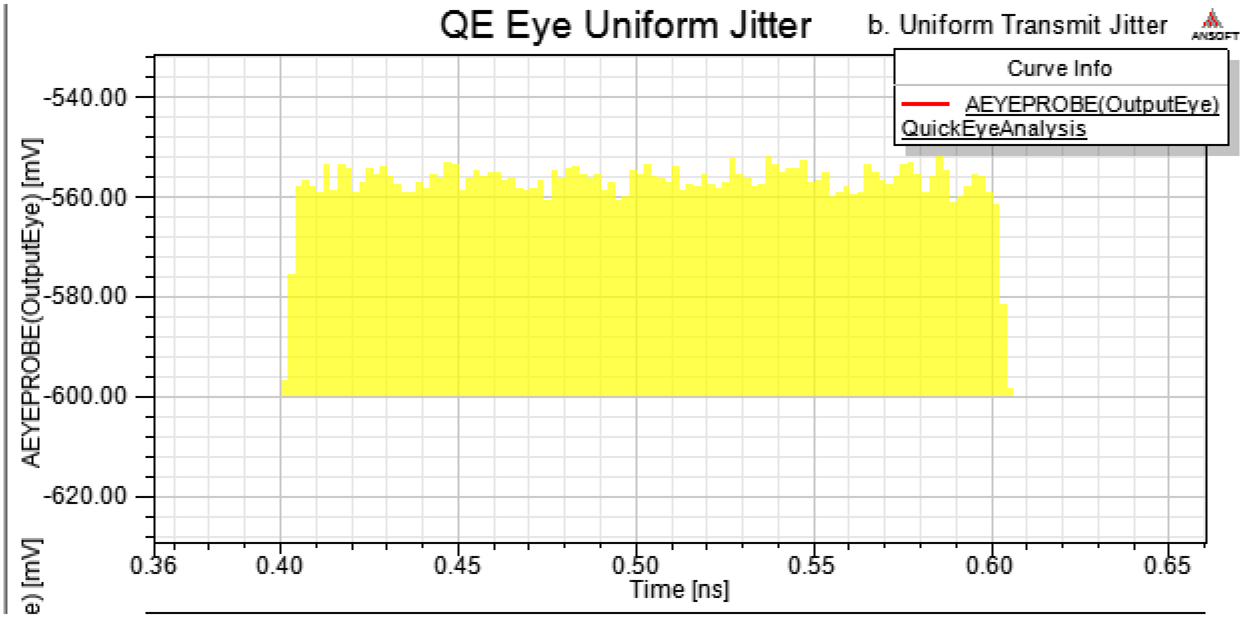
Quick Eye analysis uses the nominal value of the variable, 0.2ns. The VerifEye analysis has been set up with a sweep of the variable through three values, 0.1ns, 0.2 ns, and 0.3ns.

4. Click the Analysis icon in the Project window and select Analyze to run the QuickEye and VerifEye analyses.
5. When the analyses have run to completion, expand the Results icon and open the two reports.

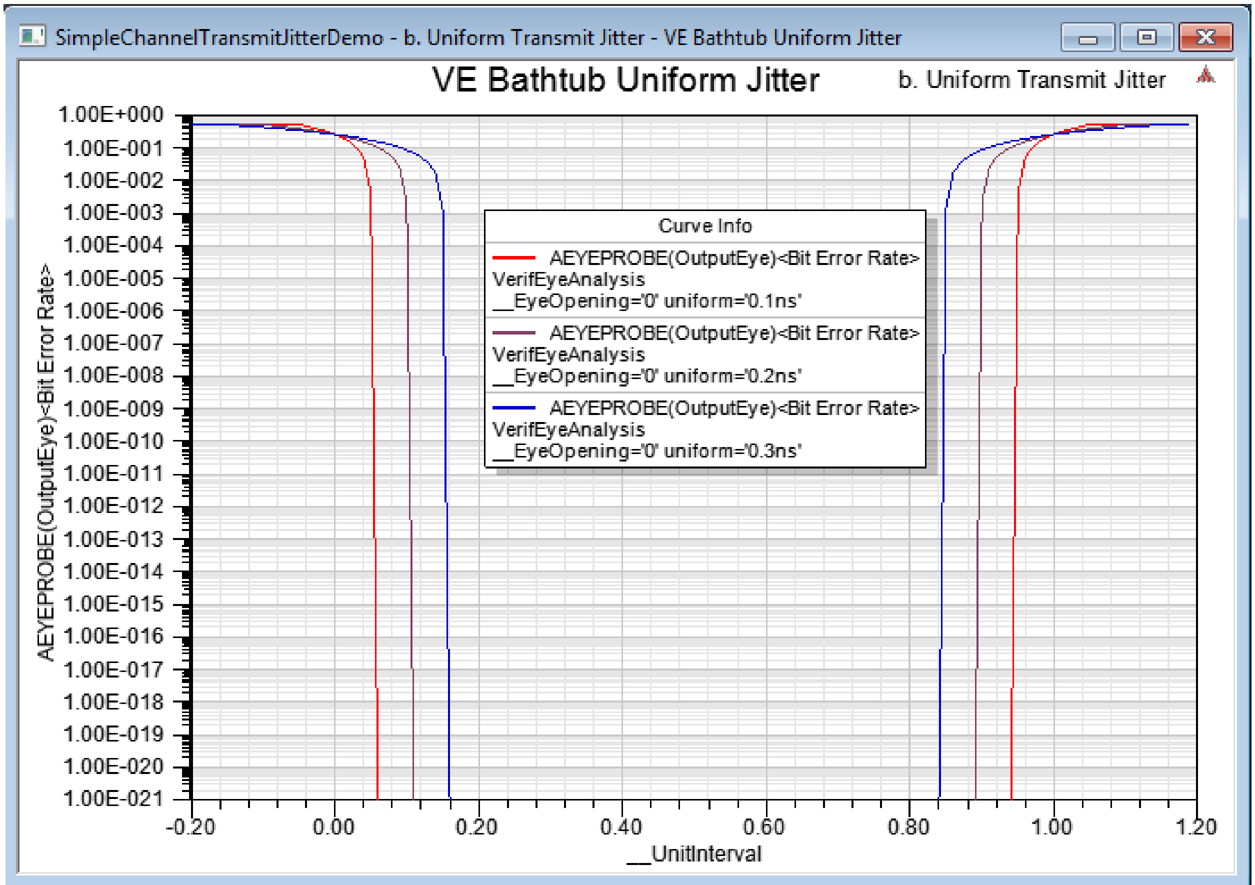
The QuickEye Eye diagram illustrates the channel response with uniform jitter:



The yellow bars under the eye diagram show the histogram of the jitter distribution. Here is a zoomed-in view:



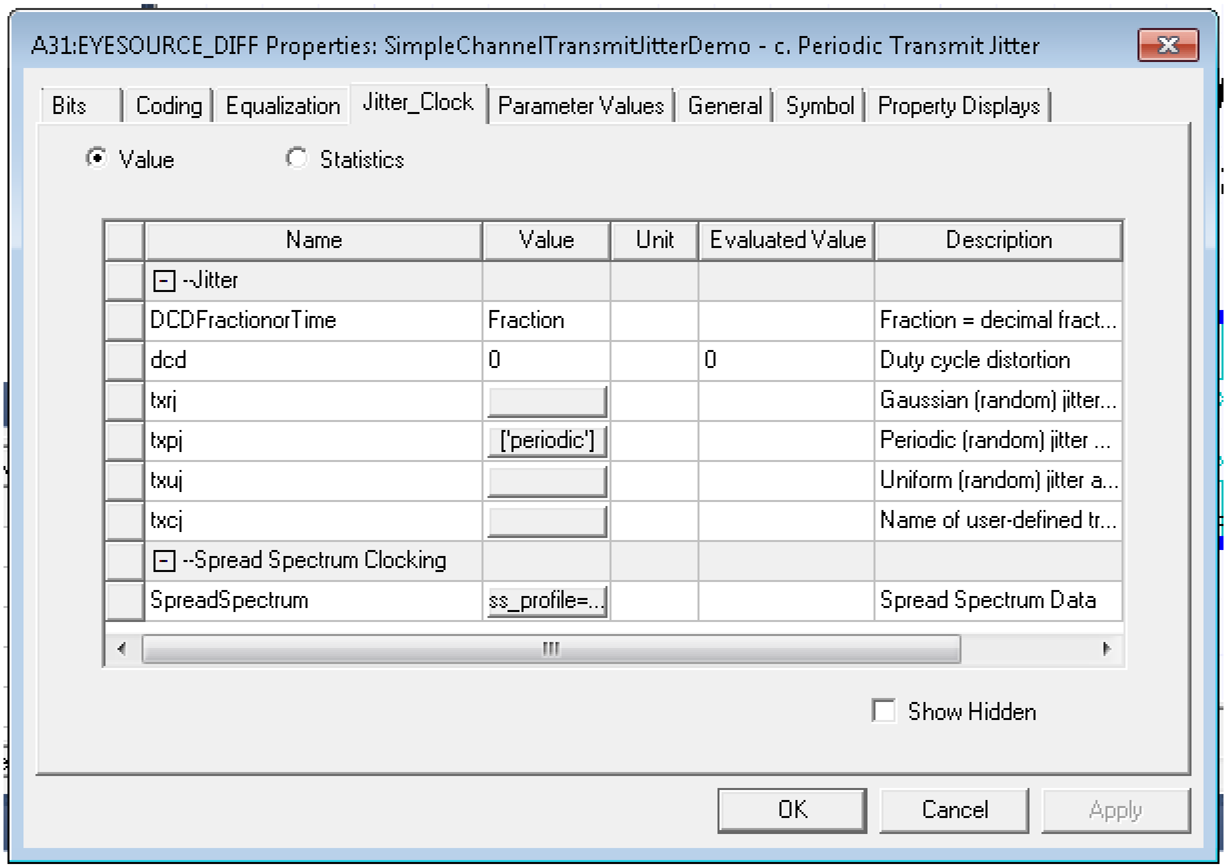
The VerifEye Bathtub diagram illustrates the channel Bit Error Rate (BER) with three levels of uniform jitter:



Channel Example with Periodic Transmit Jitter

Next, the effect of periodic transmit jitter.

1. Click the design **Periodic Transmit Jitter** to open the schematic.
2. Click the Eye Source to open the Properties. Click the **Jitter_Clock** tab:

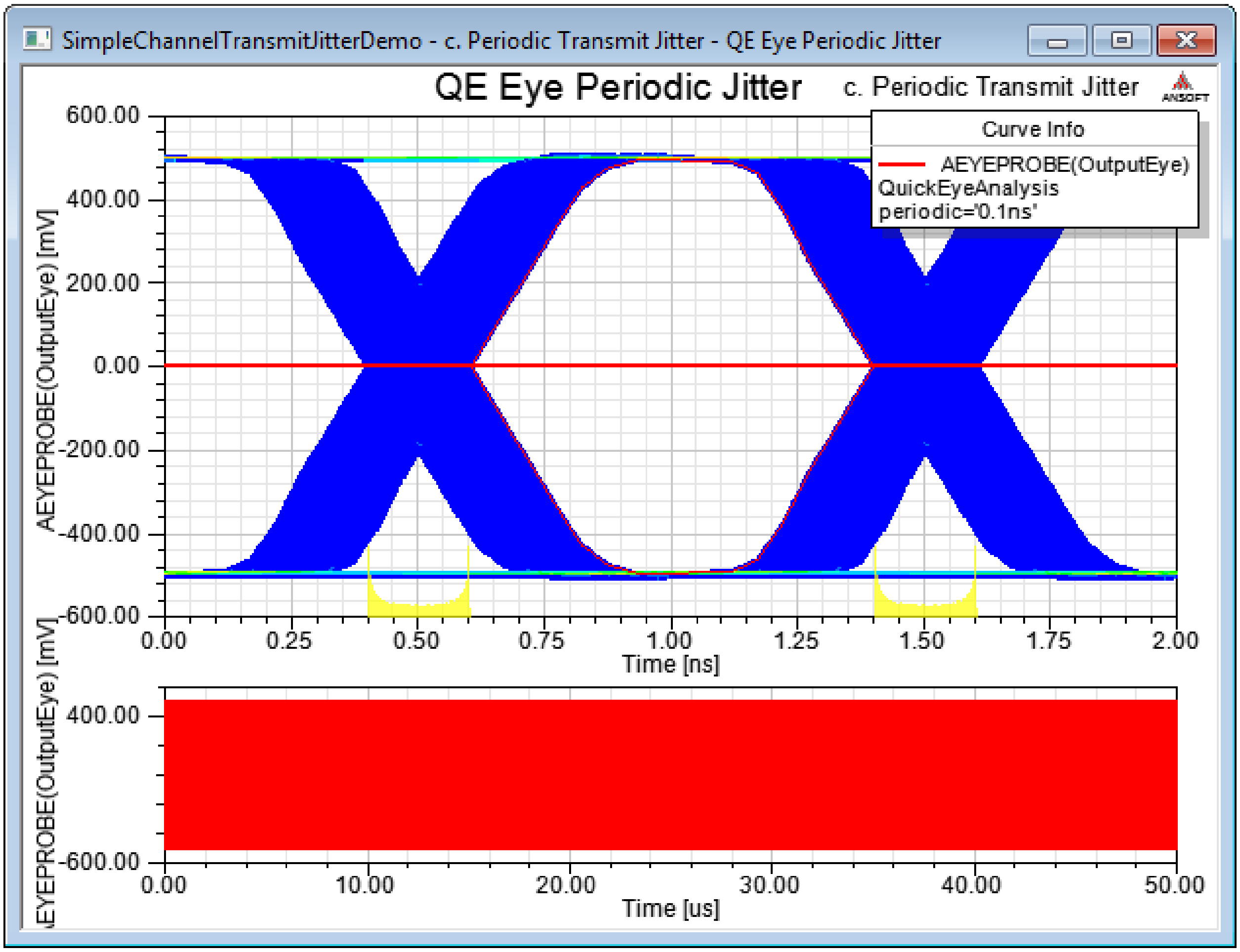


3. The **txpj** parameter sets the bounds of the periodic jitter distribution. A local variable, ‘periodic,’ has been set up to allow us to change or sweep the parameter.

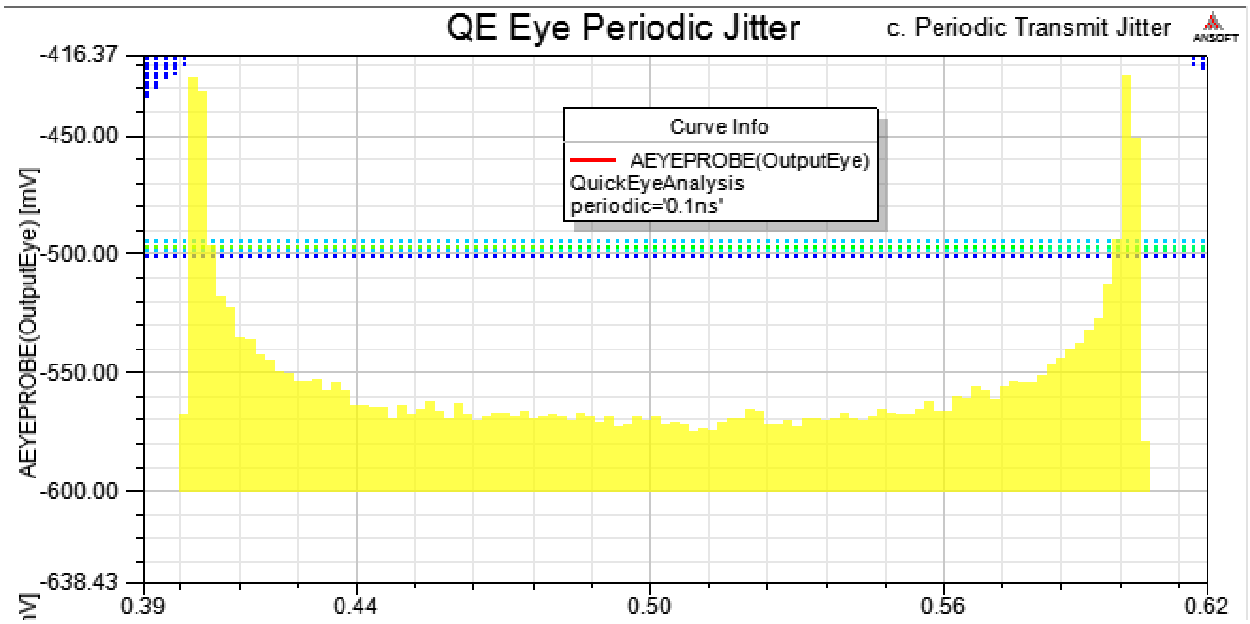
Quick Eye analysis uses the nominal value of the variable, 0.1ns. The VerifEye analysis has been set up with a sweep of the variable through three values, 0.05ns, 0.1ns, and 0.15ns.

4. Click the Analysis icon in the Project window and select Analyze to run the QuickEye and VerifEye analyses.
5. When the analyses have run to completion, expand the Results icon and open the two reports.

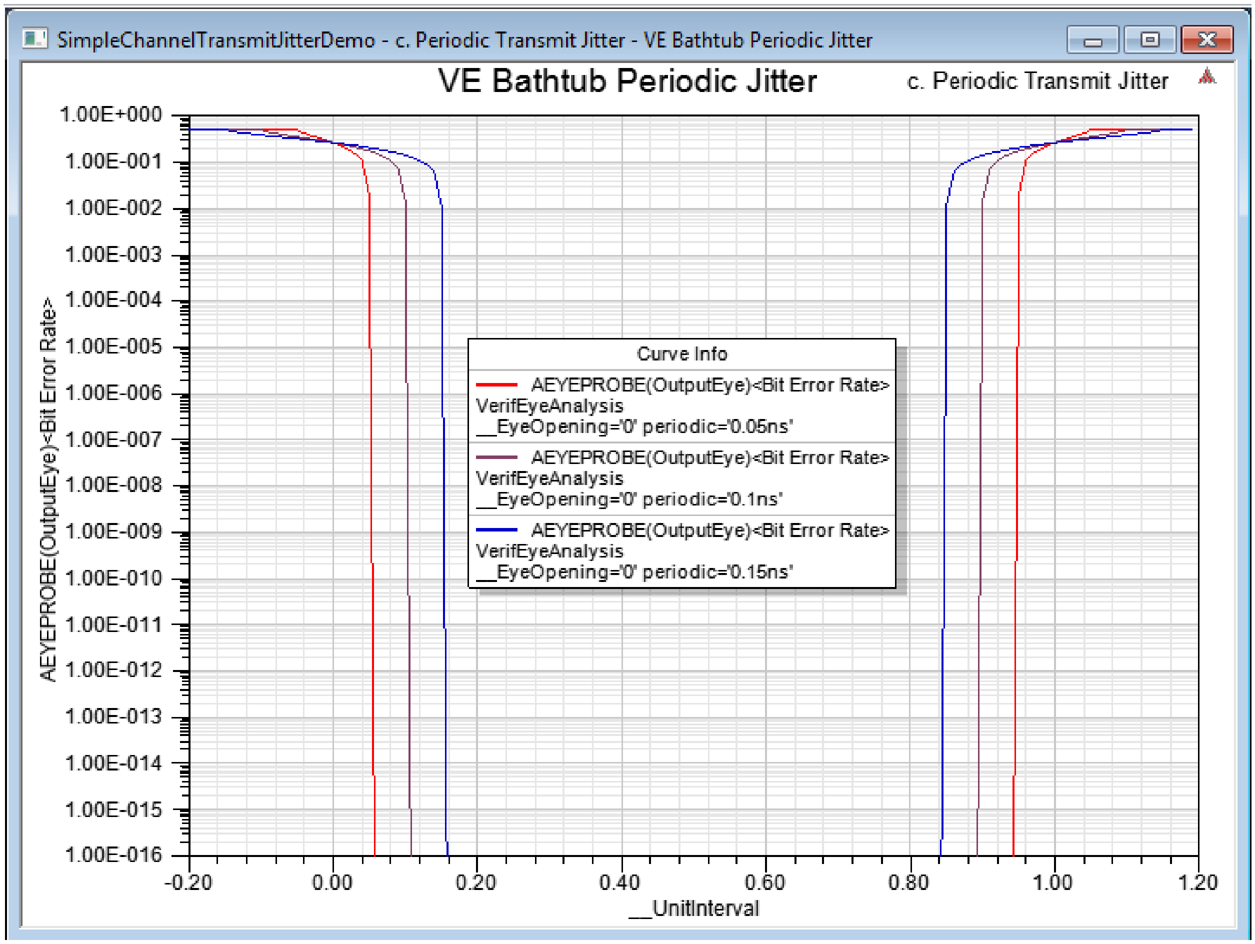
The QuickEye Eye diagram illustrates the channel response with periodic jitter:



The yellow bars under the eye diagram show the histogram of the jitter distribution. Here is a zoomed-in view:



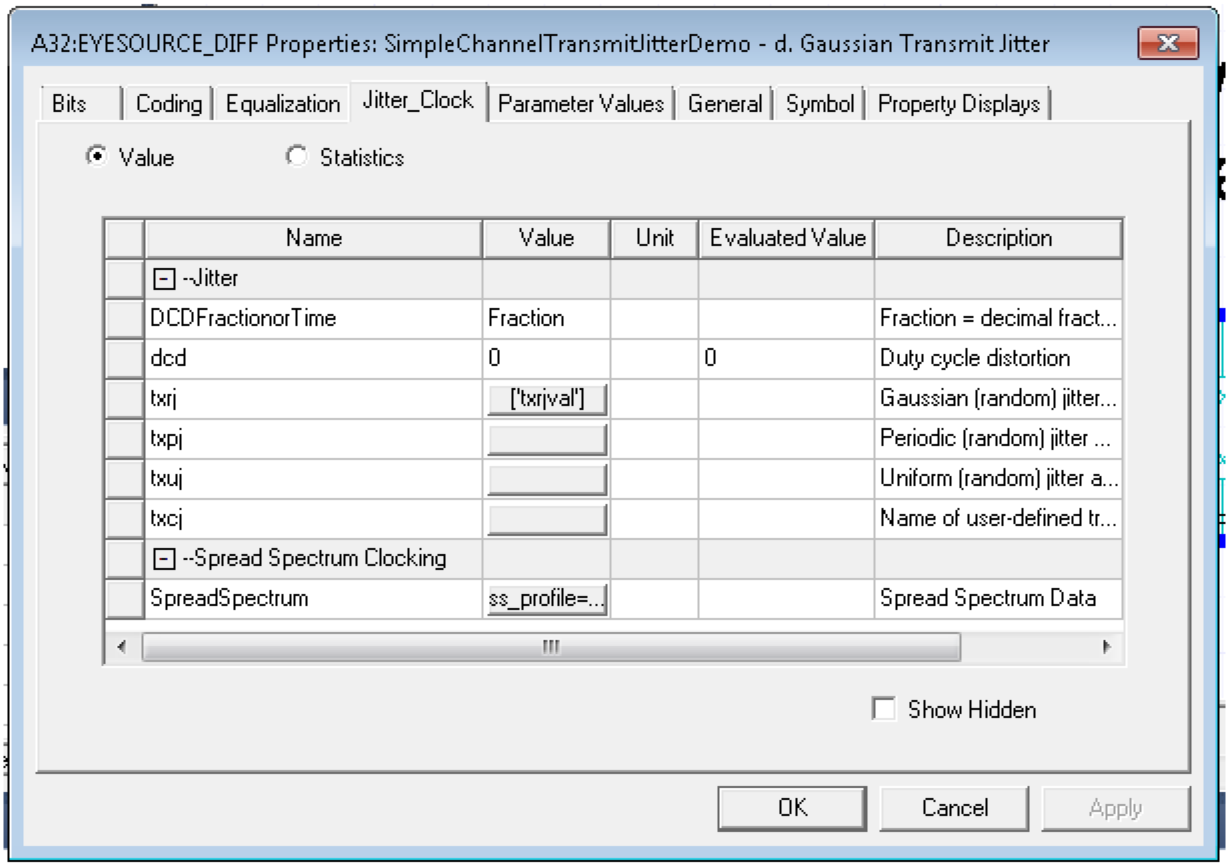
The VerifEye Bathtub diagram illustrates the channel Bit Error Rate (BER) with three levels of periodic jitter:



Channel Example with Gaussian Random Transmit Jitter

Next, the effect of Gaussian random transmit jitter.

1. Click the design **Gaussian Transmit Jitter** to open the schematic.
2. Click the Eye Source to open the Properties. Click the **Jitter_Clock** tab:

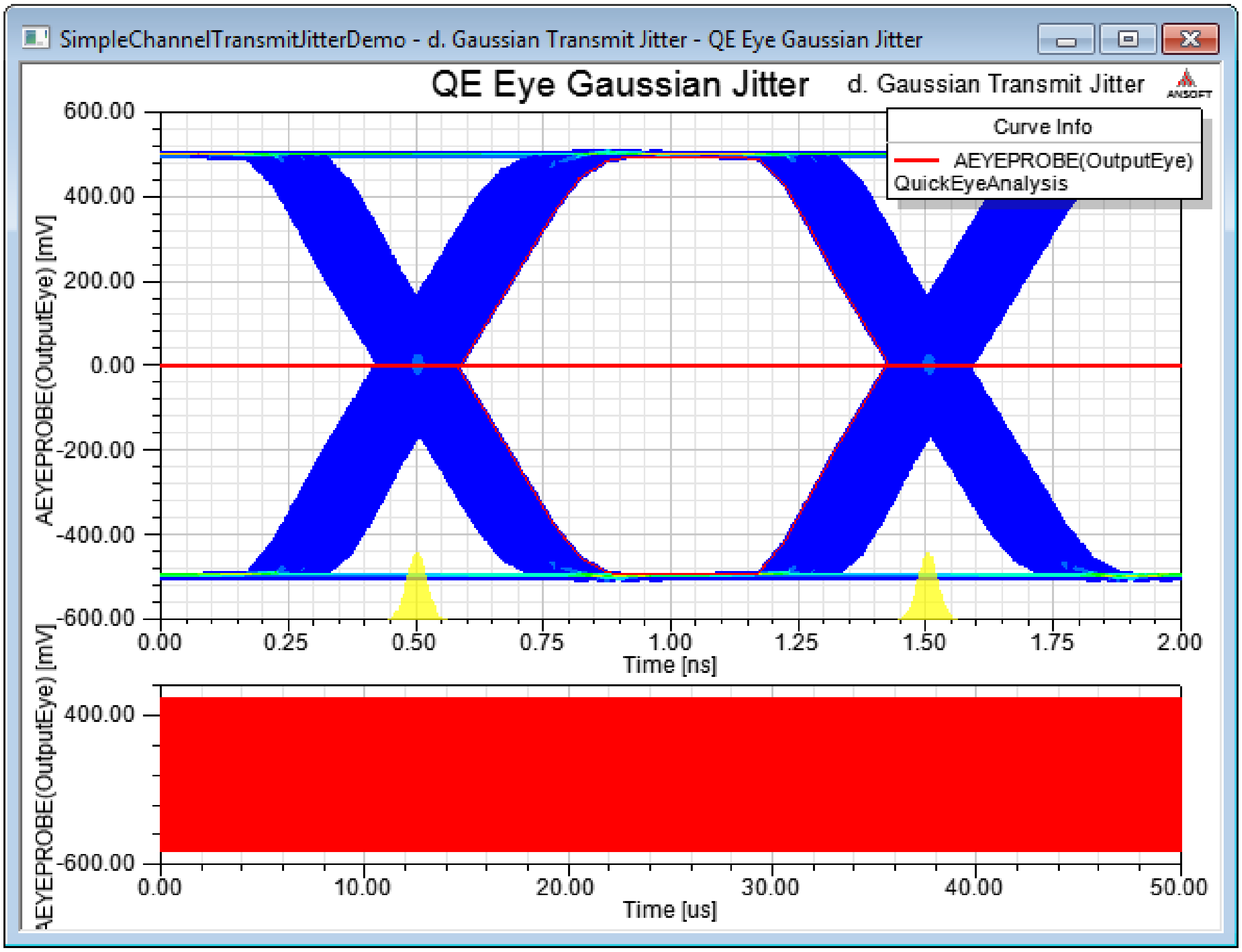


3. The **txrj** parameter sets the standard deviation of the Gaussian jitter distribution. A local variable, ‘txrjval,’ has been set up to allow us to change or sweep the parameter.

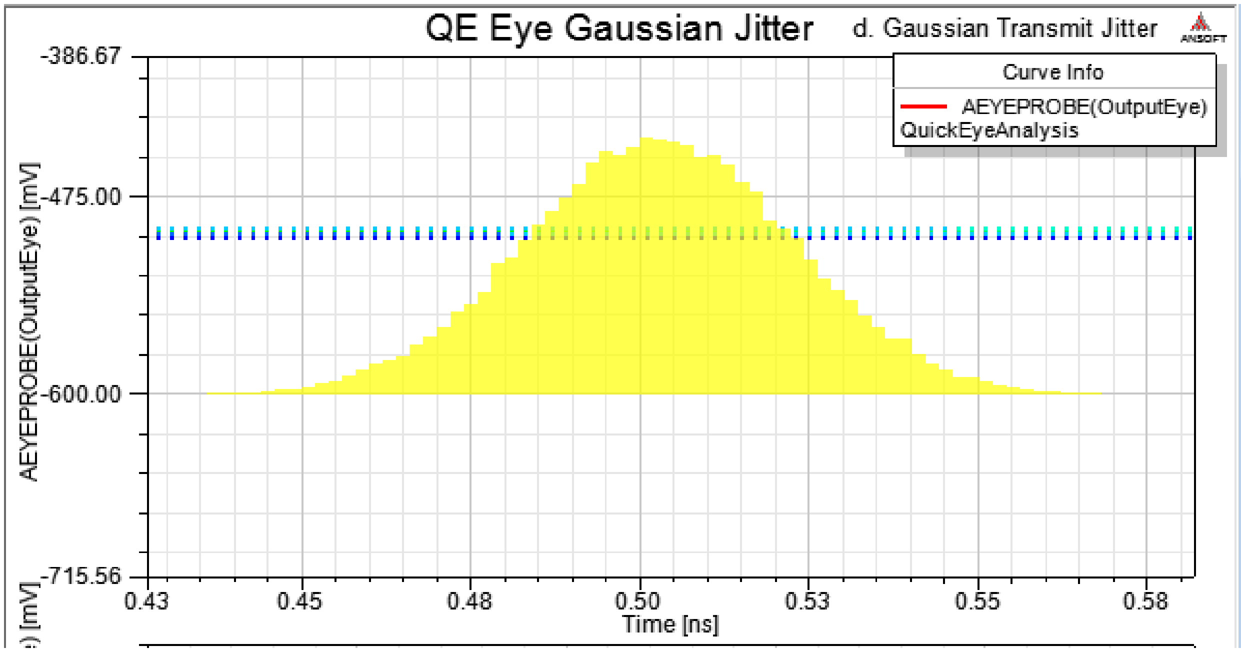
Quick Eye analysis uses the nominal value of the variable, 20ps. The VerifEye analysis has been set up with a sweep of the variable through three values, 10ps, 20ps, and 30ps.

4. Click the Analysis icon in the Project window and select Analyze to run the QuickEye and VerifEye analyses.
5. When the analyses have run to completion, expand the Results icon and open the two reports.

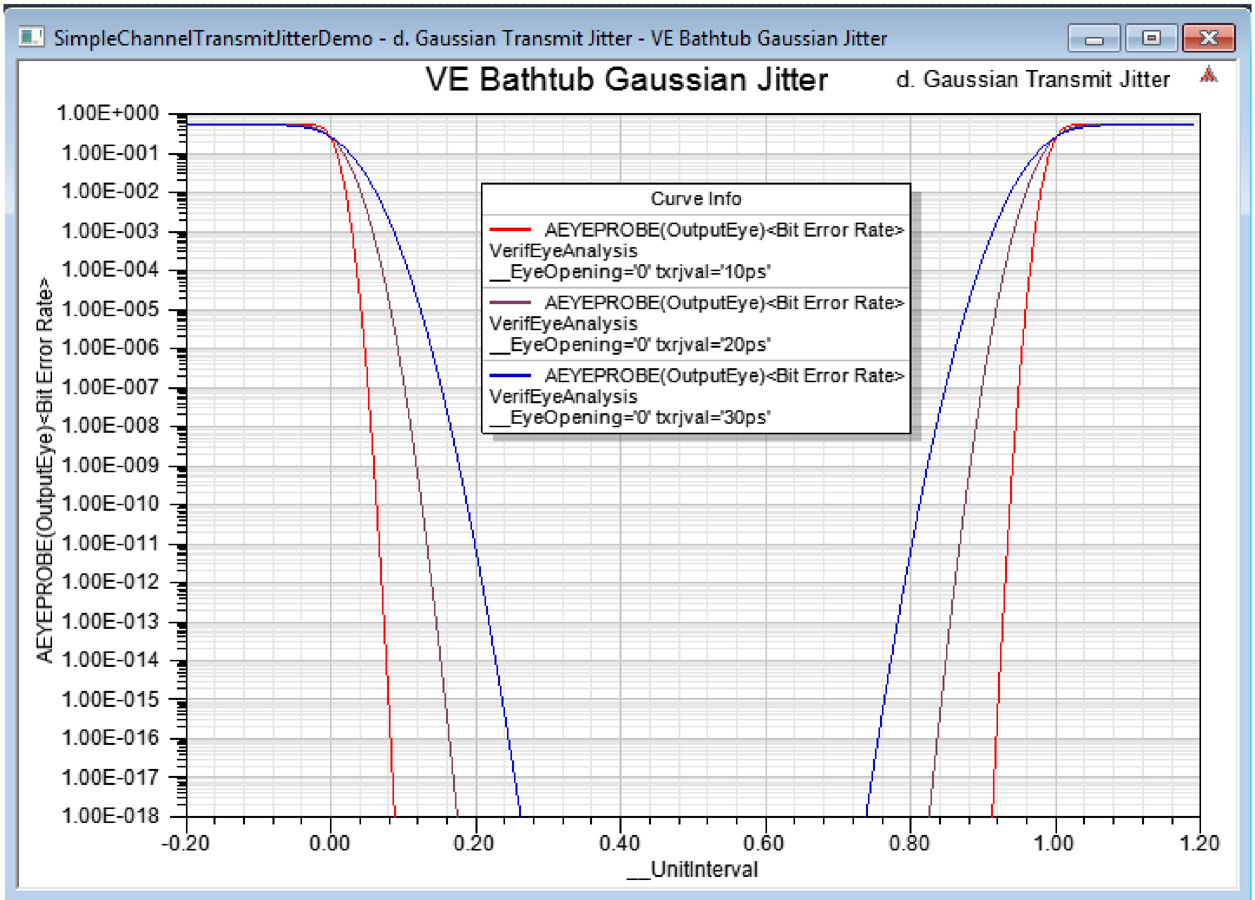
The QuickEye Eye diagram illustrates the channel response with Gaussian random jitter:



The yellow bars under the eye diagram show the QuickEye Eye diagram. Here is a zoomed-in view:



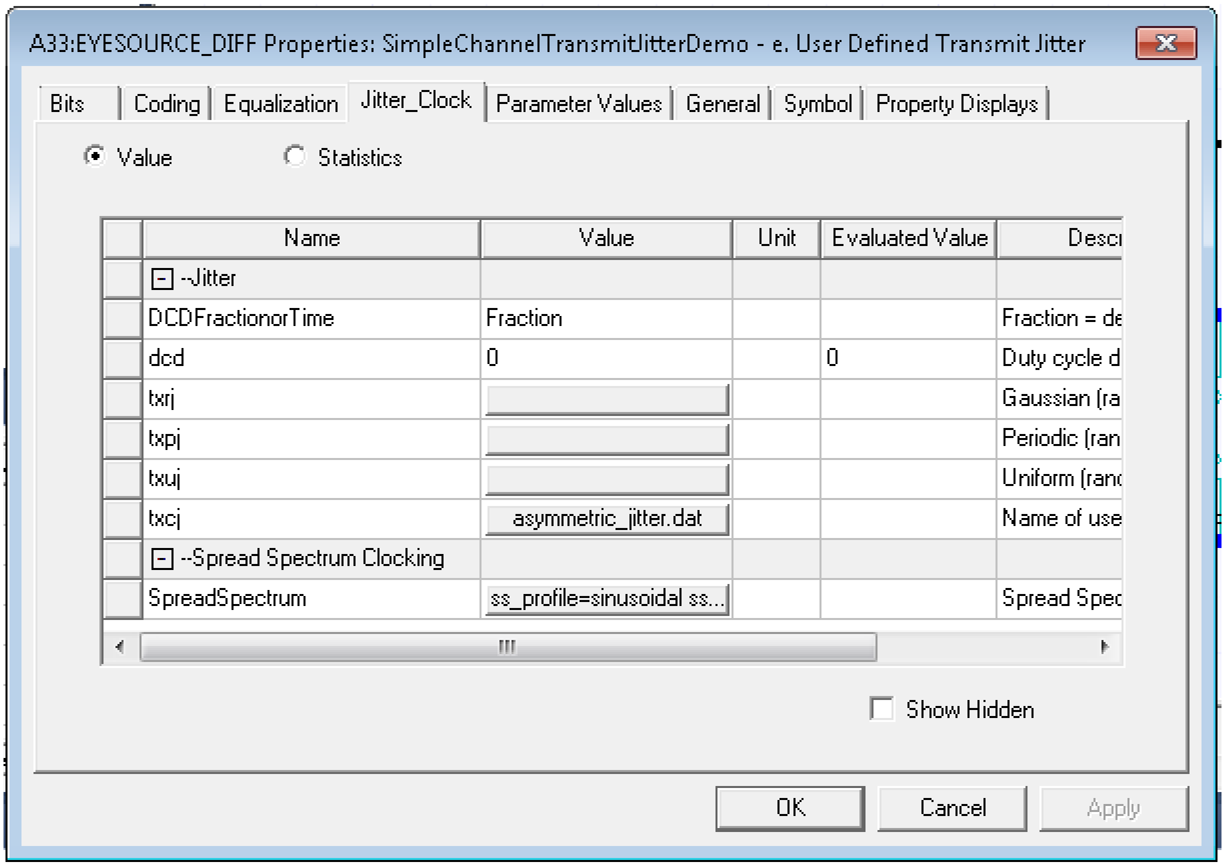
The VerifEye Bathtub diagram illustrates the channel Bit Error Rate (BER) with three levels of Gaussian random jitter:



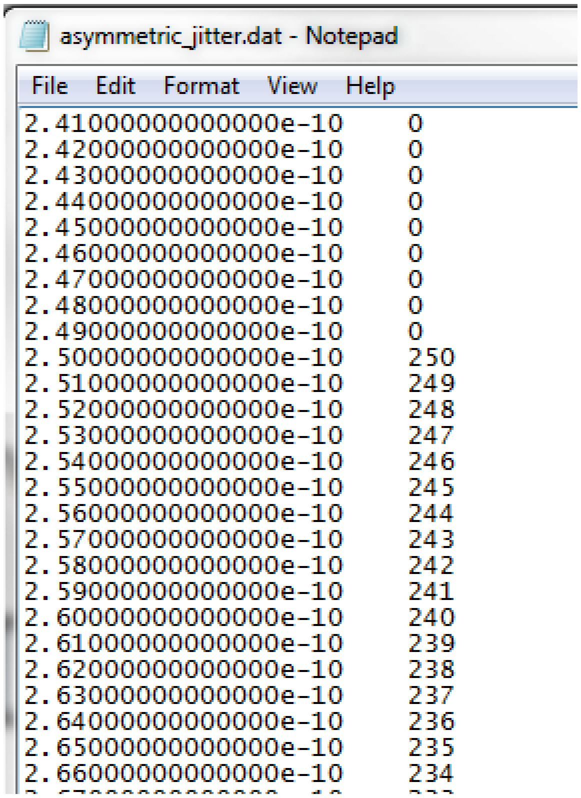
Channel Example with Custom Transmit Jitter

Next, add a custom transmit jitter distribution.

1. Click the design **user-defined Transmit Jitter** to open the schematic.
2. Click the Eye Source to open the Properties. Click the **Jitter_Clock** tab:



3. The **txcj** parameter specifies a file containing the custom jitter distribution. The Examples directory contains several examples of custom transmit jitter files (Examples/Circuit/SignalIntegrity/models/SimpleChannelTransmitJitterDemo/*.dat). Here is a portion of the file asymmetric_jitter.dat:


asymmetric_jitter.dat - Notepad

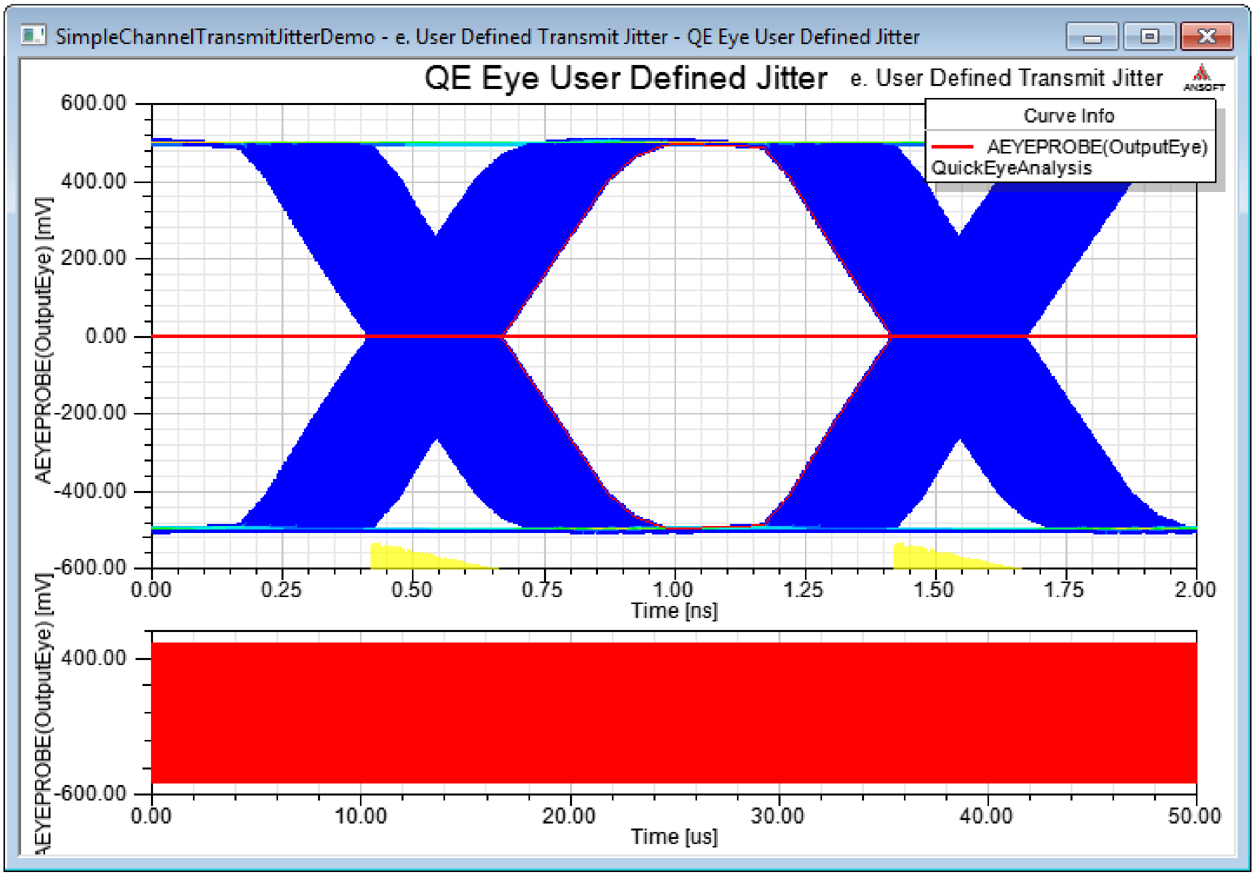
File Edit Format View Help

```
2.41000000000000e-10    0
2.42000000000000e-10    0
2.43000000000000e-10    0
2.44000000000000e-10    0
2.45000000000000e-10    0
2.46000000000000e-10    0
2.47000000000000e-10    0
2.48000000000000e-10    0
2.49000000000000e-10    0
2.50000000000000e-10    250
2.51000000000000e-10    249
2.52000000000000e-10    248
2.53000000000000e-10    247
2.54000000000000e-10    246
2.55000000000000e-10    245
2.56000000000000e-10    244
2.57000000000000e-10    243
2.58000000000000e-10    242
2.59000000000000e-10    241
2.60000000000000e-10    240
2.61000000000000e-10    239
2.62000000000000e-10    238
2.63000000000000e-10    237
2.64000000000000e-10    236
2.65000000000000e-10    235
2.66000000000000e-10    234
```

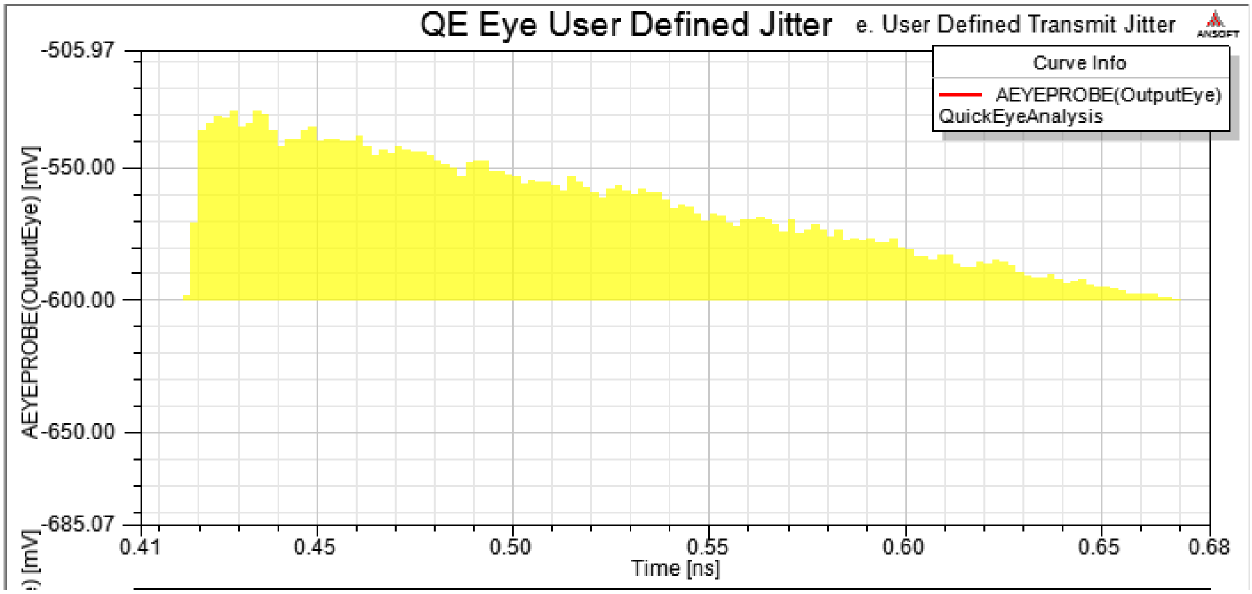
Each line specifies a time and the relative magnitude of the jitter probability distribution function at that time. Nexxim centers the distribution's mean value at the edges of the unit interval.

4. Click the Analysis icon in the Project window and select Analyze to run the QuickEye and VerifEye analyses.
5. When the analyses have run to completion, expand the Results icon and open the two reports.

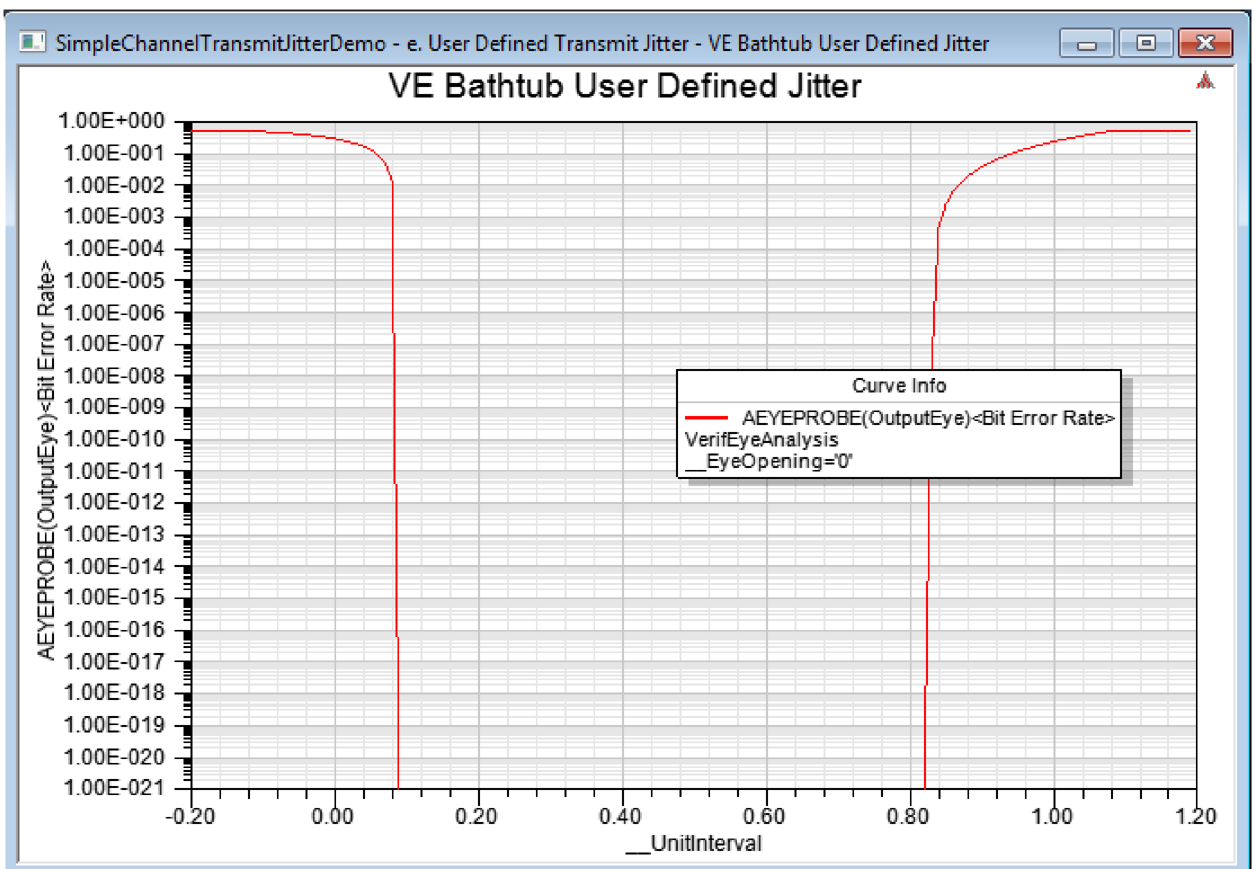
The QuickEye Eye diagram illustrates the channel response with custom jitter:



The yellow bars under the eye diagram show the histogram of the jitter distribution. Here is a zoomed-in view:



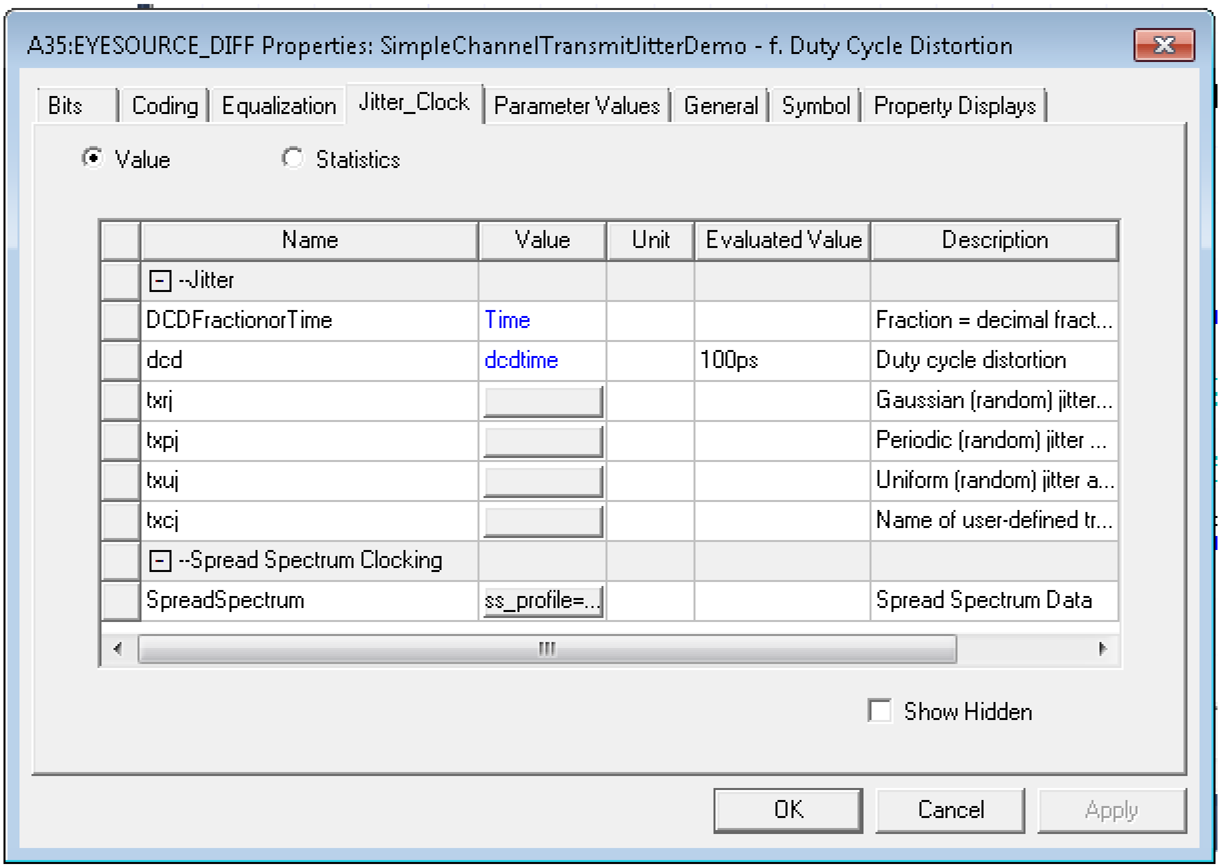
The VerifEye Bathtub diagram illustrates the channel Bit Error Rate (BER) with custom jitter:



Channel Example with Duty Cycle Distortion

Next, the effect of duty cycle distortion (asymmetry in the bit timings). The duty cycle distortion can be specified as a fraction of the unit interval (UI) or as an absolute time.

1. Click the design **Duty Cycle Distortion** to open the schematic.
2. Click the Eye Source to open the Properties. Click the **Jitter_Clock** tab:

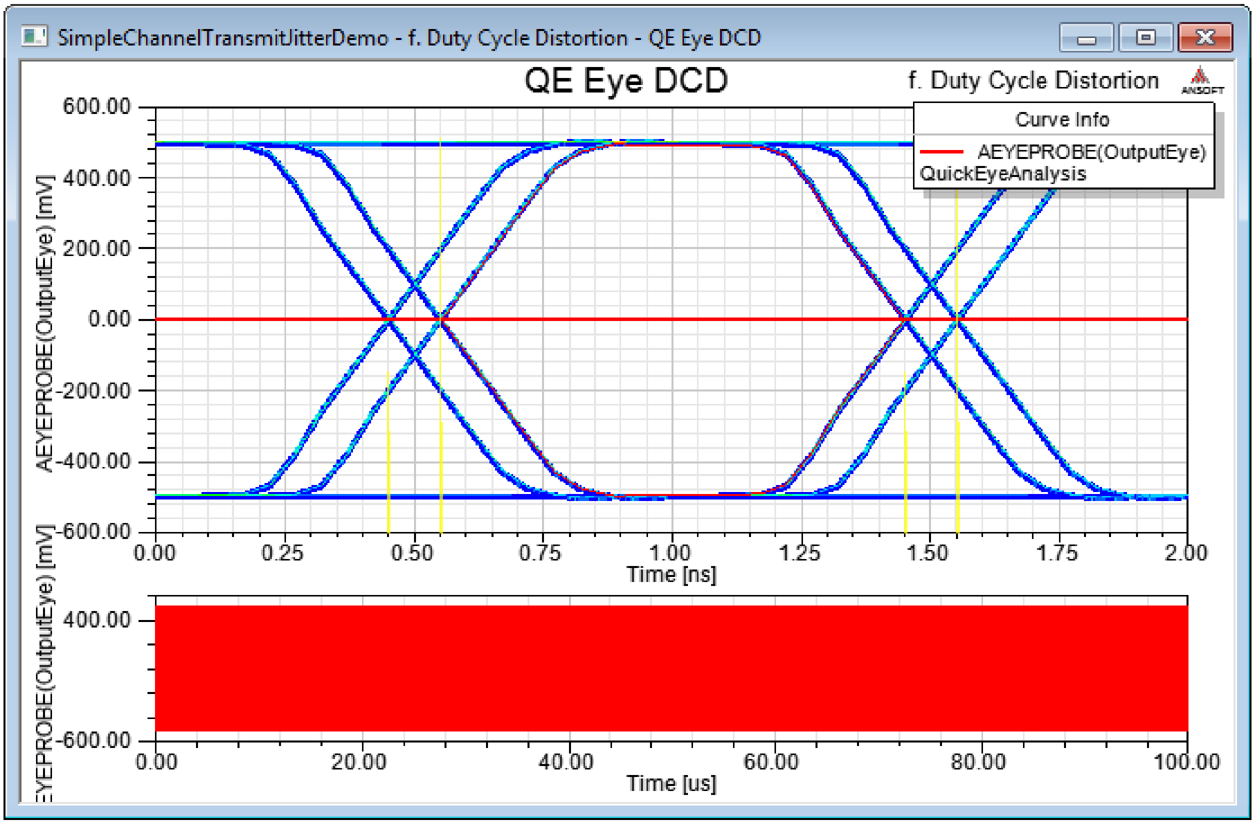


3. The parameter **DCDFractionorTime** selects between the two ways to specify the DCD. The default is **Fraction**. In this mode, the **dcd** parameter sets the bounds of the duty cycle distortion to be a fraction of the UI, 0<=DCD<=1. A local variable, 'dcdval,' has been set up to allow us to change or sweep the DCD Fractional value.

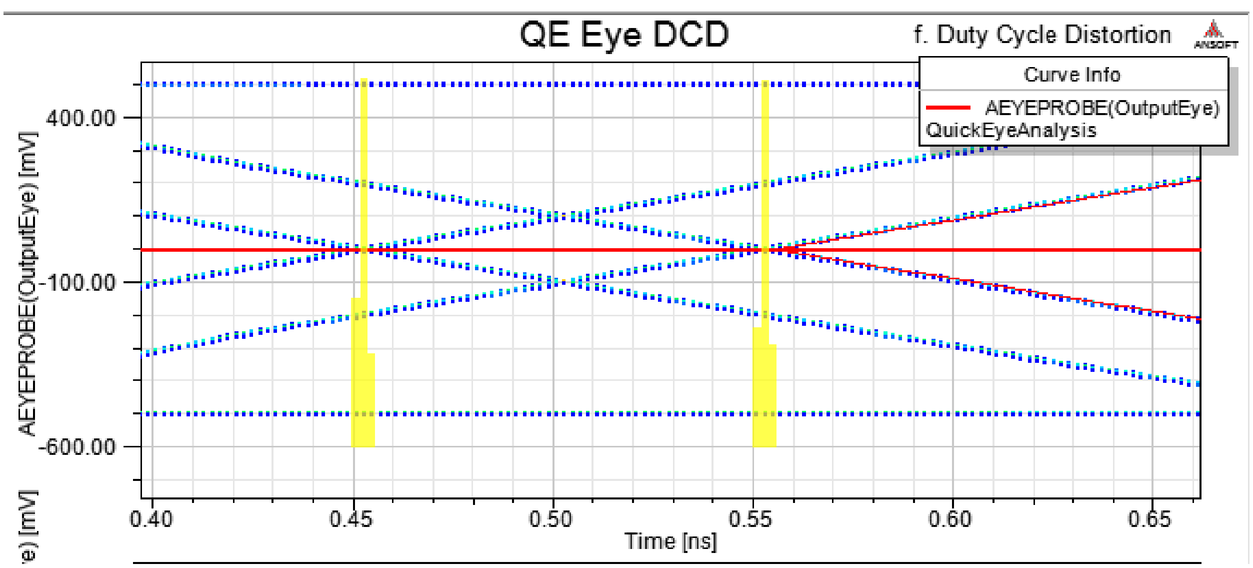
Both QuickEye and VerifEye analyses use the nominal value of the variable, 0.1 or 10% of the UI (1ns).

4. Click the Analysis icon in the Project window and select Analyze to run the QuickEye and VerifEye analyses.
5. When the analyses have run to completion, expand the Results icon and open the two reports.

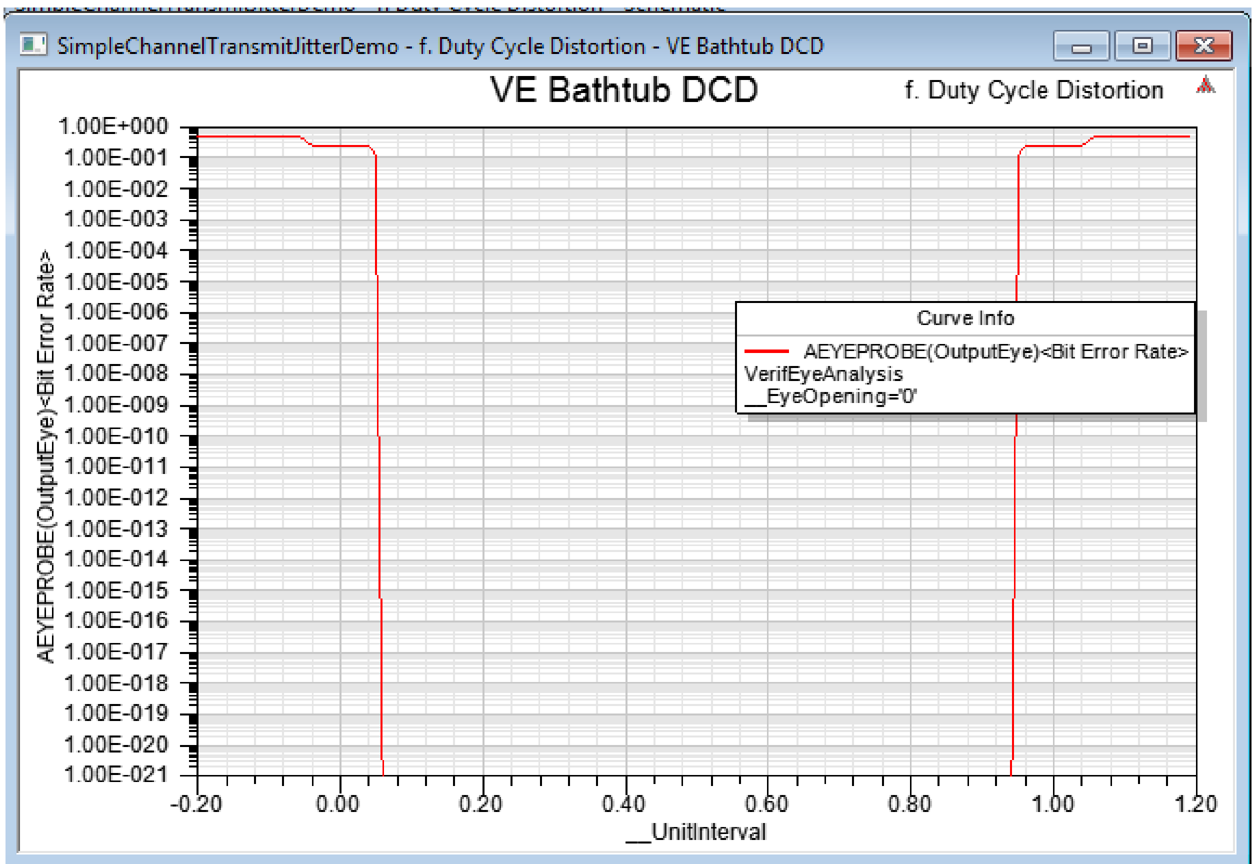
The QuickEye Eye diagram illustrates the channel response with duty cycle distortion:



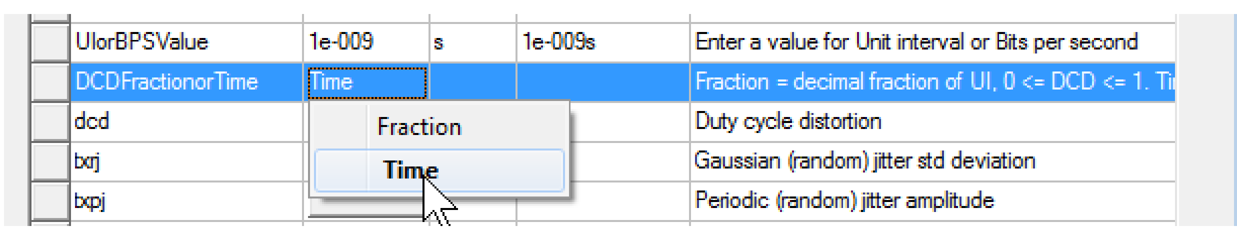
The yellow bars under the eye diagram show the histogram of the DCD distribution. Here is a zoomed-in view:



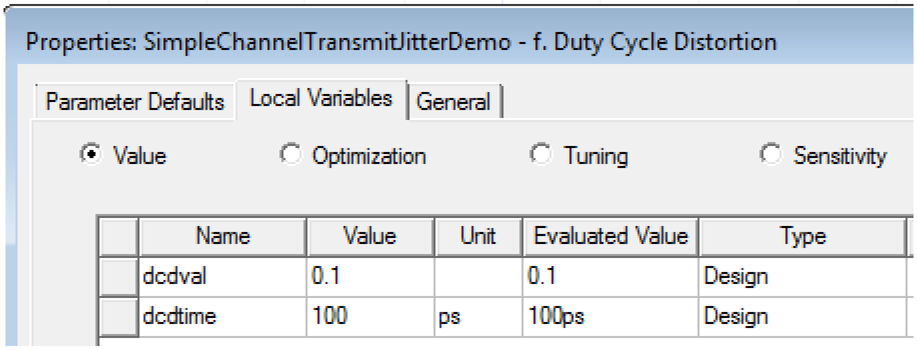
The VerifEye Bathtub diagram illustrates the channel Bit Error Rate (BER) with duty cycle distortion:



Alternatively, you may wish to specify the DCD as an absolute time between 0 and the size of the UI. Click the **DCDFractionorTimeValue** field and select Time on the menu.



For the **dcd** property value, enter **dcdtime**. This local variable has been defined as 100ps, equivalent to the fraction 0.1 when the UI is 1ns.



Here is the **Properties** window with the **dcdtime** variable specified.

Name	Value	Unit	Evaluated Value	Description
resistance	100	ohm	100ohm	Resistance (Ohm)
vlow	-1	V	-1V	Low voltage of the source
vhigh	1	V	1V	High voltage of the source
trise	5e-010	s	5e-010s	Rise time of the source
tfall	5e-010	s	5e-010s	Fall time of the source
phase_delay	0	s	0s	Phase of the source in seconds
UIorBPS	UnitInterval			Choose either Unit Interval or Bits per Second.
UIorBPSValue	1e-009	s	1e-009s	Enter a value for Unit interval or Bits per second
DCDFractionorTime	Time			Fraction = decimal fraction of UI, 0 <= DCD <= 1. Ti
dcd	dcdtime		100ps	Duty cycle distortion
txrj				Gaussian (random) jitter std deviation

Run the analyses again and verify that the results are exactly the same.

CTLE Gain with USB3.0 Parameters

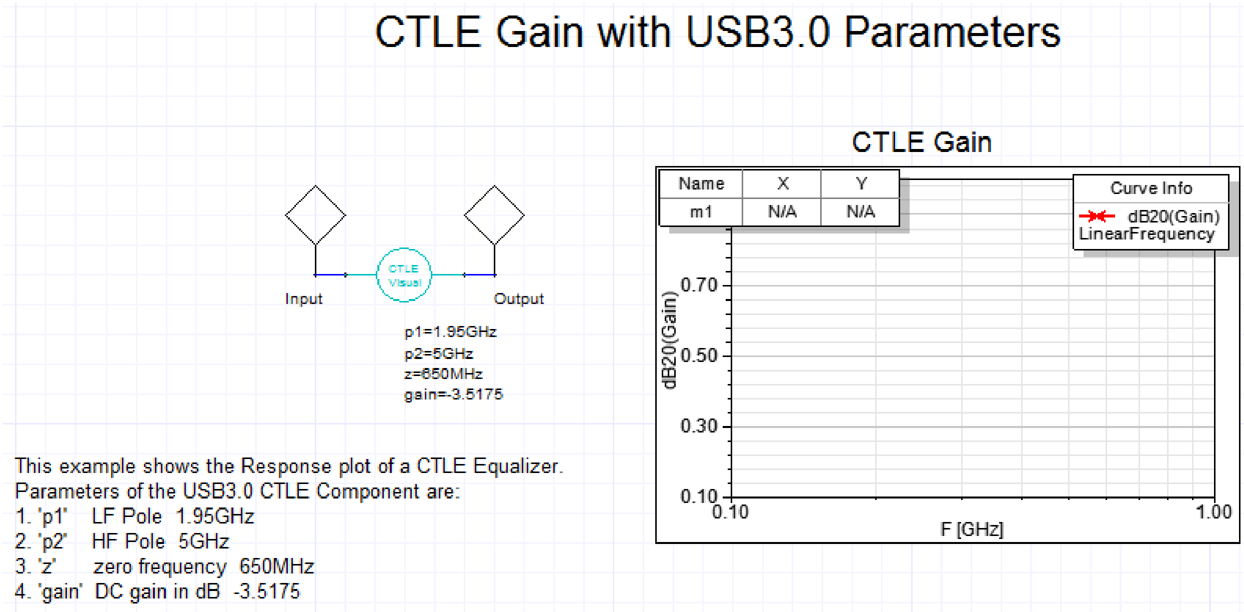
This example plots the response of a continuous-time linear equalizer (CTLE) using the *CTLE Visualization* element on the Circuit Component library. The CTLE boosts the high-frequency components of the received data. The CTLE parameters are pole 1 (low bound of HP), pole 2 (high bound of HP), and pole z (start of upslope). The gain can also be specified.

The initial values (p1=1.95GHz, p2=5GHz, z=650MHz, gain=-3.5175), are on the example given in the USB 3 Compliance Standard.

The schematic for the circuit is available in the Examples directory.

Open the project from its file in the Examples directory.

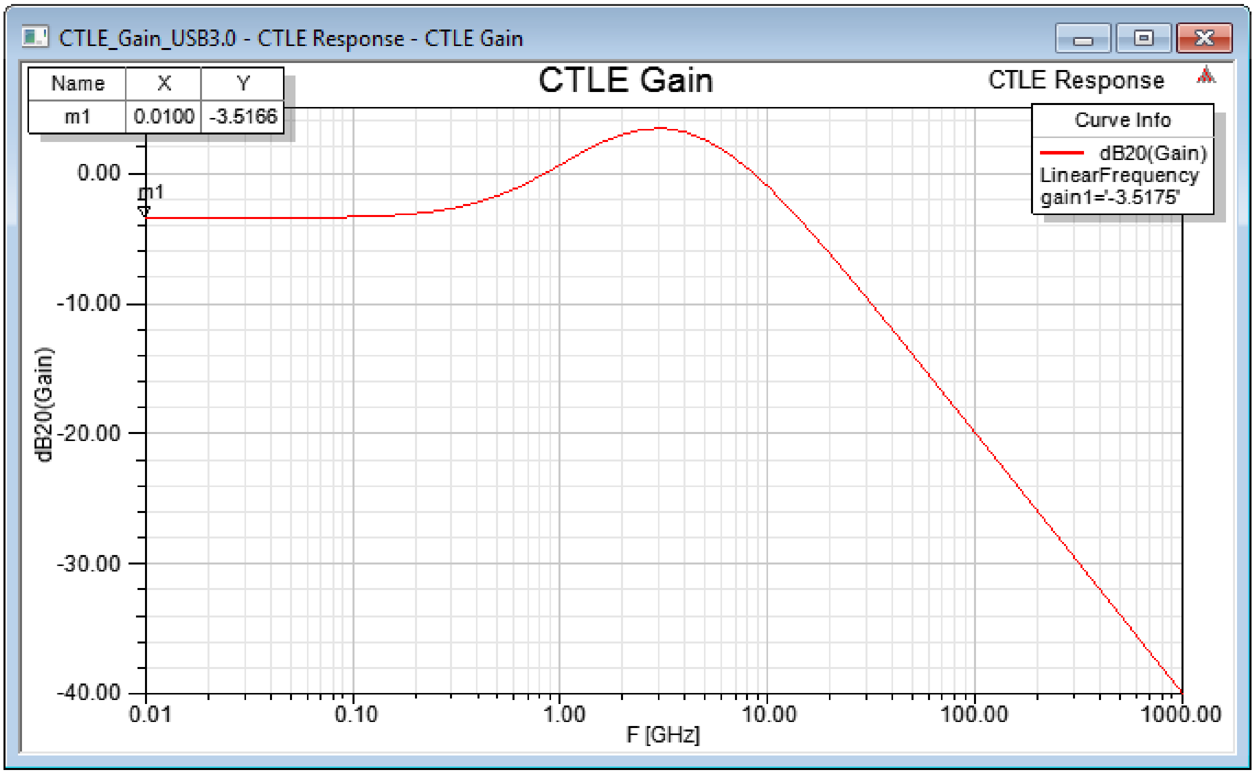
1. From the **File** menu, select **Open Examples** to open an explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **CTLE_Gain_USB3.0.aedt**, or select **CTLE_Gain_USB3.0.aedt** and click **Open**.



The plot on the schematic is filled in when the analysis is run.

3. Use **File>Save As** to save the project in a directory where you have write permission.
4. Click the CTLE Visual component to verify the parameter settings.
5. The Linear Network analysis is already set up. Expand the **Analysis** icon, double-click the **Linear Frequency** setup, and select **Analyze** (you may need to use **Force Analysis**).

6. When the analysis is complete, the schematic plot is filled in. You can view a larger plot by expanding the **Results** icon, then clicking on the **CTLE Gain** report:



The marker shows the gain at the lowest plotted frequency (10MHz), close to the DC gain of -3.5175.

7. Use **File>Save** to save the project, and **File>Close** to close it.

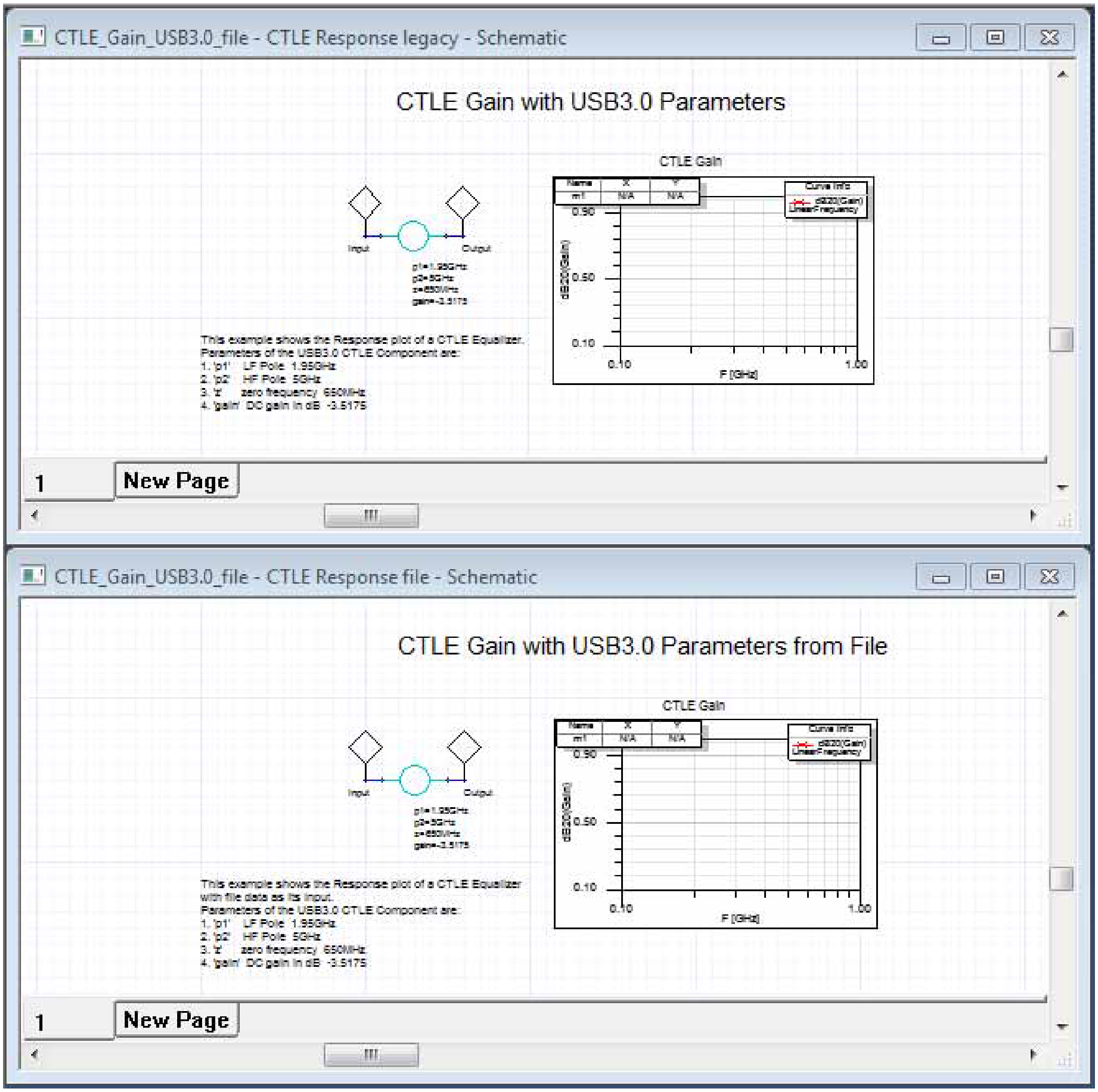
CTLE Gain with USB3.0 Parameters from a File

This example compares the response of the CTLE Visualization element with fixed USB3.0 parameters to the response obtained from a file with CTLE data derived on the same USB3.0 specification.

The schematic for the circuit is available in the Examples directory.

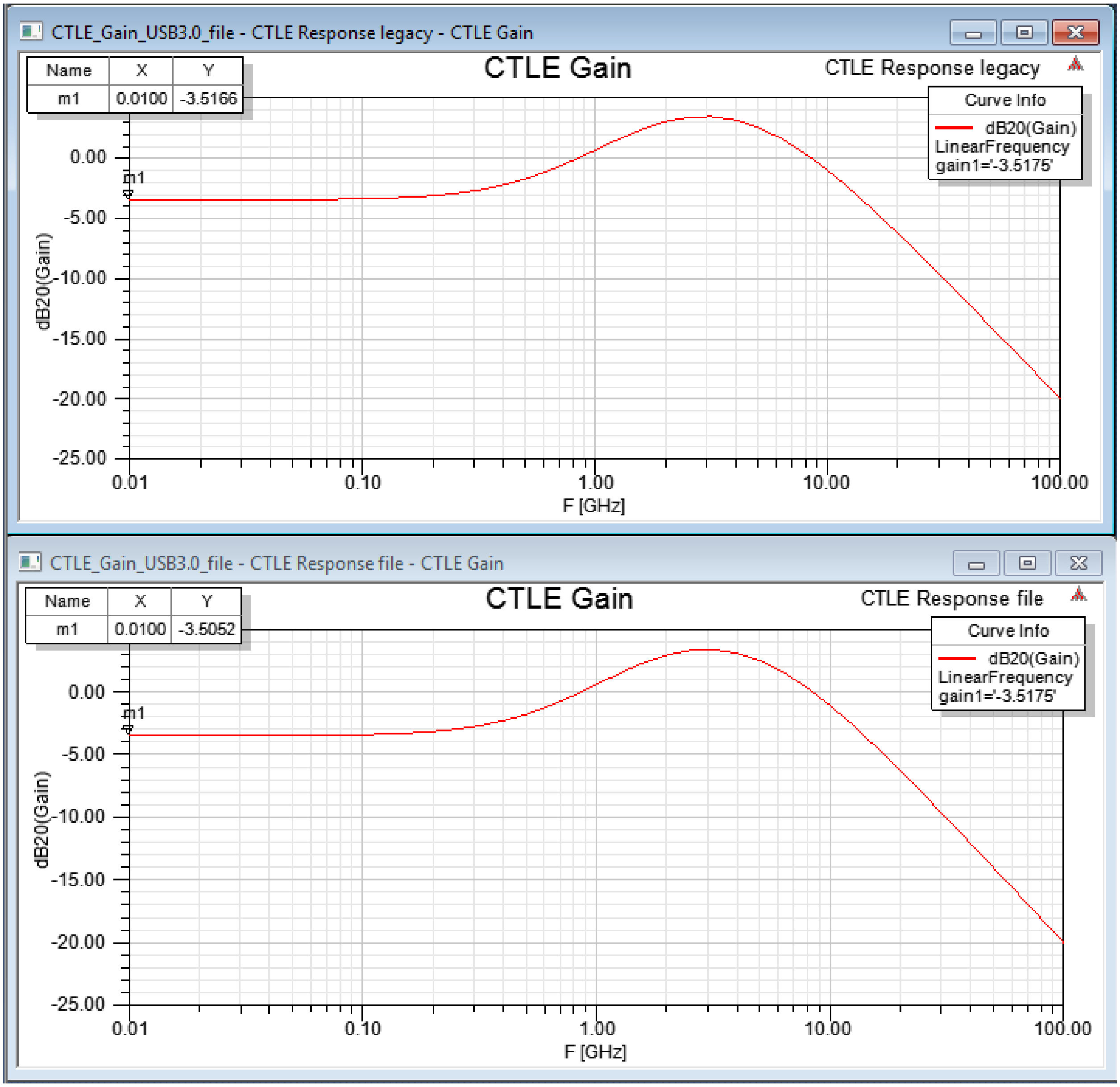
Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **CTLE_Gain_USB3.0_file.aedt**, or select **CTLE_Gain_USB3.0.aedt** and click **Open**.



3. In the project **CTLE Response legacy**, view the Eye Source properties and verify that the values (p1=1.95GHz, p2=5GHz, z=650MHz, gain=-3.5175), are the same as in the example given for the USB 3.0 Compliance Standard.
4. Click the **Linear Frequency** analysis, and click Analyze. The **CTLE Gain** plot for that project is displayed.
5. In the project **CTLE Response file**, view the Eye Source properties and verify that the file parameter points to the data file **usb3.ctle** (in a subdirectory of the Examples folder.)

6. Click the **Linear Frequency** analysis, and click Analyze. The **CTLE Gain** plot for that project is displayed.
7. Verify that the two plots are identical.



CTLE Gain with PCIe3.0 Parameters

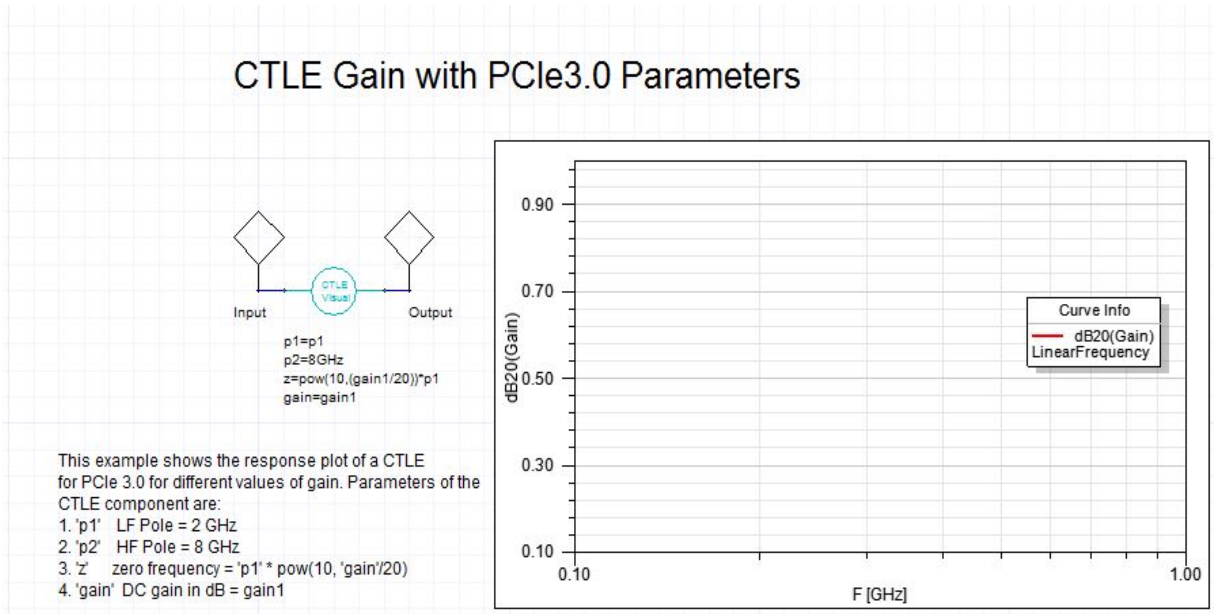
This example plots the response of a continuous-time linear equalizer (CTLE) using the *CTLE Visualization* element on the Circuit Component library. The CTLE boosts the high-frequency components of the received data. The CTLE parameters are pole 1 (low bound of HP), pole 2 (high bound of HP), and pole z (start of upslope). The gain can also be specified and, to implement the PCIe3.0 values in Nexxim, the frequency for pole z is calculated on the gain.

The initial values are p1=2GHz, p2=8GHz, gain=-6, and z=pow(10,(gain/20))*p1, representing the Nexxim implementation of the PCIe3.0 standard.

The schematic for the circuit is available in the Examples directory.

Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **CTLE_Gain_PCIe3.0.aedt**, or select **CTLE_Gain_PCIe3.0.aedt** and click **Open**.

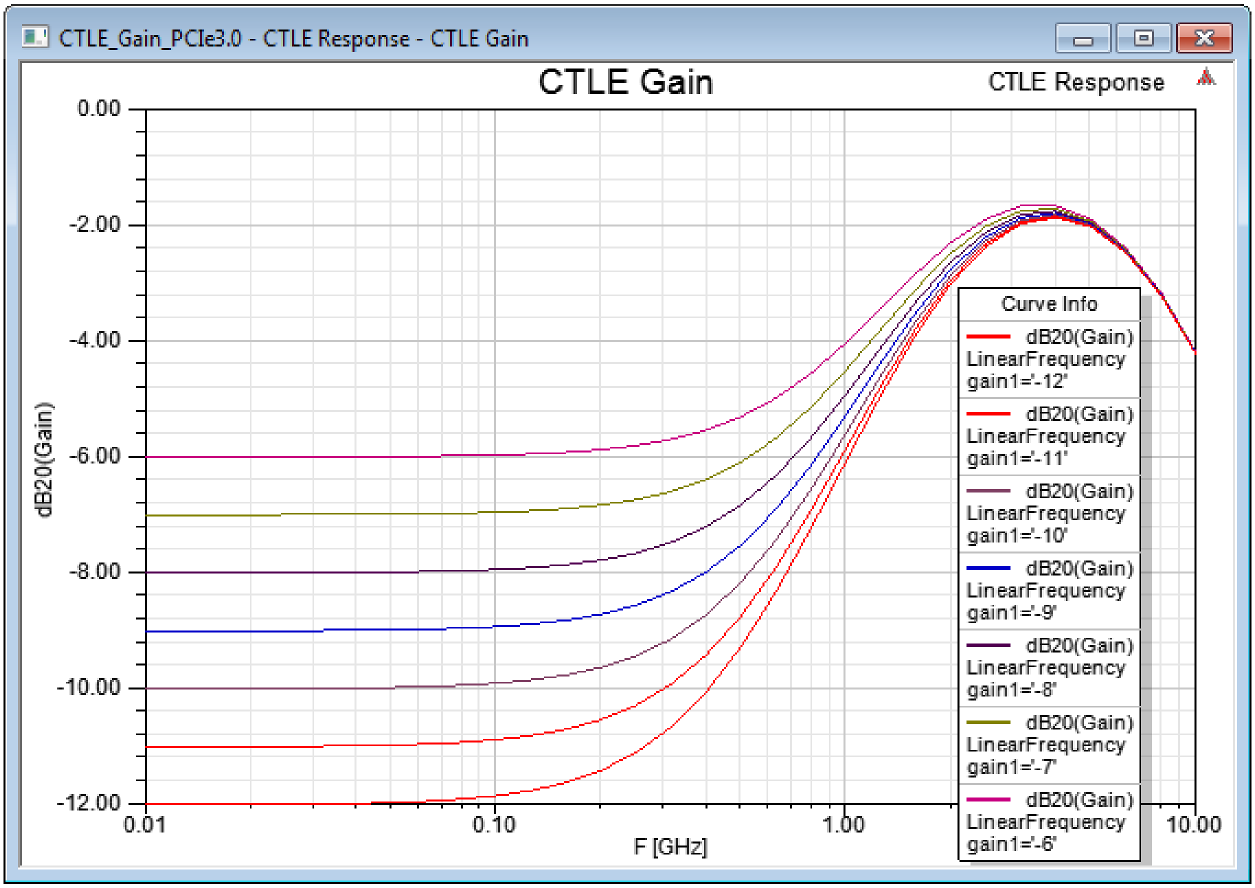


3. Use **File>Save As** to save the project in a directory where you have write permission.

The plot on the schematic is filled in when the analysis is run.

4. Click the CTLE Visual component to verify the parameter settings. The gain parameter is set with a local variable, gain1, that is swept from -6 to -12 dB.
5. The Linear Network analysis is already set up.
6. Expand the **Analysis** icon and double-click the **Linear Frequency** setup. Verify the sweep settings. Click OK to close the setup.
7. Right-click the **Linear Frequency** setup and select **Analyze** (you may need to use **Force Analysis**).

8. When the analysis is complete, the plot is generated. You can view a larger plot by expanding the **Results** icon, then clicking on the **CTLE Gain** report:



9. Use **File>Save** to save the project, and **File>Close** to close it.

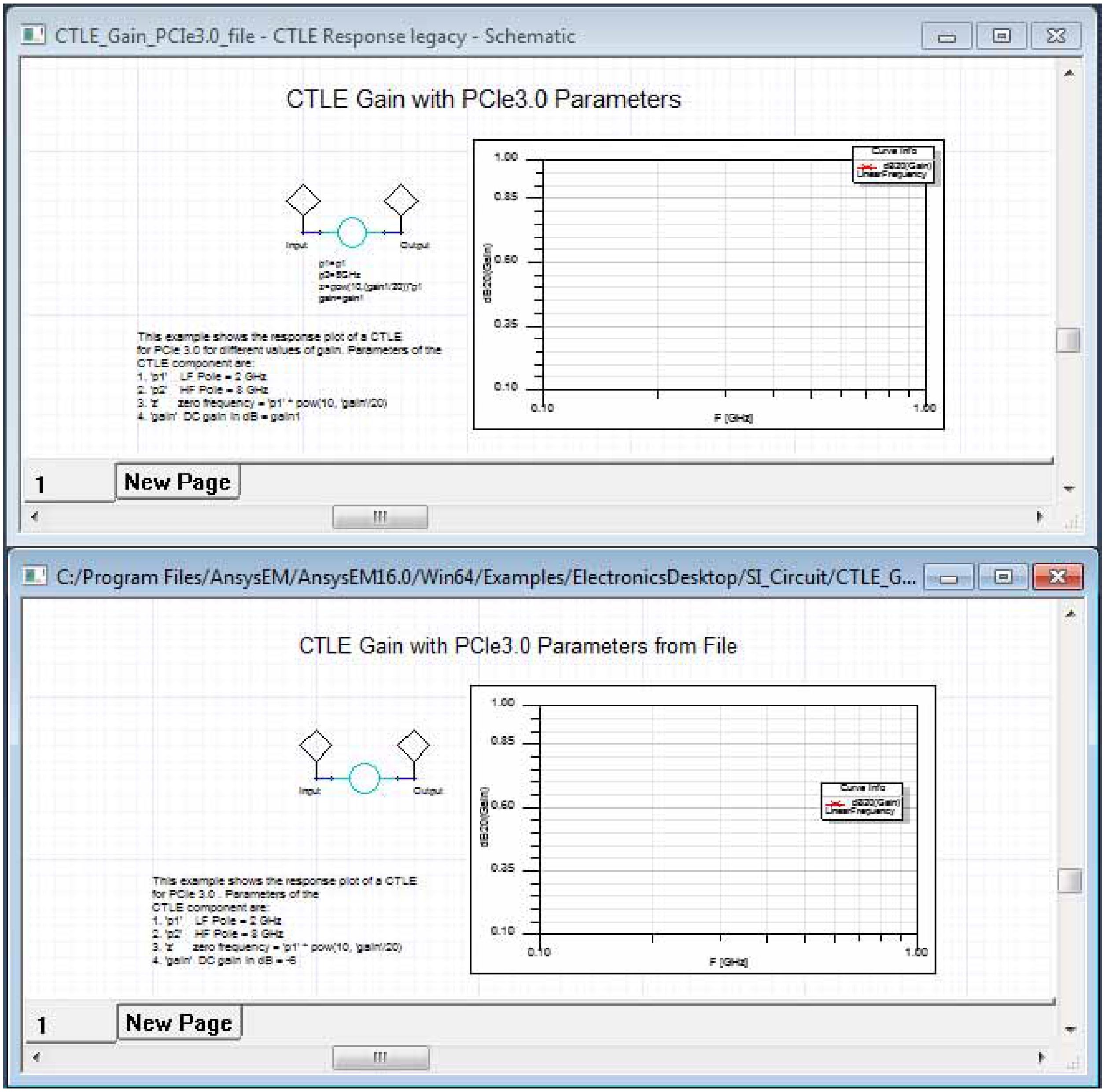
CTLE Gain with PCIe3.0 Parameters from a File

This example compares the response of the CTLE Visualization element with fixed PCIe3.0 parameters to the response obtained from a file with CTLE data derived on the same PCIe3.0 specification.

The schematic for the circuit is available in the Examples directory.

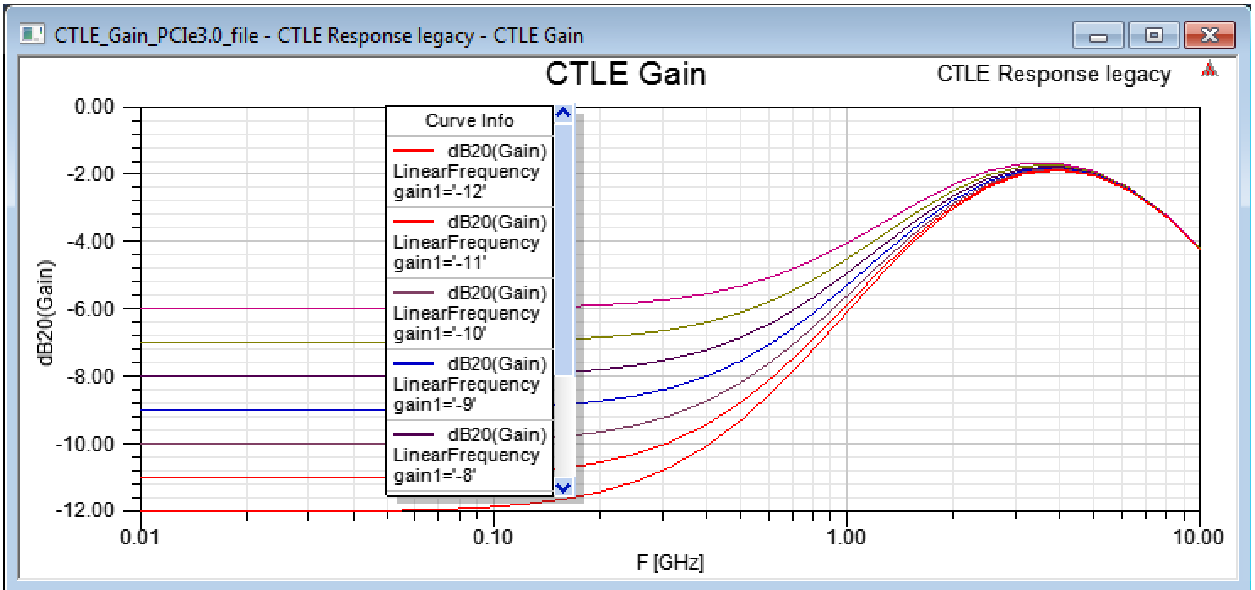
Open the project from its file in the Examples directory.

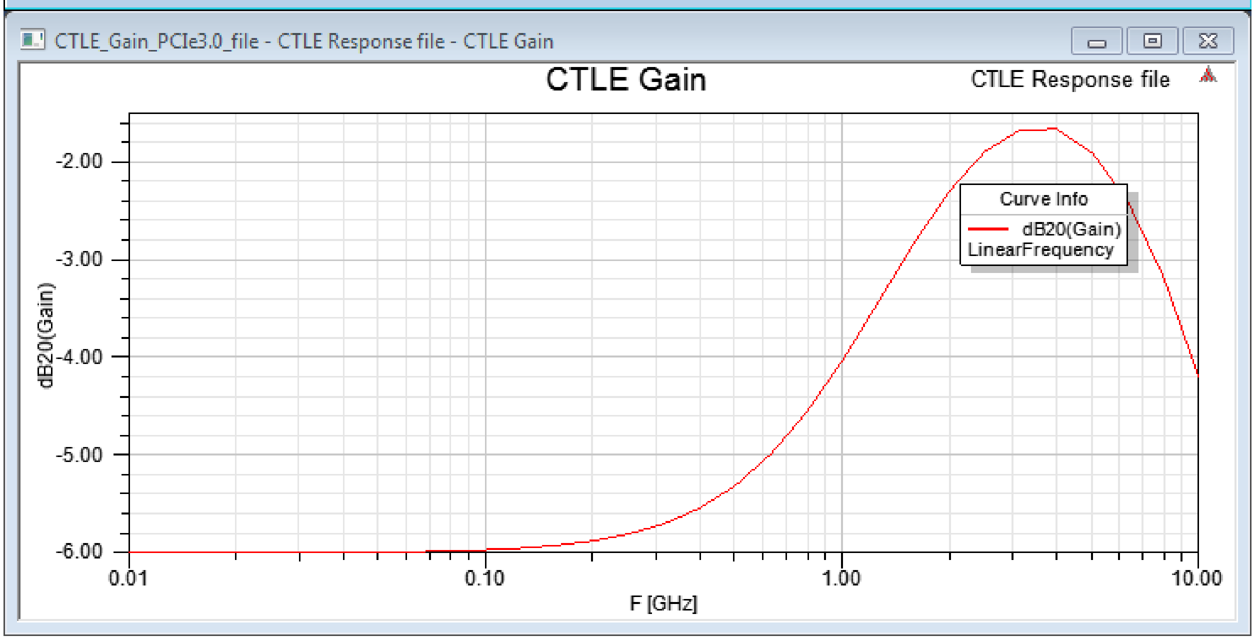
1. From the **File** menu, select **Open Examples** to open an explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **CTLE_Gain_PCIe3.0.aedt_file**, or select **CTLE_Gain_PCIe3.0.aedt_file** and click **Open**.



3. In the project **CTLE Response legacy**, view the Eye Source properties and verify that the values (p1=2GHz, p2=8GHz, gain=-6, and z=pow(10,(gain/20))*p1, are the same as for the example given earlier using the PCIe3.0 standard.
4. The Linear Frequency analysis has been set to sweep the variable **gain1** from -6 to -12. Click the **Linear Frequency** analysis, and click Analyze. The **CTLE Gain** plot for that project is displayed.

Verify that the two plots for gain=-6 are identical.





CTLE Gain using a Generic Rational Function

This example shows how to Click the **Generic rational function** group box on the CTLE data window to specify the transfer function for Continuous Time Linear Equalization (CTLE). Earlier versions of the CTLE data window offered one method, the “legacy” method. These examples show how the newer method can do everything that the legacy method could do, and more.

The CTLE data window also allows you to read the CTLE data from a file. See CTLE Gain with USB3.0 Parameters from a File and CTLE Gain with PCIe3.0 Parameters from a File for examples of the file method.

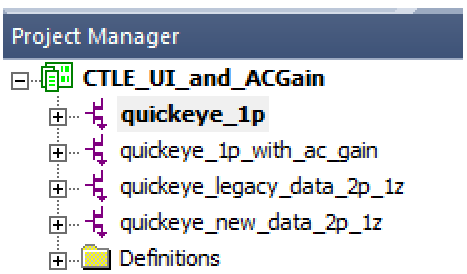
The schematic for the circuit is available in the Examples directory.

Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. From the **Examples** directory, select **Circuit** > **Signal Integrity**. Then double-click **CTLE_UI_and_ACGain.aedt_file**, or select **CTLE_UI_and_ACGain.aedt_file** and click **Open**.

The circuit is the same as the Serial_Channel used for the Eye Analysis Example.

3. There are four Circuit projects in this example. Each one shows a different way to specify a CTLE filter function:



4. Open the **quickeye_1p** project. View the properties for the Eye Source.

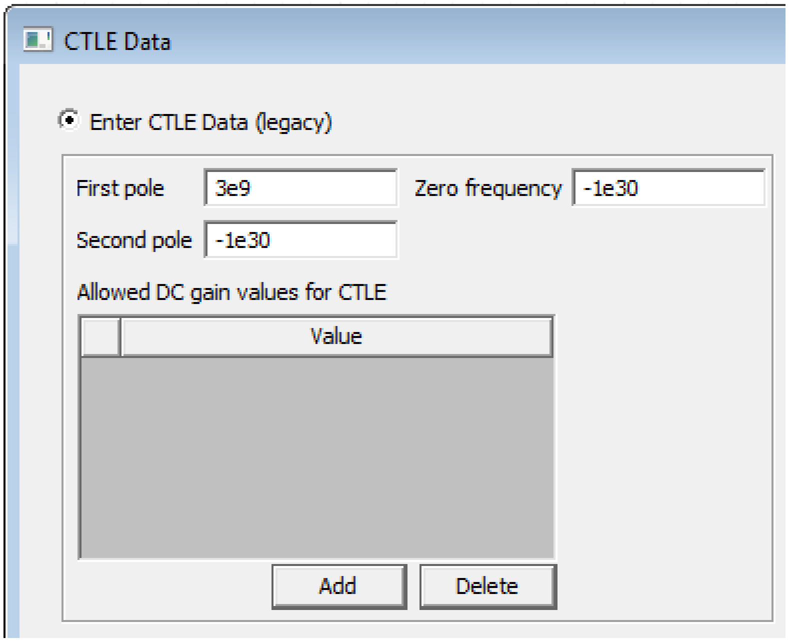
5. All the projects use the same Eye Source setup:

Name	Value	Unit	Evaluated Va...	
resistance	100	ohm	100ohm	Resistance (Ohm)
vlow	-850	mV	-850mV	Low voltage of the source
vhigh	850	mV	850mV	High voltage of the source
trise	50	ps	50ps	Rise time of the source
tfall	50	ps	50ps	Fall time of the source
phase_delay	0	s	0s	Phase of the source in seconds
modulation	NRZ			Serial data modulation, NRZ (two voltage levels, 1 bit p
coding_PAM4_only	Gray			Relevant for PAM-4 only. Coding for mapping low-to-hig
UIorBPS	BitsPerSecond			Choose UnitInterval or BitsPerSecond. UnitInterval is th
UIorBPSValue	6000000000		6000000000	Enter a value for the Unit Interval (time duration of one :
DCDFractionorTime	Fraction			Fraction = decimal fraction of UI, 0 <= DCD <= 1. Time
dcd	0		0	Duty cycle distortion
txrj	['10ps']			Gaussian (random) jitter std deviation
txpj				Periodic (random) jitter amplitude
txuj				Uniform (random) jitter amplitude
txcj				Name of user-defined transmit jitter file
repeat_count	0		0	Number of times to repeat the bit pattern.
step_resp_num_ui	100		100	Num of UIs for initial step response
do_encoding	0		0	Encoding type: 0=none; 1=8b10b; 2=64b66b
hold_last_bit	☐			Hold last bit or repeat the sequence.
FFE_data				FFE data
BitPattern	random_bit_count=1000...			Bit pattern for QuickEye
COMPONENT	EYE_SOURCE			

The transmitter includes a Gaussian jitter with a standard deviation of 10ps to be overcome by equalization. The transmitter itself contributes two taps of feed-forward equalization (FFE).

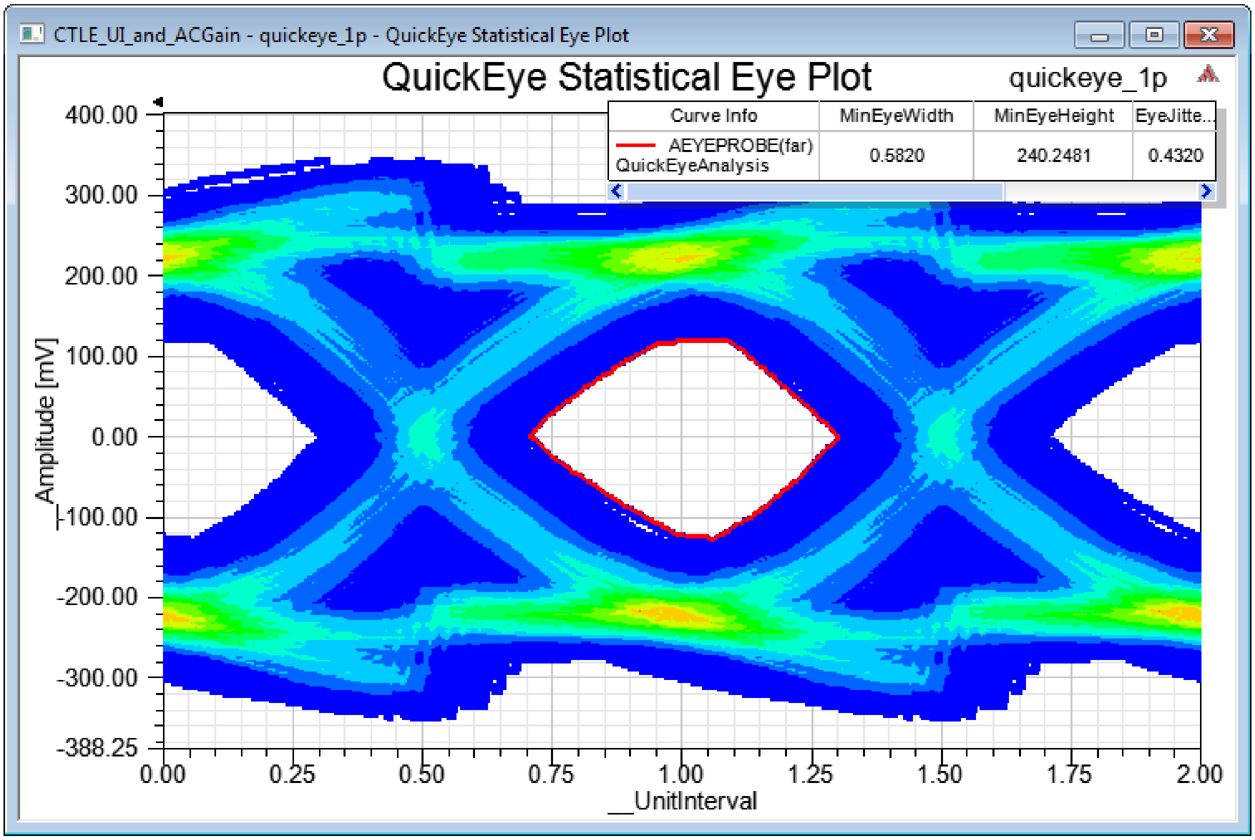
6. View the properties for the Eye Probe. The Eye Probe includes two taps of Decision Feedback Equalization (DFE).

7. Click **CTLE_data**.



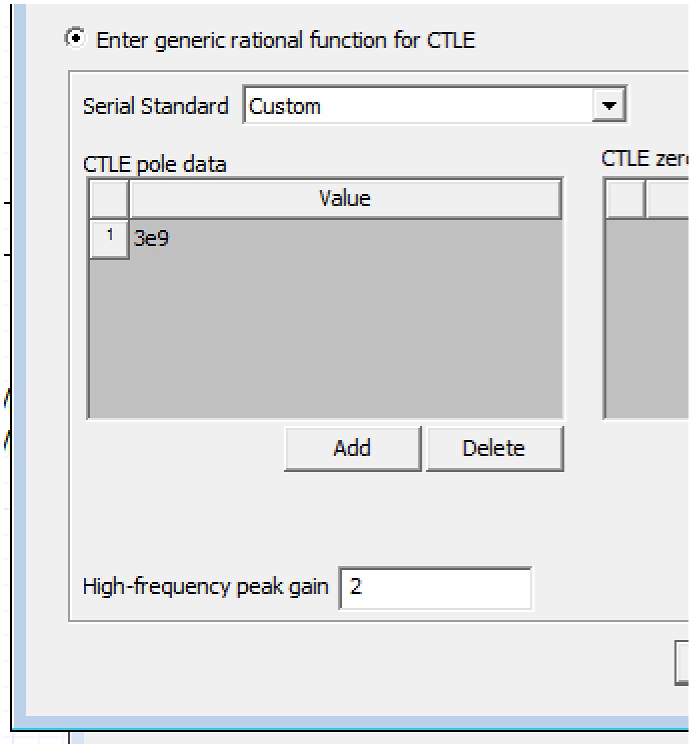
With the “legacy” method, you specify one or two pole frequencies and, optionally, a zero frequency. For this example, **Enter CTLE Data (legacy)** is selected, and the **First pole** is set to 3GHz.

8. Click the **Analysis** icon. A simple Quick Eye analysis has been set up. Click **Analyze**.
9. Under the results, icon, open the **QuickEye Statistical Eye Plot**.



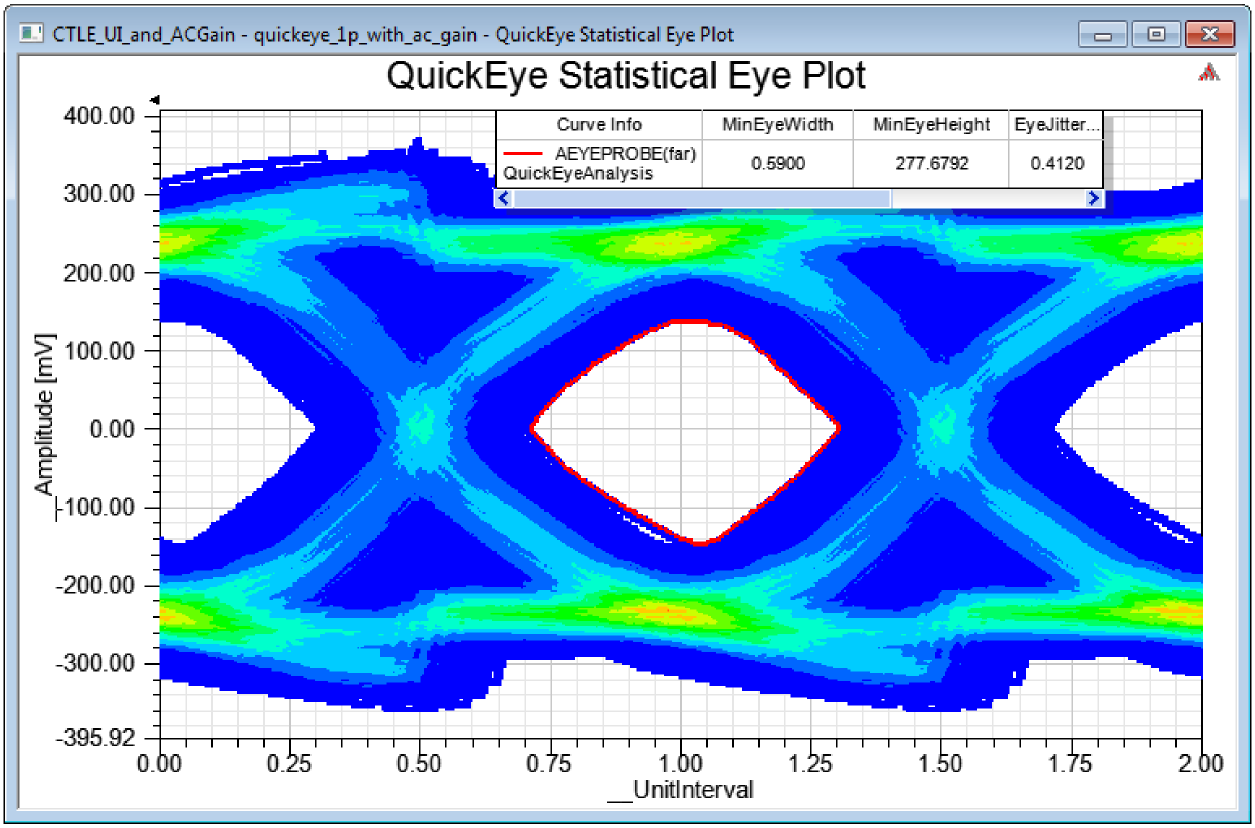
You can see the effect of the random jitter.

10. Close the **quickeye_1p** project and open the **quickeye_1p_with_ac_gain** project. The only change is to the CTLE data window on the Eye Probe:



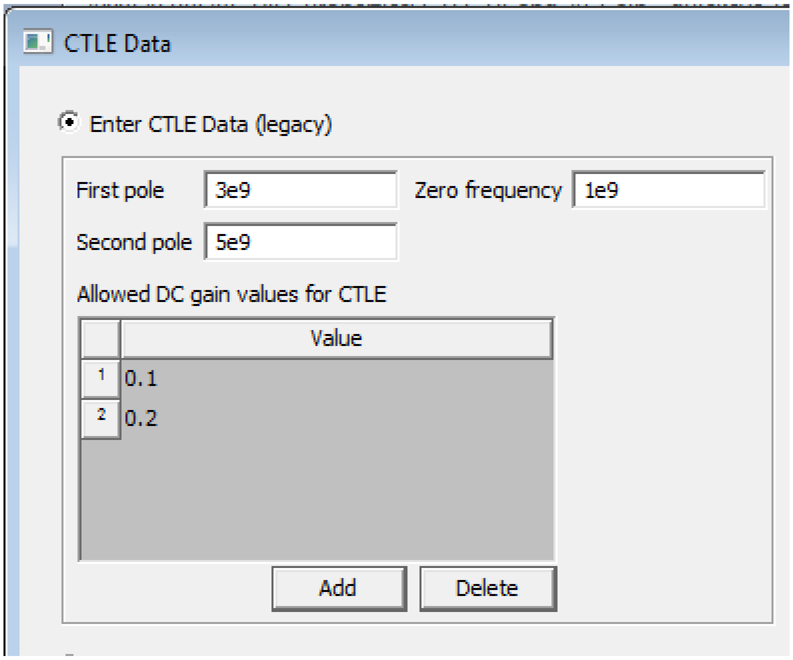
The setup uses the **Enter generic rational function for CTLE** group box instead of the legacy method, but the setup is for the same single pole at 3GHz as before. Using the generic setup, In this example, the High-frequency peak gain or AC gain value, has been added. This sets the maximum value of the CTLE transfer function. The legacy method does not support AC gain.

11. Run the QE analysis for this project, and view the result with AC gain added:



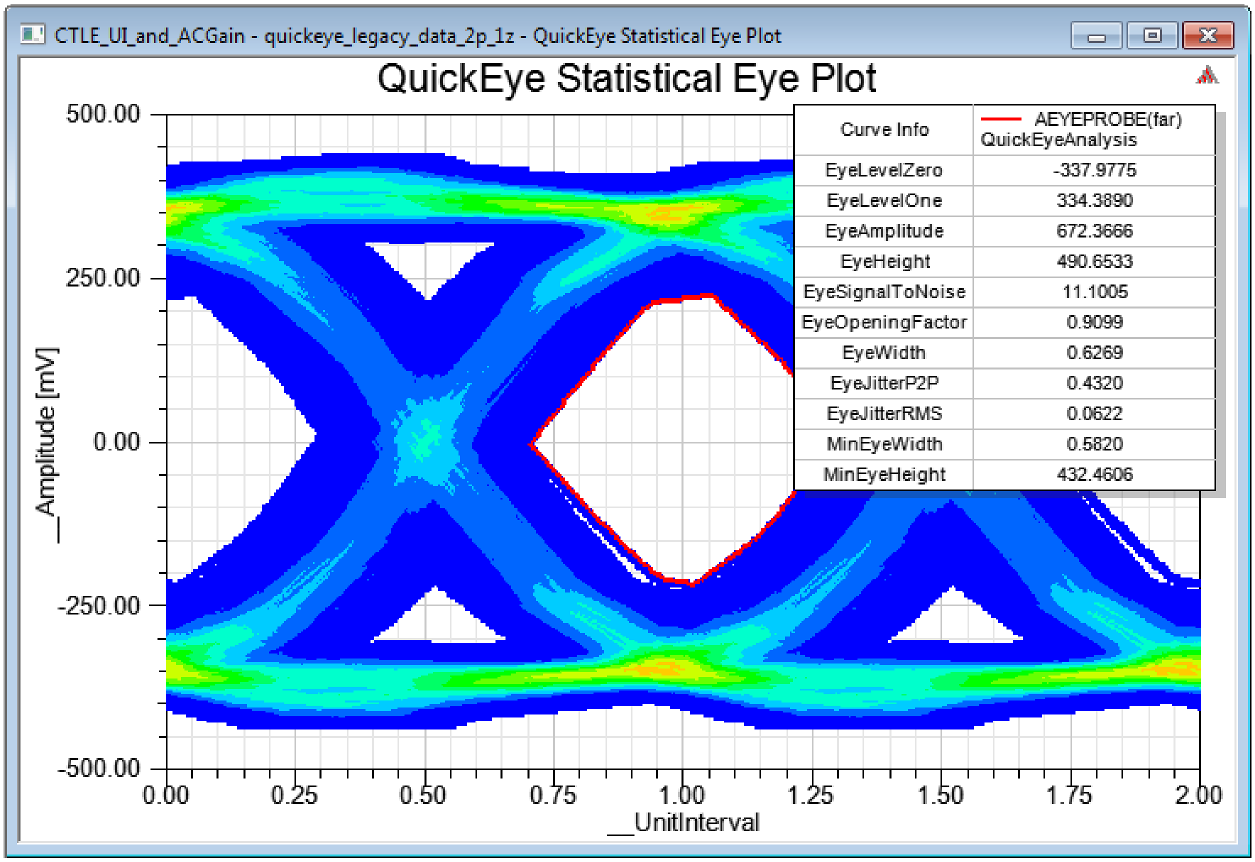
Note the slight improvement in the eye height and width. over the plot with no AC gain in the CTLE.

12. Close the **quickeye_1p_with_ac_gain** project and open the **quickeye_legacy_data_ 2p_1z** project. View the CTLE data window in the Eye Probe properties.

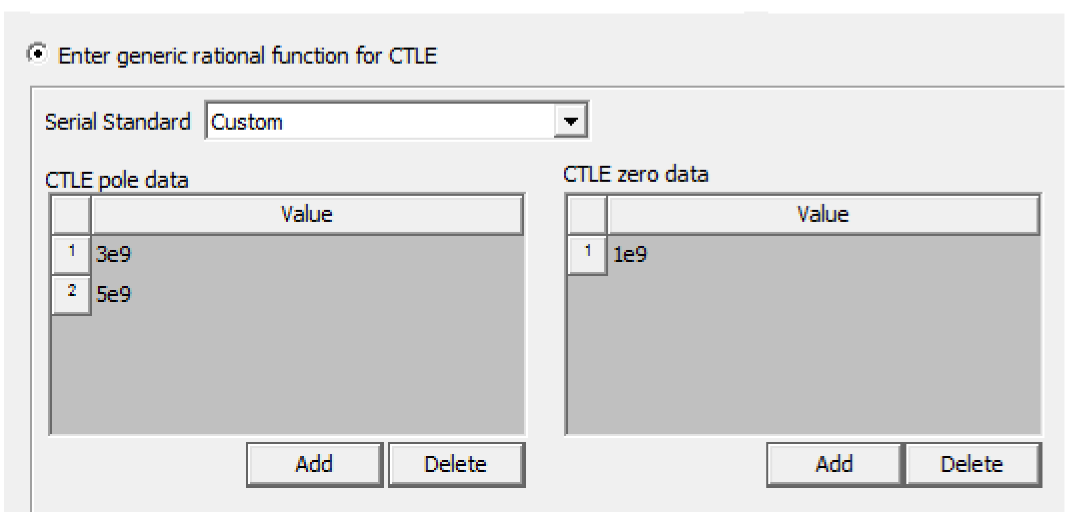


The legacy CTLE data group box has been set with poles at 3gHz and 5gHz, and a zero at 1gHz. The legacy group box also allows you to specify a range of DC gain values. Nexxim tries all the gain values and uses the one that gives the best eye. Here, two DC gain values of 0.1 and 0.2 are specified.

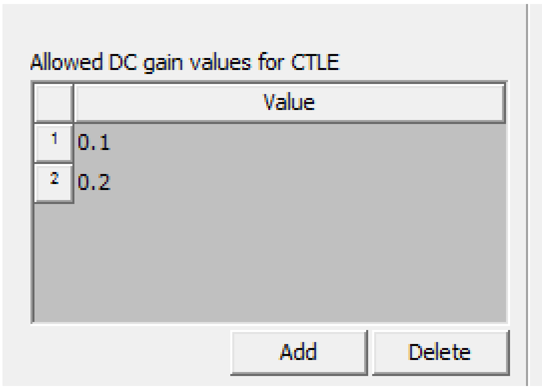
13. Run the Quick Eye analysis and view the report:



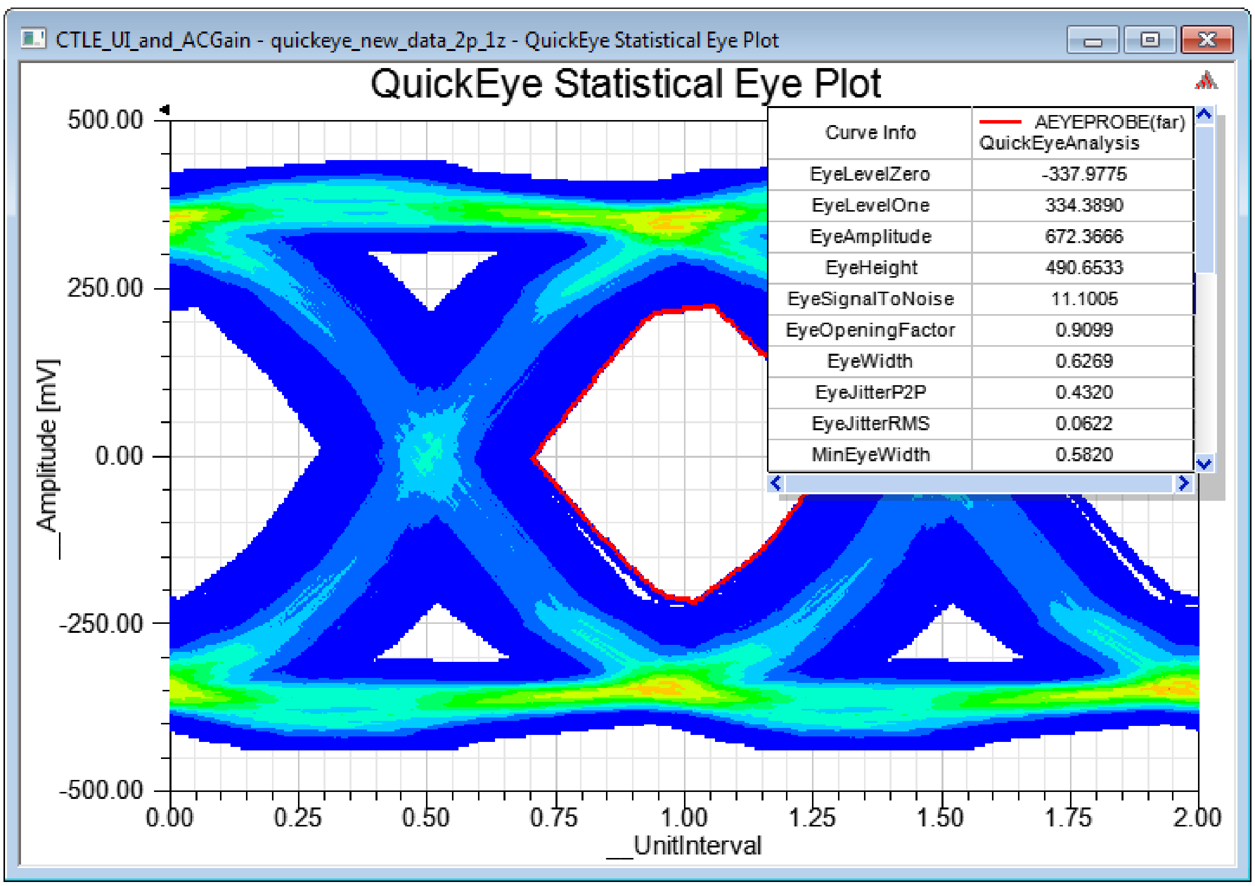
14. Close the quickeye_legacy_data_2p_1z project and open the **quickeye_new_data_2p_1z** project. View the CTLE data window:



15. This setup is identical to the legacy 2pole, 1 zero specification. The same two Allowed DC gain values (0.1 and 0.2 dB) have been added to the specification.



16. Nexxim tries each of these and use the one that gives the best eye.
17. Run the Quick Eye analysis and view the results.



This should be very similar to the two-pole, one-zero legacy setup shown earlier.

PCI Crosstalk Example

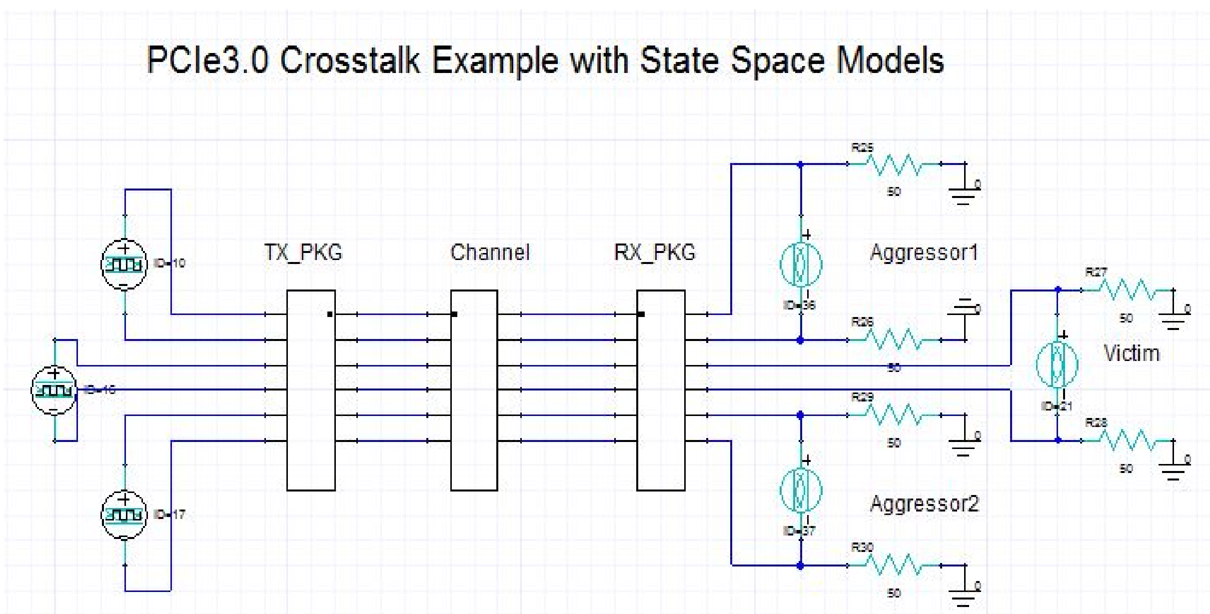
This example analyzes crosstalk on a channel with equalization at both transmitter and receiver. The receiver equalization includes PCIe3.0-specific CTLE. For a simpler example of using PCIe3.0 CTLE see CTLE Gain with PCIe3.0 Parameters.

- The setup uses three differential line pairs so crosstalk between pairs can be evaluated.
- The outer two pairs are the aggressors, and the inner pair is the victim. All three channels use equalization.
- The gain on all channels is swept from (-6dB) to (-12dB), and the reports show how the eye is affected.

The models for this demonstration have been converted on their original S-parameters (.sNp) to state-space fittings (.sss). The complete simulation takes under one minute with state-space models, noticeably faster than the original version with S-parameter models. The original TX and RX Touchstone models are on the PCI-SIG Seasim software package.

The schematic for the circuit is available in the Examples directory. Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **Signal_Integrity**, then select the file **PCIe_StateSpace.aedt**. Click **Open**.



3. Use **File>Save As** to save the project and its subordinate files to a directory where you have write permission.

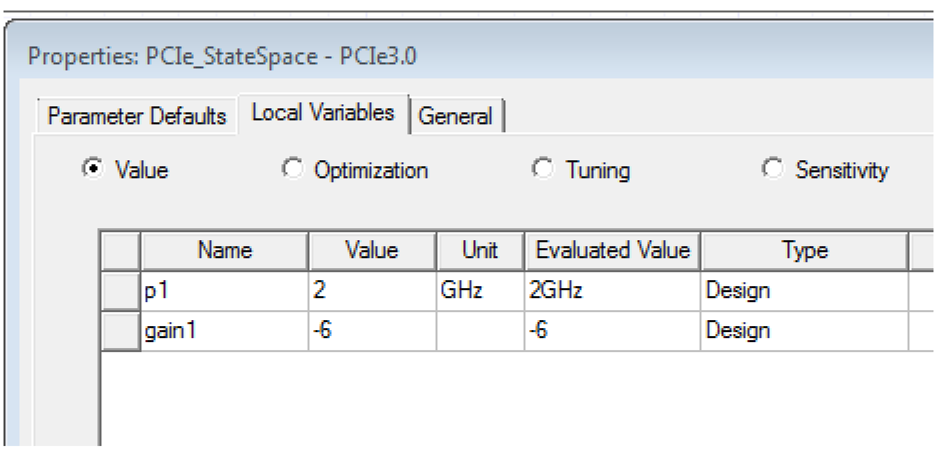
4. Click any of the Eye Sources to view the Properties. All three sources use the same setup:

- Logic levels 0.8V for high, -0.8V for low.
- 125ps unit interval with 10ps rise and fall times.
- Three-tap feed-forward equalization (FFE).
- 7-bit PRBS as the bit pattern.

5. Click any of the Eye Probes to display the Properties. All three probes use the same setup:

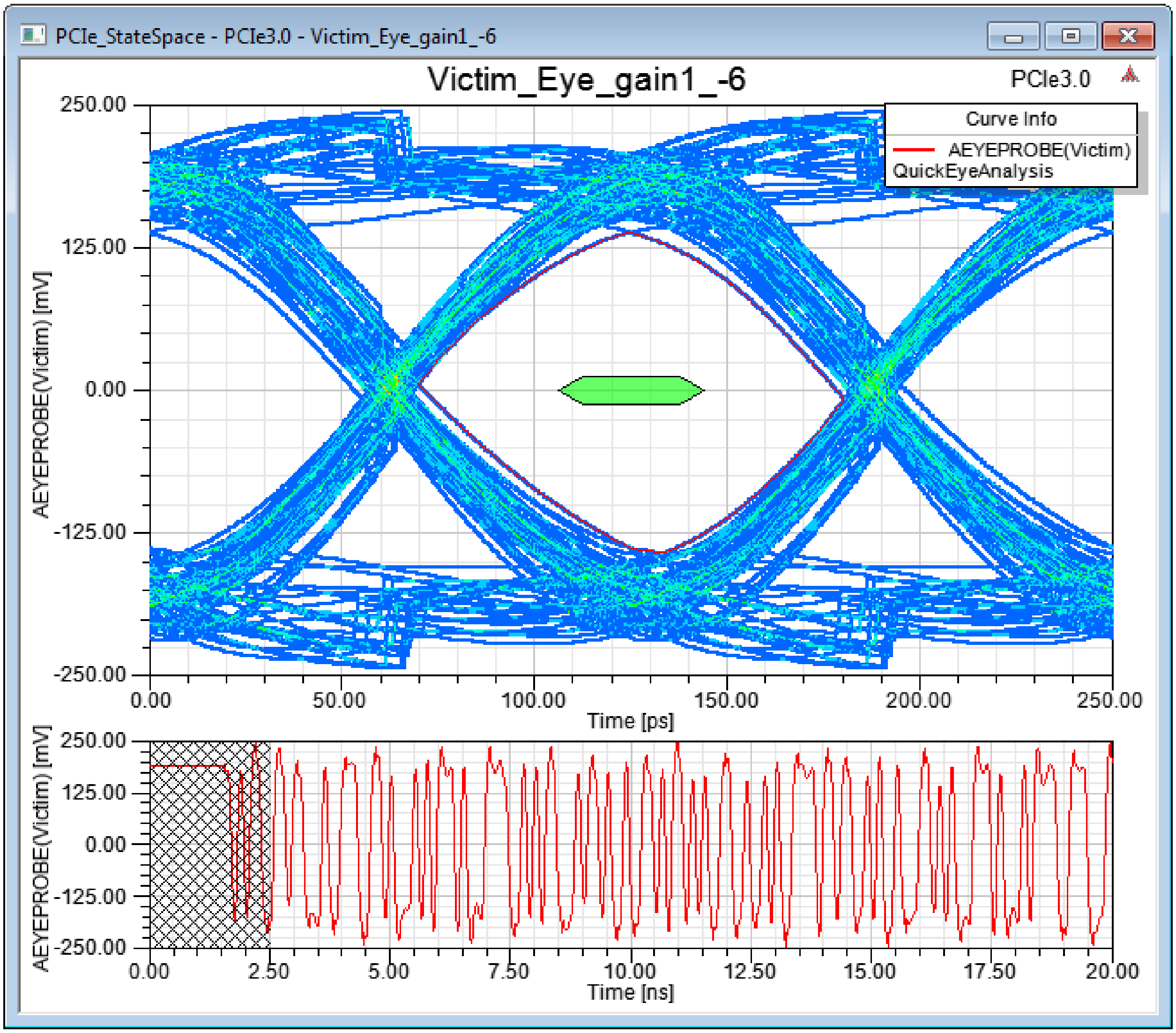
- One-tap decision-feedback equalization (DFE).
- Continuous-time linear equalization (CTLE). The CTLE boosts the high-frequency components of the received data. The CTLE parameters are pole 1, pole 2, and pole z. For the Nexxim implementation of the PCIe3.0 standard, the frequency for pole z is calculated from the gain. The initial values, pole 1 (p1) at 2GHz, pole2 at 8GHz, gain (gain1) at -6dB, and pole z=pow(10,(gain/20))*p1, represent the Nexxim implementation of the PCIe3.0 standard.

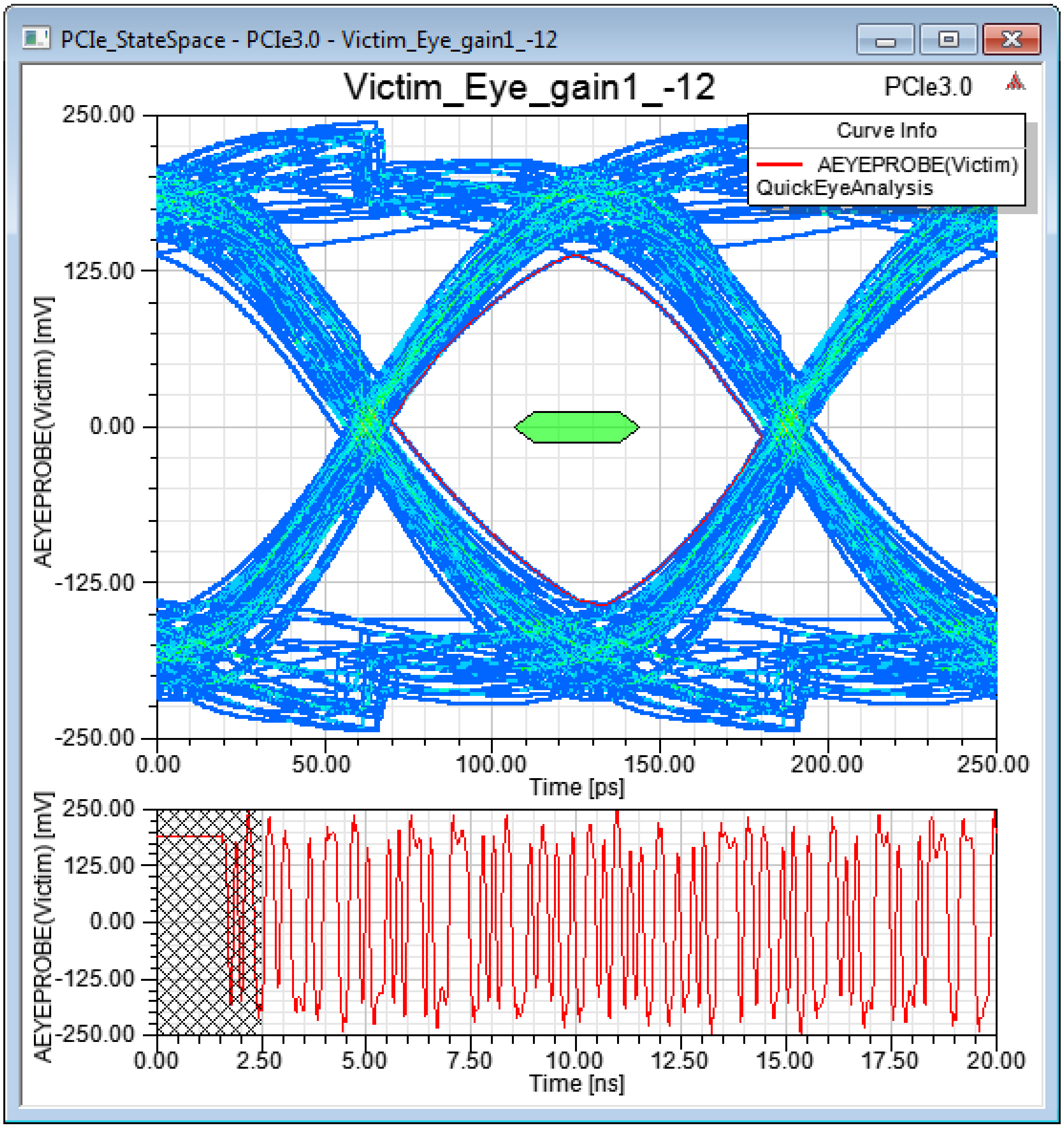
6. The values for pole 1 and gain are set up as local variables, **p1** and **gain1**, so they may be swept. In this example, only the gain is swept. To view the local variables, right-click the **PCI3.0** project, then select **Design Properties**, and click the **Local Variables** tab:



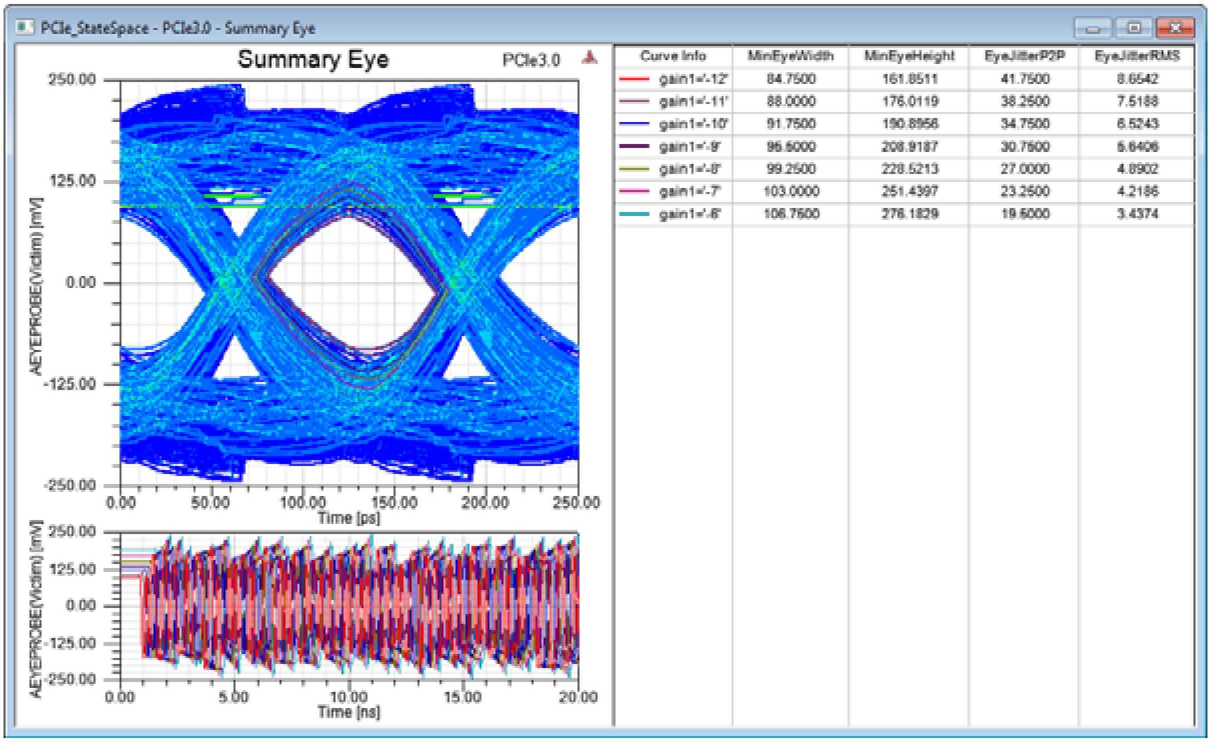
7. The Quick Eye analysis is already set up. The local variable, **gain1**, is swept from -6 to -12 dB.
8. Expand the **Analysis** icon and double-click the **QuickEye** setup. Verify the sweep setting. Click **OK** to close the setup.
9. Right-click the **QuickEye** setup and select **Analyze** (you may need to use **Force Analysis**).
10. When the QE analysis has run to completion, click the reports to see the Eye Diagrams.

The reports show the eye closing as the gain is swept.


PCIe_StateSpace - PCIe3.0 - Victim_Eye_gain1_-6
Victim_Eye_gain1_-6
PCIe3.0
Curve Info
AEYEPROBE(Victim)
QuickEyeAnalysis
250.00
125.00
0.00
-125.00
-250.00
AEYEPROBE(Victim) [mV]
0.00
50.00
100.00
150.00
200.00
250.00
Time [ps]
AEYEPROBE(Victim) [mV]
250.00
125.00
0.00
-125.00
-250.00
0.00
2.50
5.00
7.50
10.00
12.50
15.00
17.50
20.00
Time [ns]



11. The Summary Report overlays all the plots, and provides key metrics for the different gains: Min Eye Height, Min Eye Width, P2P Jitter, RMS Jitter.



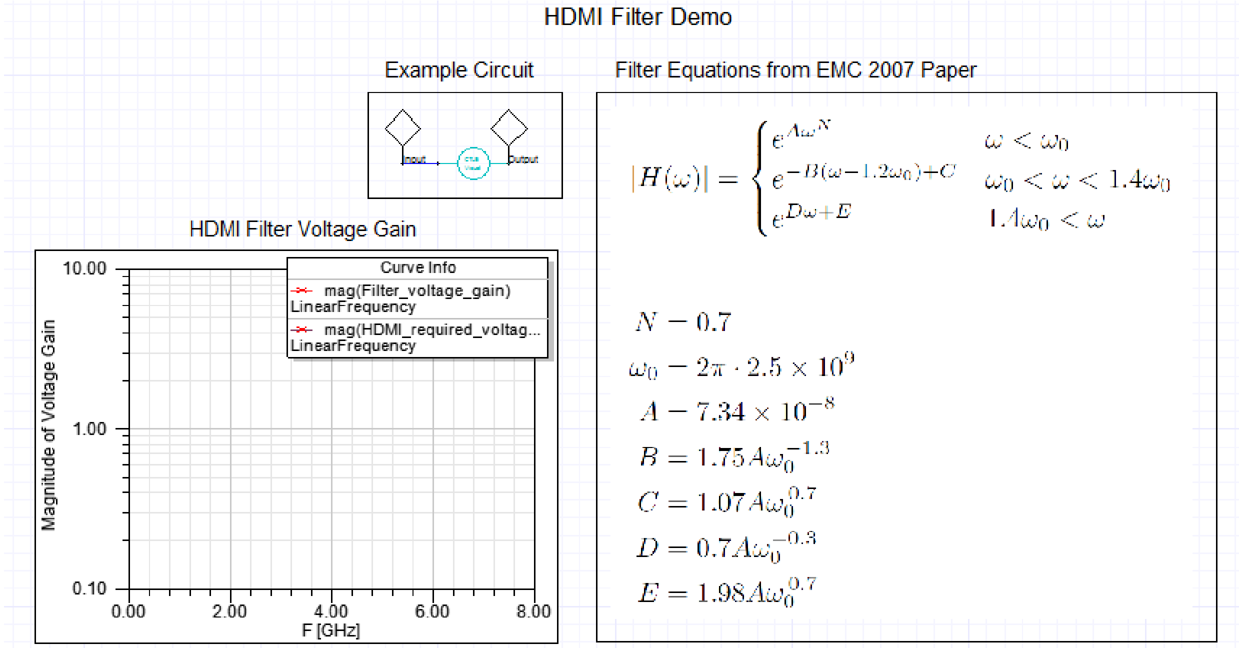
12. Use **File>Save** to save the project, and **File>Close** to close it.

HDMI Filter Demonstration

This example implements a filter for a High-Definition Multimedia Interface (HDMI) cable. The filter equations are from [1]. The *"Causality Enforcement"* paper describes the causality issues that can arise during transient analysis with the HDMI filter.

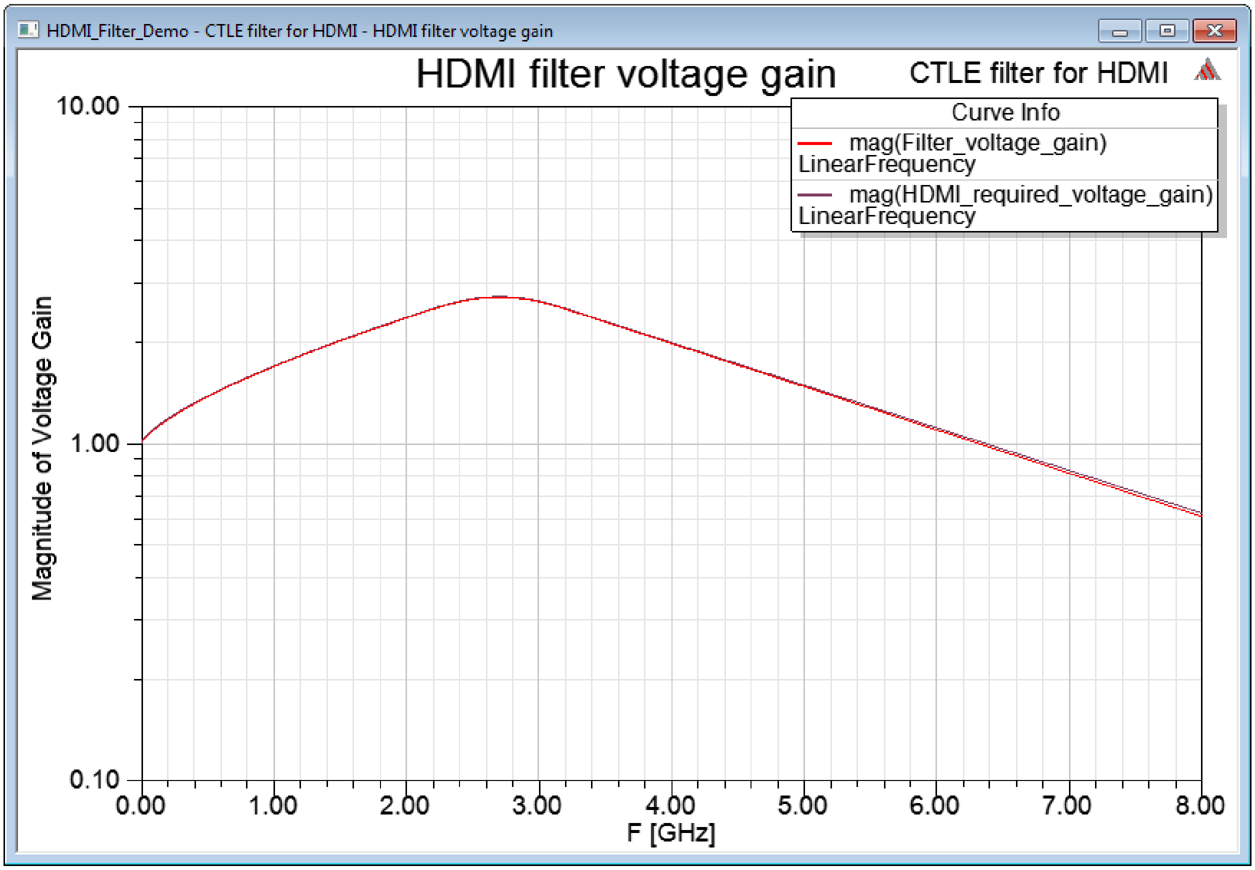
The schematic for the circuit is available in the Examples directory. Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **Signal_Integrity**, then select the file **HDMI_Filter_Demo.aedt**. Click **Open**.



In this example, the filter is realized by fitting a state-space model to a causal transfer function provided in a CTLE data file, HDMI_transfer_function.ctle. The magnitude of this transfer function matches, approximately, the magnitude specified in the equations shown next to the CTLE component.

3. Open on the **Analysis** folder and run the **LinearFrequency** analysis. If needed, use **Force Analysis** to run the LNA.
4. When the LNA has completed, the report in the schematic is updated. The report shows the magnitude of the CTLE component's transfer function and the magnitude specified in the equations, on the same plot. Note that the quantity being compared is the voltage gain.



HDMI Reference

[1] Eakhwan Song, Jeonghyeon Cho, Joungho Kim, and Don Gun Kam, "Causality Enforcement in Transient Simulation of HDMI Interconnects with Magnitude Equalization." *IEEE International Symposium on Electromagnetic Compatibility*, 2007.

Transient Parallel Bus Speedup Demonstration

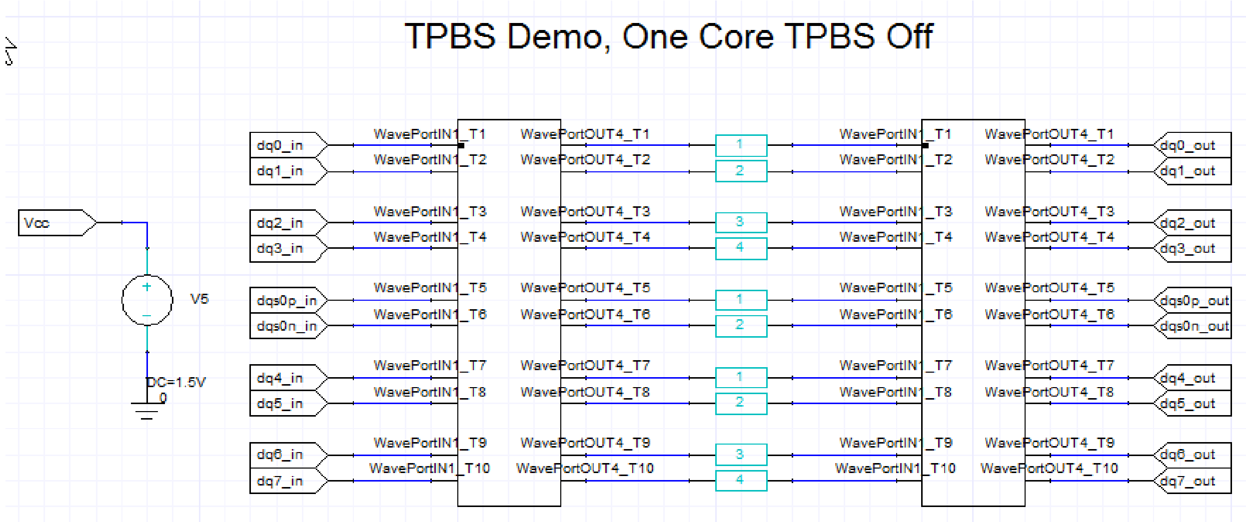
Highly non-linear circuits do not simulate well with QuickEye or VerifEye, so transient analysis must be used. To speed up transient solutions for large non-linear designs that require simulating millions of bits, Nexxim offers the Transient Parallel Bus Speedup (TPBS) option.

Transient parallel bus speedup operates by dividing the simulation length into equal-length overlapping windows; simulation of windows is performed in parallel. After calculating the response within each window, the overlaps are removed using an iterative error-minimizing algorithm. Due to the overhead required, transient parallel speedup should be used only for bit-pattern problems requiring from a few hundred to millions of bits to determine. As shown in this demonstration, TPBS can achieve a significant speedup of a transient simulation.

To run this example, your computer system must have an HPC license and at least four CPU cores. Parallel speedup also requires at least 20GB of free memory, 30GB recommended.

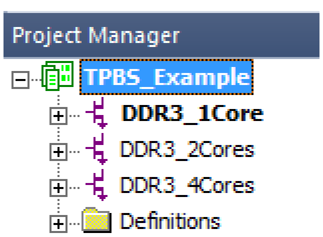
Open the project from its file in the Examples directory.

1. From the **File** menu, select **Open Examples** to open an explorer window.
2. Open the **Electronics Desktop** directory, then **Circuit**, then **Signal_Integrity**, then select the file **TPBS_Example.aedt**. Click **Open**.

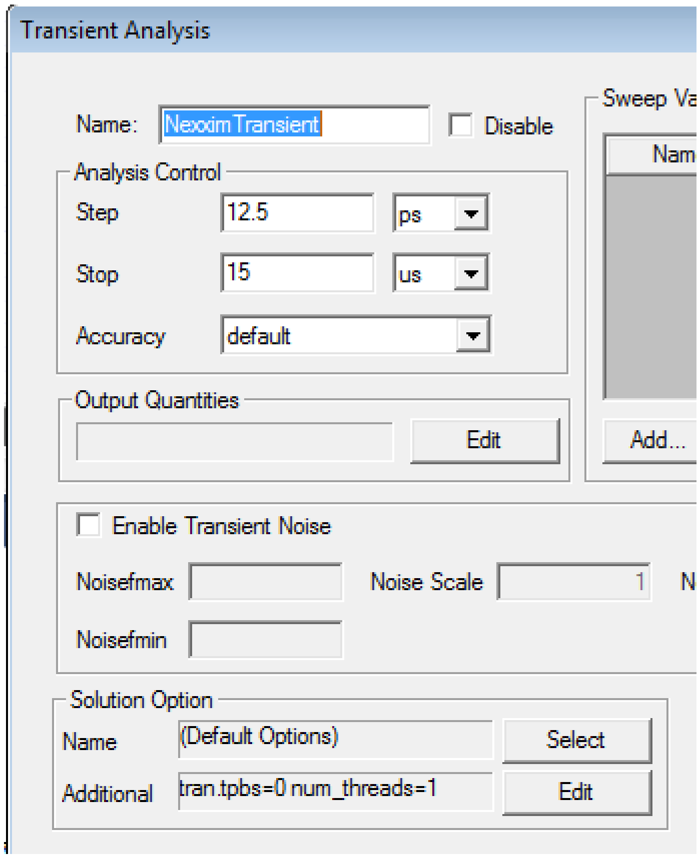


The demonstration uses three identical copies of the DDR3 project to provide a large-scale transient simulation. This does not represent a typical application for TPBS, but it serves as a demonstration.

There are three projects:



3. Expand the **DDR3_1Core** project. Expand the **Analysis** icon and double-click the **NexximTransien** t setup to open it:



The transient analysis runs for 15µs with a maximum step size of 12.5ps.

In the **Solution Options** group box, the **Additional** options have been set to:

```
tran.tpbs=0 num_threads=1
```

This turns TPBS off and sets Nexxim to run with one core, locally to this transient analysis.

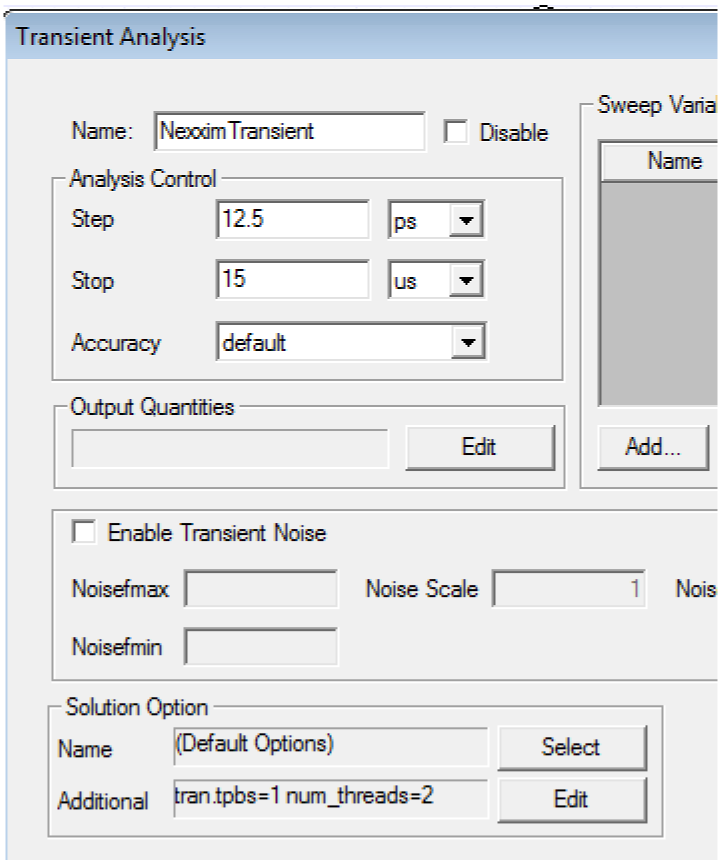
4. Click **OK** to close the Transient Analysis window.
5. Right-click the **NexximTransient** setup and select **Force Analysis**. With one core, the analysis takes nearly 20 minutes.
6. When the analysis is complete, scroll the **Message Manager** to locate the entry **TRAN_ cpu_time**:

```
(info): Simulation succeeded. Total simulator time: 0:18:39 (11:42:10
AM  Apr 27, 2017)
```

With one core, the transient analysis took 18 minutes, 34 seconds.

Now let us see what TPBS can do for us. (Note: the analysis times you obtain may differ in magnitude from machine to machine, but the relative improvements in time with TPBS should be very similar.)

1. From the **Project Manager** window, open the project **DDR3_2Cores**.
2. The only difference is in the **NexximTransient** setup.
3. Expand the **Analysis** icon and double-click the **NexximTransient** setup to open it:



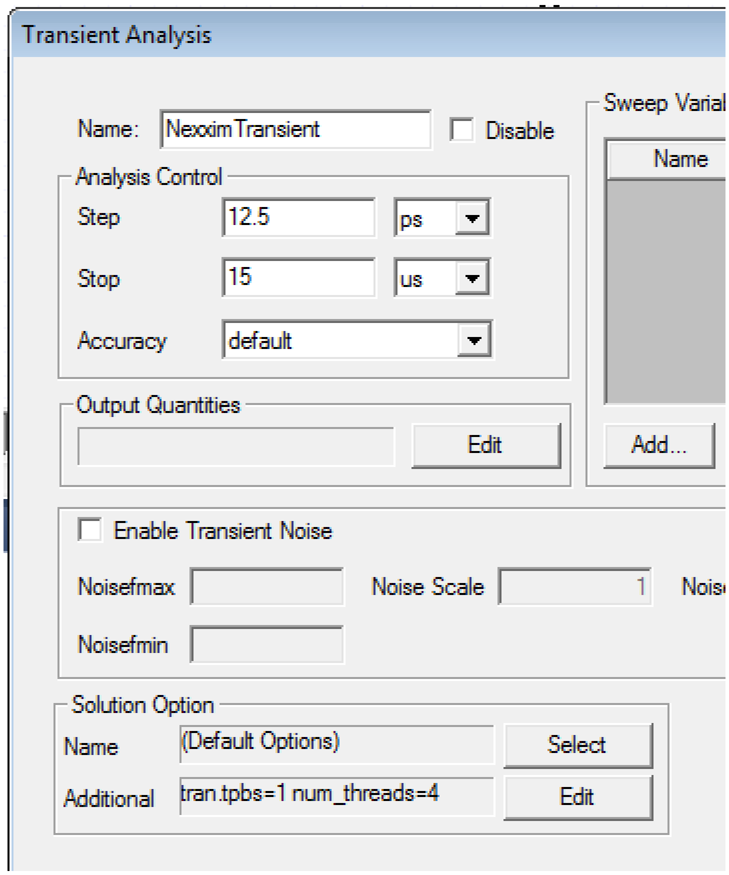
Now, the option tran.tpbs=1 turns on transient parallel bus speedup, and num_threads=2 requests two cores.

4. Right-click the **NexximTransient** setup and select **Force Analysis**. With two cores, the analysis takes about 10 minutes.
5. When the analysis is complete, scroll the **Message Manager** to locate the entry **TRAN_ cpu_time**:

```
(info): Simulation succeeded. Total simulator time: 0:09:47 (11:52:57
AM  Apr 27, 2017)
```

The CPU time of 9 minutes, 47 seconds with TPBS and two cores is a speedup of nearly 2x over the one core time. Now let us see what four cores can do.

1. From the **Project Manager** window, open the project **DDR3_4Cores**.
2. The only difference again is in the **NexximTransient** setup.
3. Expand the **Analysis** icon and double-click the **NexximTransient** setup to open it:



Now, the option tran.tpbs=1 turns on transient parallel bus speedup, and num_threads=4 requests four cores.

4. Right-click the **NexximTransient** setup and select **Force Analysis**. With four cores, the analysis takes about 5 minutes.
5. When the analysis is complete, scroll the **Message Manager** to locate the entry **TRAN_cpu_time**:

```
(info): Simulation succeeded. Total simulator time: 0:05:00
(11:58:46 AM  Apr 27, 2017)
```

TPBS with four cores completes in about 5 minutes, nearly twice the performance of two cores, and nearly 4x the speed for one core.

6. If appropriate, Click the **File>Save As ...** menu item to save the demonstration.

For this demonstration, TPBS was enabled for each transient simulation setup. More commonly, TPBS is enabled for all transient simulations, using the Global Analysis Options window. For more information, see Transient Parallel Bus Speedup in the Circuit Time Domain topic.

21 - Circuit and Layout Definition Libraries

This topic describes how to configure and manage **Electronics Desktop** definition libraries. You can use these topics to create and edit circuit and layout library elements.

The typical Electronics Desktop user requires these library tools only rarely. The Desktop automates the creation of many kinds of components. This topic covers only the operations needed to create and maintain components like the built-in Nexxim and Planar EM components.

- For information on creating component models, see Circuit and Layout Import and Export Operations.
- For information on creating components to be simulated via dynamic links and Solver on Demand, see Co-simulation.
- For information on creating and modifying substrates for use by the layout field solvers, see the *Nexxim Component Models* topics for each kind of distributed element (coplanar waveguide, grounded coplanar waveguide, microstrip, offset stripline, rectangular waveguide, slotline, stripline, and suspended stripline).

Related Topics

Working with Definition Libraries

Using the Library Editor

Using the Material Editor

Using the Script Editor

Using the Component Editor

Using the Padstack Editor

Using the Symbol Editor

Using the Footprint Editor

Vendor Library Components

Encrypted Libraries

Working with Definition Libraries

In Electronics Desktop, a component is a object placed in a design to be solved. Built-in Ansys components contain netlist and footprint information representing the native circuit and EM solver models. Other components can reference external models using frequency-dependent network data or other model formats.

The definition of a built-in component contains parameter data to pass to the solvers. The component definition also references separately stored data for the schematic symbol, fabrication material, layout footprint, and layer stackup definitions. These component dependencies are stored as objects in their own definition libraries.

System, User, and Personal Library Directories

The stock 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.

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 foundry support, user-defined models, and any custom or proprietary sets of components or simulation models. Customarily, **userlib** is a network repository for proprietary or corporate definitions available to all seats in an enterprise. **personalLib** contains project and circuit-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.

Changing the Locations of Libraries

You can relocate the base library directory to a new valid library location. To specify a new library directory:

1. Click **Tools** > **Options** > **General Options**.
2. In the **Directories** group in the **Project Options** tab, type the new folder location in the **Library Directory** group box, or click **Browse ...** to specify it.
3. Click **OK**.

To return the library directory specification to its default value, click **Reset Library Directory**.

The personal library (**PersonalLib**) folder is located at **<Project Directory>\PersonalLib**, where **<Project Directory>** is the location you specified for your Electronics Desktop projects. To specify a new location for the **PersonalLib** folder, you must specify a new project directory location. To do this:

1. From the **Tools** menu, select **Options** > **General Options...**.
2. In the **Directories** group in the **Project Options** tab, type the new folder location in the **Project Directory** group box, or click **Browse ...** to specify it.
3. Click **OK**.

Library and Project Definitions

In Electronics Desktop, library component definitions are intended to be accessed once per component — when you place the first instance of a component onto a schematic. After it is placed in a schematic, the definition for the component transfers on the library to the project file. Editing and updating a component definition is then controlled on the *project definition*, which is listed in the **Components** folder in the **Project Manager** window. In other words, you do not edit component definitions in the library, but in the project. If you want your edits to be reflected in the library definition (for use in another design) you must export the edited components to a component library.

Once a definition is used, it is transferred to the current project, and remains in the project unless it is explicitly removed. To see the definitions included in a project, expand its **Definition** folders and subfolders in the **Project Manager** window:

Modifying component and dependency definitions in libraries, or installing libraries with modified component and dependency definitions, does not automatically update those definitions in projects that contain them. To update project definitions from library definitions, see Updating Project Definitions from Library Definitions.

Updating Project Definitions from Library Definitions

Once a library component is placed into a design, the parent component definition resides in the project, and the link to the library is broken. To update a project component definition with a definition from a library file, select the project you want to update in the **Project Manager** window, then click **Project Tools>Update Definitions** on the **Tools** menu.

The **Update Definitions** window opens, listing any library definitions that have been edited since they are added to the project, and their original library locations. Use the window to select the item to update, and click **Update** to finish.

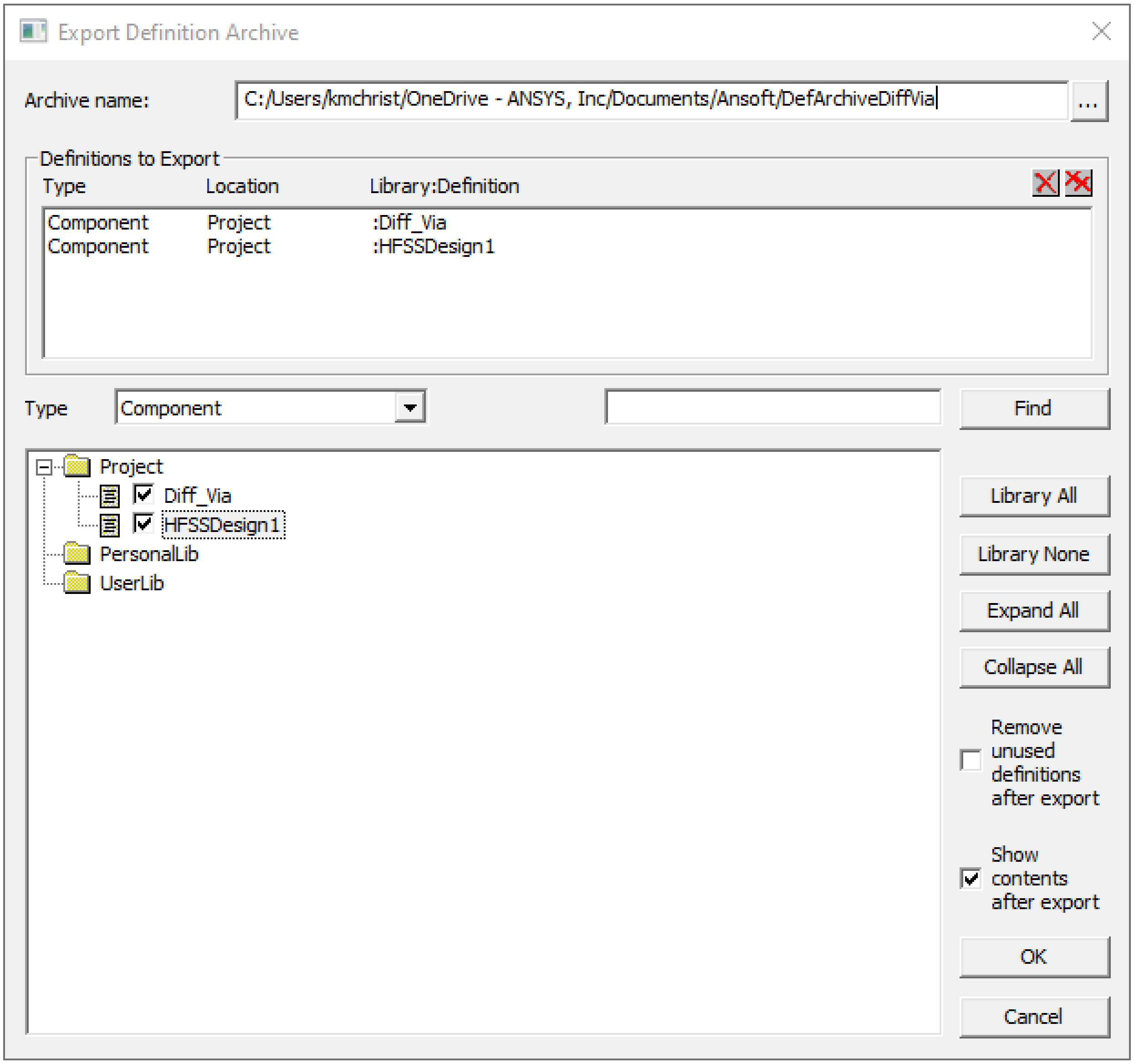
If you wish to remove project definitions that are unused, choose the **Project Tools>Remove Unused Definitions** command on the **Tools** menu which opens the Unused Definitions window.

Exporting and Importing Definition Archives

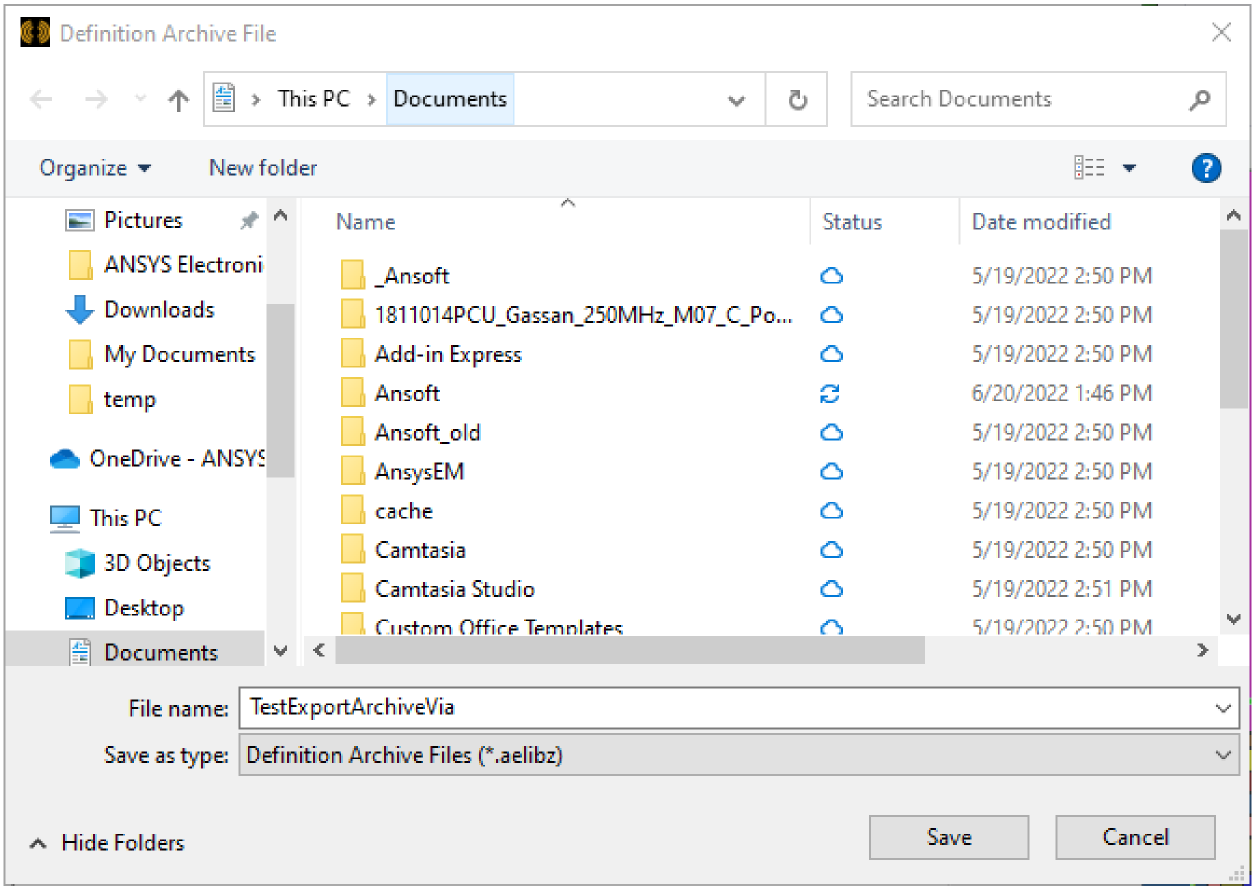
The Import and Export Definition Archive commands allow you to select individual component, model, or other library definitions – or even entire libraries – to package and send to another user. Referenced definitions, data files, and DLLs are included automatically. All selected definitions (either those in an entire library or those selected separately) is written to one set of libraries (e.g., a component library and associated symbol and model libraries).

Exporting Definition Archives

Select **Tools** > **Library Tools** > **Export Definition Archive** to open the **Export Definition Archive** window.



- To export a definition archive, first choose an archive name, either by typing the full filename in to the **Archive name** edit field, or by using the **[...]** button. This button opens a browser window.



- The archive name must have a Definition Archive Files **.aelibz** extension, which is supplied by the window.

> **Note:** An **.aelibz** archive is not readable with utilities such as WinZip®.

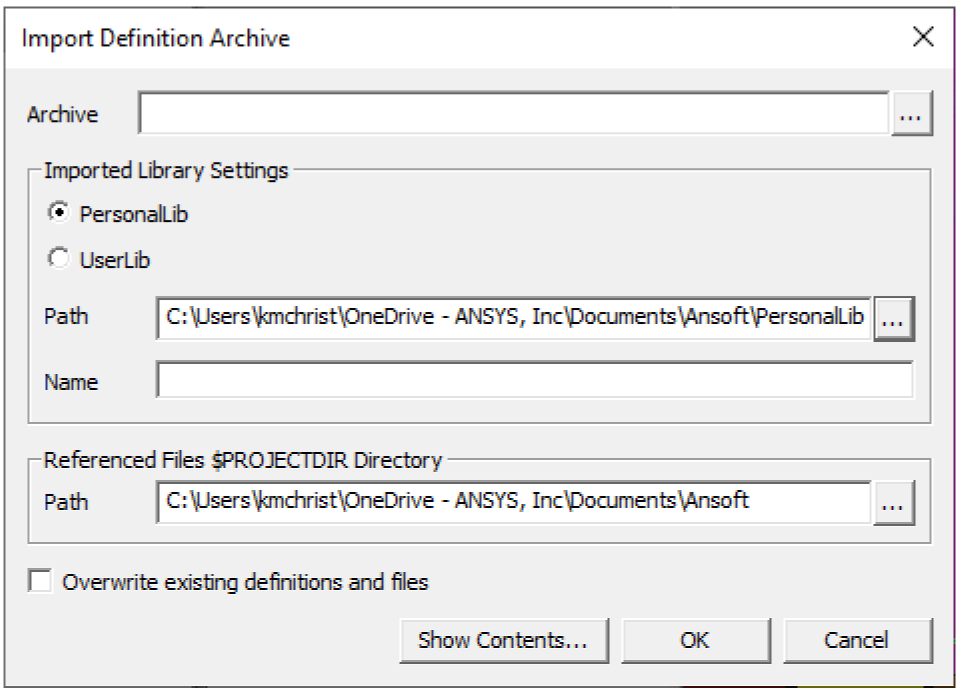
- The definitions selected for export are listed in the **Definitions to Export** listbox. Click the  and  buttons to delete selected entries, or all entries, respectively.
- The **Type** dropdown menu on the left, under the list of **Definitions to Export** controls which libraries are shown in the tree under it. Choices include: **Component**, **Symbol**, **Model**, **Package**, **Material**, and **Script**. In the tree, each leaf element (definition) has an associated check box. Checking the box adds the definition to the export list above. Clearing the box removes the definition on the export list. With a library or library element node highlighted, a user can select all definitions in that library for export by clicking **Library All**. Similarly, the user can unselect all definitions in the selected library by clicking **Library None**. **Expand All** and **Collapse All** expand or collapse the entire tree.
- Typing a string into the field next to the **Find** button allows the user to scroll to the first definition whose name matches the string. Searches are not case-sensitive. You can also use the “wildcard” asterisk character in search strings. For example: entering P* should

find, in succession, every component that starts with P (or p). Use multiple asterisks to further control matches (e.g., type *p* to find any definition with a 'p' in it, including as a first or last character; or *p*p* to find "pipe" and "pump", et cetera).

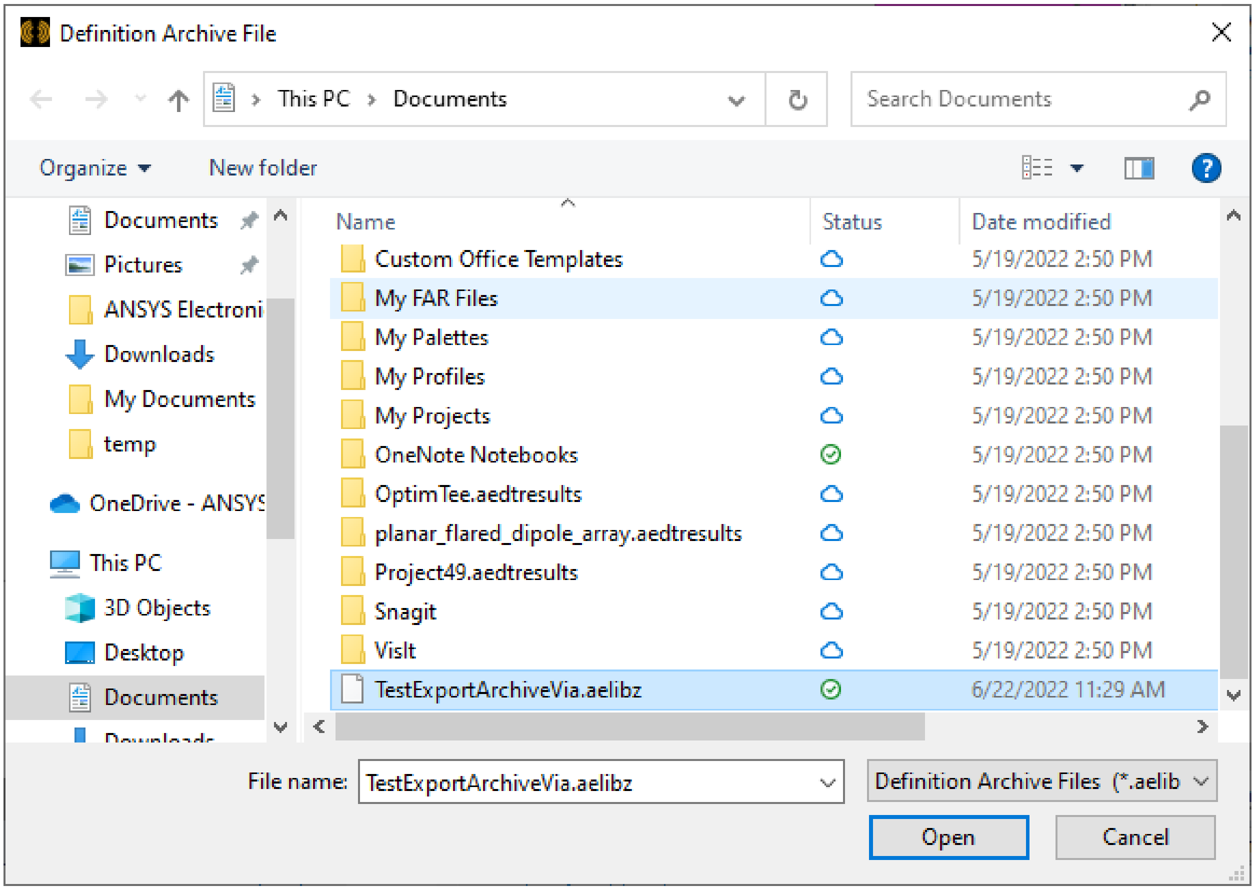
- The **Remove unused definitions after export** check box allows the user to clean up the current project after export is complete.
- When the **Show contents after export** box is checked, a **Contents** window shows a list of what was saved to the archive. This list may include multiple libraries (due to definition references) or external file references. Save the list to a file by clicking **Save , or** click **Done** when you are finished viewing.
- Click **OK** to complete the process. The selected definitions, libraries, and referenced files are zipped, and given a Definition Archive FIles extension, **.aelibz**. Users can then e-mail or otherwise transfer the archive file to others. The archive file is not encrypted, though models and packages contained in it may be.

Importing Definition Archives

To import a definition archive, choose **Tools>Library Tools> Import Definition Archive** to open the **Import Definition Archive** window.



- First select a definition archive either by typing the path and definition archive file name in the **Archive** field, or click **[...]** to browse for the Definition Archive File created using the **Tools>Export Definition Archive** command.



- In the **Imported Library Settings** group box, click **PersonalLib** or **UserLib** to choose the basic location for the imported archive. If appropriate, the user can further choose a subdirectory **Path** under **PersonalLib** or **UserLib** as a location for the libraries. The user can also enter a **Name** for the imported libraries. The default name is the base name of the archive file.
- On import, data files is moved to appropriate **bin** and **data** directories. External **Referenced Files** that are not appropriate to **bin** and **data** library directories is referenced using **$PROJECTDIR**. The user can click the **[...]** button to choose the directory for these files. The default is the Project Directory specified in **Tools>Options>General Options**.
- The **Overwrite existing definitions and files** check box (unchecked by default) allows the user to replace current definitions and files with archive elements, permissions allowing.
- Click **Show contents** to view a window showing a list of what is in the archive. This list may include multiple libraries (due to definition references) or external file references. Save the list to a file by clicking **Save**, or click **Done** when you are finished viewing.

Managing Library Files

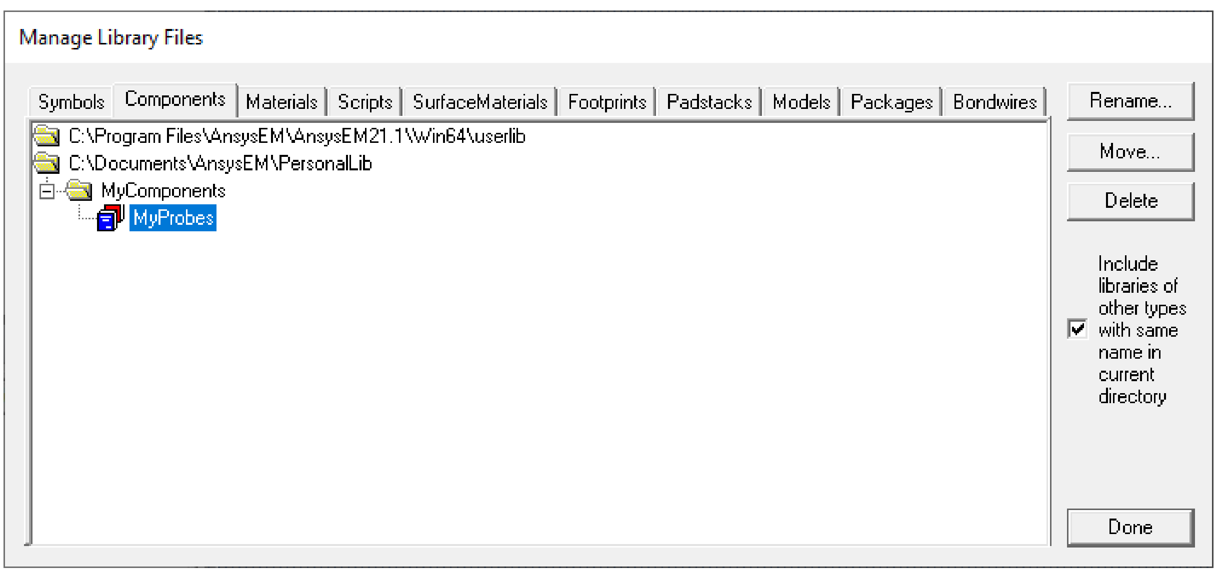
You can use **Manage Files** to change the names of libraries in the UserLib and PersonalLib directories; move libraries to another location in the directories specified for UserLib or PersonalLib, or delete libraries. You can also change the names of sub-directories of the UserLib and PersonalLib directories, move directories to another location in the directories specified for UserLib or PersonalLib, or delete directories.

> **Note:**
>
> - You cannot move libraries out of the UserLib or PersonalLib hierarchies.
> - To move a library or directory to a subfolder, that subfolder must already exist.

Components and other definitions in libraries directly affected by an action which See definitions changing library name or location has those references adjusted. Definitions in libraries indirectly affected– that is, those not having their names or locations modified by an action–with references to definitions in directly affected libraries are updated.

To manage library files:

1. Click **Tools >Library Tools >Manage Files**. The **Manage Library Files** window opens on the Components tab. The **Manage Library Files** window opens on the Components tab. There is a tab for each library type: Symbols, Components, Materials, Scripts, Surface Materials, Footprints, Padstacks, Models, Packages, and Bondwires.



2. Click the tab you want.
3. Select an individual library or a directory on which you want to perform the action.

4. Click **Rename**, **Move**, and **Delete**, as necessary.
5. Select the **Include libraries of other types with same name in current directory** check box to cause the action to be performed on other types with the same name in the same directory. For example, if you select “MyProbes” on the Components tab, the action is extended to a symbol library called “MyProbes.aslb” in the same directory.
6. Click **Done** when finished managing libraries.

Library Search Precedence

When you place a new component in a design, Electronics Desktop searches in the current project for a definition with the corresponding name. If an appropriate definition exists in the project, that definition is used to satisfy the placement *regardless of whether additional instances, even more recent ones, may exist in external libraries.*

If Electronics Desktop cannot locate the required definitions in the project, it searches in personal (**PersonalLib**), user (**userlib**), and system (**syslib**) library files, *in that order*. Once a definition of the correct name has been located, the search ends, and that definition is used to satisfy the placement request.

Component Creation Sequence

Every component definition uses an associated bitmap (for the **Project Manager** window icon) and a schematic symbol. Many components also require footprints, which in turn depend on material definitions and padstacks to characterize their physical presence in and electrical connections to a layout.

To save time in editing or creating new components, you should identify and modify or create its dependent elements in the following sequence as you define your component. All of the dependencies listed may not be necessary for every component you create or modify.

Materials

Material information defines the physical properties of a substance — such as substrate, metallization, or solder mask layer — for its inclusion in a *stackup*, the physical foundation of a manufacturable layout or Planar EM model. Material definitions are stored in library files with the extension **.amat**. See Using the Material Editor for how to create and modify materials.

Layers

The substrate technology defines the name and parameters of the layout stackup to use for all distributed elements in the design. You create or define a substrate type by clicking **Circuit>Add Reference Data>Add Substrate Definition**. You can view and modify the current layout stackup by clicking **Schematic>LayourStackup**.

Padstacks

A padstack defines the physical structure and electrical connectivity of a *pad*, the region where a component pin connects to a copper trace. A padstack must be associated with each pin of a component. Padstack definitions are stored in library files with an extension of **.pslb**. See Using the Padstack Editor for how to create and modify padstacks.

Footprints

A footprint defines the planar space consumption, physical orientation, metal usage, and electrical connectivity of a component in a layout. Footprint definitions are stored in library files with the extension **.aflb**. See Using the Footprint Editor for how to create and modify footprints.

Symbols

A symbol is graphical representation of a component in a schematic. Symbol definitions are stored in library files with the extension **.aslb**. See Using the Symbol Editor for how to create and modify symbols.

Bitmaps

A bitmap is a small picture that is displayed when the ACLB containing the component is opened in the schematic editor. Bitmaps are stored in syslib\Bitmaps*.bmp. To create or modify a bitmap, use a bitmap editor such as MS Paint(r).

Components

The Electronics Desktop **Component Editor** supports the viewing, modification, and saving of information for component elements and graphical primitives. After creating or modifying a component, the information can be saved, used to render a component, or exported. Component definitions are stored in library files with an extension of **.asty**. See Using the Component Editor for how to create and modify components.

Scripts

The Electronics Desktop **Script Editor** supports the viewing, modification, and saving of script file information for footprint elements and graphical primitives. After modifying a script, the information can be saved, used to render a footprint, or exported. Script definitions are stored in library files with an extension of **.dsc**. See Using the Script Editor for how to create and modify scripts.

Using the Library Editor

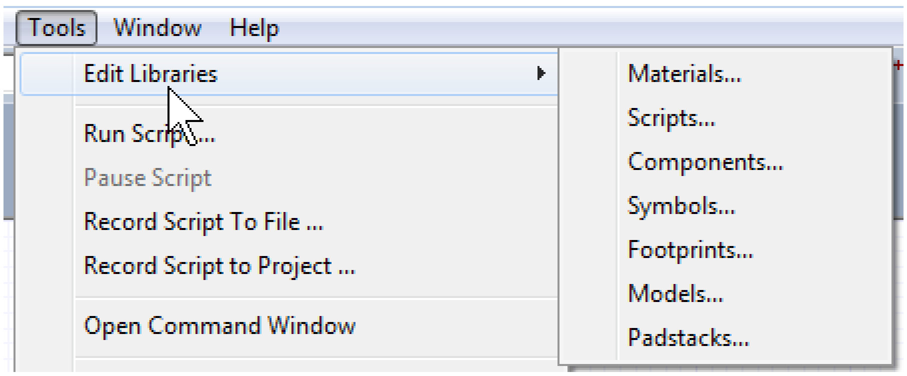
The Electronics Desktop **Library Editor**, also called the **Edit Libraries** window, allows you to access, manage, save, and control the contents of various development libraries which contain information and properties for the following design entities:

- Materials
- Scripts
- Components

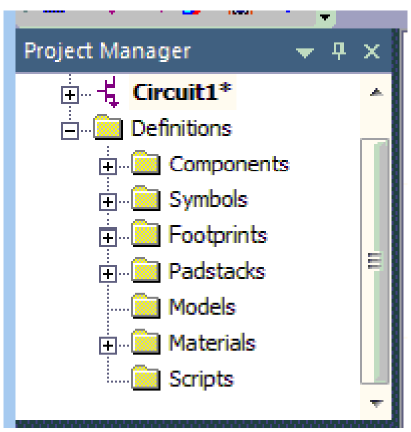
- Symbols
- Footprints
- Models
- Padstacks

Starting the Library Editor

To start the **Library Editor**, select **Tools > Edit Libraries**, then select the type of library you wish to access on the following menu:

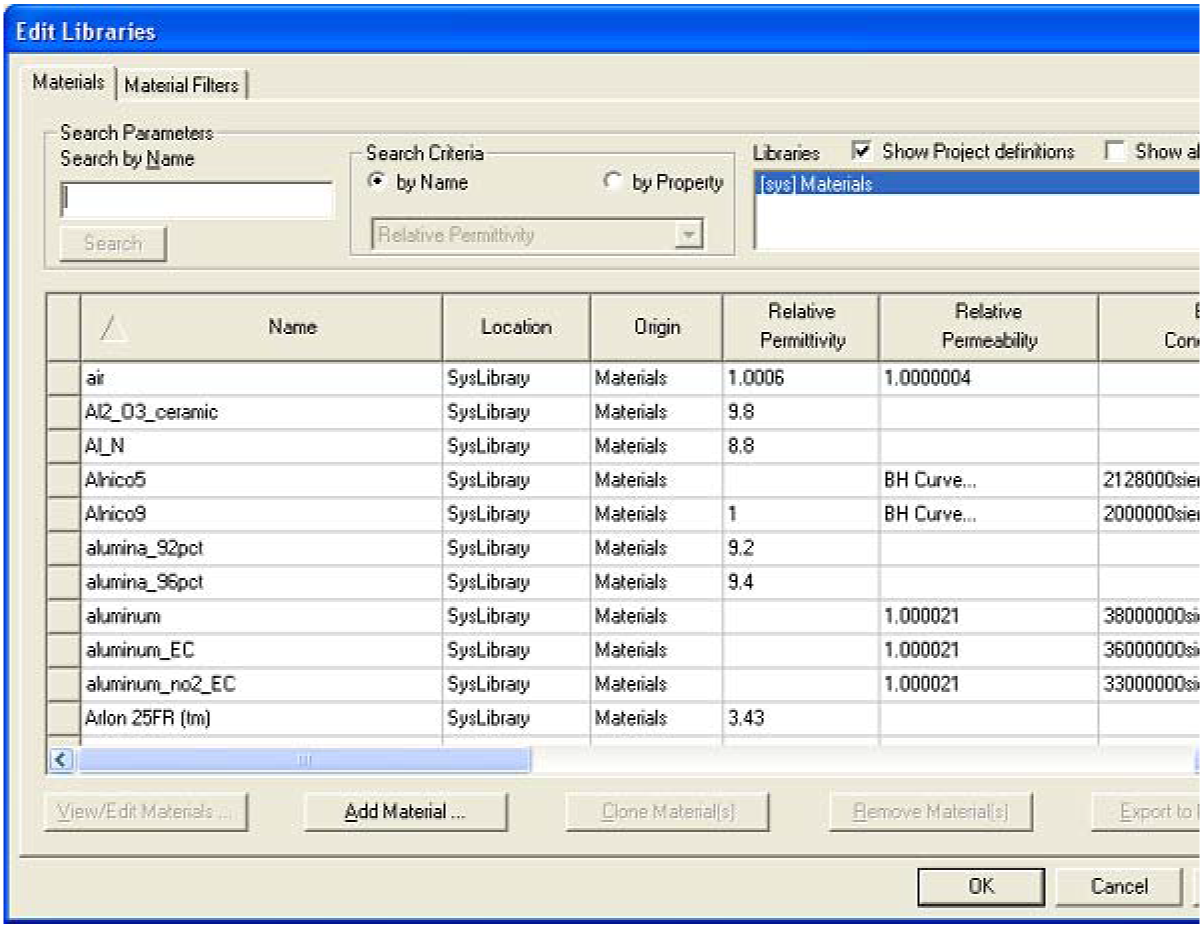


You may also right-click any of the folders listed in the **Definitions** directory of the **Project Tree** in the **Project Manager** window, then select **Edit Library**.



Edit Libraries window

After choosing a library to edit, the **Edit Libraries** window opens with a tab open to the library you selected, such as the **Materials Browser**.



The **Edit Libraries** browser searches library objects using filters based on a number of attributes:

- Name
- Property
- Model type

You can customize the **Edit Libraries** window by clicking **Show Project definitions** and **Show all libraries** (to override filtering).

The following **Edit Libraries** browser controls are available:

- **Edit** – Edits properties/attributes for the selected object. The resulting object can have the same name as the original, but it is saved to the current project, rather than back to the library. In effect, you have “checked out” the part in order to edit it. If you want to write it back to the library and overwrite the original part, use **Export to Library**.
- **Add** – Creates a new object in the selected library.

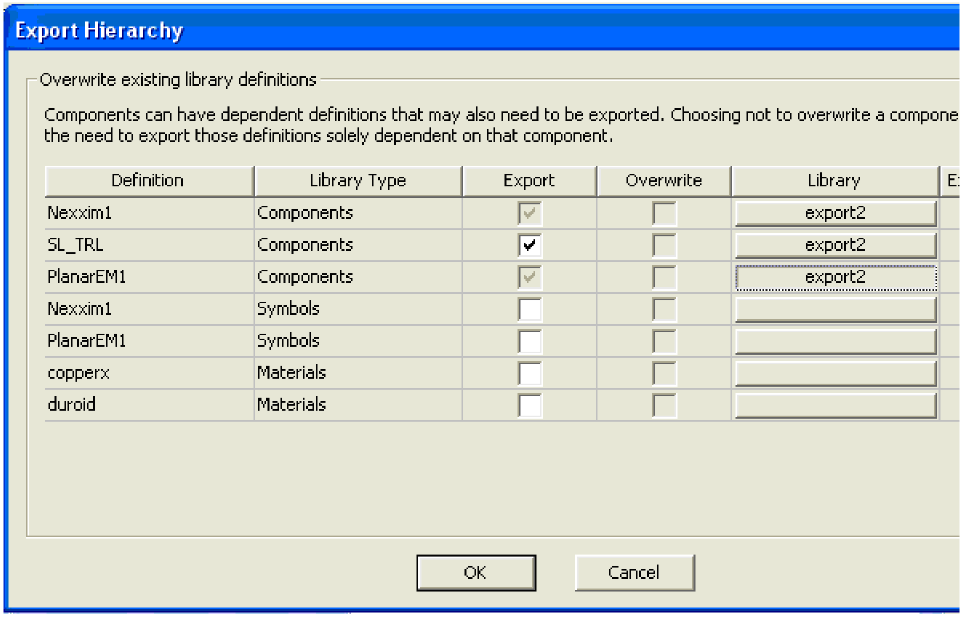
- **Clone** – Creates a copy of a selected object with a different name in the selected library.
- **Remove** – Removes an object on the library
- **Export** – Exports edited objects from a project to a library, or exports selected objects from one library to another. If the component you are exporting is hierarchical, after you click to **Export** and specify a library name in the **Export to user library** window to open the **Export Hierarchy** window. For more information see Exporting Hierarchical Components.

Note:

1. When you modify a library component, the modified definition is automatically transferred to the components available to the current project. In effect, you have “checked out” the component to modify it, and it is now in the project. The original library definition is intact. To update the library definition, you must do the extra step of exporting the modified component on the project to the library.
2. Subcircuits automatically inherit higher-level library configurations.
3. Project variables and project datasets used in designs represented by hierarchical components is saved to the library with the component and restored when the component is used. If the variable or dataset already exists in the project (determined by name), the library version is ignored. When creating project variables or datasets for hierarchical component designs to be saved in a library, ensure the names of the variables and datasets are appropriate and not likely to clash with those of other hierarchical components or common defaults.

Exporting Hierarchical Components

Electronics Desktop allows you to export hierarchical components, along with their solutions. Click **Export** to export a hierarchical component in the Library Editor and specify a library name in the **Export to user library** window to open the **Export Hierarchy** window.



- The component selected to be exported appears in the top-most row of the window. The library to which it is exported is listed beneath the **Library** column (in this example, “export2”). To select a different library to export to, click the button in the **Library** column (in this example, “export2”).
- Dependent definitions appear underneath the top-most row of the window and may also be exported. Definitions that are mandatory for export has an unavailable **Export** check box.
- You can export a component’s solution by checking its **Export** group box. At import time, the solution is re-associated with the design.
- Symbols need not be exported, they is auto-generated at import time.
- Each definition is exported as a definition in its own right. That is to say, at import time, using the above example, "Planar EM1" is visible in the "export2" library as a component independent of "Nexxim1".
- Components that originated in libraries (e.g., the "SL_TRL" component) need not be exported. At import time, each library component is retrieved from its original library and re-associated with the design.

Using the Material Editor

The Electronics Desktop **Material Editor** supports the viewing, modification, and saving of material information for substrate elements and stackup primitives. After modifying a material, the information can then be saved, used to define a design, or exported.

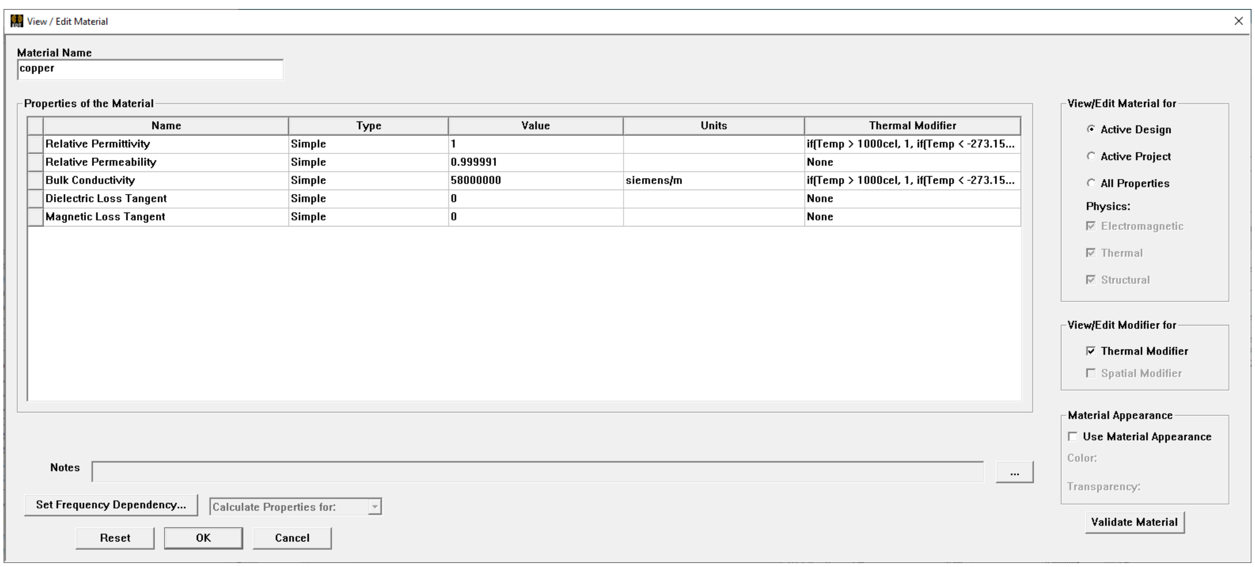
A material is a set of information that defines the physical properties of a substance — such as that used as a substrate, metallization, or solder mask layer — for its inclusion in a *stackup*, the physical foundation of a manufacturable Planar EM model. Material definitions are stored in library files with an extension of **.amat**. You can access material definitions using the **Edit Material** window, also referred to as the **Material Editor**.

To start the **Material Editor**, select **Tools > Edit Libraries > Materials**. When the Edit Libraries window opens, use its controls to open the **Material Editor**.

The View / Edit Material Window

From the **View / Edit Material** window, control the property values of materials. Open the **View/Edit Material** window in one of two ways:

- From **Tools**, select **Edit Libraries** > **Materials...** to open the **Edit Libraries** window. Then search for and double-click the name of the appropriate material, or highlight the name and select **View/Edit Materials**.
- From the **Project Manager** window, expand the project tree > **Definitions** > **Materials** folder. Then right-click the appropriate material and select **Edit Material...**.



Material Name — This box sets the material name. To specify or change the name, click in the box, then type the appropriate name.

Property Headings

Name — Displays the property name. The values in this column are not editable.

Type – Displays and sets the property type (**Simple** [the default] or **Anisotropic**). To change a properties Type setting, click in the **Type** cell and select the appropriate type.

Value — Displays and sets the property value. To change a property value, click the value you want to change, then type a value or parameter name.

Units – Displays and, where applicable, sets the unit that applies to the Value entry. (For example, the unit of magnetic saturation can be set to **Gauss**, **uGauss**, **Tesla**, or **uTesla**.) To change a unit, click it, then select the appropriate unit.

Property Names

Relative Permittivity – This box sets the material relative permittivity. To specify or change the value, click in the box, then type the appropriate value.

Relative Permeability – This box sets the material relative permeability. To specify or change the value, click in the box, then type the appropriate value.

Bulk Conductivity – This box sets the material bulk conductivity. To specify or change the value, click in the box, then type the appropriate value.

Dielectric Loss Tangent — This box sets the material dielectric loss tangent. To specify or change the value, click in the box, then type the appropriate value.

Magnetic Loss Tangen t — This box sets the material magnetic loss tangent. To specify or change the value, click in the box, then type the appropriate value.

Magnetic Saturation – This box sets the material magnetic saturation. To specify or change the value, click in the box, then type the appropriate value.

Lande G Factor – This box sets the material Lande G factor. To specify or change the value, click in the box, then type the appropriate value.

Delta H – This box sets the material delta H. To specify or change the value, click in the box, then type the appropriate value.

View/Edit Material For

The check boxes in this group select the Ansys product(s) for which material properties are shown.

- Active Design
- This Product
- All Products

View/Edit Modifier For

The check boxes in this group select the Ansys product(s) for which modifier properties are shown.

- Thermal Modifier

Validate Material

Click **Validate Material** to validate the current material property values for **Electronics Desktop** product(s) checked. If validation succeeds, a green check mark appears under **Validate Now**. If validation fails, a red X appears instead, and an error message informs you of which parameter value(s) are invalid for which product(s), and why.

Set Frequency Dependency

Click **Set Frequency Dependency** to open a window that allows you to set options for the following:

- Piecewise Linear Input
- Loss Model Input
- Low-Loss Dielectric Model Input
- Enter Frequency Dependent Data Points

After you set options for any of the above, the **Value** column of the **MaterialEditor** indicates that a material's value is set to be frequency dependent, rather than set to a constant.

> **Note:** The **Set Frequency Dependency** option is available only with Planar EM. To add a material with frequency dependence, or to edit the frequency dependence of a pre-existing material, you must first ensure that the active design selected in the **Project Tree** is a Planar EM design, otherwise **Set Frequency Dependency** does not be available. For more information see Defining Frequency-Dependent Material Properties.

Material Editor window Controls

- Click **Reset** to restore all changed properties to their values prior to opening the **Material Editor**.
- Click **Cancel** to close the window without committing changes and return to the **MaterialBrowser**.
- Click **OK** to save changes and return to the **MaterialBrowser**.

For more information see Viewing and Modifying Material Attributes.

> **Note:** If you like the changes you have made in a material to be available for use in other projects, you must export the material to a library as described in the Edit Libraries window topic.

Editing an Existing Material

To edit an existing material, display its properties in the Edit Material window by doing either of the following:

- Open the **Material Browser** by clicking **Tools > Edit Libraries > Materials**, then search for and select the material you want to edit, and either double-click its name or click **View/Edit Materials**.
- In the **Definitions** /**Materials** subfolder of the **Project Tree**, locate the icon for the material you want to edit, then either double-click the icon or right-click it and select **EditMaterial**.

The Edit Material window opens (also called the **Material Editor**).

Creating a New Material

To create a new material, first open the **Material Browser** by clicking **Tools > Edit Libraries > Materials**, then do either of the following:

- Click **Add Material**. The Edit Material window opens (also called the **Material Editor**).
- Locate and select an existing material definition on which to base the new definition, then click **Clone Material(s)**. The selected definition is copied under a new name. To edit the new material, double-click its name to open the Edit Material window (also called the **Material Editor**).

Using the Models Editor

1. To open the window for a model library, click **Tools** > **Edit Libraries** > **Models**. The **Edit Libraries** window opens on the **Models** tab.
2. From the **Models** tab, Click the **Libraries** scrolling window to select the library file for contents display. Use the buttons described to manage selected library objects:
   - **Edit Model** - Edits properties or attributes for a selected model in the corresponding model editor. The resulting model may have the same name as the original, but it is saved to the current project, rather than back to the library. In effect, you have “checked out” the part to the project in order to edit it. If you want to write it back to the library, overwriting the original part, use **Export to Library**.

     (Click the **Show Project Definitions** check box to include current project elements in the **Edit Libraries** window display.)

     Some models - such as those that have been encoded - are not editable (button is unavailable). Encrypted models can be edited after the password has been entered to unlock it.

   - **Add Model** – Creates a new model object in the selected library. The **Add Model** window allows you to name the model, and choose the model type you want to add. By default, the model is added via the appropriate model editor.
   - **Clone Model(s)** – Creates a copy of selected object(s) with a different name in the selected library. Encoded models cannot be cloned (button is unavailable).

- **Remove Model(s)** – Removes selected object(s) on the library. System library models cannot be removed (button is unavailable).
- **Export to Library** – Exports a selected object to a different library. (Use also to export edited objects from the project to the library if necessary.) System library models cannot be exported (button is unavailable).

3. When finished, click **OK** to close the **Edit Libraries** window box.

Using the Padstack Editor

A padstack is a set of information that defines the physical structure and electrical connectivity of a pin or via (the region in which a component pin connects to a layout or a connection between layers). A padstack includes information about the layers involved in making the connection, on the size and shape of the layout area to the type and dimensions of the accompanying hole. Padstack definitions are stored in library files with an extension of **.pslb**.

The Electronics Desktop Padstack Editor supports the viewing, modification, and saving of padstack definition information for physical structure and connectivity. After modifying a padstack definition, the information can then be used to define a via or pin. Padstack definitions can be exported to library files.

To open the **Padstack Editor**, load a project and select **Tools** > **Edit Libraries** > **Padstacks**. When the Edit Libraries window opens, use its controls to open the **Padstack Editor**.

Creating a New Padstack

To create a new padstack, do one of the following:

- Select **Tools** > **Edit Libraries** > **Padstacks**, then click **Add Padstack**.

  The **Definition Name** window opens and allows you to specify the name for the new padstack.

  The **Choose Layout Technology** window opens, and you may browse for a technology file to define the original layers of the definition. If no technology file is chosen, the definition is created with initial layers named **Start, Stop, and Default** .

- Select **Tools** > **Edit Libraries** > **Padstacks**, then locate and select an existing padstack definition on which to base a new definition.

  Search for and select a padstack name and then click **Clone Padstack(s)**.

- From the **Project Manager** window, right-click the **Padstack** folder and select **Edit Library** to open the Padstack tab of the **Edit Libraries** window. Click **Add Padstack** or **Clone Padstack(s)** to open the **Edit Padstack Definition** window.
- In the Project Tree, right-click **Padstacks** and select **Add Definition**.

After a padstack is created, edit the padstack.

Editing a Padstack

Edit and define a padstack in the **Edit Padstack Definition** window. Open this window in one of the following ways:

- Select **Tools** > **Edit Libraries** > **Padstacks**. Find and select the padstack you want to edit, then double-click its name or click **Edit Padstack**.
- In the **Definitions** directory of the **Project Manager** window, right-click the **Padstack** folder and select **Edit Library**. Find and select the padstack you want to edit, then double-click its name or click **Edit Padstack**.
- Select a via or pin that uses the padstack. In the **Properties** window, click **Padstack Definition** to open the **Padstack Usage and Definition** window. When opened on the **Properties** window, information related to the padstack definition is highlighted in blue.

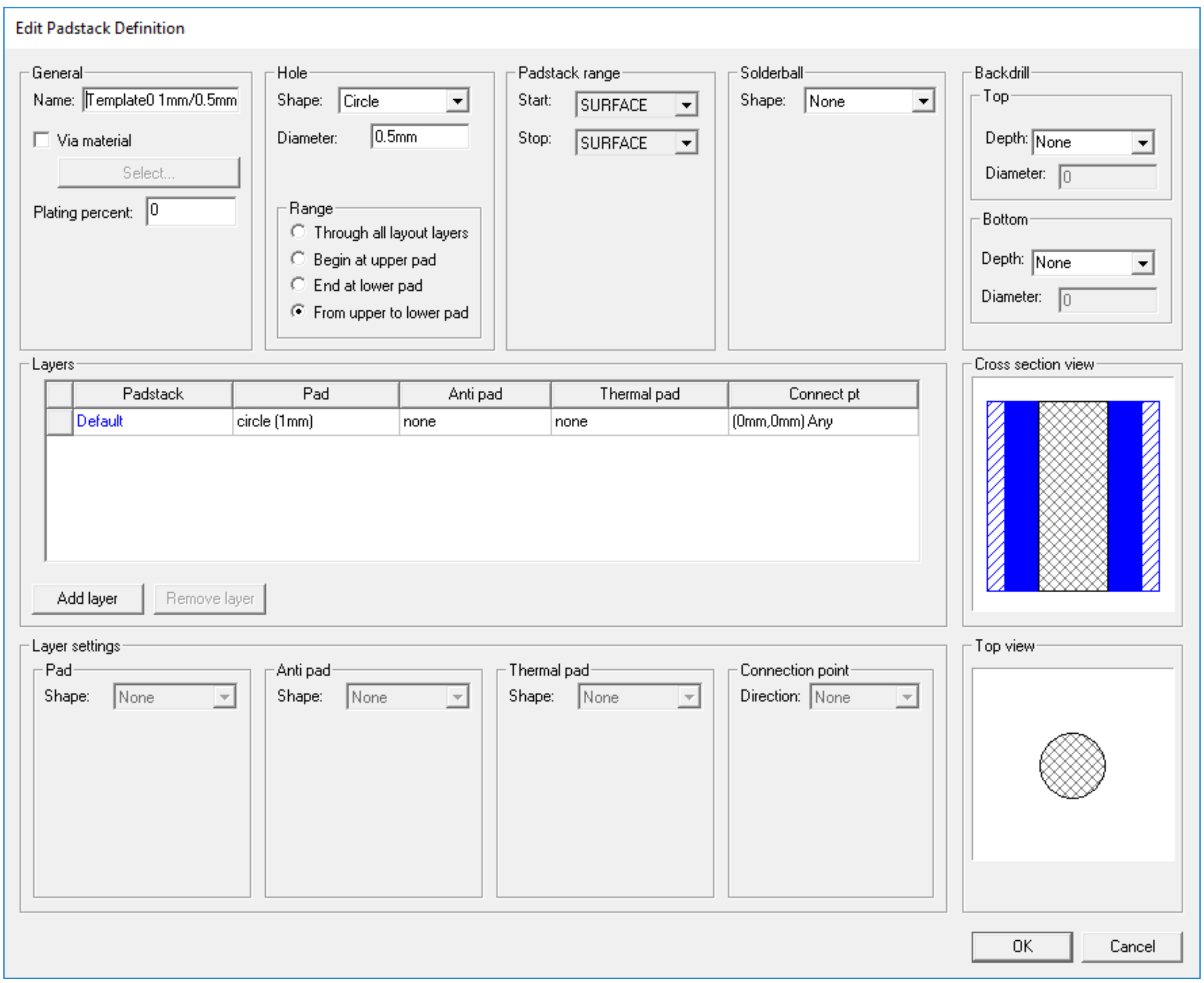


Figure 21-1 The Edit Padstack Definition window

> **Important:** Some options may be unavailable or may not appear, depending upon the context and which padstack is being edited.

Options include:

General

- **Name** – the name of the padstack.
- **Via Material** – the material for plating the via or pin hole. Click **Select** to choose from available materials.
- **Plating Percent** – the percent of the hole's radius that is filled by the plating material. At 0%, there is no plating. At 100%, plating fills the hole. The **Cross section view** and **Top view** areas show the vertical cross-section and top of the layer stackup. This setting is used by SIwave Solvers and Icepak **Electronics Desktop**.

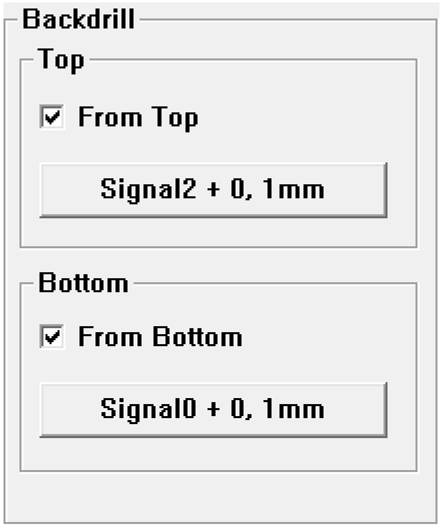
Hole

- **Shape** – choices are None, Circle, Square, Rectangle and Polygon. The Polygon choice is currently only supported for padstacks; polygon holes may not be defined through the window. Polygons can be modified by selecting a layer and clicking **Edit Polygon** to open the **Polygon Editor** . Selecting a shape enables additional options:
  - **Size / Diameter / XY Size** – specifies the size of the hole.
  - **Range** – can specify the hole range to be:
    - Through all the layout stackup layers
    - Beginning at the upper pad of the padstack and continuing to the lowest elevation of the layout stackup layers
    - Beginning at the highest elevation of the layout stackup layers and ending at the lowest pad of the padstack
    - Beginning at the upper pad and ending at the lowest pad of the padstack

Padstack Range

This area contains controls to specify the **Start** and **Stop** layers for the padstack.

Backdrill



Backdrilling creates empty holes on one or both sides of a via. From the **Backdrill** group box, define backdrills from a selected via's top layer, bottom layer, or both. Refer to Backdrilling. Backdrills are rendered in the **Cross section view** and **Top view** group boxes as empty space. In the following script example, the backdrill is recorded as a block:

```
Array("NAME:Backdrill Top Diameter", "MustBeInt:=", false, "Value:=",
"0.4mm").
```

- **From Top** – define a backdrill that starts on the selected via's top layer. Check the adjacent box to activate or remove the check to deactivate. Click the associated button to proceed to the **Backdrill Definition** window.
- **From Bottom** – define a backdrill that starts on the selected via's bottom layer. Check the adjacent box to activate or remove the check to deactivate. Click the associated button to proceed to the **Backdrill Definition** window.

  Refer to Define Backdrill Parameters.

Solderball

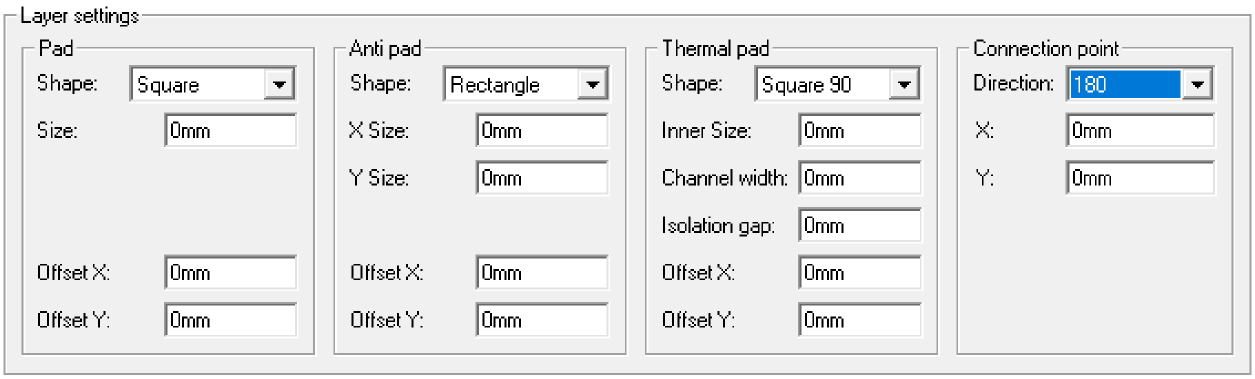
- **Shape** – specifies the shape of the solderball; choices are none, cylinder, and spheroid.

- **Diameter** – specifies the diameter of the solderball.
- **Mid diameter** – for spheroid solderballs, specifies the mid diameter.
- **Solder** – click to open the **Material Editor** for the solderball in order to specify the material used.
- **Connection** – select to specify the location of the solderball either above or below the padstack hole.

Layers

Layer names are used when a padstack definition is used in a via or pin. A default mapping of padstack definition layers to layout layers occurs based on the layer names.

- None of the **Layers** group box grid control cells are directly editable.
- When one or more rows in the **Layers** group box is selected, the **Layer settings** group box is enabled and can be used to change the pad settings for selected rows.



- **Add Layer** inserts a new layer above the selected row.
- **Remove Layer** removes the selected row(s). You may also rearrange rows to be any order.

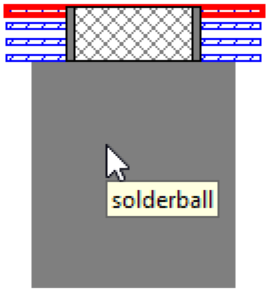
Layer Settings

The **Layer settings** area displays pad settings for padstack definition layers. The Polygon choice is currently only supported for imported padstacks; polygon pads, anti pads, and thermal pads may not be defined through the window. Dimension controls appear to support the shape choice for each pad. Controls to specify the pad offset are also available.

- **Pad** and **Antipad** shape choices are None, Circle, Square, Rectangle, Oval, Bullet, N-Sided Polygon, and Polygon. Polygons can be modified by selecting a layer and clicking **Edit Polygon** to open the **Polygon Editor** .
- **Thermalpad** shape choices are None, Polygon, Round45, Round90, Square45, and Square90. Polygons can be modified by selecting a layer and clicking **Edit Polygon** to open the **Polygon Editor** .
- **Connection point** allows you to choose a direction in degrees or specify None. There are also controls to specify the X,Y location of the connection point.

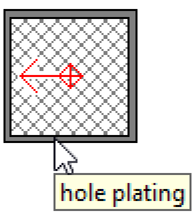
Cross Section View

This area displays all definition layers and their pads from a side view. Selected layers are highlighted. Hover the mouse over this area to see tooltips indicating padstack components.



Top View

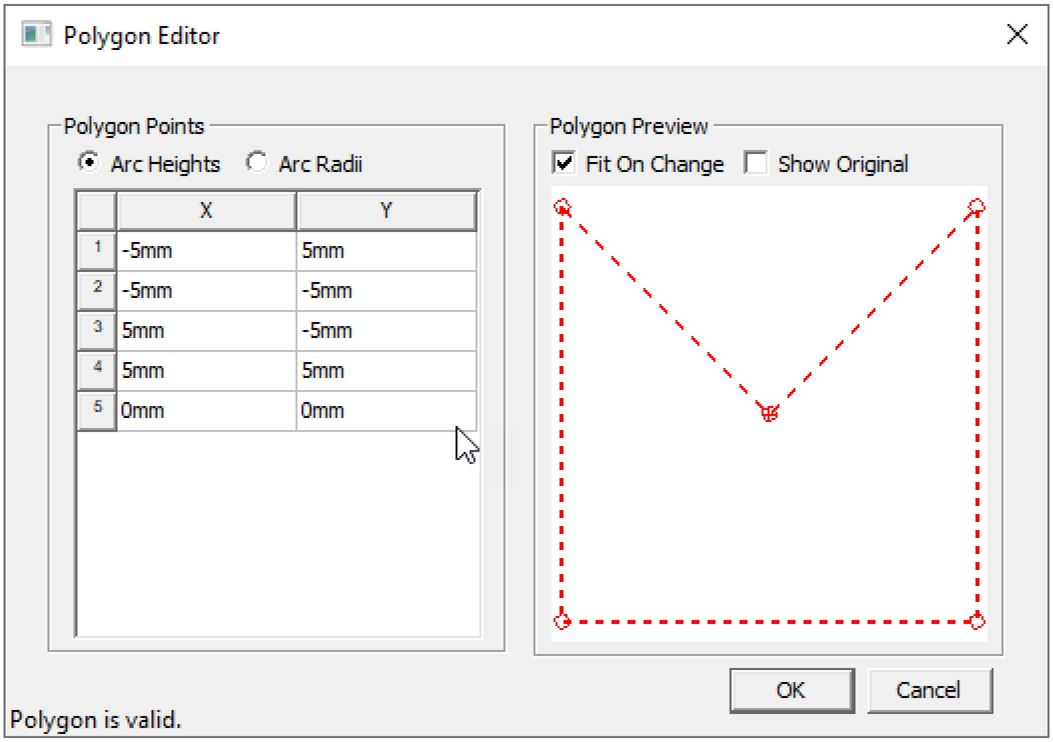
This area displays the pads of selected layers from a top down perspective. Hover the mouse over this area to see tooltips indicating padstack components.



Editing a Polygon Pad

Pads, antipads, thermal pads, and holes can have polygon shapes which can be modified in the **Polygon Editor**. This editor lists the points of the polygon in a grid and displays a diagram of the polygon. The **Polygon Editor** is available on the **Edit Padstack Definition** and the **Padstack**

Usage and Definition windows.



Each row in the grid defines a vertex or an arc segment. Vertices have **X** and **Y** coordinates that are located in relation to the center point of the pad, antipad, thermal pad, or hole. The center point is marked by a plus sign (**+**) on the diagram. Enter a distance on the center point for each coordinate. Select a row to find its corresponding point in the diagram.

Arc segments are curved segments defined by three points: a beginning, end, and apex. Each arc segment must have a vertex before and after it to define its beginning and end. Thus, arc segments cannot be adjacent to one another. Each arc segments has an arc height or arc radius which can be changed in the grid.

The grid has a shortcut menu where you perform the following tasks. Different options appear depending on whether you select a row and whether you Right-click a vertex or arc segment.

- **Append Vertex** adds a new row for a vertex to the grid with the coordinates 0 mm, 0mm.
- **Insert Vertex Before** inserts a point before the current selection.
- **Insert Vertex After** inserts a point after the current selection.
- **Remove Vertex/Segment** removes the row you Right-clicked on.
- **Convert to Arc Segment** converts a vertex to an arc segment.
- **Insert Arc Segment Before** inserts an arc segment before the current selection.
- **Insert Arc Segment After** inserts an arc segment after the current selection.

The diagram displays a rendering of the polygon and the object's center point. Hover over a point to see its coordinates. Click a point to select it in the grid. Right-click the diagram to zoom in or out or to resize it to show the entire polygon. The diagram offers two options:

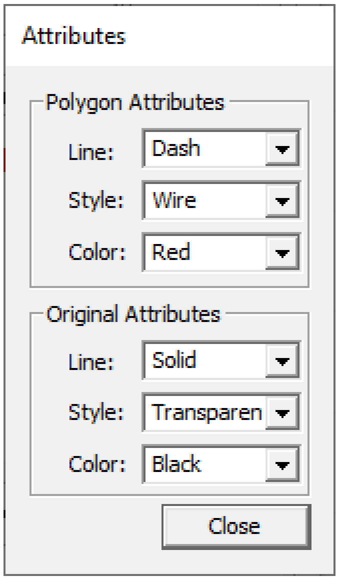
- **Fit On Change** redraws the diagram to fit the polygon whenever you make a change.
- **Show Original** displays an outline of the original polygon to compare it with any changes.

Edit the Polygon

1. Add new vertices as necessary.
   a. Highlight the row of the point just before where you want to add a vertex.
   b. Right-click and choose **Insert Vertex After**. A new row appears in the grid with the coordinates 0 mm, 0mm.
   c. Enter coordinates in the **X** and **Y** columns of the new row.
2. Add new arc segments as necessary.
   a. Highlight the row of the point just before where you want to add an arc segment.
   b. Right-click and choose **Insert Arc Segment After**. A new row appears in the grid.
   c. Above the grid choose whether you want to work with **Arc Heights** or **Arc Radii**.
   d. Enter the arc height or arc radius of the new segment in the new row.
3. Remove unnecessary vertices or segments.
   a. Highlight the row of the unwanted vertex or segment.
   b. Right-click and click **Remove Vertex/Segment**.
4. Check your work in the diagram.
5. Click **OK** to close the **Polygon Editor**. The polygon appears in the **Top view** diagram of the **Edit Padstack Definition** window. The **Cross section view** diagram is unchanged.

Change how the diagram looks

Right-click the diagram and click **Show/hide attributes** to open an **Attributes** window. Here customize the lines, fill, and color of the polygon. Controls are available for both the edited polygon and the original polygon if **Show Original** is enabled.

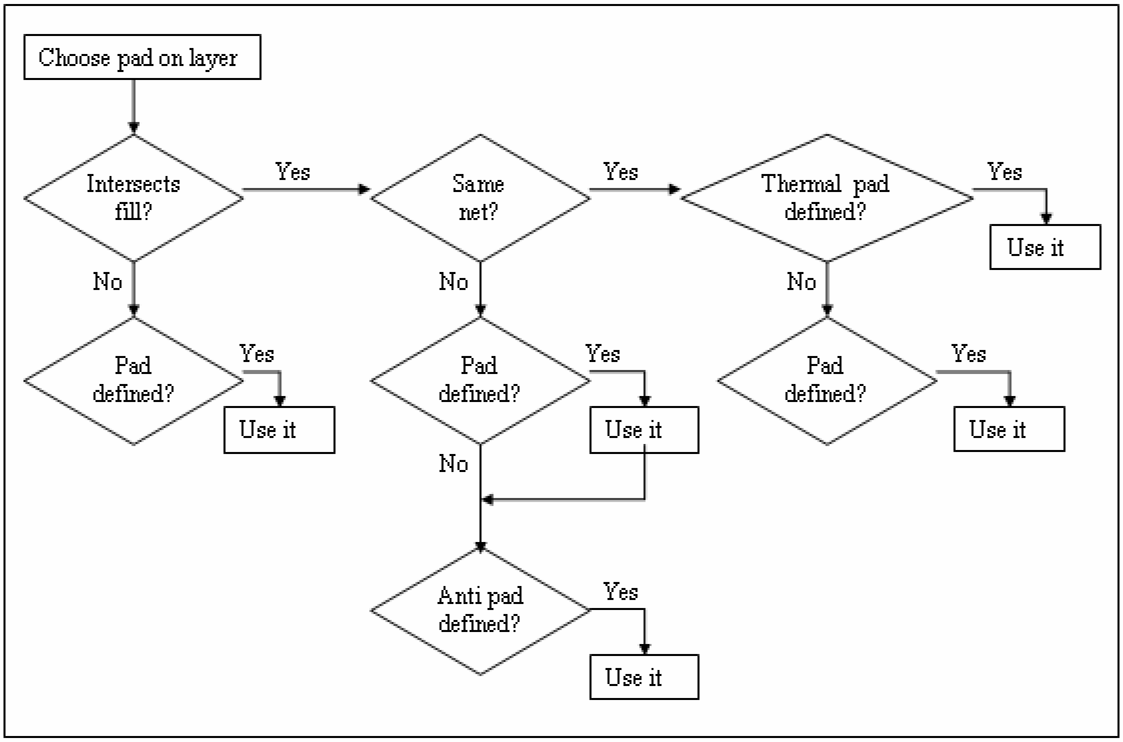


Pad Behavior in Vias and Pins

The pad(s) used in a particular via or pin are determined by the padstack definition and by the intersected primitives in their nets. Multiple pads are sometimes defined for the same layer.

- The intersection point between the central axis of a via or pin and the intersected layers is inspected to determine its metal type (fill, trace, or un-present). By definition, ground layers are fill-metal with primitives defining scratch areas.
- If a fill is intersected, the net of the fill is also taken into consideration.

- Based on the intersection of the fill/net/pads that are defined in the padstack definition, the choice of pad that is used is determined using the following algorithm:

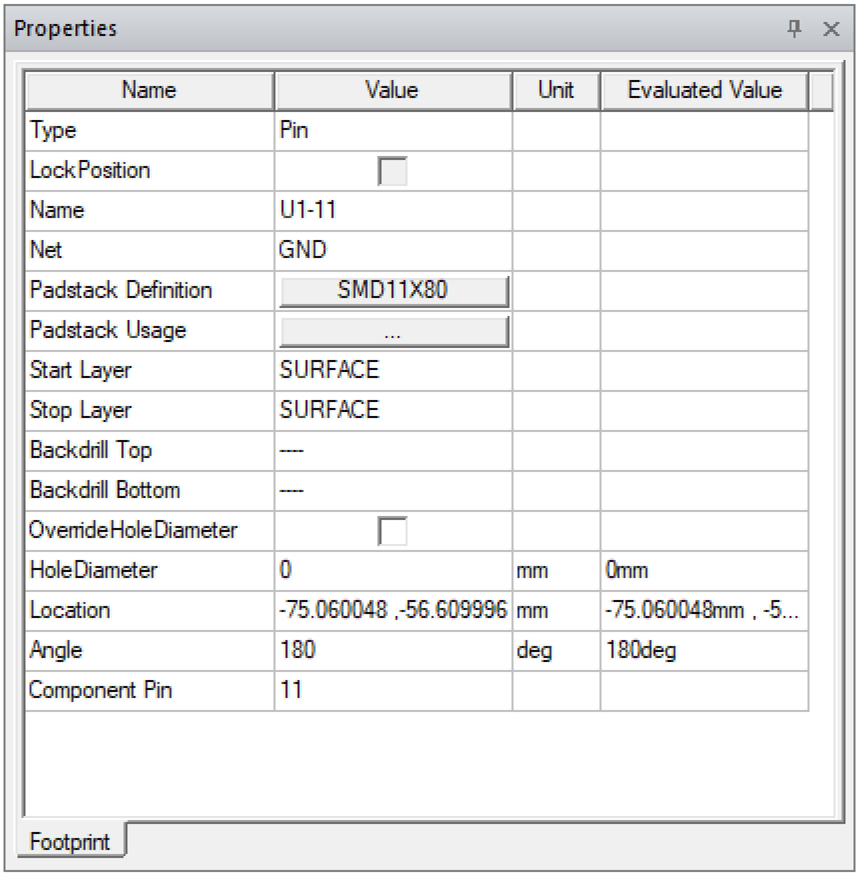


- The pad choice on the algorithm is subject to suppressing of non-functional internal pads as described in Layout Editor Options: Object Panel.

> **Note:**
>
> Anti pads subtract material on the fill area.

Editing Via and Pin Padstacks in Layout

To edit a via or pin padstack, click the via or pin to display its properties in the **Properties** window.



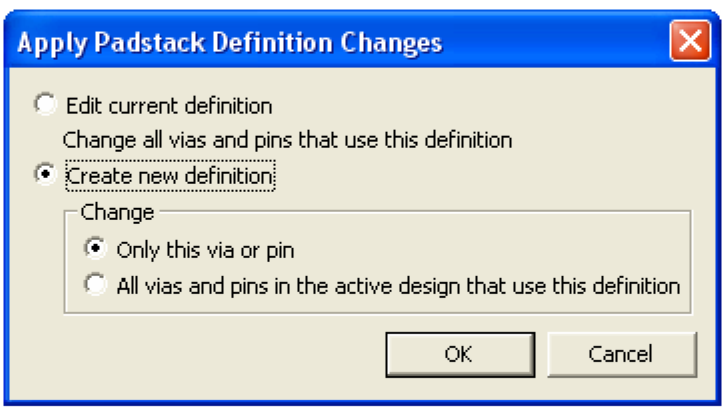
- **Name** is the name of the object.
- **Net** lists the net to which the object is assigned.
- **Override Hole Diameter** and **Hole Diameter** are used to override the diameter of the padstack definition for this particular via or pin.
- **Padstack Definition** opens the Edit Libraries window, which allows you to switch the padstack definition that is used by the via or pin.
- **Padstack Usage** opens the **Padstack Usage and Definition** window, to edit the padstack. When opened on the Properties window, information related to the padstack definition is highlighted in blue.
- **Type** determines whether the object definition is a via or pin.

Initial Padstack Definition Layers

Special padstack definition layer names (**Start**, **Stop**, and **Default**) are provided to allow you some control over default mapping of the definition layers to stackup layers when the padstack is placed into a layout as a via or pin. Using these special layer names is optional. When a via or pin is placed, a default mapping occurs in the following order:

1. Any layout stackup signal layer before the start layer of the padstack or after the stop layer of the padstack does not get mapped to any definition layer.
2. Any layout stackup signal layer whose name matches a definition layer name is mapped to that definition layer.
3. If the start layer of the padstack is not yet mapped, and there is a definition layer named "Start", they are mapped.
4. If the stop layer of the padstack is not yet mapped, and there is a definition layer named "Stop", they are mapped.
5. If there is a definition layer named "Default", any remaining layout stackup signal layers are mapped to it.
6. If there are remaining layout stackup signal layers, they are mapped (in order) to the remaining definition layers.
7. Any layout stackup signal layers that are unmapped, remain that way.

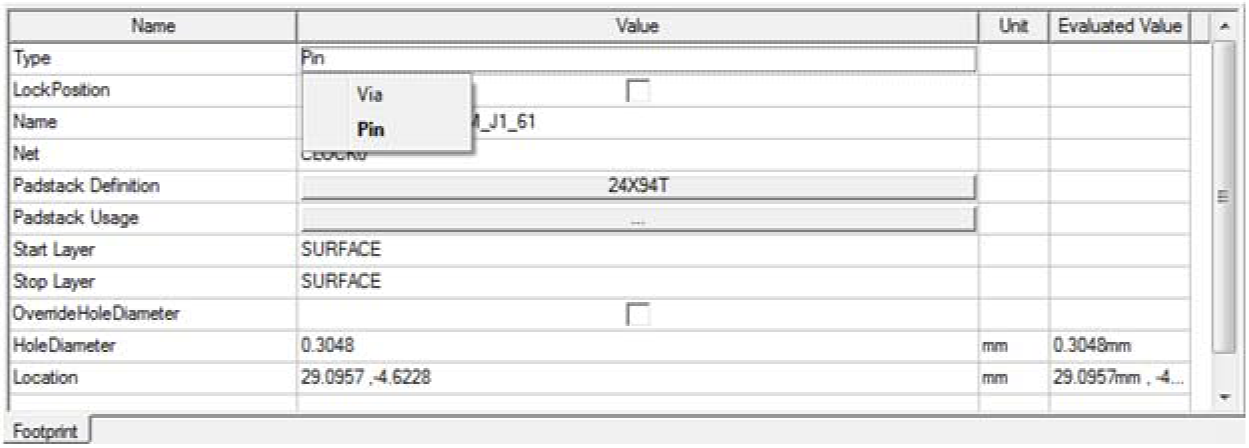
When the **Padstack Usage and Definition** window is closed by clicking **OK**, you are given a choice of how to apply the definition changes that are made; the **Apply Padstack Definition Changes** window opens only if changes are made to the padstack definition (not all changes to the **Padstack Usage and Definition** window change the padstack definition).



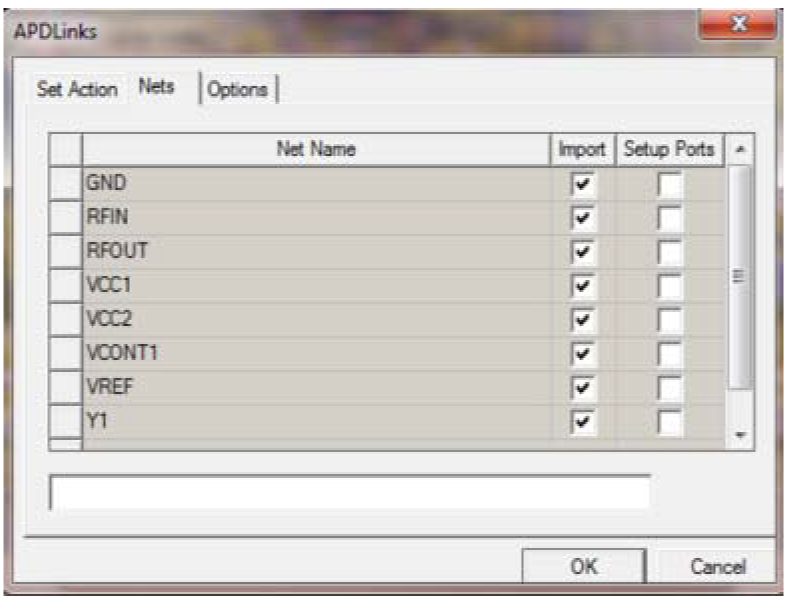
- **Edit current definition** — changes all vias and pins that use this definition
- **Create new definition** – allows you select whether the changes apply to **Only this via or pin** or whether the changes apply to **All vias and pins in the active design that use this definition** of the original padstack.

Pin and Net Padstack Operations

Padstack instances are pins or vias; either type may have an associated port. The "pin" designation does not imply that a port must be associated with the padstack. When a port is removed from a pin, the object retains its "pin" type. The type is changed through the "Type" property:



Both APDLinks and ANFV2 imports correctly retain the "pin" and "via" type for padstacks. For APDLinks, only those nets for which "Setup Port" has been selected, has associate ports with the pins.



The following commands act on pins and nets. These are accessed through the shortcut menu on the Layout tab under "Nets" and also on the right-click layout menu. See Pin and Net Padstack Operations for information about the following scripts.

AddPortsToNet

Use:

— Add ports to all the pins on the designated nets.

Command:

— In Layout, **Right-click > Port > Create Ports on Net**

— Layout tab under Nets, **Right-click > Create Ports**

RemovePortsFromNet

Use:

— Removes ports from all the pins on the designated nets.

Command:

— In Layout, **Right-click > Port > Remove Ports from Net**

— Layout tab under Nets, **Right-click > Remove Ports**

AddPortsToAllNets

Use:

— Adds ports to all the pins in all the nets.

Command:

— Layout tab under Nets, **Right-click > Create Ports**

RemovePortsFromAllNets

Use:

— Removes ports from all the pins in all the nets.

Command:

— Layout tab under Nets, **Right-click > Remove Ports**

Using the Footprint Editor

The Desktop **Footprint Editor** supports the viewing, modification, and saving of footprint information for components and layout primitives. After modifying a footprint, the information can be saved, used to define a design, or exported.

A footprint is a set of information that defines the planar space consumption, physical orientation, metal usage, and electrical connectivity of a component. Footprint definitions are stored in library files with an extension of **.aflb**. You can access material definitions using the **Footprint Editor**.

Creating or editing a footprint involves using the footprint editor to:

- Draw graphical primitives, such as rectangles, circles, and arcs, using options on the Draw menu
- Add pins using the Pin option on the Draw menu
- Add text labels using the Text option on the Draw menu
- Update the current project with the new or revised footprint definition using the **Update Project** option on the **Footprint** menu

You can export a footprint to a footprint library (**.aflb**) file for use in other projects. For information on how to do this, see the Edit Libraries window topic.

Opening the Footprint Editor

You can create a new footprint definition or edit an existing footprint.

Creating a New Footprint

To create a new footprint, do one of the following:

- Open the **Tools > Edit Libraries > Footprints** window, then click **Add Footprint**.

  — The **Definition Name** window opens and allows you to specify the name for the new footprint.

  — Type a name for the footprint into the **Enter the name for this new Footprint** group box, then click **OK** or press **Enter**. The **Choose Technology** window opens and allows you to specify the original layers of the definition

  — Double-click the entry for the new footprint, or click **Edit Footprint**.

- Select **Add Definition** on the right-click menu in the **Footprints** folder of the **Project Manager** window.

  — The **Definition Name** window opens and allows you to specify the name for the new footprint.

  — Type a name for the footprint into the **Enter the name for this new Footprint** group box, then click **OK** or press **Enter**. The **Choose Technology** window opens and allows you to specify the original layers of the definition

- Open the **Tools > Edit Libraries > Footprints** window, then locate and select an existing footprint on which to base your new footprint.

  — Click **Clone Footprint(s)** to open the **Get Name** window.

  — Type a name for the new footprint into the **Enter the name for this new Footprint** group box, then click **OK** or press **Enter**.

  — Double-click the entry for the new footprint, or click **Edit Footprint** to open the **Footprint Editor**.

Editing an Existing Footprint

To edit an existing footprint, do one of the following:

- From the **Project Manager** window, expand the **Definitions** /**Footprints** subfolder for the project that contains the footprint you want to edit. Double-click the entry for the footprint you want to edit, or right-click the entry and select **Edit Footprint**.
- Open the **Tools > Edit Libraries > Footprints** window, then locate and select the Project version of the footprint that you want to edit. Click **Edit Footprint**, or double-click the selected entry.
- Select the footprint you wish to edit and click **Edit Footprint** in the shortcut menu.

The footprint editor runs and opens the selected footprint for editing.

Load a Footprint into the Current Project

To load a footprint into the current project:

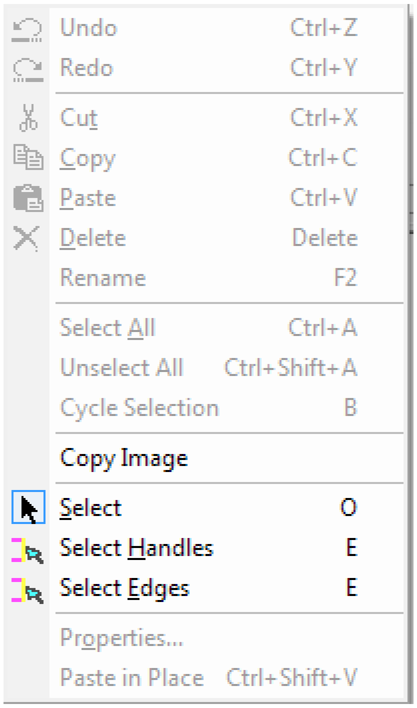
1. With a new local project open, select **Tools > Edit Libraries > Footprints**.
   This opens the Edit Libraries window.
2. Select a component in the Name column, for instance, BENDA, and click **Edit Footprint**.
   This loads the component into the local project.

Footprint Editor Operations

The footprint editor opens when a footprint is created or modified. Like the schematic and symbol editors, the footprint editor is a graphical tool that keeps the placement of footprint elements uniform by snapping them to points on a grid.

With the footprint editor, add and manipulate graphical primitives, text labels, pins, vias, and padstacks, and/or adjust the resolution, color, and visibility of the editor grid. Once changes are made, update the current project with any changes to the footprint. Options related to these operations are available on the **Footprint** menu and the Footprint Draw menu.

Footprint Edit Menu



The fields on this menu become active as appropriate to the type of footprint being edited.

Undo<*last edit operation*> — Undo the last edit operation.

Redo<*last edit operation*> — Redo the last edit operation.

Cut – Delete the selected object, and retain a copy for pasting into a layout in the same application.

Copy – Create a local copy for pasting into a layout in the same application.

Paste – Put the object on the last cut or copy into the layout.

Delete – Delete the selected object without retaining a copy.

Rename – Click to rename the selected object in the layout.

Select All – Select all objects in the layout.

Unselect All – Unselect all objects in the layout.

Cycle Selection – Cycle selection of two or more objects that overlap. Each time you click **Cycle Selection**, a different one of the overlapping objects is selected.

Copy Image — Create a global copy of the selected objects on the clipboard for pasting into a different application.

Select – Sets cursor mode to selection of objects.

Select Handles – Sets cursor mode to selection of object handles.

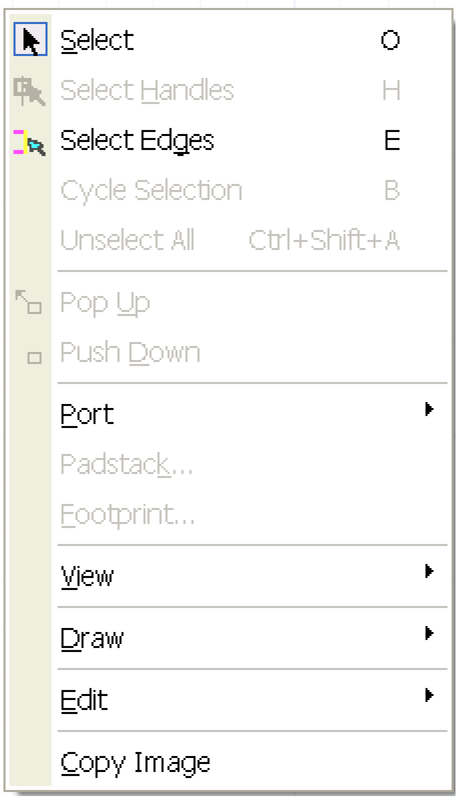
Select Edges – Sets cursor mode to selection of object edges.

Properties – Opens the **Properties** window to view and edit the properties of the selected object.

Paste in Place – Paste object to same location, without any X,Y offset/displacement.

Footprint shortcut menu

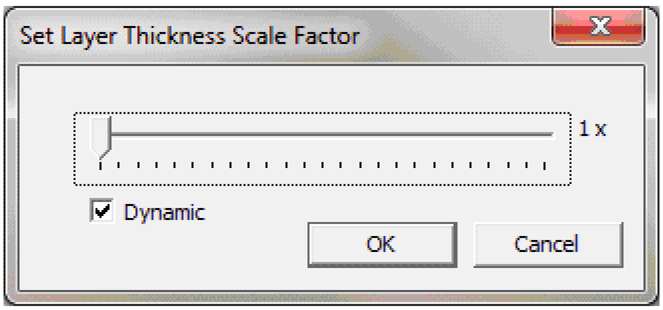
When a footprint design is active, right-click to open the shortcut menu for the Footprint Editor, similar to the following:



The fields on this menu become active as appropriate to the type of footprint being edited.

Footprint View Menu

- Control the display by selecting the check box of the following: **Status Bar**, **Message Manager, Project Manager window, Properties Window, Progress Window, Component Manager, Layout Window.**
- **Variables**: View the project and design variables with their values in a grid view.
- **Rotate**: **click+drag** to rotate the view; the same double-click options hold true for the alt+drag editor display options.
- Change the zoom display by selecting one of the following: **Zoom In, Zoom Out, Zoom Area, Zoom Previous.**
- Fit or Pan by selecting: **Fit Drawing, Pan.**
- Select **Stretch Z** to open the **Set Layer Thickness Scale Factor** window.



- Dragging the slider applies a Z scaling factor to the current view. If **Dynamic** is selected, the scaling happens in real time otherwise it is applied when **OK** is chosen.

Footprint Draw Menu

The footprint editor **Draw** menu presents options for drawing graphical primitives and text, and manipulating selected objects. The **Draw** menu options include:

Primitive > Arc

This option, also available on the toolbar, initiates creation of an arc. To draw an arc, click the footprint editor grid at the two points that determines the arc's ends, and then drag the arc to the appropriate radius.

Once you have created an arc, adjust its radius or move its endpoints by clicking the arc to select it, and dragging the appropriate handle.

Primitive > Circle

This option, also available on the **Footprint Draw** toolbar, initiates creation of a circle. To draw a circle, click the footprint editor grid to select a center point, then drag the circle to the appropriate diameter.

- Click the circle to select it, then drag one of its handles.
- Double-click the circle to open its **Properties** window, type a new value for the **Radius** parameter, and click **OK**.

Primitive > Line

This option, also available on the **Footprint Draw** toolbar, initiates creation of a polyline with one or more segments. To draw a line:

2. To complete the line, do one of the following after defining its final segment:
    - Press the **Spacebar**.
    - Right-click and click **Finish**.

After you have completed a line, change the endpoints of its segments as follows:

- Select the line and drag the appropriate handle(s).
- Double-click the line to open its **Properties** window, type new values for the parameters you want to change, and click **OK**.

Primitive > Rectangle

This option, also available on the **Footprint Draw** toolbar, initiates creation of a rectangle. To create a rectangle, click the editor grid to specify the position of one corner, then drag the rectangle to the appropriate size.

Once you have drawn a rectangle, Edit its height and width, the position of its center, and its angle (its rotation, in degrees, relative the handles of its bounding box) as follows:

- Double-click the rectangle, then edit its properties in the **Properties** window.

Primitive > Polygon

This option, also available on the **Footprint Draw** toolbar, initiates creation of a polygon. To create a polygon:

1. Click the footprint editor grid to specify the position of one vertex, then click wherever you want to place additional vertices.
2. To complete a polygon, specify the position of its final vertex, then do either of the following:

Once you have drawn a rectangle, Edit its vertex positions and other properties, including its fill style, as follows:

- Click the rectangle, then edit its properties in the **Properties** window.
- Double-click the rectangle, then edit its properties in the **Properties** window.

Primitive > Text

This option, also available on the **Footprint Draw** toolbar, adds to the footprint an editable “Default text” label in 12-point Arial. To edit the default text string immediately after placement, type the text you want, then press **Enter** or click elsewhere in the editor grid.

To change just the text of an existing label:

1. Click the label.
2. Click the label again to open its text for editing.
4. Press **Enter** or click elsewhere in the editor grid.

To change other properties of a label, including its font and size:

1. Click the label and view its properties in the **Properties** window, or double-click the label and view its properties in the **Properties** window.
2. Click in the **Value** cell for the property you want to modify.
3. Modify the value.
4. Click **OK**, or click in another **Value** cell to commit the change and keep editing values.

Void > Circle

Create a void, or remove material from an object, in the shape of a circle, rectangle, polygon, or line using the Void > Circle command. One object can have multiple voids, but each void can only be associated with one “parent” object.

Void > Line

Create a void, or remove material from an object, in the shape of a circle, rectangle, polygon, or line using the Void > Line command. One object can have multiple voids, but each void can only be associated with one “parent” object.

Void > Rectangle

Create a void, or remove material from an object, in the shape of a circle, rectangle, polygon, or line using the Void > Rectangle command. One object can have multiple voids, but each void can only be associated with one “parent” object.

Void > Polygon

Create a void, or remove material from an object, in the shape of a circle, rectangle, polygon, or line using the Void > Polygon command. One object can have multiple voids, but each void can only be associated with one “parent” object.

Port > Create

Click the Port > Create command to create a port.

Port > Remove

Click the Port > Remove command to remove a port.

Pin

This option, also available on the **Footprint Draw** toolbar, initiates placement of a footprint pin with a default stem length of 10. You can rotate a pin once you've started to place it by iteratively pressing R until the pin is oriented to your liking. Click in the editor grid to finish placing the pin.

Via

This option, also available on the **Footprint Draw** toolbar, initiates placement of a through hole (via) of the default size (1mm).

Handle

This option, also available on the **Footprint Draw** toolbar, allows you to create an explicit footprint handle that can later be used to resize or reshape the footprint based on the handle's reshaping rules.

3D Structure

Opens a menu that allows you to create and perform operations on a 3D structure in the Drawing Region.

Coordinate System

Opens a menu that allows you to create and perform operations on a coordinate system in the Drawing Region manage. For more information see Coordinate Systems in the **Layout Editor** topic.

Rotate

This option, also available on the **Footprint Draw** menu or by pressing **Ctrl**+**R** on the keyboard, rotates a selected object or group of objects 90° to the left.

Flip Vertical

This option, also available on the **Footprint Draw** menu, flips a selected object about the X axis.

Flip Horizontal

This option, also available on the **Footprint Draw** menu, flips a selected object about the Y axis.

Reverse Line

Reverse selected objects in the Drawing Region.

Align...

Position Relative

When a relative coordinate system (CS) and an object independent of that CS are selected, the “Draw > Position Relative” command positions the CS relative to that object.

Clear Position Relative

When a relative coordinate system (CS) and an object are selected, the “Draw > Clear Position Relative” command positions the CS unrelative to that object.

Duplicate

Duplicate a selected object in the Drawing Region.

Expand

Expand selected objects in the Drawing Region.

Split Polygon Region

Split selected polygon regions in the Drawing Region.

Geometry Healing

Geometry heal selected objects in the Drawing Region.

Stitch Lines

Stitch together connected/crossing lines into polygons and lines.

Defining Footprint Handles

Handles appear around the perimeter of a selected element as small squares in the selection color. If the cursor is moved over one of these handles, it changes to indicate its proximity to the handle. When clicking on a handle, the user can then resize or reshape the element based on the handle's reshaping rules.

Component footprints can also have handles which can be specified in the footprint editor. These handles can likewise be used to reshape a component footprint, and at the same time altering one or more of its electrical parameters. In other words, the user can use handles to directly manipulate an element's geometry corresponding to electrical parameters.

Defining a Handle

Open the footprint editor for the footprint appropriate. From the **Draw** menu, choose **Handle**. Next, click to place the handle anywhere in the drawing, exact placement does not matter.

Editing Handle Location

Next, open the property window (**Edit > Properties**). There is one property listed: location. For this property, edit the X & Y locations to place the handle in the correct position based on the parameters.

Specifying Parameters to Change

Next, click **Add** in the **Property** window. In the responding window, specify the name of the electrical parameter, and the expression, based on the X & Y location of the handle that sets this parameter. For example:

Edit the footprint for the MSTRL component. Place a handle that is on the top edge of the line. The component has the parameters P for length and W for width. The handle has its location (X,Y) set to: (P/2, W/2). One property is added called "W" and its expression is "abs(Y*2)". The "abs" (absolute value) of "Y*2" ensures that as Y approaches zero, the line width stays positive. The two entries in the window should look like the following:

Location: P/2, W/2

W: abs(Y*2)

How it works

When the user drags the handle, for each new location of the cursor, each of the handles properties are recalculated for that new position. For the TRL example, the y position of the top-side handle is used to set the line's "w", which of course is twice that of y. After the new parameter values are calculated, the handle's own position is recalculated based on the updated parameters and the expressions for its location. Thus, as the cursor goes up, the line gets wider, and the handle tracks with the new width.

Using Scripts to Define Footprints

Footprints in Layout can use scripts and primitives in defining the geometry needed. Scripts can also move and/or create ports. Scripts can be written in any Windows supported languages. Universally supported languages include JavaScript and VBScript.

When Layout needs to retrieve footprint geometry, it invokes the script, passing an object called LayoutHost. This object provides the interface to the needed functions within Layout necessary for creating geometry and accessing information within Layout. The script then calls the member functions of this host object to define the footprint appropriate.

Documentation on scripting within Windows including language definition for JavaScript and VBScript can be found at: http://msdn.microsoft.com/scripting/.

Layout Host Object

LayoutHost is an object within Layout which provides access to necessary Layout functions for use in a script. This topic describes the interface for this function.

LayoutHost Properties

Pars

Returns an ElementPars object. This property is used to set/get the component parameters declared in the footprint.

Example:

var pars = LayoutHost.Pars;

LayoutPars

Returns an ElementPars object. This property is used to set/get the locally defined parameters in the footprint.

Example:

var layPars = LayoutHost.LayoutPars;

LengthUnits

Returns a string. This property is used to set/get the internal length units in the current script. By default, length units in the script are meters. When this property is changed, any subsequent calls to the LayoutHost are interpreted in the new units.

Example:

LayoutHost.LengthUnits(“mm”);

MID

A constant used to query an edge for its mid point.

Example:

var x = rect.Edge(1).X(LayoutHost.MID);

START

A constant used to query an edge for its start point.

Example:

var x = rect.Edge(1).X(LayoutHost.START);

END

A constant used to query an edge for its end point.

Example:

var x = rect.Edge(1).X(LayoutHost.END);

LayoutHost Methods

The following LayoutHost methods are available for use in **Electronics Desktop**.

LayoutHost General Methods

GetDefinitionType()

Returns an integer which indicates the type of shape used to define a shape-based footprint. This type of footprint allows the user to draw a shape such as a rectangle or polygon , the script uses this definition to create a footprint. The method GetDefinitionPoints is then used to retrieve the points input by the user to define the shape’s outline.

Example:

var shapeType = LayoutHost.GetDefinitionType();

Possible shapes are:

0: Single point

1: Rect

2: Polyline

3: Polygon

GetDefinitionPoints()

Returns a PointsObject containing the points defining an outline for a shaped-based footprint. This method is used along with GetDefinitionType to retrieve the information needed regarding the base shape used in order to define the footprint.

Example:

var outline = LayoutHost.GetDefinitionPoints();

GetLastError()

Returns an integer id regarding the last function call which changes the footprint. Functions which change the footprint include those creating geometry, and those altering pins. A non-zero result indicates an error occurred. Zero is returned for success.

Example:

var error = LayoutHost.GetLastError();

GetLayerID(name)

Returns an integer id for the given name. If a layer with “name” is not found, then a -1 is returned. The name is not case-sensitive.

Example:

var layerID = LayoutHost.GetLayerID(“Top”);

CreatePointsObject()

Creates and returns an empty PointsObject. Points can then be added to this object which in turn is passed as a parameter to certain primitive creation functions.

Example:

var pts = LayoutHost.CreatePointsObject();

LayoutHost Shape Creation Methods

These methods create virtual shapes in the footprint. While these shapes mirror actual primitives, they are drawn only and are not selectable or editable. All length values are in the current units of the footprint.

NewArc(layer, cx, cy, r, ang1, ang2, lw)

Creates an arc in the footprint, returns a geometry object.

Parameters:

Layer: In integer id retrieved from GetLayerID

Cx: Center x coordinate

Cy: Center y coordinate

R: Radius of arc centerline

Ang1: Start angle of arc (in degrees)

Ang2: Stop angle of arc going CCW

Lw: Line width of arc

Example:

var geom = LayoutHost.NewArc(layerID, 0, 0, r, 0, 90, w);

NewArcLine(layer, cx, cy, r, ang1, ang2, lw, end)

Creates an arc line in the footprint, returns a geometry object.

Parameters:

Layer: In integer id retrieved from GetLayerID

Cx: Center x coordinate

Cy: Center y coordinate

R: Radius of arc centerline

Ang1: Start angle of arc (in radians)

Ang2: Stop angle of arc going CCW

Lw: Line width of arc

End: End style. Possible end styles are: flat, extended, round.

Example:

var geom = LayoutHost.NewArcLine(layerID, 0, 0, r, 0, 90, w, “flat”);

NewCircle(layer, cx, cy, r)

Creates a circle in the footprint, returns a geometry object.

Parameters:

Layer: In integer id retrieved from GetLayerID

Cx: Center x coordinate

Cy: Center y coordinate

R: Radius of circle

Example:

var geom = LayoutHost.NewCircle(layerID, 0, 0, r);

NewLine(layer, points, lw, join, end)

Creates a polyline in the footprint, returns a geometry object.

Parameters:

Layer: In integer id retrieved from GetLayerID

Points: A PointsObject populated with the points appropriate

Lw: Linewidth

Join: Join style. Possible join styles are: corner, miter, round.

End: End style. Possible end styles are: flat, extended, round.

(Passing an empty string defaults to round for Join or End style.)

Example:

var geom = LayoutHost.NewLine(layerID, pts, w, “corner”, “flat”);

NewPoly(layer, points)

Creates a polygon in the footprint, returns a geometry object.

Parameters:

Layer: In integer id retrieved from **GetLayerID**.

Points: A PointsObject populated with the points appropriate

Example:

var geom = LayoutHost.NewPoly(layerID, pts);

NewRect(layer, cx, cy, width, height, angle)

Creates a rectangle in the footprint, returns a geometry object.

Parameters:

Layer: In integer id retrieved from GetLayerID

Cx: Center x coordinate

Cy: Center y coordinate

Width: Width of rect

Height: Height of rect

Angle: Rotation angle to draw rect (in radians)

Example:

var geom = LayoutHost.NewRect(layerID, 0, 0, w, h, 0);

NewText(layer, text, x, y, font, size, angle, just)

Creates text in the footprint, returns a geometry object.

Parameters:

Layer: An integer id retrieved from GetLayerID

Text: The text string

X: Text x placement position

Y: Text y placement position

Font: Text font; may be left blank. On system layers, a system font must be used. Otherwise, specify a plotter font (e.g., "RomanDuplex").

Size: Text height (in length units for a plotter font; in points for a system font)

Angle: The text angle (in radians)

Just: Text justification relative to the placement position: "LeftTop", "LeftBase", "LeftBottom", "CenterTop", "CenterBase", "CenterBottom", "RightTop", "RightBase", "RightBottom"

Example:

var text = LayoutHost.NewText(layer, "Text", 2, -1, "", 1, 0, "LeftBase");

LayoutHost Relative Shape Placement Methods

These methods place shapes relative to each other

SnapEdges(edge, edge_pos, to_edge, to_edge_pos);

Places one geometry object against another so the specified edges are parallel and points touching

Parameters:

Edge: Specifies both the object to be moved and the edge to snap.

Edge_pos: The position on the source edge (the object being moved) that is to be snapped to the target edge; specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

To_edge: Specifies the target object and edge (this object is not moved)

To_edge_pos: The target snap point. The source edge and position is snapped to this location; specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

Example:

// Snap 'rect' so as that edge 1 is placed against edge 2 of 'rect2'

LayoutHost.SnapEdges(rect.Edge(1), LayoutHost.MID, rect2.Edge(2), LayoutHost.MID);

SnapEdgesWithOffset(edge, edge_pos, to_edge, to_edge_pos, x_off, y_off, rel_angle);

Same as for SnapEdges but the target snap point is specified by an edge position with an offset and angle. Associated with the target edge position is a normal vector and a tangent vector. The normal vector is perpendicular to the edge and points outwards (to the right in polylines). The tangent vector points in the direction of the edge. These two vectors define the local coordinate system in which x_off, y_off, and angle are specified.

Parameters:

Edge: Specifies both the object to be moved and the edge to snap.

Edge_pos: The position on the source edge (the object being moved) that is to be snapped to the target edge; specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

To_edge: Specifies the target object and edge (this object is not moved)

To_edge_pos: The target snap point. The source edge and position is snapped to this location; specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

X_off: An offset along the normal vector for the target edge.

Y_off: An offset along the tangent vector for the target edge.

Rel_angle: An additional rotation (specified as a counter-clockwise rotation in radians) relative to the normal vector for the target edge.

Example:

// Snap 'rect' so as that edge 1 is offset 1 unit from edge 2 of 'rect2'

LayoutHost.SnapEdgesWithOffset(rect.Edge(1), LayoutHost.MID, rect2.Edge(2), LayoutHost.MID, 1, 0, 0);

LayoutHost Port and Pin Methods

These methods create and move ports/pins as needed.

MovePort(name, cx, cy, angle)

Moves an existing port of that name. All scripted footprints must call MovePort for every pin in the footprint. This call should be made prior to defining any port geometry.

Parameters:

Name: Name of the port

Cx: Center x coordinate

Cy: Center y coordinate

Angle: Rotation angle to draw port (in radians)

Example:

LayoutHost.MovePort(“1”, x, y, 2*Math.tan(theta));

SnapPort(name, edge, edge_pos, rel_angle)

Moves a port to the specified edge position. The angle is relative to the normal vector (perpendicular to the edge and pointing outwards, or to the right in poly-lines).

Parameters:

Name: Name of an existing port

Edge: The edge on which the port is to be placed

Edge_pos: The position on the edge where the port is to be placed; specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

Rel_angle: An additional rotation (specified as a counter-clockwise rotation in radians) relative to the normal vector for the edge.

Example:

LayoutHost.SnapPort(“1”, rect1.Edge(1), LayoutHost.MID, 0);

SetPortLayers(name, start_layer, end_layer)

Sets the layer range (and hence placement) for a port

Parameters:

Name: Name of the port

Start_layer: An integer id retrieved from GetLayerID

End_layer: An integer id retrieved from GetLayerID

Example:

var top = LayoutHost.GetLayerId(“top”);

var bottom = LayoutHost.GetLayerId(“bottom”);

LayoutHost.SetPortLayers(“a”, top, bottom);

AddPortEdge (layer, name, points)

Defines the edge of the port to the polyline defined by “points”. Defining the port edge is necessary for co-simulation of a scripted footprint. Note, if you define the port edge, you must also have set the port position via “MovePort” first. Multiple edges may be added by calling this function repeatedly.

The points for the edge should be defined with the port as the origin. For example, if the edge is a vertical line 10 mils long running thru the port, the two points defining the edge is (0, 5) & (0, -5), regardless of the location of the port itself.

Parameters:

Layer: In integer id retrieved from GetLayerID

Name: Name of the port

Points: A PointsObject populated with the points appropriate

Example:

LayoutHost.AddPortEdge(layerID, “n1”, pts));

SetPortWidth (layer, name, w)

Defines the edge of the port to be a single line segment of width “w”, centered on the port position, and rotated to the port angle. This function is an alternate means of defining a port edge. Defining the port edge is necessary for co-simulation of a scripted footprint. Note, if you define the port edge, you must also have set the port position via “MovePort” first.

Parameters:

Layer: An integer id retrieved from GetLayerID

Name: Name of the port

W: Width of the port edge

Example:

LayoutHost.SetPortWidth(layerID, “n1”, w));

LayoutHost Via and Padstack Methods

NewVia(name, start_layer, end_layer, cx, cy, padstack, diameter)

Creates a via in the footprint and returns it as a via object.

Parameters:

Name: Name of the via

Start_layer: Integer id retrieved from GetLayerID

End_layer: Integer id retrieved from GetLayerID

Cx: Center x coordinate

Cy: Center y coordinate

Padstack: Name of the padstack to use. Passing an empty string (“”) results in “No Pad SMT East” by default.

Diameter: Diameter of the via

Example:

var via = LayoutHost.NewVia(“via_name”, layerID1, layerID2, x, y, “Planar EMVia”, diameter);

SnapVia(via, to_edge, to_edge_pos, x_off, y_off, rel_angle);

Moves a via to the specified edge position. The target snap point is specified by an edge position with an offset and angle. Associated with the target edge position is a normal (perpendicular to the edge and pointing outwards, or to the right in poly-lines) and a tangent vector (pointing in the direction of the edge). These two vectors define the local coordinate system in which x_off, y_off, and angle are specified.

Parameters:

Via: Specifies the via object to be moved

To_edge: Specifies the target object and edge

To_edge_pos: The target snap point. The via position is snapped to this location, specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

X_off: An offset along the normal vector for the target edge.

Y_off: An offset along the tangent vector for the target edge.

Rel_angle: An additional rotation (specified as a counter-clockwise rotation in radians) relative to the normal vector for the target edge.

Example:

// Place ‘via’ 1 unit inside edge 2 of ‘rect’

LayoutHost.SnapVia(via, rect.Edge(2), LayoutHost.MID, -1, 0, 0);

ElementPars Object

This object permits retrieval of parameters & local variables defined in the footprint for use in a script. Retrieval of this object was discussed under LayoutHost.

Count

Contains the number of parameters in this object. This property is read-only.

Example:

var pars = LayoutHost.Pars;

var numPars = pars.Count;

Item

Contains the given parameter in this object. The item may be retrieved by index (zero based) or name. The name is not case sensitive. This property is read-only.

Example:

var pars = LayoutHost.Pars;

var w = pars.Item(0);

var p = pars.Item(“p”);

ElementPars Methods

There are no methods for the ElementPars object.

Related Topics

ElementPars Properties

ElementPars Methods

Points Object

This object permits retrieval of parameters & local variables defined in the footprint for use in a script. Retrieval of this object was discussed under LayoutHost.

Points Object Properties

Count

Contains the number of points in this object. This property is read-only.

Example:

var pts = LayoutHost.CreatePointsObject();

var numPts = pts.Count; // returns 0 in this case

X(index)

Contains the X coordinate for a given index (zero based).

Example:

var pts = LayoutHost.CreatePointsObject();

x = 10;

pts.Add(x, y);

x2 = pts.X(0) * 2;

pts.X(0) = x2;

Y(index)

Contains the Y coordinate for a given index (zero based).

Example:

var pts = LayoutHost.CreatePointsObject();

y = 10;

pts.Add(x, y);

y2 = pts.Y(0) * 2;

pts.Y(0) = y2;

Points Object Methods

Add(x,y)

Adds a new point to the end of the object's array.

Example:

var pts = LayoutHost.CreatePointsObject();

pts.Add(5, 10);

Remove(index)

Removes a point on the object's array, given the (zero-based) index.

Example:

var pts = LayoutHost.CreatePointsObject();

pts.Add(5, 10);

pts.Remove(0);

Geom Object

Geometry objects are returned by the shape creation methods (NewRect, NewCircle, etc.) Geometry objects may be manipulated in a variety of ways through the methods exposed by their interface. All coordinates and length related values are passed/returned in the current default script units. Unless changed through LayoutHost.LengthUnits, all values are by default assumed to be in meters (SI units).

Example:

var rect = LayoutHost.NewRect(layer, 0, 0, l, w, 0);

Geom Object Properties

Layer

Get/Set the layer on which a geometry object is placed.

Example:

rect.Layer = LayoutHost.GetLayerID('top');

Edge(index)

Return the edge object for the edge specified. Edge indexes are zero based and count in a counter-clockwise direction around the object. For rectangles, edge zero is the lower edge. Circles are defined by two arc edges, a lower edge (edge zero) and an upper edge (edge 1); the start/end points of these edges are directly to the left/right of center. Lines and Polygon edges are defined by the user when the object is created. A rotated box giving the text extent is used to define the edges for a text object; edge zero is the line beneath the text.

Example:

var len = poly.Edge(5).Length

Count

Returns the number of edges in the geometry object.

Example:

for (var i = 0; i < poly.Count; ++i)

Geom Object Methods

MoveBy(dx, dy)

Translates the geometry object by the specified offset.

Parameters:

Dx: X offset

Dy: Y offset

Example:

rect.MoveBy(10, 20);

MoveTo(x, y)

Moves the geometry object to the specified location; useful for rectangles, circles and arcs (the center point is moved)

Parameters:

X: New X location

Y: New Y location

Example:

rect.MoveTo(10, 20);

Rotate(x, y, angle)

Rotates the geometry object counter-clockwise about the specified location by the angle specified (in radians).

Parameters:

X: X center

Y: Y center

Angle: Rotation angle in radians

Example:

rect.Rotate(0, 0, Math.PI/4);

Scale(x, y, factor)

Scales the geometry object about the specified location by the factor specified.

Parameters:

X: X center

Y: Y center

Factor: Scale factor (2 implies 2 times, 4.5 implies 4 and a half times, etc.)

Example:

rect.Scale(0, 0, 2);

MirrorX(x)

Mirrors the X values of the object (Y values are unchanged) about the specified X value (X values falling on this line are unchanged).

Parameters:

X: Center of reflection (X values on this line are unchanged).

Example:

rect.MirrorX(0);

MirrorY(y)

Mirrors the Y values of the object (X values are unchanged) about the specified Y value (Y values falling on this line are unchanged).

Parameters:

Y: Center of reflection (Y values on this line are unchanged).

Example:

rect.MirrorY(0);

Copy()

Create and return a copy of the geometry object.

Example:

var rect_copy = rect.Copy();

AddVoid(void_geom)

Adds a void object to an existing geometry object.

Example:

Var rect = LayoutHost.NewRect(layer, 0, 0, l, w, 0);

Var void_rect = LayoutHost.NewRect(layer, 0, 0, l/2, w/2, 0);

rect.AddVoid(void_rect);

Edge Object

Edge objects are returned by an Edge query on a Geometry object. Note: edges may be arcs or straight segments.

Related Topics

Edge Object Properties

Edge Object Methods

Edge Object Properties

Edge objects are returned by an Edge query on a Geometry object. Note: edges may be arcs or straight segments.

Example:

var edge = geom.Edge(1);

Length

Return the true edge length (if an arc, the curvature is taken into account).

Example:

var len = geom.Edge(1).Length;

Angle(pos)

Return the angle of the tangent (in radians), relative to the standard X axis, of the edge point specified by 'pos'. Angles are returned using the usual 'count-clockwise is positive' convention. Tangents point in the direction of the edge.

Parameters:

Pos: Edge position; usually specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

Example:

var angle = geom.Edge(1).Angle(LayoutHost.START);

Normal(pos)

Return the angle of the normal (in radians), relative to the standard X axis, of the edge point specified by ‘pos’. Angles are returned using the usual ‘count-clockwise is positive’ convention. Normals are outward facing (or to the right in poly-lines) of the edge.

Parameters:

Pos: Edge position; usually specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

Example:

var normal = geom.Edge(3).Normal(LayoutHost.END);

X(pos)

Return the X value of the edge point specified by ‘pos’.

Parameters:

Pos: Edge position; usually specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

Example:

var x = geom.Edge(3).X(LayoutHost.END);

Y(pos)

Return the Y value of the edge point specified by ‘pos’.

Parameters:

Pos: Edge position; usually specified as LayoutHost.START, MID, or END (or a value < 0, 0, or > 0).

Example:

var y = geom.Edge(3).Y(LayoutHost.END);

Edge Object Methods

There are no methods for the Edge object.

Via Object

Via objects are returned by the LayoutHost method NewVia. They may be manipulated in a variety of ways through the methods exposed by their interface. All coordinates and length related values are passed/returned in the current default script units. Unless changed through LayoutHost.LengthUnits, all values are by default assumed to be in meters (SI units).

Example:

var via = LayoutHost.NewVia(“via_name”, layerID1, layerID2, x, y, “Planar EMVia”, diameter);

Via Object Properties

StartLayer

Get/Set the start layer of the via.

Example:

via.StartLayer = LayoutHost.GetLayerID(‘top’);

EndLayer

Get/Set the end layer for the via.

Example:

var layer = via.EndLayer;

X

Get/Set the x position for the via.

Example:

v1.X = v2.X;

Y

Get/Set the y position for the via.

Example:

v1.Y = v2.Y;

HoleDiameter

Get/Set the hole diameter for the via. A negative diameter turns off the diameter override and the value on the padstack definition is used.

Example:

v1.HoleDiameter = v2.HoleDiameter;

Rotation

Get/Set the via rotation. The rotation is in radians.

Example:

v1.Rotation = Math.PI/4;

Via Object Methods

ClearLayerMapping(fp_layer)

Clears the mapping between a footprint layer and a layer in the padstack definition.

Parameters:

fp_layer: An integer id retrieved from GetLayerID

Example:

var v1 = LayoutHost.NewVia("via1", layer, bottom, 0, 0, "Round 1mm/0.5mm", 0.2);

v1.ClearLayerMapping(layer);

ClearLayerMappings

Clears the mapping between all footprint and padstack definition layers.

Example:

var v1 = LayoutHost.NewVia("via1", layer, bottom, 0, 0, "Round 1mm/0.5mm", 0.2);

v1.ClearLayerMappings();

Copy()

Create and return a copy of the via.

Example:

var via_copy = via.Copy();

DefaultLayerMapping

Reset the mapping between all footprint and padstack definition layers to the default mapping.

Example:

var v1 = LayoutHost.NewVia("via1", layer, bottom, 0, 0, "Round 1mm/0.5mm", 0.2);

v1.DefaultLayerMapping();

MirrorX(x)

Mirrors the via position about the specified X value (Y value is unchanged; an X value falling on this line is unchanged).

Parameters:

X: Center of reflection (X values on this line are unchanged).

Example:

via.MirrorX(0);

MirrorY(y)

Mirrors the via position about the specified Y value (X values are unchanged; a Y value falling on this line is unchanged).

Parameters:

Y: Center of reflection (Y values on this line are unchanged).

Example:

via.MirrorY(0);

MoveBy(dx, dy)

Translates the via by the specified offset.

Parameters:

Dx: X offset

Dy: Y offset

Example:

via.MoveBy(10, 20);

MoveTo(x, y)

Moves the via to the specified location; useful for rectangles, circles and arcs (the center point is moved)

Parameters:

X: New X location

Y: New Y location

Example:

via.MoveTo(10, 20);

Rotate(x, y, angle)

Rotates the via counter-clockwise about the specified location by the angle specified (in radians).

Parameters:

X: X center

Y: Y center

Angle: Rotation angle in radians

Example:

via.Rotate(0, 0, Math.PI/4);

Scale(x, y, factor)

Scales the via position about the specified location by the factor specified.

Parameters:

X: X center

Y: Y center

Factor: Scale factor (2 implies 2 times, 4.5 implies 4 and a half times, etc.)

Example:

via.Scale(0, 0, 2);

SetLayers(start_layer, end_layer)

Sets the via layer range

Parameters:

Start_layer: An integer id retrieved from GetLayerID

End_layer: An integer id retrieved from GetLayerID

Example:

var top = LayoutHost.GetLayerId(“top”);

var bottom = LayoutHost.GetLayerId(“bottom”);

via.SetLayers(top, bottom);

SetLayerMapping(fp_layer, ps_layer, pad_type);

Creates or adjusts the mapping between a footprint layer and the padstack definition layer.

Parameters:

fp_layer: An integer id retrieved from GetLayerID

ps_layer: The name of the layer in the padstack definition

pad_type: Unused

Example:

var v1 = LayoutHost.NewVia("via1", layer, bottom, 0, 0, "Round 1mm/0.5mm", 0.2);

v1.ClearLayerMappings();

v1.SetLayerMapping(layer, "template", "");

v1.SetLayerMapping(bottom, "template", "");

Using the Symbol Editor

A symbol is a set of information that defines the graphical representation and electrical connectivity of a component. Symbol definitions are stored in library files with an extension of **.aslb**.

You can access material definitions using the **Symbol Editor**. The Electronics Desktop **Symbol Editor** allows you to view, modify, and save symbol information for components and layout primitives. After modifying a symbol, the information can either be saved, used to define a design, or exported.

When a symbol is used in a project, it is copied to the project. Thus new projects do not list any symbols in the Project Manager. If a change is made to the symbol, it is actually made to the project's copy of the symbol. So each project has its own set of symbols based on AEDT's symbol library. When you open an existing project, the Project Manager displays the symbols in use in the project.

To start the **Symbol Editor**, open a project and select **Tools** > **Edit Libraries** > **Symbols**. When the Edit Libraries window opens, use its controls to open the **Symbol Editor**.

Creating a New Symbol

1. Click **Tools** > **Edit Libraries** > **Symbols**.
2. From the **Symbols** tab of the **Edit Libraries** window, either:
   - Click **Add Symbol** to open the **Get Name** window.
   - Select an existing symbol to base your new symbol and click **Clone Symbol(s)** to open the **Get Name** window.
3. Type a name for the new symbol into the **Enter the name for this new Symbol** field and click **OK**.
4. Then double-click the entry for the new symbol or click **Edit Symbol** to open the symbol editor.
5. Use the symbol editor to:

- Draw graphical primitives, such as rectangles, circles, and arcs, using options on the Draw menu.
- Add pins for electrical connections using the Pin option on the **Draw** menu.
- Modify pin properties in the Pin List dialog box.
- Add labels. To include component properties or other variables in a label, use the Property Display Setup option on the Symbol menu. For unchanging text, use the text option on the Draw menu. Unusual labels are most likely represent a component property.
- Update the current project with the new or revised symbol definition using the Update Project option on the **Symbol** menu

You can also export a symbol to a symbol library (**.aslb**) file for use in other projects. For information on how to do this, see the Edit Libraries.

Editing an Existing Symbol

When a symbol is used in a project, it is copied to the project from a global library. Any changes made to the symbol are made to the project's copy of the symbol and only affect the symbol in the project and not in any other project. To edit an existing symbol, either:

- From the **Project Manager** window, open the project that contains the symbol you want and open **Definitions** > **Symbols**. Either double-click the symbol you want to edit or right-click the symbol and select **Edit Symbol**.
- Open **Tools** > **Edit Libraries** > **Symbols**. From the **Symbols** tab of the **Edit Libraries** window, either double-click the symbol that you want to edit or right-click the symbol and choose **Edit Symbol**.

The symbol editor opens with the selected symbol ready for editing.

Use the symbol editor to:

- Draw graphical primitives, such as rectangles, circles, and arcs, using options on the Draw menu.
- Add pins for electrical connections using the Pin option on the **Draw** menu.
- Modify pin properties in the Pin List dialog box.
- Add labels. To include component properties or other variables in a label, use the Property Display Setup option on the Symbol menu. For unchanging text, use the text option on the Draw menu. Unusual labels are most likely represent a component property.
- Update the current project with the new or revised symbol definition using the Update Project option on the **Symbol** menu

You can also export a symbol to a symbol library (**.aslb**) file for use in other projects. For information on how to do this, see the Edit Libraries.

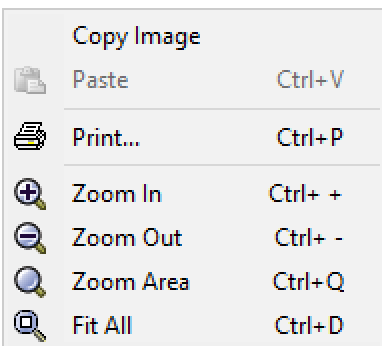
Symbol Editor Operations

The symbol editor opens when you begin to create or modify a symbol. Like the **Schematic Editor**, the symbol editor is a graphical tool that keeps the placement of symbol elements uniform by snapping them to points on a grid. The window controls are the same as those in the schematic editor: press the arrow keys and **Page Up** and **Page Down** to pan, or **Ctrl +D** to fit the view.

With the symbol editor, add, modify, and manipulate graphical primitives, text labels, pins, and property displays. You can also adjust the resolution, color, and visibility of the editor grid. When you're done, update the current project with any changes you've made to the current symbol. Options related to these operations are available on the Symbol menu and the Symbol Draw menu.

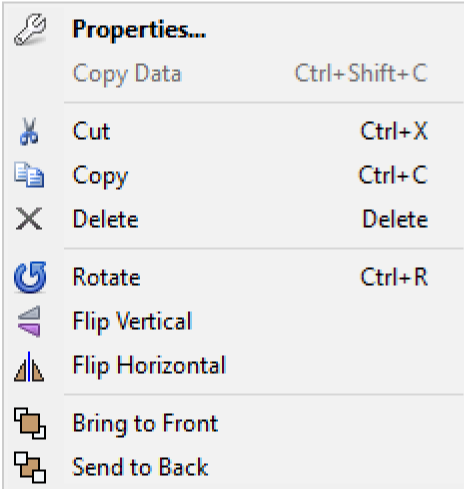
Symbol Editor Shortcut Menus

When you right-click in the symbol editor grid, this shortcut menu appears:



These options are a subset of those available on the **View** menu.

When you right-click an object in the symbol editor, this shortcut menu appears:



Click **Properties** to view the properties of the selected object. The other options are a combination of entries on the **Edit**, **View**, and **Draw** menus.

Symbol Draw Menu

The symbol editor **Draw** menu presents options for drawing graphical primitives and text, and manipulating selected objects. The **Draw** menu options include:

- Arc
- Circle
- Line
- Polygon
- Rectangle
- Text
- Image
- Ellipse
- Curve
- Pin
- Rotate
- Align Horizontal
- Align Vertical
- Flip Vertical
- Flip Horizontal
- Bring to Front
- Send to Back

Arc

This option, also available on the **Symbol Draw** toolbar , initiates creation of an arc. To draw an arc, click the symbol editor grid at the two points that determines the arc's ends, then drag the arc to the appropriate radius.

Once you have created an arc, adjust its radius or move its endpoints by clicking the arc to select it, and dragging the appropriate handle.

Circle

This option, also available on the **Symbol Draw** toolbar , initiates creation of a circle. To draw a circle, click the symbol editor grid to select a center point, then drag the circle to the appropriate diameter.

Once you have created a circle, adjust its diameter in either of two ways:

- Click the circle to select it, then drag one of its handles.
- Double-click the circle to open its **Properties** window, type a new value for the **Radius** parameter, then click **OK**.

By default, a new circle is hollow. You can fill a circle with solid color or parallel lines in one of several styles as follows:

a. With the circle's properties displayed, click in the **Value** cell for the **FillStyle** property.
b. Select the appropriate fill style on the list.
c. Click **OK**.

Line

This option, also available on the **Symbol Draw** toolbar , initiates creation of a polyline with one or more segments. To draw a line:

1. Click the symbol editor grid where you want the line to start, then click at one or more points to continue the line.
2. To complete the line, do one of the following after defining its final segment:
   - Press the **Spacebar**.
   - Right-click, then click **Finish**.

After you have completed a line, change the endpoints of its segments by doing the following:

- Select the line and drag the appropriate handle(s).
- Double-click the line to open its **Properties** window, type new position values (in the form X: Y) for the appropriate vertices, and click **OK**.

Alternately, change the endpoints of a selected line by clicking **Layout > Line Styles** to open the **Line Styles** window, then click in the Cap Type field to change the cap or line-ending type (Flat, Extended, or Round).

Since a line has nonzero width, its CapType determines how its ends behave:

- **Extended** – Extends the line by its own width past the endpoint you click.
- **Flat** - Cuts the line perpendicularly at the endpoint you click.
- **Round** - Ends the line in a semicircle.

The existing technology files define line/trace styles, including a CapType (typically Flat) for each style. If you select None as the technology type, you'll automatically get a palette of four styles with a variety of widths, BendTypes and CapTypes.

Polygon

This option, also available on the **Symbol Draw** toolbar , initiates creation of a polygon. To create a polygon:

1. Click the symbol editor grid to specify the position of one vertex, then click wherever you want to place additional vertices.
2. To complete a polygon, specify the position of its final vertex, then do either of the following:
   - Press the **Spacebar**.
   - Right-click, then click **Finish**.

Once you have drawn a rectangle, Edit its vertex positions and other properties, including its fill style, as follows:

- Click the rectangle, then edit its properties in the **Properties** window.
- Double-click the rectangle, then edit its properties in the **Properties** window.

By default, a new polygon is hollow. You can fill a polygon with solid color or parallel lines in one of several styles as follows:

a. With the polygon’s properties displayed, click in the **Value** cell for the **FillStyle** property.
b. Select the appropriate fill style on the list.
c. Click **OK**.

Rectangle

This option, also available on the **Symbol Draw** toolbar , initiates creation of a rectangle. To create a rectangle, click the editor grid to specify the position of one corner, then drag the rectangle to the appropriate size.

Once you have drawn a rectangle, Edit its height and width, the position of its center, and its angle (its rotation, in degrees, relative the handles of its bounding box) as follows:

- Click the rectangle, then edit its properties in the **Properties** window.
- Double-click the rectangle, then edit its properties in the **Properties** window.

By default, a new rectangle is hollow. You can fill a rectangle with solid color or parallel lines in one of several styles as follows:

1. With the rectangle's properties displayed, click in the **Value** cell for the **FillStyle** property.
2. Select the appropriate fill style on the list.
3. Click **OK**.

Text

This option, also available on the **Symbol Draw** toolbar , adds to the symbol an editable "Default text" label in 12-point Arial. To edit the default text string immediately after placement, type the text you want, then press **Enter** or click elsewhere in the editor grid.

To change just the text of an existing label:

1. Click the label.
2. Click the label again to open its text for editing.
3. Type the appropriate text.
4. Press **Enter** or click elsewhere in the editor grid.

To change other properties of a label, including its color, font or size:

1. Click the label and view its properties in the **Properties** window, or double-click the label and view its properties in the **Properties** window.
2. Click in the **Value** cell for the property you want to modify.
3. Modify the value.
4. Click **OK**, or click in another **Value** cell to commit the change and keep editing values.

Image

This option, also available on the **Symbol Draw** toolbar , adds an image to the symbol. To add an image, select an image file in the **Select Image** window and click **Open**. Click the editor grid to place the image, then drag the mouse until the image is the appropriate size, and click the mouse again.

Once you have placed an image, Edit its height and width, the position of its center, and its angle (its rotation, in degrees, relative the handles of its bounding box) as follows:

- Click the image to select it, then drag one of its handles.
- Click the image and edit its properties in the **Properties** window.
- Double-click the image and edit its properties in the **Properties** window.

To flip the image horizontally, select the image and in the **Properties** window, click **Mirrored**.

By default, a new image is not linked its original file. To ensure that the image is updated when changes are made to the file, select the image and in the **Properties** window, click **Link to File**.

A new image does not have a border. Add one as follows:

1. With the image's properties displayed in the **Properties** window, click the **Value** cell for the **DisplayBorder** property.
2. Set a **BorderWidth**.
3. Click the **Value** cell for the **BorderColor** property to add a color in the **Color** window.
4. Click **OK** to close the **Color** window.

Ellipse

This option, also available on the **Symbol Draw** toolbar , initiates creation of an ellipse. To draw a ellipse, click the editor grid to specify the top and left most position of the shape , then drag the ellipse to the appropriate size.

Once you have drawn an ellipse, Edit its height and width, the position of its center, and its angle (its rotation in degrees relative the handles of its bounding box) as follows:

- Click the ellipse to select it , then drag one of its handles.
- Click the ellipse and edit its properties in the **Properties** window.
- Double-click the ellipse and edit its properties in the **Properties** window.

By default, a new ellipse is hollow. You can fill an ellipse with solid color or parallel lines in one of several styles as follows:

1. With the ellipse's properties displayed, click in the **Value** cell for the **FillStyle** property.
2. Select the appropriate fill style on the list.
3. Click **OK**.

Curve

This option, also available on the **Symbol Draw** toolbar , initiates creation of a polycurve with one or more segments. To draw a curve:

1. Click the symbol editor grid where you want the curve to start.
2. Clicke or more points to continue the curve.

3. To complete the curve, mark its final segment and do one of the following:
   - double-click.
   - Press the **Spacebar**.
   - Right-click and click **Finish**.

After you have completed a curve, change the endpoints of its segments by doing the following:

- Select the curve and drag the appropriate handle(s).
- Double-click the curve to open its **Properties** window, type new position values (in the form X: Y) for the appropriate vertices, and click **OK**.

Pin

This option, also available on the **Symbol Draw** toolbar, initiates placement of a symbol pin with a default stem length of 10. You can rotate a pin once you've started to place it by iteratively pressing R until the pin is oriented to your liking. Click in the editor grid to finish placing the pin.

Rotate

This option, also available on the **Symbol Draw** menu and by pressing **Ctrl**+**R** on the keyboard, rotates a selected object or group of objects 90° to the left.

Align Horizontal

This option horizontally aligns the uppermost edges of the members of a group of selected objects with the uppermost edge of the first-selected object. To align multiple objects horizontally:

1. Press **Ctrl**, then click the object with which you want to align the others.
2. Still pressing **Ctrl**, click the additional objects in turn to add them to the selection.

   Note that the first-selected object is highlighted in red, and that the subsequently selected objects are highlighted in dark red.

3. From the **Draw** menu, select **Align Horizontal**.

Align Vertical

This option vertically aligns the leftmost edges of the members of a group of selected objects with the leftmost edge of the first-selected object. To align multiple objects vertically:

1. Press Ctrl, then click the object with which you to align the others.
2. Still pressing Ctrl, click the additional objects in turn to add them to the selection.

   Note that the first-selected object is highlighted in red, and that the subsequently selected objects are highlighted in dark red.

3. From the **Draw** menu, select **Align Vertical**.

Flip Vertical

This option, also available on the **Symbol Draw** toolbar, flips a selected object about the X axis.

Flip Horizontal

This option, also available on the **Symbol Draw** toolbar, flips a selected object about the Y axis.

Bring to Front

This option, also available on the **Symbol Draw** toolbar, places a selected object in front of all other elements in the editor, i.e., closest to the user.

Send to Back

This option, also available on the **Symbol Draw** toolbar, places a selected object behind all other elements in the editor, i.e., furthest on the user.

Symbol Menu

The Symbol menu lists options for symbol naming, property displays, and file operations. Its entries include:

Update Project

This option updates the current project with changes made in the current symbol.

> **Note: Update Project** updates your symbol changes to the current project in memory and does not save your changes to disk. To save your changes to disk, you must also click **Save** on the **File** menu. If you like the changes you have made in a symbol to be available for use in other projects, you must export the symbol to a library as described in the Edit Libraries topic.

Set Name

This option opens a **Properties** window to rename the current symbol.

Property Display Setup

This option opens the Properties Display so add or modify display of the names and/or values of properties associated with the component represented by the current symbol.

Pin List

This option gives you access to settings related to pin properties by opening the Pin List window.

Grid Setup

This option gives you access to settings related to the symbol editor grid resolution, color, and visibility by opening the symbol editor Grid Setup window.

Import File

The Import File command allows you to import an existing SVG-formatted symbol into the symbol editor.

Export File

This option saves the current symbol as a Microsoft Enhanced Meta File (**.emf**) graphic.

Edit Component

This option launches the Edit Component window to modify the component associated with the active symbol.

Normalize Symbol on Save

Allows you to normalize the symbol you are editing when you save after making modifications.

List

Opens the **Design List** window, which lists the objects in the layout. Each column may be sorted by clicking on the heading. One or more object rows can be selected and operated by clicking **Delete** and **Properties**.

Import File (Symbol)

The **Symbol** > **Import File** command allows you to import an existing SVG-formatted symbol into the symbol editor. The SVG (Scalable Vector Graphics) format is an open, xml-based web standard graphics format supported by many third-party tools.

To import an SVG (**.svg**) file when creating a new symbol definition do the following:

1. From the **Project Manager** window, right-click **Definitions** > **Symbols** in the appropriate project and select **Add Definition**.
2. In the **Definition Name** window, enter a name for the new symbol and click **OK** to open a new blank **Symbol Editor** window.

3. From the main menu bar, select **Symbol** > **Import File** and choose an existing **.svg** symbol file to import it into the symbol editor. A window opens advising that importing the

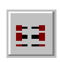
.svg file removes any existing symbol graphics in the editor, and asking for confirmation to continue the import.

> **Note:**
>
> - There is no “Undo” for the import operation. Consequently if you make a mistake on import, you must close the project without saving, then reopen it to recover.
> - Currently **.svg** symbols can be imported into any symbol editor window - including those containing pre-existing component symbols in which case the current symbol in the editor is replaced by the imported symbol, so ensure that you import the symbol in the appropriate place.

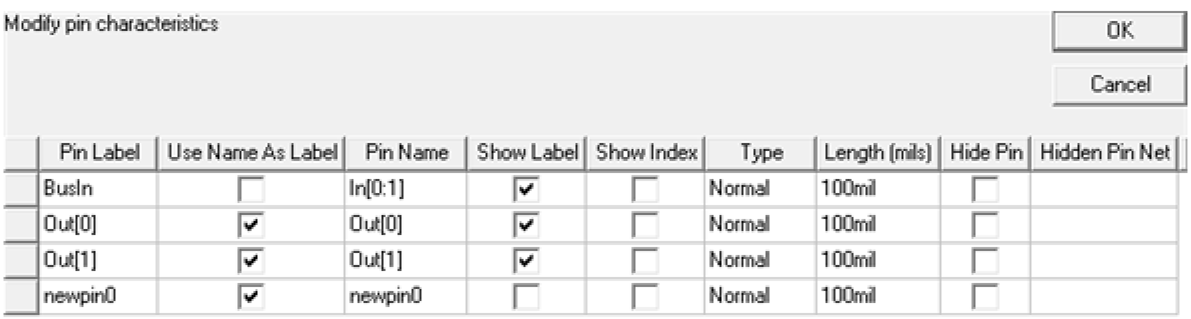
4. Click **OK** to import the **.svg** symbol for further editing, or **Cancel** to abort the operation. Once the new symbol has been created, attach it to a component via the Property Display Setup command.

Editing Pin Properties

To edit a property of a symbol pin, open the symbol editor’s **Pin List** window by doing either of the following:

- In the **Symbol** menu, select **Pin List**.
- From the **Symbol** toolbar, click the **Pin list** icon .

The **Pin List** window displays and sets values for the properties of symbol pins.



Pin Label	Use Name As Label	Pin Name	Show Label	Show Index	Type	Length (mils)	Hide Pin	Hidden Pin Net
BusIn	☐	In[0:1]	☑	☐	Normal	100mil	☐	
Out[0]	☑	Out[0]	☑	☐	Normal	100mil	☐	
Out[1]	☑	Out[1]	☑	☐	Normal	100mil	☐	
newpin0	☑	newpin0	☐	☐	Normal	100mil	☐	

Pin properties include:

- **Pin Label** – provides a field to add a user-defined label for the chosen pin on a component symbol without having to change the actual **Pin Name** associated with the chosen pin. Pin labels for pins in a component should be unique.

To specify or change the label of a pin, click in the **Pin Label** cell for that pin, type the name, and do either of the following:

- Press **Enter**.
- Click in another cell.

The **Pin Label** can be displayed for the chosen pin by unchecking **Use Name As Label**.

- **Use Name As Label** – controls whether the **Pin Name** or **Pin Label** is displayed for the chosen pin when **Show Label** is enabled. For a new pin, **Use Name As Label** is checked by default.
- **Pin Name** – sets the actual pin name. To specify or change the name of a pin, click in the **Pin Name** cell for that pin, type the name, and do either of the following:
  - Press **Enter**.
  - Click in another cell.
- **Show Label** – controls visibility of the either the **Pin Name** or **Pin Label** text for the pin (as determined by **Use Name As Label**). For a new pin, it is unchecked by default.
- **Show Index** – controls visibility of the pin number in the schematic editor. For a new pin, it is unchecked by default.
- **Type** – drop-down menu sets the pin type. Options include **Normal** (the default), **ANSI In**, **ANSI Out**, and **Zero Length**. To change a pin's type, click in its **Type** cell, then click the appropriate option.
- **Length (mils)** – sets the length of the pin stem (the graphical line associated with the pin port symbol). By default, new pins have a pin stem length of 100 mils. To change the length of a pin, click in its **Length** cell, type a new value, and do either of the following:
  - Press **Enter**.
  - Click in another cell.
- **Hide Pin** – controls whether or not a pin is visible in the schematic editor. For new pins, it is unchecked by default.
- **Hidden Pin Net** – editable when **Hide Pin** is checked, specifies the net (circuit node) to which a pin is connected if it is hidden. To specify a net, click in the **Hidden Pin Net** cell, type a net name, then do either of the following:
  - Press **Enter**.
  - Click in another cell.

Edit Symbol Pin Locations

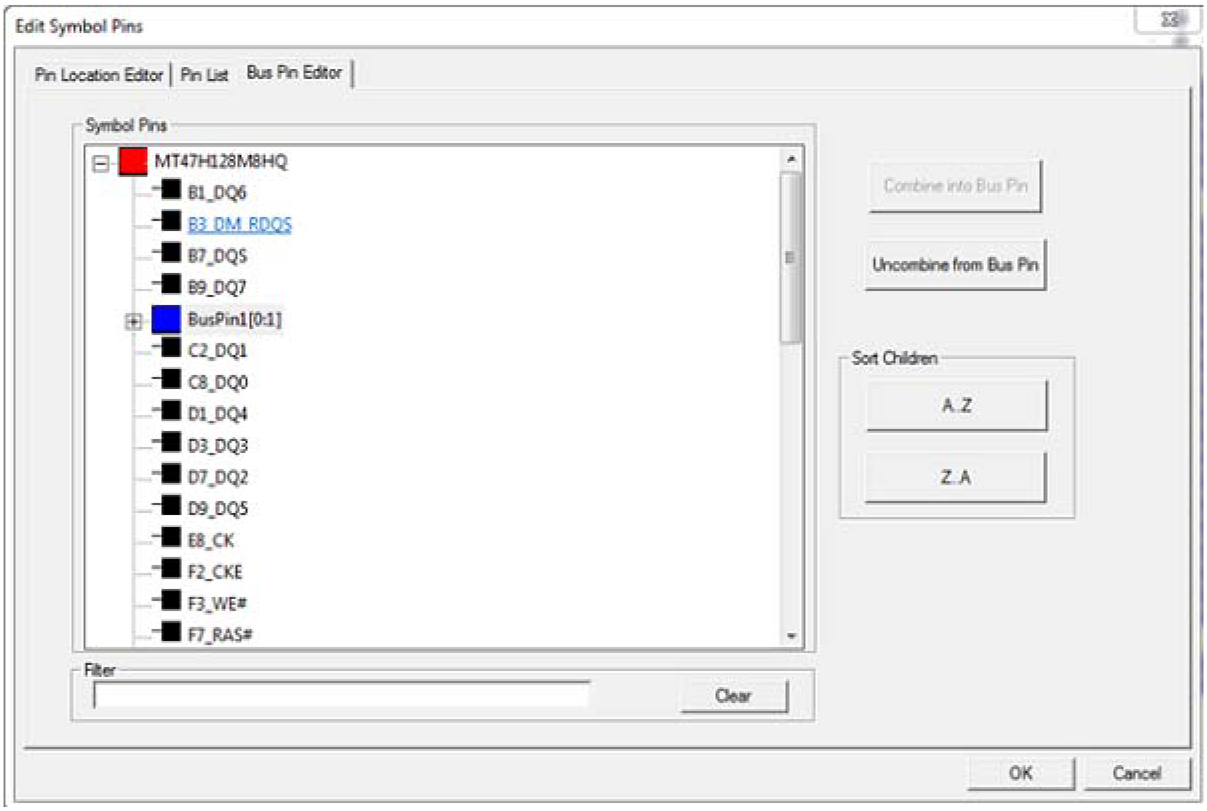
To edit the location of a symbol pin, open the **Edit Symbol Pins** window by doing one of the following:

- In the Symbol Editor **Symbol** menu, select **Edit Pins**.
- In the component instance shortcut menu, select **Edit Symbol Pins**.

The window has three tabs: **Bus Pin Editor**, **Pin Location Editor**, and **Pin List**. Bus Pin Editor and Pin Location Editor appear only for symbols that are drawn as a rectangle with pins.

Bus Pin Editor

The **Bus Pin Editor** tab allows you to create, edit, and delete bus pins from regular pins. The pin names are the terminal names associated with the component model and cannot be changed, but the bus pin names can be anything the user wants.

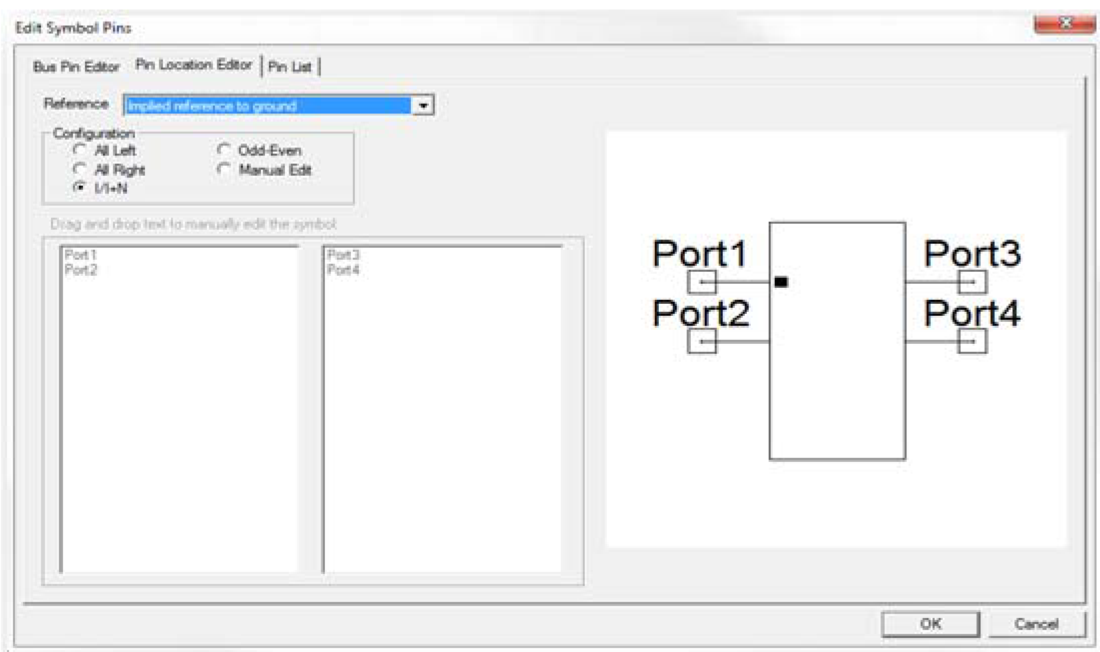


- **To create a new bus pin**
  - Select the pins needed for the bus pin and click **Combine in to Bus Pin**. Both regular pins and bus pins can be combined to create the new bus pin.
  - Pins can be selected with Ctrl + Click or Shift + click.
  - Pins can be dragged and dropped onto an existing bus pin.
- **To remove pins from a bus pin**
  - Select the whole bus pin or individual pins inside the bus pin and click **Uncombine from Bus Pin**.
  - Pins can be dragged out of the bus pin to the root level to remove on the bus pin.

- **Re-ordering Pins**
  - The pins inside a bus pin can be reordered by dragging them to the appropriate location. The order is important since the first pin with index 0 connects to bus[0] when a bus is connected, and so on.
  - When either the root node (red symbol) is selected OR a bus pin is selected, Sort features become available. Click the buttons to order the pins and bus-pins immediately under the selected node by name in either alphanumeric or reverse-alphanumeric order.
- **Searching the Tree**
  - The tree can be filtered by the pin name by typing in the filter box.
  - The filter text supports wildcard search (e.g., "P*1" should search for everything that begins with a "P" and ends with a "1").
  - Search works for both the root level and leaf level of pins (e.g., if a search matches the pin name under a bus pin, the bus pin is displayed in the search result).

Pin Location Editor

The **Pin Location Editor** allows you to change the locations of the symbol pins to the left or right side according to the settings chosen in the window.



Selecting a new option on the **Reference** drop-down menu changes the symbol displayed to show or hide additional reference pins. The **Reference** list is not shown in the winodw if the symbol does not support reference pins. The options are:

- **Implied reference to ground**: the global ground is the reference port for each N port connection. Its voltage is always 0V.
- **Common reference port**: an additional reference pin is added that is a common reference for the definition of port voltages. This reference can have any voltage and corresponds to the net for that extra pin.
- **Add individual hidden reference pin per port**: an additional reference pin is added for each port. These reference pins are hidden, set to 0V by default, and can each have their own reference voltage. You can connect each reference pin to its own net. Select this option if you do not want to see the reference pins.
- **Add individual reference pin per port**: an additional reference pin is added for each port. Every reference pin can have its own reference voltage and be connected to its own net.

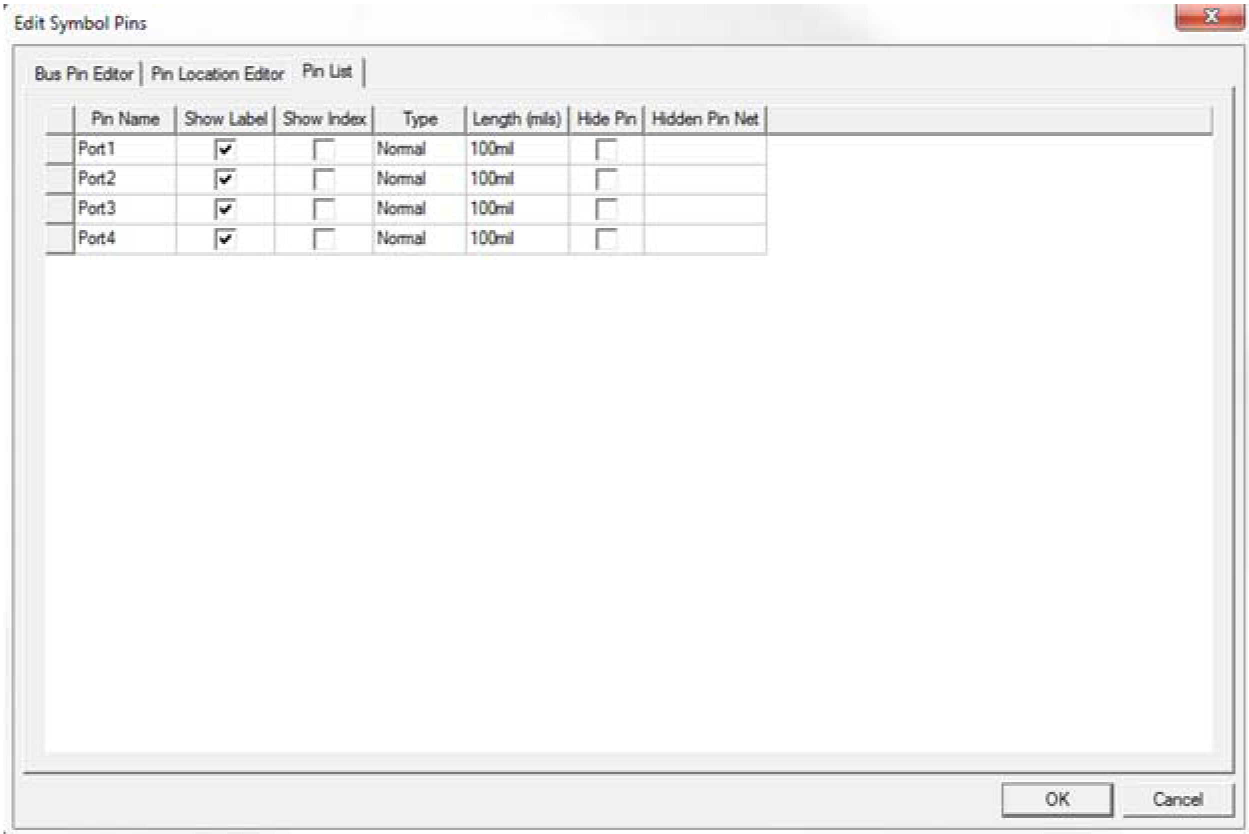
> **Note:**
>
> See Reference Nodes on S-Parameter Elements in the S-parameter Technical Notes for how the Circuit solver handles reference node configurations.

Clicke of the five **PinDirection** options to change the symbol shown to a fixed configuration.

- **All Left** puts all pins on the left side
- **All Right** puts all pins on the right side
- **I/I+N** puts the first half of the pins in order on the left side and the remaining pins on the right side
- **Odd-Even** puts every other pin on the left side and the remaining pins on the right side
- **Manual Edit** enables the two list boxes in the bottom left quadrant. When those boxes are enabled, the user can drag pins from one box to the other to switch sides and drag pins within a box to switch the pin order.

Pin List

The **Pin List** window displays and sets values for the properties of symbol pins.



The available pin properties include:

- **Name** – This editable cell displays and sets the pin name. To specify or change the name of a pin, click in the **Name** cell for that pin, type the name, then press **Enter** or click in another cell.
- **ShowLabel** – This check box controls visibility of the pin name in the schematic editor. For a new pin, it is unchecked by default.
- **ShowIndex** – This check box controls visibility of the pin number in the schematic editor. For a new pin, it is unchecked by default.
- **Type** – This drop-down menu sets the pin type, the options for which include **Normal** (the default), **ANSI In, ANSI Out,** and **Zero Length.** To change a pin's type, click in its **Type** cell, then click the appropriate option.
- **Length** – This editable cell sets the length of the pin stem (the graphical line associated with the pin port symbol). By default, new pins have a pin stem length of 10. To change the length of a pin, click in its **Length** cell, type a new value, then press **Enter** or click in another cell.
- **HidePin** – This check box controls whether or not a pin is visible in the schematic editor. For new pins, it is unchecked by default.

- **HiddenPinNet** – This cell, editable when **Hide Pin** is checked, specifies the net (the circuit node) to which a pin is connected if it is hidden. To specify a net, click in the **Hidden Pin Net** cell, type a net name, then press **Enter** or click in another cell.

Changing Pin Properties

To change a pin property, either:

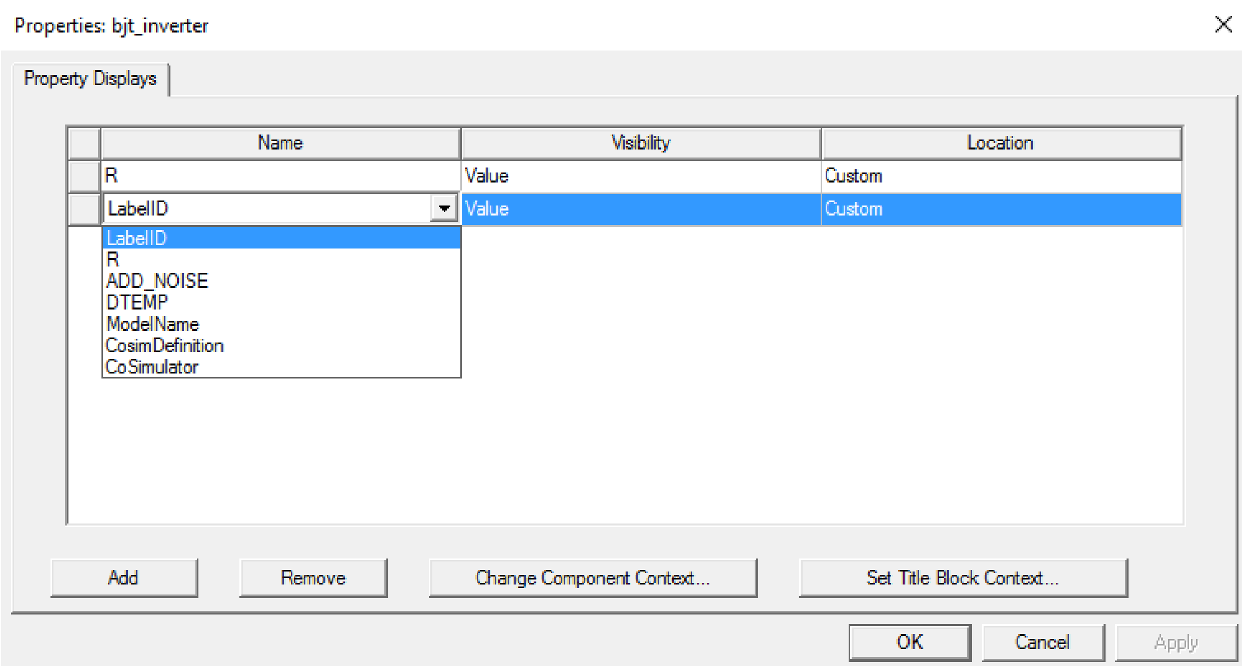
- Click the cell that displays the pin property value that you want to change and modify the property value appropriate to its type as described above.
- To commit the new value and continue editing pin properties, click another property value.

Committing Changes

Clicking **OK** in the window commits changes to the component symbol made in all three tabs.

Symbol Property Display Setup

The **Symbol** > **Property Display Setup** command opens a **Properties** window in which you can add or modify how the names, values, and locations of properties associated with the component or object (such as a title block) are represented by the current symbol.



The Symbol Property Displays Window

This window lists symbol display properties and controls their presence, visibility, and position. Every entry in this window corresponds to a property on a component associated with the symbol in question. Whoever creates or edits a component determines which property to add.

Some properties such as [LabelID] can appear in multiple components. The component is set automatically if you are editing an existing symbol. To change the reference component, click **Change Component Context**.

> **Note:** [LabelID] may display as **[InstanceName]** or *** in the Symbol editor.

Columns in the grid are:

- **Name** – Click in this cell to type a property display name, or to select an existing property display from a list. The component selected via **Change Component Context** determines which property displays are listed for selection.
- **Visibility** – Click in this cell to set the visibility (**None**, **Name**, **Value**, or **Both**) of a property display.
- **Location** – Click in this cell to set the location (**Left**, **Top**, **Right**, **Bottom**, **Center**) of the property display relative to the symbol center.

> **Note:** You can further position a property display by rotating it (**Ctrl+R**), or by dragging it to another location. When you do this for a given property display, its Location value is set to **Custom**.

The actions available in this window are:

- Click **Add** to add a new property display entry to the list.
- Click **Remove** to remove the selected property display.
- Click **Change Component Context** to open the **Select Definition** window, to select the component definition on which property displays are based. The properties of the selected component are then available in the **Name** cell properties list.

> **Note:** Because a symbol can be used by multiple components with differing properties, any property name can be entered by the user. If a component using the symbol does not have a property with that name, the symbol property isn't used in that component.

- Click **Set Title Block Context** to make the list of default page properties, and others as specified, available for display. Default properties include: ProjectPath, Project, Design, Title, Author and Date. This context is selected by default if the symbol has three or more propdisplays for default page properties.

Symbol Grid Setup window

The **Symbol** > **Grid Setup** window sets visibility, colors, resolution, and snapping for the Symbol Editor's alignment grid. It contains the following controls.

- **Major** — Specifies the spacing of the major grid lines (default in mils). To change the color of the major grid lines, click the **Major** color button, specify a color in the **Color** window, then click **OK**.
- **Minor** — Specifies the spacing of the minor grid lines (default in mils). To change the color of the minor grid lines, click the **Minor** color button, specify a color in the **Color** window, then click **OK**.
- **Show Grid** — Toggles grid line visibility.
- **Snap to Grid** — Controls whether graphics and text placed on the grid automatically snap to the nearest grid intersection.
- **Background Color** — To set the background color used for symbol editing, click the color box to open its window, specify a color, then click **OK**.
- **Save as Default** — Save the current **Grid Setup** values as defaults for future Electronics Desktop sessions.
- **Defaults** — Restore all **Grid Setup** settings to their defaults.
- **OK** — Commit changes and close the window.
- **Cancel** — Close the window without committing changes.

Note:

- To ensure electrical connectivity among schematic elements, placed symbol pins snap to a 100-mil grid (2.54-millimeter) regardless of the graphical Major and Minor grid line settings. This snapping cannot be deactivated, and the spacing of the connectivity grid cannot be adjusted.
- The units used for grid line spacings are controlled by the **Tools** > **Options** > **General Options** window.
- Use of the **Minor** grid setting varies depending upon the active editor. In the Layout Editor, the minor grid-line setting specifies the number of units between each minor grid division. However in the Schematic and Symbol Editors, the **Minor** grid-line setting specifies the number of minor grid lines that appear between major grid lines.

Using the Component Editor

The Electronics Desktop **Component Editor** supports the viewing, modification, and saving of component information for use in the development of circuit designs. You can create new components, edit existing components within a project file, and manage components within external libraries. After modifying a component, the information can be saved, used to define a design, or exported. Component definitions are stored in library files with an extension of **.aclb**. You can access component definitions using the **Component Editor**.

To start the **Component Editor**, open a project and select **Tools > Edit Libraries > Components**. When the Edit Libraries window opens, use its controls to open the Edit

Component window, then create a new component, or select an existing one, and click **Edit** to open the **Component Editor**.

Creating and Editing Components

Use the Edit Component window to:

- Specify or change the component name, description, associated bitmap, and default property values, including those necessary for netlisting, co-simulation, and **Solver-on-Demand** operation, as appropriate
- Specify or respecify a graphical symbol to represent the component in the schematic editor
- Specify or change the component terminal properties
- Specify or respecify a footprint to represent the component in the layout editor

The changes you make when creating or editing a library component become part of the current project. The newly edited component is saved to the project, not back to the library. If the edited component is used in any schematics in the project, all default values of any modified parameters are updated immediately.

- To make a new or modified component available for use in other projects, save it to a component library (**.aclb**) file using the **Export** command in the Edit Libraries window.
- To update another project with your updated component(s), select its icon in the **Project Manager** window, then click **Project Tools** > **Update Definitions** on the **Tools** menu.

> **Note:** If you are creating a new component, it is useful to first identify and, if necessary, create the necessary dependencies before defining your component. See Component Creation Process in the introduction for further information.

Related Topics

Editing an Existing Component

Creating a New Component

Editing an Existing Component

To edit an existing component, do either of the following:

- From the **Project Manager** window, expand the **Definitions** /**Components** subfolder for the project that contains the component you want to edit. Double-click the entry for the component you want to edit, or right-click the entry and select **Edit Component**.
- Open the **Tools > Edit Libraries** > **Components** window, then locate and select the component that you want to edit. Click **Edit Component**, or double-click the selected entry.

The Edit Component window opens, displaying the definition of the selected component for editing.

Creating a New Component

To create a new component, do either of the following:

- Select **Tools** > **Edit Libraries** > **Components**, and click **Add Component** to open the **Edit Component** window.
- Open the **Tools > Edit Libraries > Components** window, then locate and select an existing component on which to base your new component. Click **Clone Component**. and a renamed copy of the selected component is added to the list. Double-click the entry for the new component, or click **Edit Component** to open the **Edit Component** window.

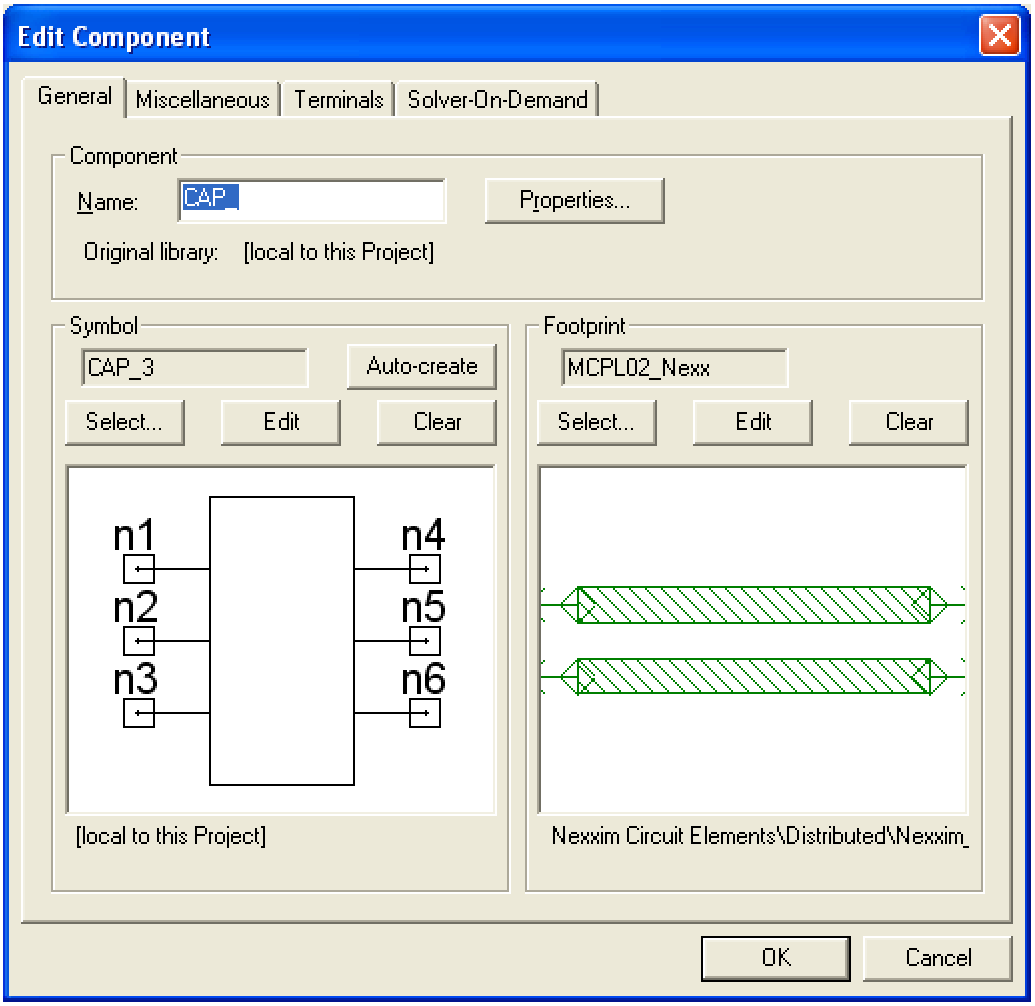
The Edit Component Window

The **Edit Component** window is used to create new components, or edit existing components, in a project file. To show the **Edit Component** window, do either of the following:

- Double-click a component definition in the project tree, or double-click a component listed in the **Edit Library** window.
- Open a project and select **Tools > Edit Libraries > Components**. When the Edit Libraries window opens, use its controls to open the **Edit Component** window.

Components General Tab

Click the **General** tab to specify all parameter values for the component and assign symbols and footprints on their respective libraries. You can also start the symbol and footprint editors on the window, however, you must exit the component editor to complete the symbol or footprint edits. The symbol and footprint preview windows at the bottom of the window are automatically updated to reflect whatever changes you make to the symbols and footprints you modify.



Component Panel

The Component group box of the General Tab window displays the component Name. The Original Library name is displayed if found. To access and edit the component properties, click **Properties** next to the **Name** text window. In the Properties window, add, edit or remove default or local properties and values.

Symbol Panel

The Symbol group box of the General Tab window allows you to search by symbol name, and also displays a picture of the symbol. Choose any symbol which has an equal or greater number of pins than the component specified in the Component Name field. You can then choose to **Auto-create** the component.

The library of origin of the component is displayed beneath the component display in the **Symbol** area; if the component is **local to this Project**, that message is displayed. When you click **Select** (**...**), the Select Definition window opens and allows you to choose a definition for the symbol from a list that is displayed. Click **Edit** to modify the symbol. A confirmation message is displayed with a reminder that the Component and Definition editors is closed (if open) and that all changes are saved if you choose to continue. Click **Clear** to clear all selections and displays in the Symbol area.

Footprint Panel

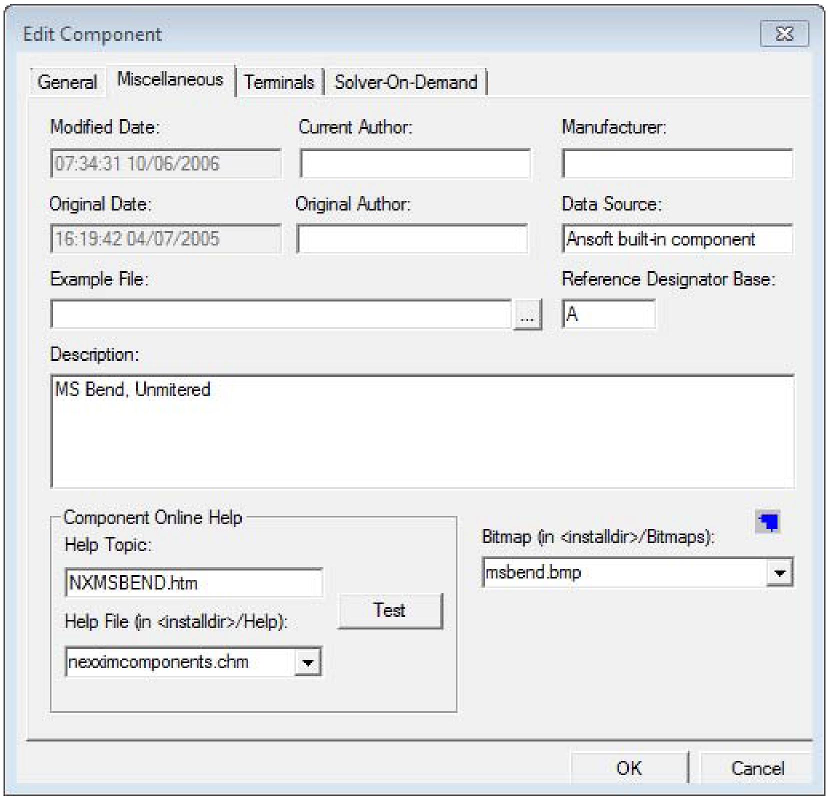
The Footprint group box of the General Tab window allows you to search by footprint name, and is used to display a picture of the footprint. Choose any footprint which has an equal or greater number of pins than the component specified in the Component Name field.

The library of origin of the footprint is displayed beneath the component display in the Footprint area; if the component is **local to this Project**, that message is displayed. When you click **Select**, (**...**) the Select Definition window opens and allows you to choose a definition for the footprint from a list that is displayed. Click **Edit** to modify the footprint. A confirmation message is displayed with a reminder that the Component and Definition editors is closed (if open) and that all changes are saved if choose to continue. Click **Clear** to clear all selections and displays in the Footprint area.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Components Miscellaneous Tab

Use the miscellaneous fields to enter component information such as the name, manufacturer, description, ref des, etc. You can also specify a bitmap for the **Project Tree** and an associated help description (if you use your own documentation). Help files must be located in **<Installation Directory>\Electronics Desktop\help**.

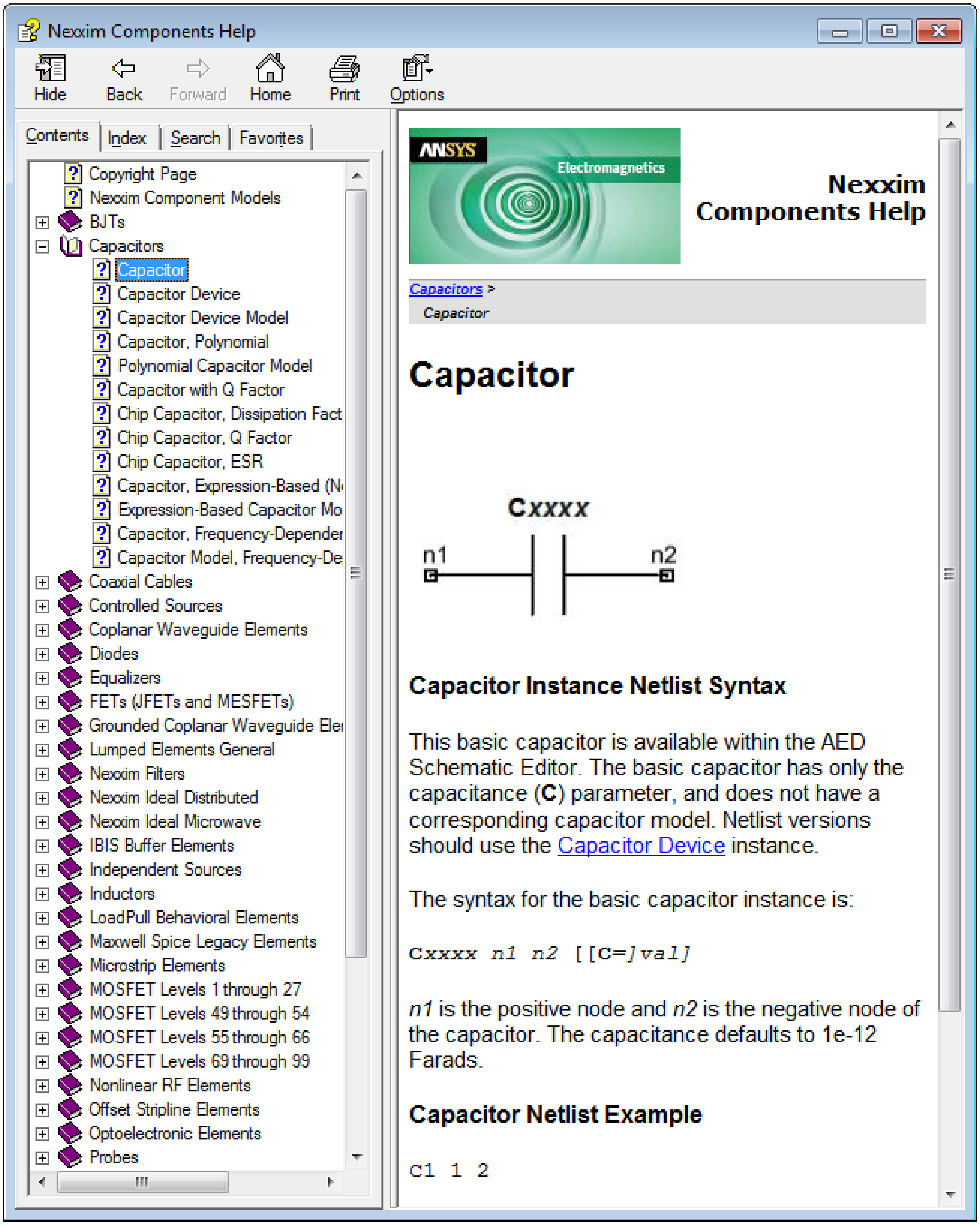


The top two rows of the **Miscellaneous** tab window displays the **Modified Date** and **Original Date** of the component, and allows you to specify the following:

- Current Author
- Manufacturer
- Original Author
- Data Source

You may also specify an **Example File** directory for the component, browse for the directory if necessary (...) and specify it's **Reference Designator**, or enter a text **Description** for the component. The default **Bitmap** of the component is displayed at lower-right, and you may use the adjacent drop-down menu to specify a different bitmap file.

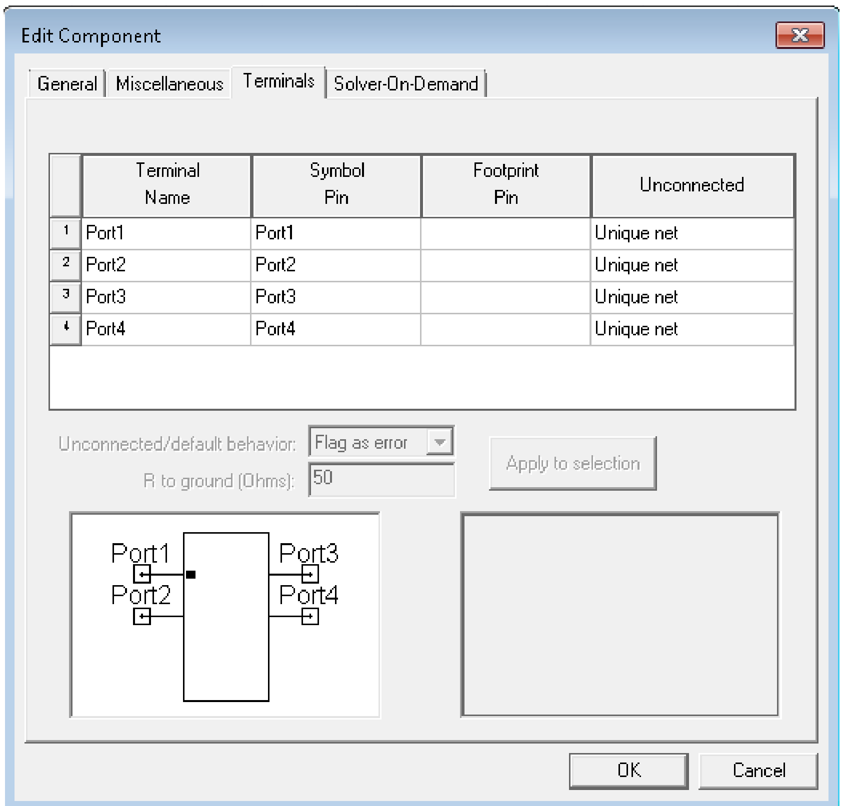
The **Component Help** area displays the default HTML help file associated with the component, and allows you to specify an alternative help file of your own. You may also **Test** the specified help file, which displays the component's HTML help page.


Nexxim Components Help
Hide
Back
Forward
Home
Print
Options
Contents
Index
Search
Favorites
Copyright Page
Nexxim Component Models
BJTs
Capacitors
Capacitor
Capacitor Device
Capacitor Device Model
Capacitor, Polynomial
Polynomial Capacitor Model
Capacitor with Q Factor
Chip Capacitor, Dissipation Fact
Chip Capacitor, Q Factor
Chip Capacitor, ESR
Capacitor, Expression-Based (N
Expression-Based Capacitor Mo
Capacitor, Frequency-Depender
Capacitor Model, Frequency-De
Coaxial Cables
Controlled Sources
Coplanar Waveguide Elements
Diodes
Equalizers
FETs (JFETs and MESFETs)
Grounded Coplanar Waveguide Ele
Lumped Elements General
Nexxim Filters
Nexxim Ideal Distributed
Nexxim Ideal Microwave
IBIS Buffer Elements
Independent Sources
Inductors
LoadPull Behavioral Elements
Maxwell Spice Legacy Elements
Microstrip Elements
MOSFET Levels 1 through 27
MOSFET Levels 49 through 54
MOSFET Levels 55 through 66
MOSFET Levels 69 through 99
Nonlinear RF Elements
Offset Stripline Elements
Optoelectronic Elements
Probes
ANSYS
Electromagnetics
Nexxim Components Help
Capacitors >
Capacitor
Capacitor
Cxxxx
n1
n2
Capacitor Instance Netlist Syntax
This basic capacitor is available within the AED Schematic Editor. The basic capacitor has only the capacitance (C) parameter, and does not have a corresponding capacitor model. Netlist versions should use the Capacitor Device instance.
The syntax for the basic capacitor instance is:
Cxxxx n1 n2 [[C=]val]
n1 is the positive node and n2 is the negative node of the capacitor. The capacitance defaults to 1e-12 Farads.
Capacitor Netlist Example
C1 1 2

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Components Terminals Tab

To line up component terminals in an envisioned order, Click the **Terminals** tab to associate each symbol pin with the proper terminal by first ordering the symbol pins, then selecting the corresponding symbol. This allows you to visually associate the symbol pins with the netlist properties for the simulation model. The symbol and footprint preview windows at the bottom of the window are automatically updated to reflect whatever changes you make to the symbols and footprints you modify.



The following controls are available:

In the **Terminals** display window, click one or more terminals directly, which then enables the following window options:

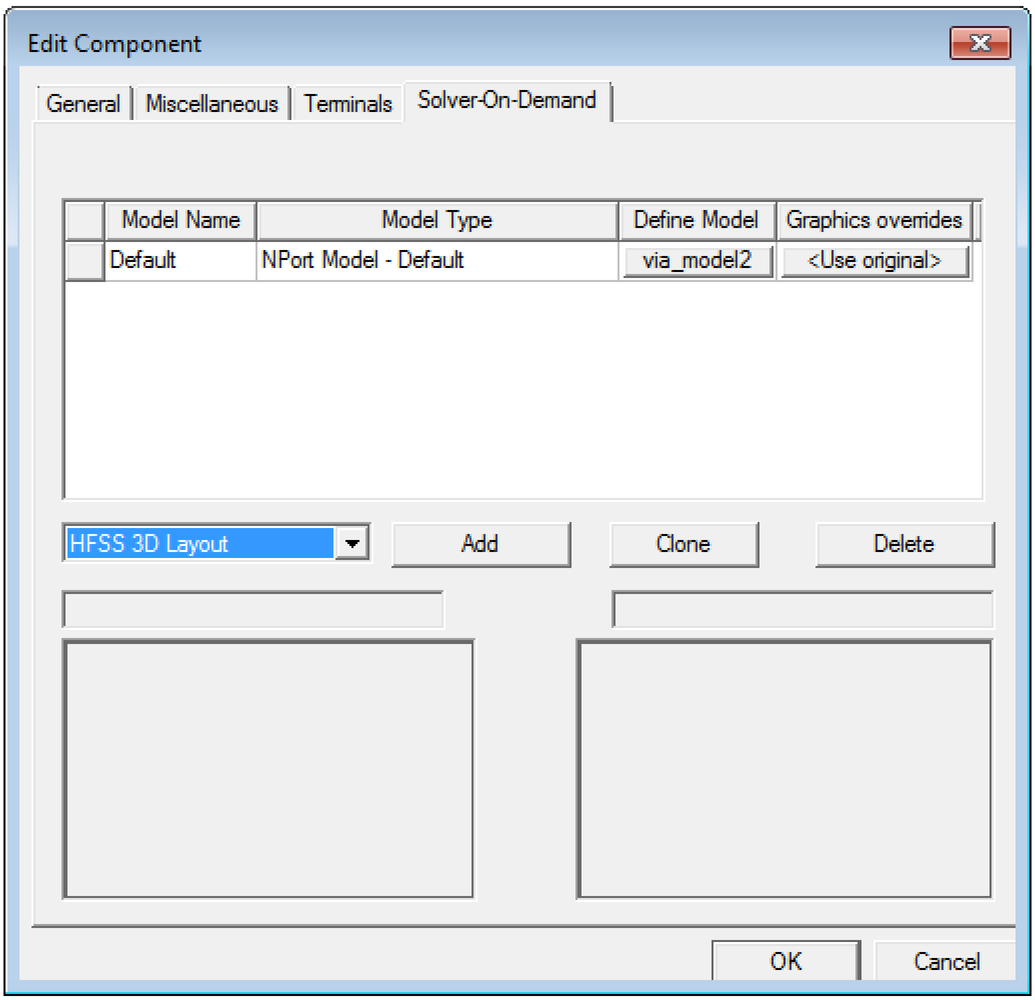
Unconnected/default behavior – Used to specify the response when a pin is not connected to a wire or port in a schematic. Choose between the following drop-down menu actions: **No action** (default), **Flag as error**, and **Grounded**. Click **Applytoselection** to apply your changes to the selected terminals.

R to Ground – If you select **Grounded** for the **Unconnected/default** behavior, then specify the resistance to ground in the **R to Ground** entry window. Click **Applytoselection** to apply your changes to the selected terminals.

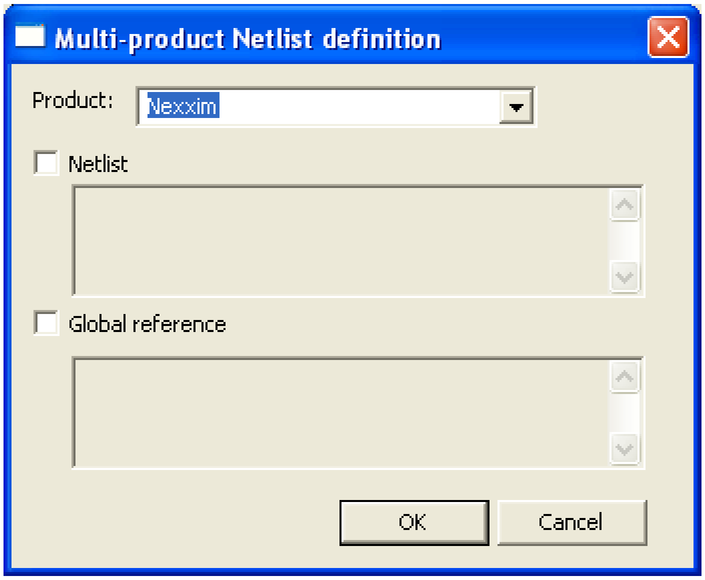
Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Components Solver On Demand Tab

Solver-on-Demand is a specialized routine to set up co-simulation using multiple products. When you first open the **Solver-on Demand** tab of the **Edit Component** window for a particular component, the only **Model Name** listed is the default multi-product type, **DefaultNetlist**.



DefaultNetlist is a default co-simulation definition type, provided automatically for each component, that allows you to define a different netlist for each Electronics Desktop product. Click the button in the **DefaultNetlist** row to open the **Multi-product Netlist definition** window.



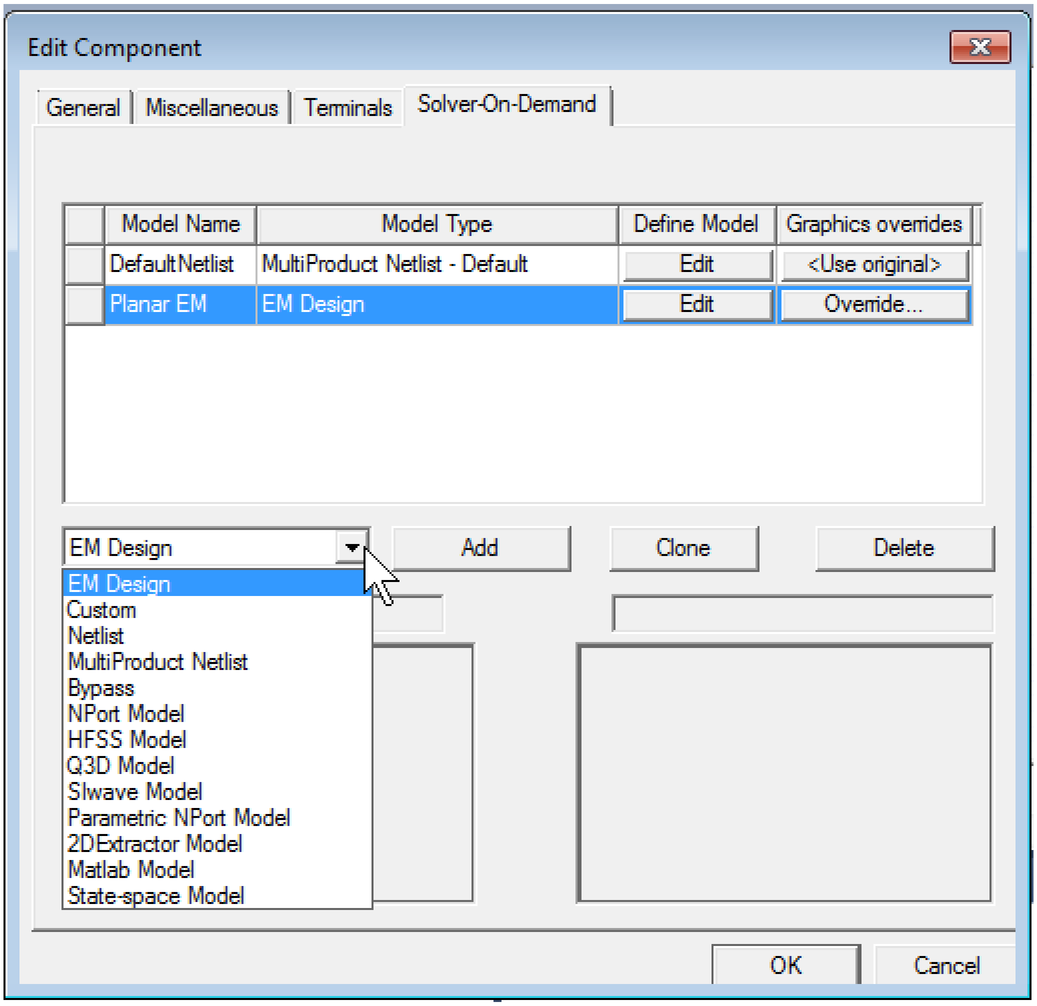
- Click the drop-down menu **Product** menu to choose which Electronics Desktop product netlist to define.
- Click **Netlist** to define a netlist.
- Click **Global reference** to define a global reference.
- Click **OK** to implement your changes, close the window, and return to the **Edit Component** window.

Multiple Co-simulation Models

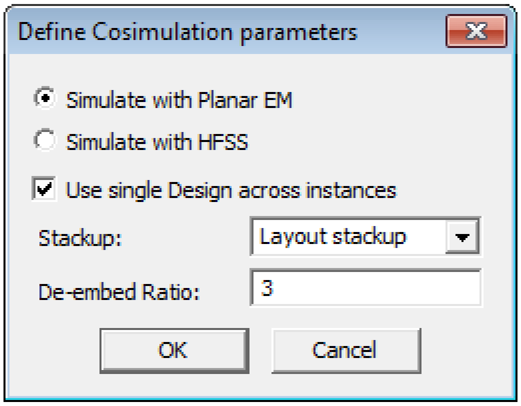
Electronics Desktop allows for multiple representations of models that can be cosimulated. In the **Solver-on-Demand** tab of the **Edit Component** window, click in the drop-down menu to choose from a list of models to add to the list of **ModelNames**.

To add a model for Solver On Demand:

1. Make a selection on the model type drop-down menu.



2. Click **Add**. A new model of the selected type is added to the list. The symbol and footprint preview windows at the bottom of the window are automatically updated to reflect whatever changes you make to a component's symbol/footprint definitions.
3. Click in the **ModelName** field for the new model and type the name you want to use for this model (or accept the default in the **ModelName** field). Note that when you highlight a **Model Name**, its symbol and footprint (if defined) are automatically previewed in the windows at the bottom of the window. These preview windows are automatically updated to reflect whatever changes you make to the component's symbol/footprint definitions.
4. Click **Edit** in the **Define Model** field (or with a **Custom** model name) to open a **Cosim** or **Netlist** definition window that is specific to the model type. Here is the window for the Planar EM model shown in the MSBEND component:



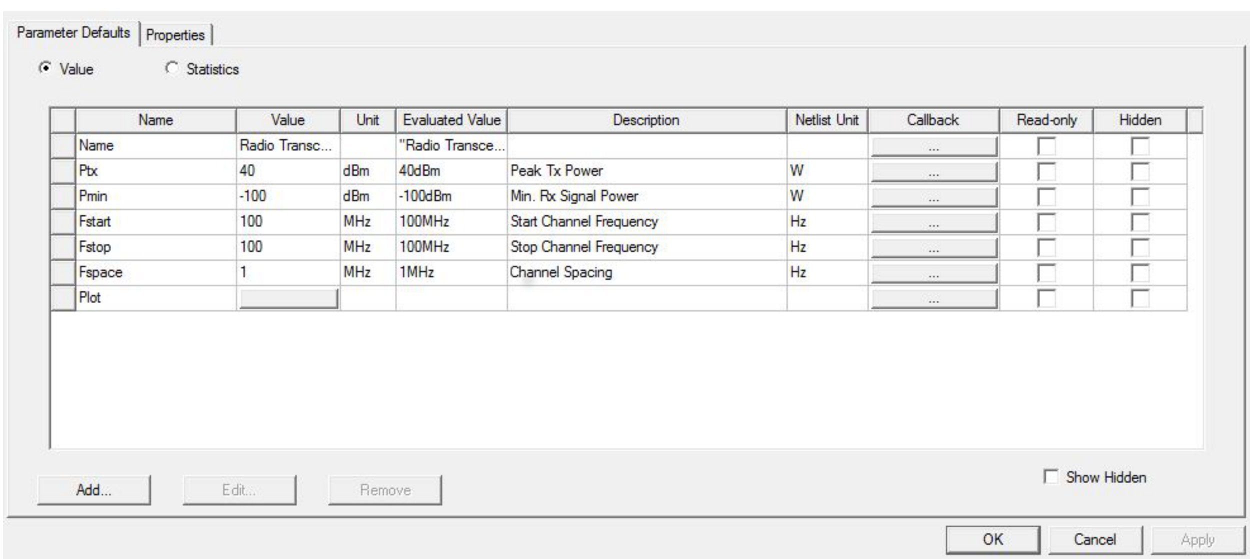
5. Repeat steps 1 through 3 to create all the models you wish to use for this type of component.
6. Click **OK** when you have defined all the models for this component type.

Repeat the above procedure to specify models for all components that is simulated using **Solver-on-Demand**.

For a complete description of the Solver On Demand tab and how to define model types, see Co-simulation.

Edit Component Properties window

To open the **Properties** window for a selected component, double-click the component definition in the **Project Manager** window to open the **Edit Components** window. Then, in the **Edit Components** window, click **Properties** to open the window.



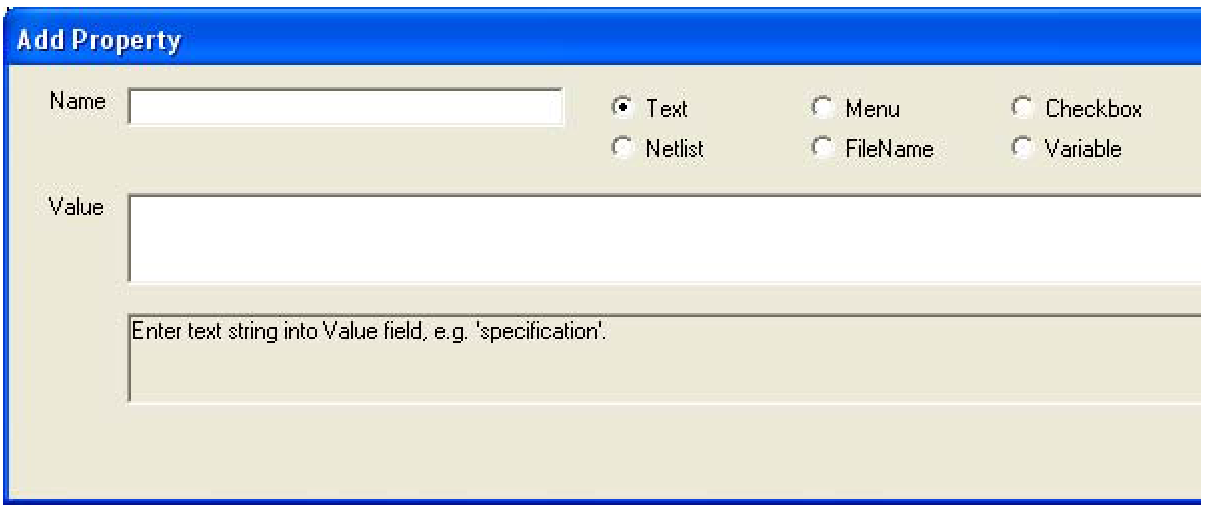
Parameter Defaults Tab

This tab displays and sets the default values for component parameters for general analysis (Value display) or statistical analysis (Statistics display).

When this option is selected, the **Parameter Defaults** tab lists, displays, and sets the default values for component parameters during general analysis.

Adding Parameters

Use **Add** to add a parameter to the definition. You are prompted to define the parameter name, value, and value type (text input, menu, check box, etc.). You may **not** add reserved system parameters to the definition.



Removing Properties

To remove a property on a component, select the property and click **Remove**. The property is deleted on the component definition (model) and from all instances in the design.

> **Warning:**
>
> Removing a property from a component from an Electronics Desktop component library may produce undesirable results when the component is simulated.

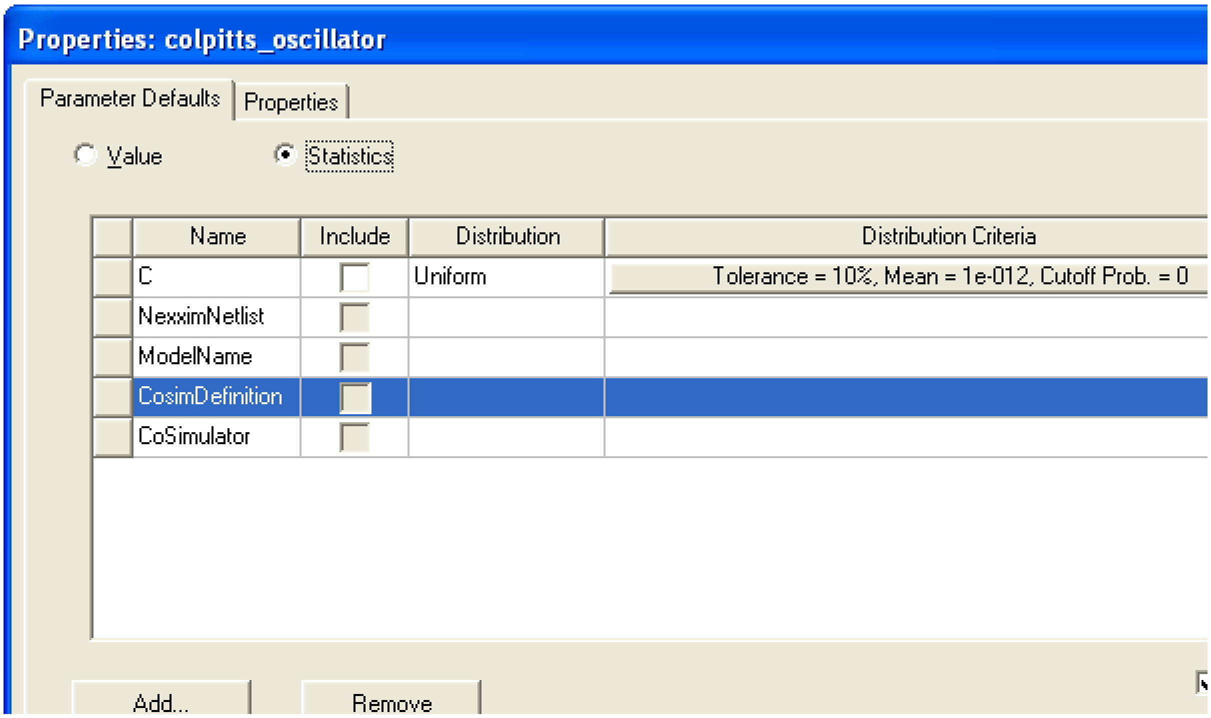
Related Topics

Parameter Statistics Display

Reserved Component Parameters

Parameter Statistics Display

When this option is selected, the **Parameter Defaults** tab lists, displays, and sets the default statistics for component parameters during general analysis.

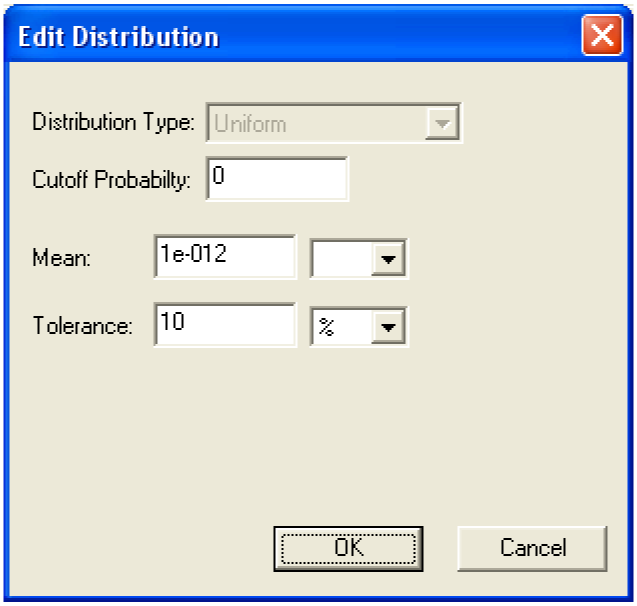


Name —Displays the names of component parameters.

Include —Controls whether or not the associated component parameter is varied during statistical analysis. Check **Include** for each parameter you want Electronics Desktop to vary during statistical analysis.

Distribution —Displays and controls whether the parameter value distribution is Uniform or Gaussian. To change the Distribution setting, click in the cell to open the options, then click the option you want.

Distribution Criteria —Click the display bar to open the **Edit Distribution** window. The **Distribution Type** drop-down menu displays whether the distribution is Uniform or Gaussian, and you can specify values for **Cutoff Probability**, **Mean**, and **Tolerance**. Units for **Mean** and **Tolerance** can be set using their adjacent dropdown menus.



Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Reserved Component Parameters

When using **Add** to add a parameter to the component definition in the **Properties** window, you are prompted to define the parameter name, value, and value type (text input, menu, check box, etc.). You may **not** add the following reserved system parameters to the definition:

"Id"

"Status"

"AnalysisSetup"

"AnalysisResults"

"PartName"

"Info"

"NPortData"

"Refbase"

"NumParts"

"CompName"

"Description"

"Manufacturer"

"Datasource"

"Date"

"Symbol"

"Footprint"

"PinCount"

"Refdes"

Component CosimDefinition Property

A netlist is a textual representation of a circuit. The Electronics Desktop simulators extract a netlist on the schematic and perform the simulation on the netlist. The Nexxim simulator uses the Nexxim netlist format. Nexsys components within a Nexxim design use a legacy Ansys netlist format for components.

When an analysis is requested for a modified design, or when **Browse Netlist** is invoked, Electronics Desktop writes the corresponding netlist to the project results folder. The netlist conveys to the simulation engine the connectivity, parameter values, analysis specifications, and other characteristics of the design(s) under analysis. To support this process, every Electronics Desktop component must include a valid definition for the **CosimDefinition** property.

To view the **CosimDefinition** for a component:

1. Open its **Properties** window.
2. Select the **Parameter Values** tab.
3. Click **Edit** for **CosimDefinition**.

The **Multi-product Netlist definition** window opens, displaying the netlist string. Use the window to edit the string if appropriate.

Related Topics

Netlist String Syntax

Netlist String Syntax

The general form for a netlist string is:

<Simulator>Netlist=[netlist_string]

where

<Simulator> is Nexxim | Planar EM | Q3D | etc.

The netlist string may contain the following netlist property syntax:

%<terminal index, 0-based>

Which specifies the name of the net connected to terminal index. A syntax error results if the terminal index is not a positive number, within range. Example:

%0, %1

%_<internal node index, 0-based>

Create a unique net for an internal node. Used when a component has nodes that are not associated with symbol terminals and are not tied to a global node. Syntax error if node index is not a positive number, within range. Example:

%_0, %_1

@<propname> | @(<propname>)

Value of property named propname. Syntax error if propname is empty or contains spaces. Example:

@R, @IDSS

?<propname>(<expr1>)[:(<expr2>)]

If property named propname exists, substitute expr1, else expr2. Also, expr1 and expr2 may contain additional substitutions. Syntax error if propname is empty or contains spaces. Example:

?IDSS(IDSS = @IDSS):(IDSS = 0.05)

~<propname>(<expr1>)[:(<expr2>)]

If property named propname doesn't exist, substitute expr1, else expr2. Syntax error if propname is empty or contains spaces. Example:

~SUB(SUB = MS)

?(<propname>==<value>)(<expr1>)[:(<expr2>)]

?(<propname>!=<value>)(<expr1>)[:(<expr2>)]

~(<propname>==<value>)(<expr1>)[:(<expr2>)]

~(<propname>!=<value>)(<expr1>)[:(<expr2>)]

Same as "?" and "~" above except the first term is evaluated for equal to (==) or not equal to (!=) and if true, substitute expr1 else expr2. Example:

?(sim==fullwave)(NSUM=@NSUM):(F0=@F0)

***<propname>(<expr1>)**

If property named propname has changed from default definition value, then substitute expr1. Syntax error if propname is empty or contains spaces. Example:

*IDSS(IDSS = @IDSS)

&(<expr>)[^(pname1,pname2, . . .)]

Add all properties that have changed from default definition value, except those in the optional exclusion list. In expr, “$” can be used to represent the property name, and “#” the property value. Example:

&($=#)^(Model)

netlists all properties that changed on their default value except the Model property.

\n

A new line marker to inform the netlister to insert a new line. Example:

R@ID %0 %_0 @R \n C@ID %_0 %1 @C

****

The backslash is used as an escape character. The character following the backslash is not processed but added to the netlist string as is. To add a single “\”, use “\\”. Example:

R:@ID %0 %1 R=\{\2*@R\}

Defined Variables

- $SYSLIB is expanded to **<InstallationDirectory>/syslib** upon netlisting, where **<Installation Directory>** is the directory into which Electronics Desktop was installed during setup.
- $USERLIB is expanded to **<InstallationDirectory>/userlib** upon netlisting.
- $PERSONALLIB is expanded to **<ProjectDirectory>/PersonalLib** upon netlisting, where **<ProjectDirectory>** is the location you specified for your Electronics Desktop projects.

Global Reference String in Electronics Desktop

A global reference property processes its value the same way as the netlist string and places the result in the top-level (global) part of the circuit file.

<Simulator>GlobalRef=[string]

where

<Simulator> is Nexxim | Q3D | etc.

Additional Examples

The following additional examples help to demonstrate netlist string syntax.

NexximNetlist= CPLE@ID %0 %1 %2 %3 ?Z(Z=@Z) ?E(E=@E) ?F(F=@F) *A(A=@A)

“Z”, “E”, and “F” use the "?" conditional because the component definition values assigned to these parameters are different from those in the engine and should always be netlisted if they are present even if the user has not changed them. The “A” parameter uses the "*" conditional because its component definition value matches that of the engine and it needs to be netlisted if the user changes the instance value.

NexximNetlist= R@ID %0 %1 @R ?TC1(TC1=@TC1 ?TC2(tc2=@TC2)) ?TJ(TJ=@TJ) ?TNOM(TNOM=@TNOM)

In this example, TC1 is netlisted only if TC1 exists, and TC2 is netlisted only if TC1 and TC2 exist.

NexximNetlist= MSBENDO@ID %0 %1 w=@Length sub=@Substrate

“w” and “sub” is netlisted.

NexximNetlist= MOSFET@ID %0 %1 %2 %3 MODEL=@Model &($=#)^(Model)

Model is netlisted. The “&($=#)^(Model)” argument designates that all parameters should be netlisted as “name=value”, except for “Model”.

NexximNetlist = D@ID %0 %1 1N914

NexximGlobalRef = .LIB $SYSLIB/Vendor/D.lib

which may netlist to:

> D22 net_3 net_4 1N914
>
> < rest of design netlists ...>
>
> .LIB $SYSLIB/Vendor/D.lib

NexximNetlist= R@ID %0 %1 R=\{\2*@R\}

If ID is 1 and R is 100, the netlist is:

> R1 net_0 net_1 R={2*100}

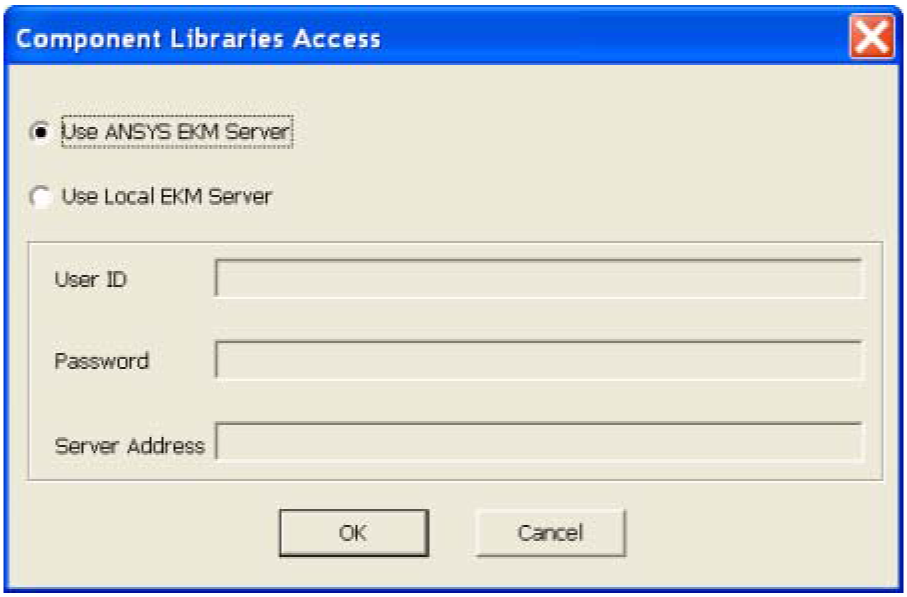
Using Vendor Components

Electronics Desktop allows you to download vendor components from a server. After being downloaded, libraries are automatically configured so these components can be used in Electronics Desktop.

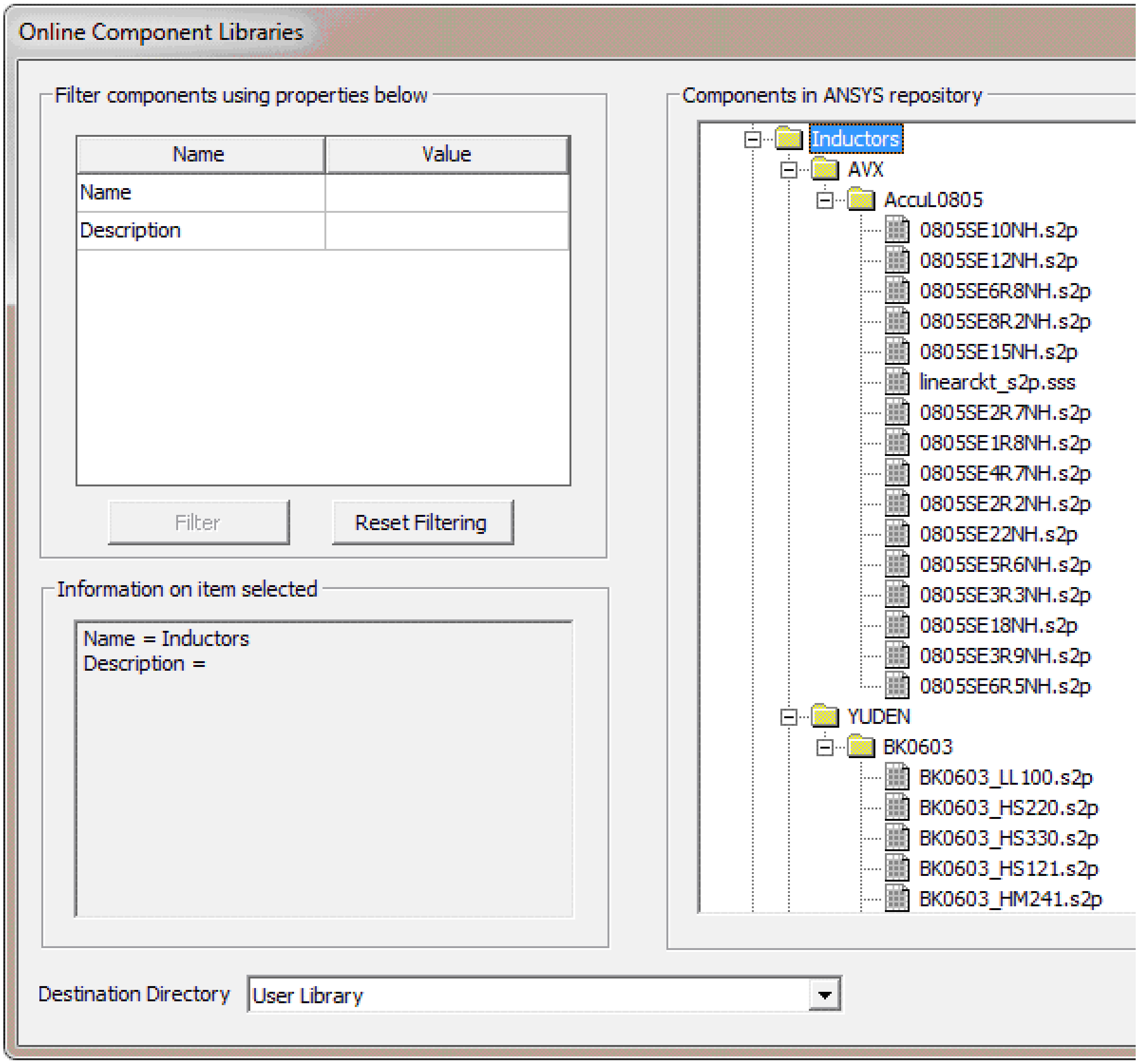
Some installations of Electronics Desktop include a set of RF Vendor Library components. To use these elements, you unzip the files into the **syslib/Vendor Elements** directory in **Electronics Desktop** installation.

Downloading Vendor Components

Electronics Desktop allows you to download vendor components from a server. After being downloaded, libraries are configured so these components can be used in Electronics Desktop. Server files to be downloaded must be set up in a predefined manner with certain property values. When a Electronics Desktop circuit is active, click **Tools > Library Tools > Download Vendor Components** to open the **Component Libraries Access** window.



Select a server, then enter the User ID, Password, and Server Address. Then click **OK** to open the **Online Component Libraries** window:



The web library window displays files as well; click on '+' next to folders for expansion.

Filter components using properties

- Shows filter criteria, and is used for choosing values for file properties, such as Vendor, Type, and NumPorts.
- Users can search for files and folders using the 'Filter' option.
- The Series property allows you to enter text, including regular expression values with wild card characters, such as "NEC202*".
- Click **Reset Filtering** to set all properties to their most inclusive values ("Any" or "All"). When you choose a new value in the **Filter components** pane, only folders satisfying the filter values are shown in the center pane.

Information on item selected

- Displays 'Name' and 'Description' property of the files and folders as the user navigates through the files in the center pane.

Components in Ansys repository

- The center pane of the window shows all the folders that satisfy the filter criteria.
- Folders are organized into a tree control, so that they can be identified correctly (you must have the full path to uniquely identify a series folder). If you move a cursor over a series folder and hover, text will appear with helpful information about that folder.
- Drag and drop any folder on the center-tree control to the right pane to copy the folder and display it in the right pane.

Components selected for download

- The right pane shows the component folders you have chosen for downloading and behaves the same as the center pane (tree control, hover information).
- If you drag something outside of the right pane and release it, the dragged folder is removed on the pane and cannot be downloaded.

Destination Directory

- For 'Vendor Components', the "Destination Directory" dropdown menu has two choices: User Library and Personal Library.
- For 'Other Files' and 'Project Archives', the drop down has three choices: User Library, Personal Library and a 'Browse…' option. 'Browse…' option allows users to download selection to a directory of their choice.
- When you are happy with the contents of the right pane, click **Download Selected** and all files in the selected folders are downloaded to the Destination.

Using RF Vendor Library Components

The RF Vendor library contains vendor-specific components for Electronics Desktop RF installations. The library is delivered as a compressed file, **Vendor_Libraries.zip**, in the **syslib/Vendor Elements** folder under your Electronics Desktop installation directory.

Here are the steps to follow to enable and place these elements.

Unzip the library directory:

1. Browse to the **syslib** folder in your Electronics Desktop RF installation. The directory path should be something like the following:

   `C:\Program Files\AnsysEM\v`*`xxx`*`\`*`Win64`*`\syslib`

The entry, *xxx*, is the Ansys EM release number. The operating system field (*Win64* in the example, for the Windows OS) is *Linux64* for the Linux OS.

2. Ensure you have write permission in the **syslib** folder.
3. Open the folder **Vendor Elements**.
4. Locate the file **Vendor_Libraries.zip**.
5. Using any compression tool, extract the contents of the zip file to the **Vendor Elements** folder.

   The folder **Vendor_Libraries** should appear in the **Vendor Elements** directory.

> **Note:**
>
> Since configuration happens automatically at Ansys Electronics Desktop startup, if you unzip the archive while **Electronics Desktop** is open, you may need to save work, close, and restart **Electronics Desktop**.

Place the Component, select Vendor, Series, and Part.

1. In the **Component Manager** group box of Electronics Desktop, you should see a **Vendor Elements** directory with the **Vendor_Libraries** folder. Expand the folder, the **vendors** folder, to display a listing of the component types in the library (Amplifier, Capacitor, etc). Each component type represents a number of components from different vendors.
2. Double-click the icon for the type of part you want to place, then drag the symbol into the design area.
3. Right-click the component, and edit the **Properties**.
4. Select a **Vendor** on the listing of available vendors for the component type.
5. Select a **Series** on the listing available series on the selected **Vendor**.
6. Verify that the **Part** property has the part you wish to use. If the **Part** menu is not populated, or if the part you want is not in the list, change the **Data Type** property (**S-parameter**, **State-Space**, or **subcircuit**) until the part appears on the Part property menu.
7. Select a **Part** on the listing of available parts in the selected **Series**.
8. Optionally, use the **Sweep** window to select a text array variable containing part names to sweep while simulating the component. The text array variable must have been defined previously. See: Adding a Project Variable Array.
9. Click **OK** to close the **Properties** window.
10. Connect the component to the circuit.

Optionally, browse through the folders under **Vendor_Libraries** in the **syslib/Vendor Elements** directory. You can see what Vendors, Series, and Parts are available for each component type. The file extensions on the Part files indicate the **Data Type**:

- S-Parameter data types use the **.sNp** file extension.
- State-Space data types use the **.sss** file extension.
- Subcircuit data types use either the **.ckt** file extension or the **.lib** file extension.

Encrypted Libraries

Electronics Desktop lets users create personal, encrypted files of design information (typically in SPICE format) that they may be added to other designs via *.lib*, *.va*, or *.inc* statements. Encrypted library files are password-protected. If the user references an imported encrypted *.lib* or *.va* file with a user-defined password, the first time a component from the file is used, **Electronics Desktop** prompts the user for the encryption password. The library file and password are then added to the list of password definitions in the **Tools** > **Password Manager** window.

> **Note:** For security reasons, encrypted components are not saved in the **Most Recently Used** list or the **Favorites** list.

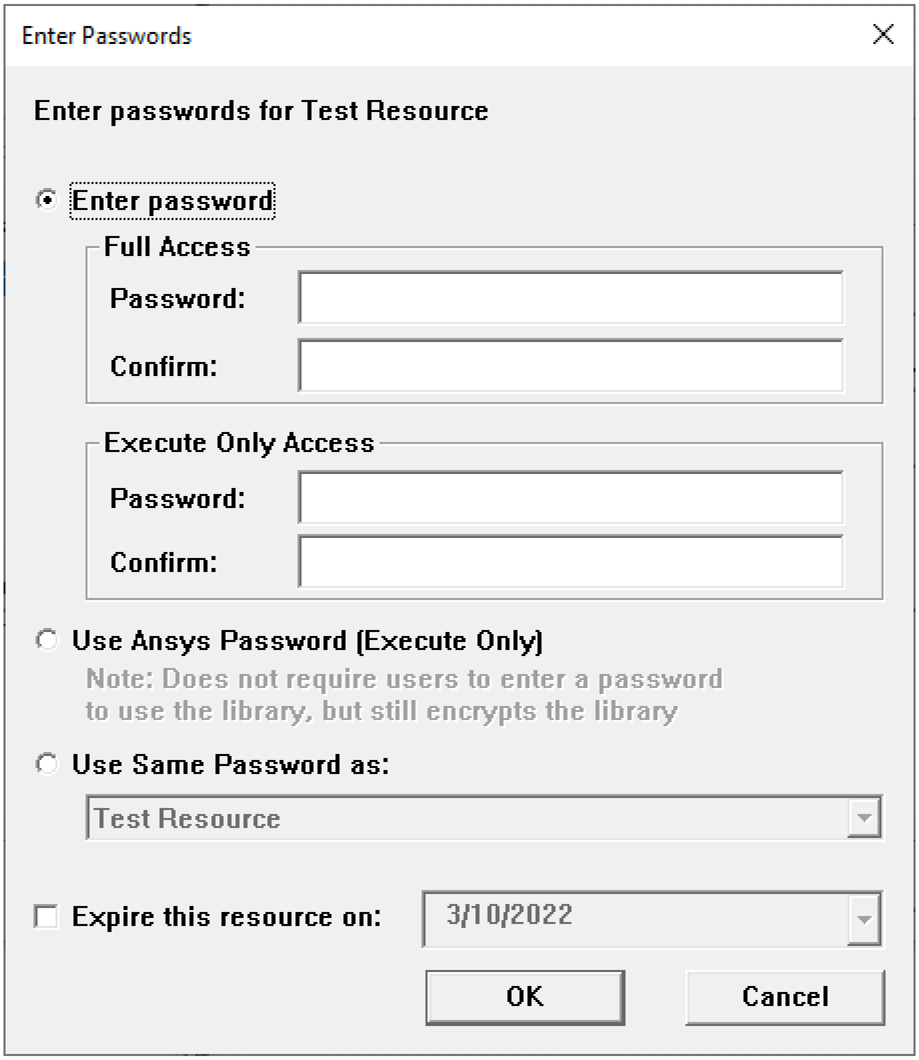
Creating and Managing Encrypted Libraries

Electronics Desktop lets users create encrypted password-protected files (typically in SPICE format) that may be added to other designs via *.lib*, *.va*, or *.inc* statements. Access the encryption tools through the **Tools** > **Password Manager** window.

Password Assignment Types

From the **Enter Passwords** window, accessible on the **Password Manager** > **New...** option, an encrypted file may be assigned a choice of password types:

- **Enter password**: User-defined passwords for an encrypted library file. Entering and confirming **Full Access** passwords grants users simulation, read, and edit permissions, while **Execute Only Access** permits users to add components and simulate a design, but they cannot read or edit the contents.
- **Use Ansys Password (Execute Only)**: A generic, hard-coded execute-only password shared across all Ansys products. Any Ansys product may read an applicable file encrypted in any other Ansys tool using the **Ansys Password**, making it the easiest way to share encrypted library files among Ansys applications if the files are not expected to be viewed or editable.
- **Use Same Password as**: Choose an existing resource on the drop-down menu to use the same password settings and synchronize passwords across multiple library files.


Enter Passwords
Enter passwords for Test Resource
Enter password
Full Access
Password:
Confirm:
Execute Only Access
Password:
Confirm:
Use Ansys Password (Execute Only)
Note: Does not require users to enter a password
to use the library, but still encrypts the library
Use Same Password as:
Test Resource
Expire this resource on:
3/10/2022
OK
Cancel

Note:

1. Passwords are not saved between sessions and must be re-entered each time **Electronics Desktop** is started. Passwords *are* available after one project is closed and another is opened, if the application is not restarted.
2. **Electronics Desktop** and Nexxim use the Advanced Encryption Standard (AES) to encrypt libraries and models. **Electronics Desktop** and Nexxim use a 128 bit block cipher with a 256 bit key.
3. You can encrypt selected parts of a file by enclosing them within a protected “block”. You can set up multiple .protect/.unprotect blocks in the same file, but you cannot nest them.

   e.g.,

   .protect

   This block is encrypted.

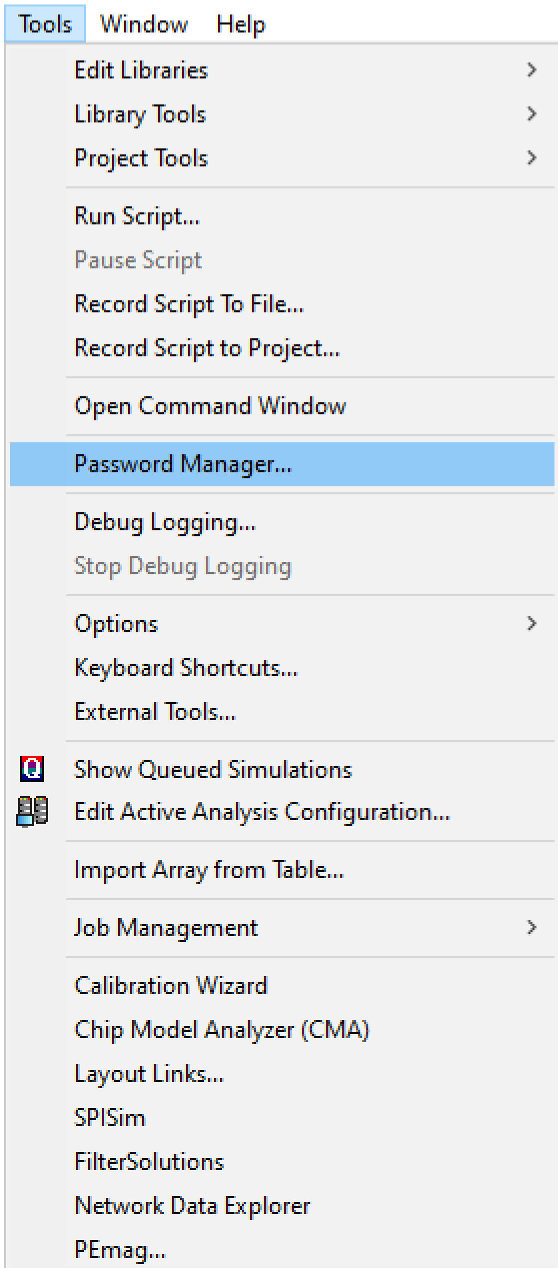
   .unprotect

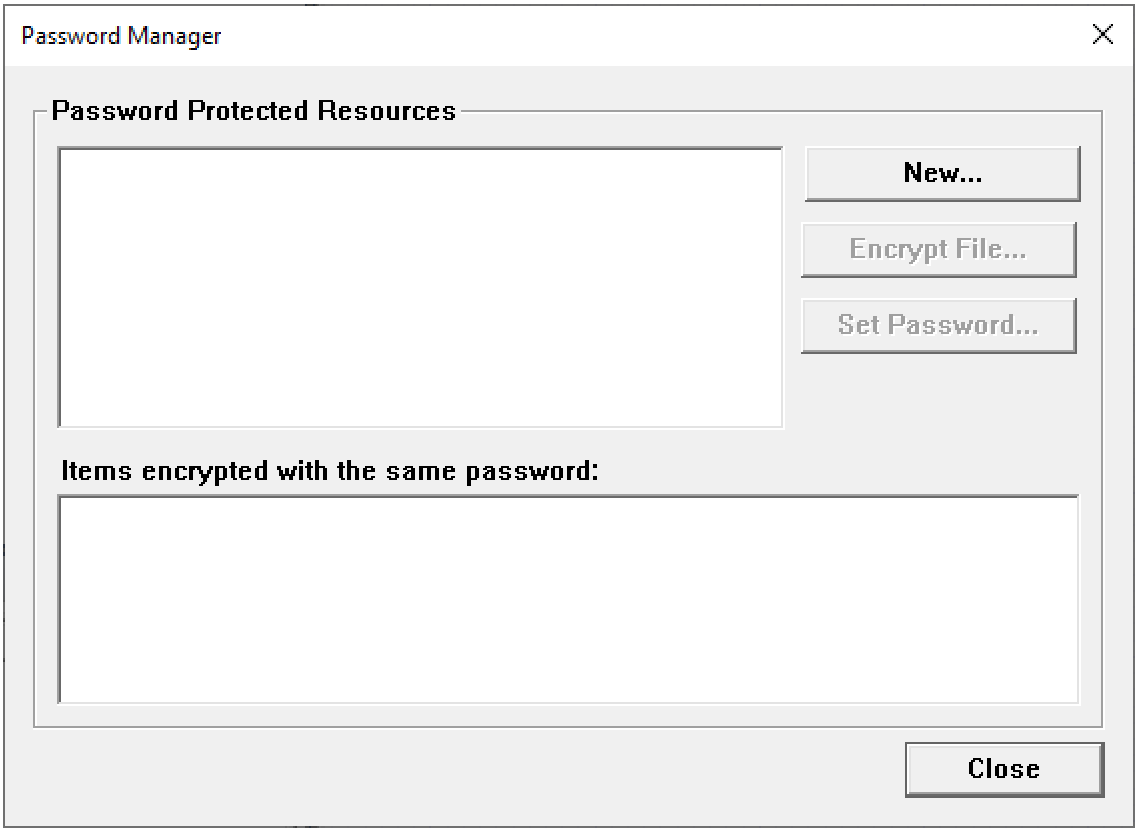
Setting Up Passwords and Encrypted Libraries

Because an encrypted library requires a password, encryption is done through the **Password Manager** window. The **Password Manager** windows lets users create a password or password set as a "password protected resource", which is a group of encrypted files which use the same password. Once the resource and its password type is defined, select it as the password for one or more newly encrypted libraries.

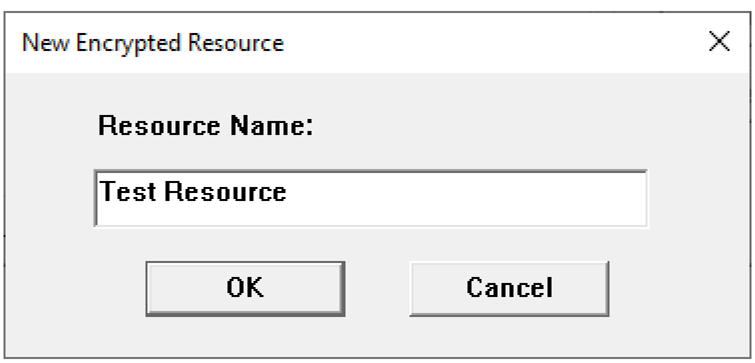
Generate a New Password Protected Resource

1. From **Tools**, select **Password Manager...** to open the **Password Manager** window.

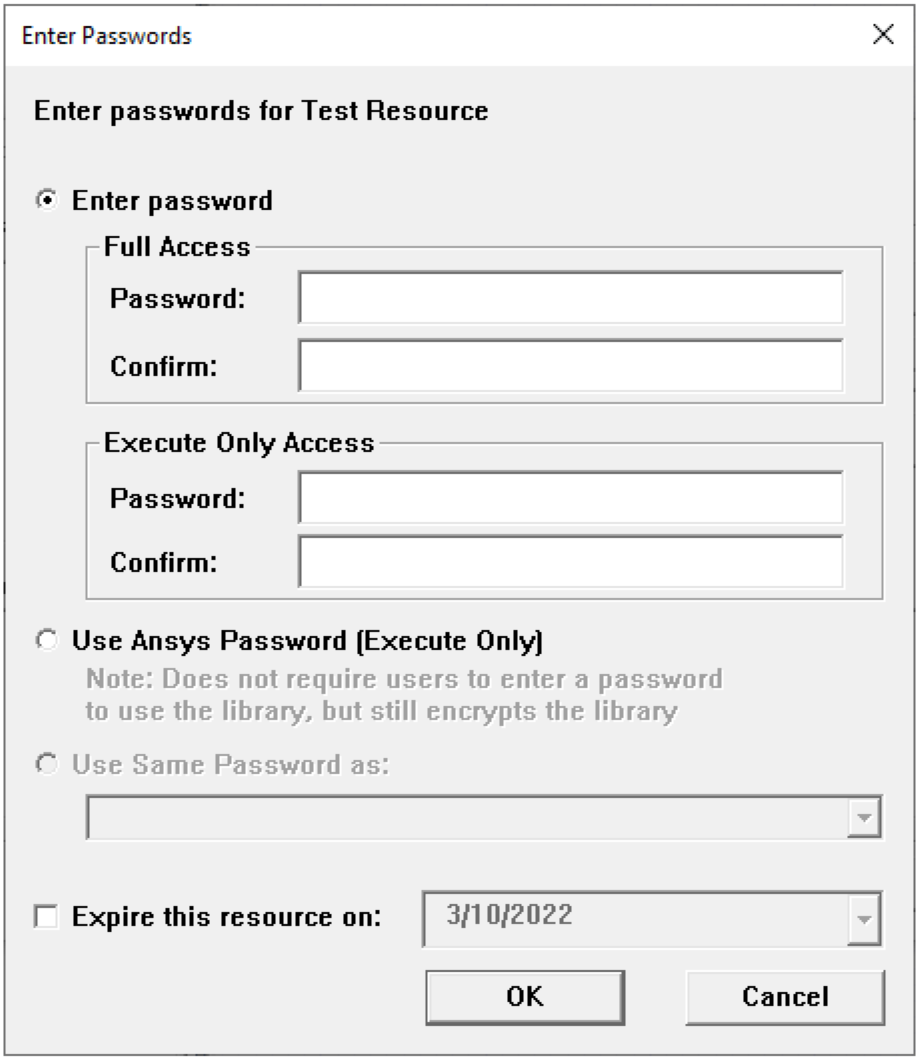

Tools
Window
Help
Edit Libraries
Library Tools
Project Tools
Run Script...
Pause Script
Record Script To File...
Record Script to Project...
Open Command Window
Password Manager...
Debug Logging...
Stop Debug Logging
Options
Keyboard Shortcuts...
External Tools...
Show Queued Simulations
Edit Active Analysis Configuration...
Import Array from Table...
Job Management
Calibration Wizard
Chip Model Analyzer (CMA)
Layout Links...
SPISim
FilterSolutions
Network Data Explorer
PEmag...



2. Click **New...** to open the **New Encrypted Resource** window.



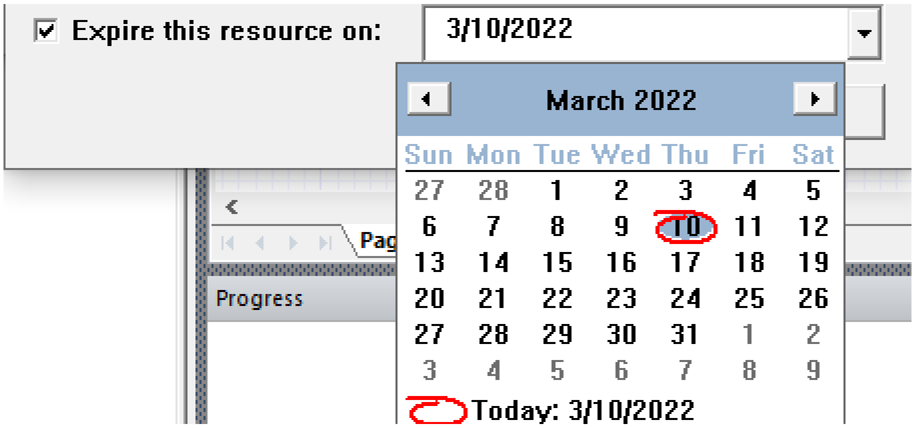
3. Type a name for the resource into the field. Then click **OK** to simultaneously close the **New Encrypted Resource** window and open the **Enter Passwords** window.



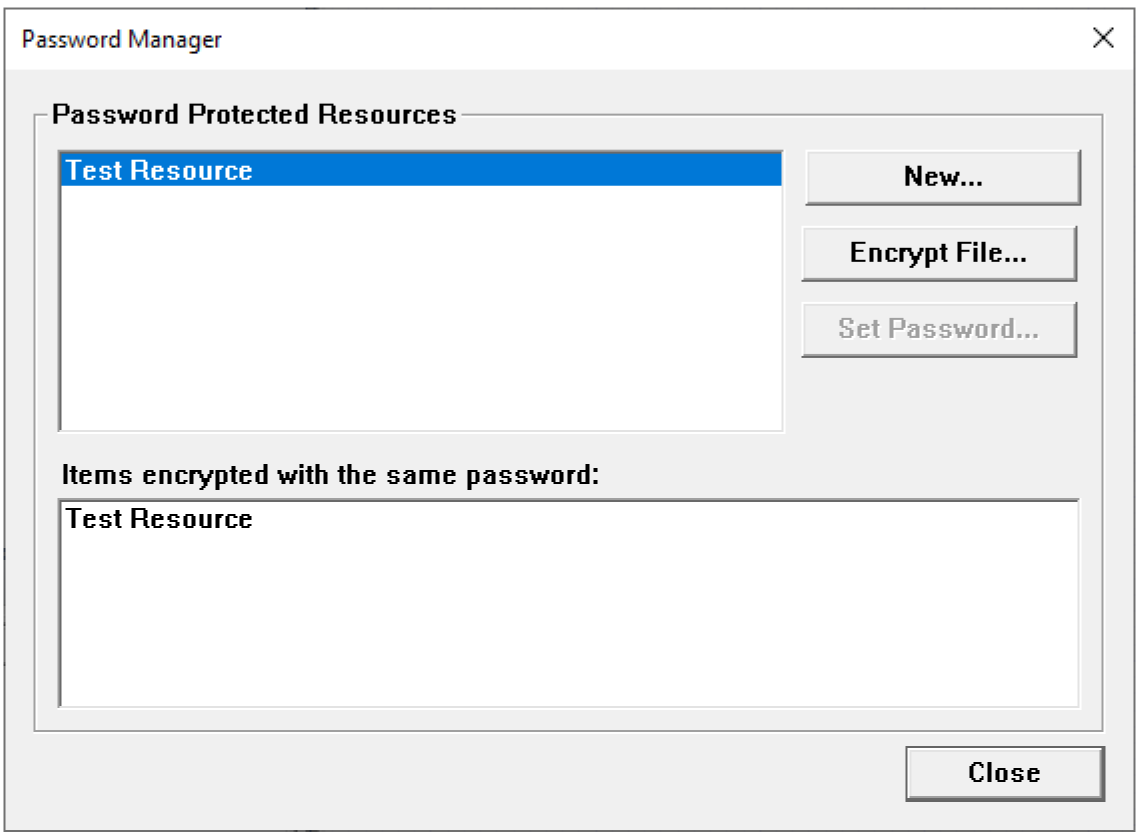
4. Choose on the following:

- **Enter password**: choose between **Full Access** and **Execute Only Access**, then enter and confirm the new password in the given text fields.
- **Use Ansys Password (Execute Only)**: as stated, this choice does not require users to enter a password to use the library, but still encrypts the library.

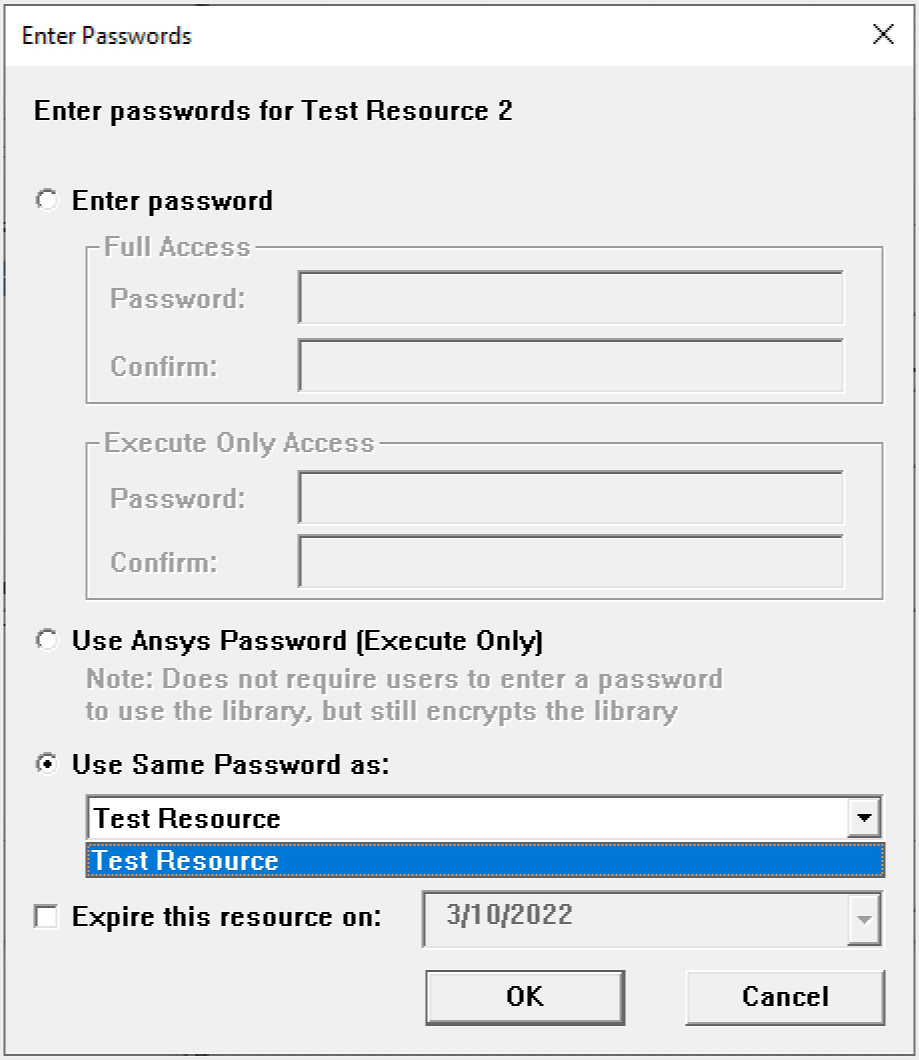
5. Optionally, assign an expiration date for the encrypted library by checking the **Expire this resource on** box and selecting a date on the drop-down calendar. If an expiration date is assigned to a library during encryption, the library cannot be used after the assigned date.



6. Click **OK** to add the new resource to the **Password Manager** list and simultaneously close the **Enter Passwords** window. Any number of libraries may be assigned to the new resource.



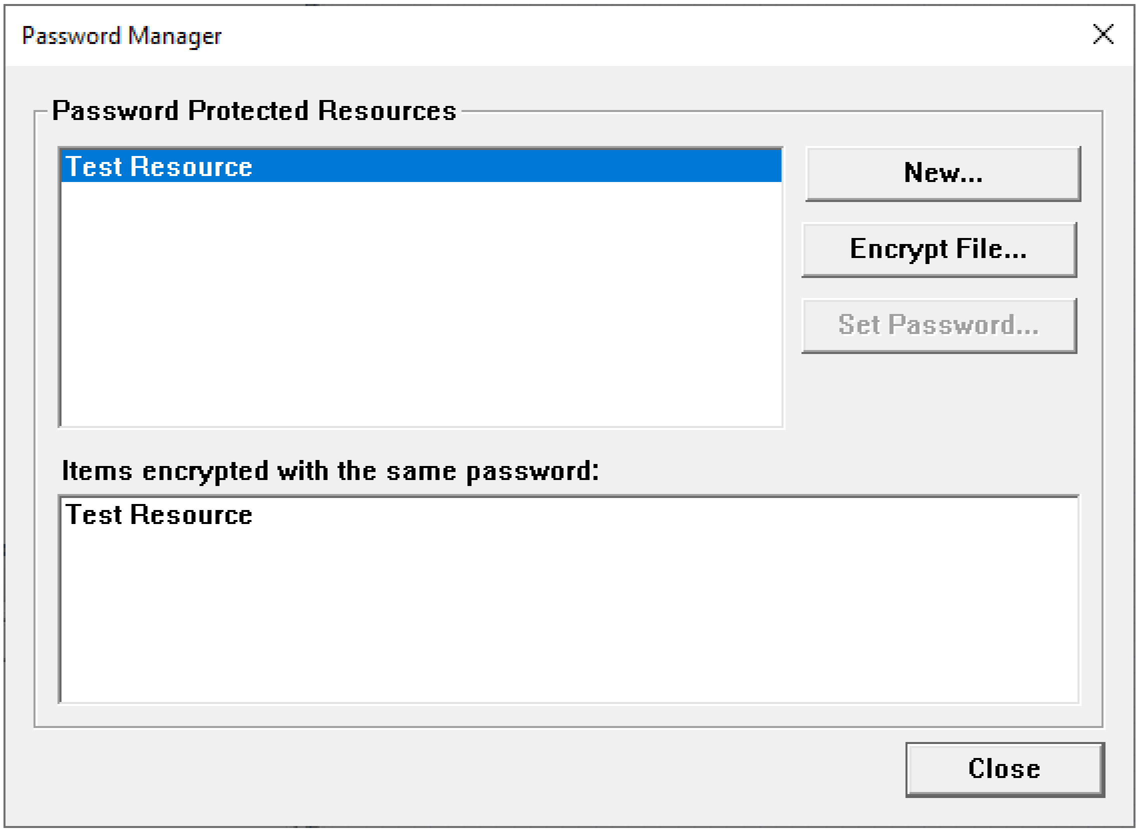
Note: Once one or more resources have been added to the **Password Protected Resources** list on the **Password Manager** window, **New** resources can use the same password profile. From the **Enter Passwords** window, select **Use Same Password as**, which was previously unavailable, and choose the existing resource on the drop-down menu.



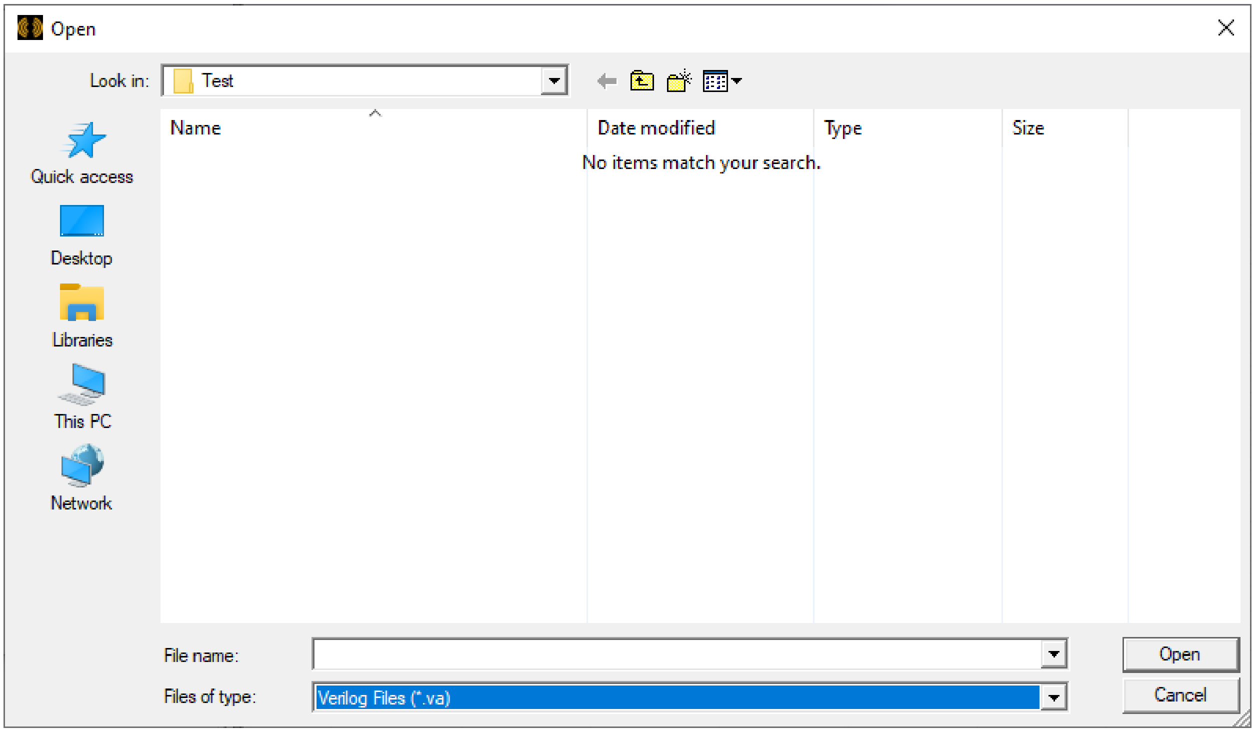
Generate a New Encrypted Library

When a library file is encrypted, the user can choose to overwrite the existing target library with the encrypted one, or save the encrypted library file with a new name and/or to a new location. During encryption, the user is prompted to choose whether the process respects .prot[ect] statements or ignores them. To generate a new encrypted library, complete the following steps.

1. From the **Password Manager** window, select a resource on the **Password Protected Resources** list to enable the **Encrypt File...** button.

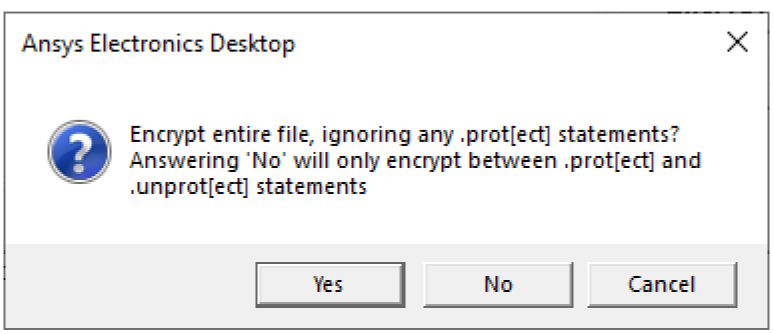


2. Click **Encrypt File...** to open an explorer window. Navigate to the appropriate library file.



Note: Only library files with **Circuit (.*lib)** and **Verilog (.va)** file extensions are valid.

3. Select a library file, or first narrow the available choices by selecting **Circuit Files (.*lib)** or **Verilog Files (.va)** on the **Files of type** drop-down menu. Then click **Open** to immediately open the prompt: **Encrypt entire file, ignoring any .prot[ect] statements?**



4. As stated in the prompt, choose on the following:

- Click **Yes** to permanently encrypt the entire library file.
- Click **No** to recognize any **.prot[ect]** statements and encrypt only the remaining aspect.

- Click **Cancel** to exit out of the prompt and return to the previous explorer window.

5. After the library file is encrypted, an explorer window will appear. Navigate to an appropriate location to save and/or rename the file, if appropriate. If the encrypted library file is saved to the same location and with the same file name as the original unencrypted library file, the unecrypted file is overwritten permanently.
6. Click **Save** to close the explorer window and create the encrypted library file.

22 - Dynamic Links and Solver On Demand

Co-simulation is a feature of Ansys Electronics Desktop that allows you to simulate a design using multiple simulators. Two co-simulation mechanisms are available: Dynamic Links to subcircuits and Solver On Demand.

- Dynamic Link co-simulation occurs whenever a schematic design contains a dynamic link subcircuit. A dynamic link subcircuit links to a project that was created using a simulator different from the top-level simulator. The dynamically linked project generates simulation results using its native simulator in parallel with the top-level simulator.
- Solver On Demand allows selection of among Circuit or Nexsys co-simulation using a component netlist, HFSS 3D Layout or Planar EM simulation using a layout or footprint, and simulation by a one of a variety of solvers accessed by connecting to a previously defined model.

Dynamic Links

Dynamic linking a subcircuit to a project allows the simulation of the subcircuit with the project native simulator in parallel to the Circuit simulator. The dynamic link subcircuit can provide simulation results on demand using its native simulator in parallel with the Circuit simulator.

The Circuit project can also import *static* subcircuits derived from external simulators. A static subcircuit is a component with a fixed set of solution data, such as Touchstone N-port data. No co-simulation is involved in the simulation of a static subcircuit. See Importing Circuit Models.

The following dynamic link project types are supported from a Circuit (Nexxim) top circuit:

- HFSS
- Q3D
- 2D Extractor
- SIwave

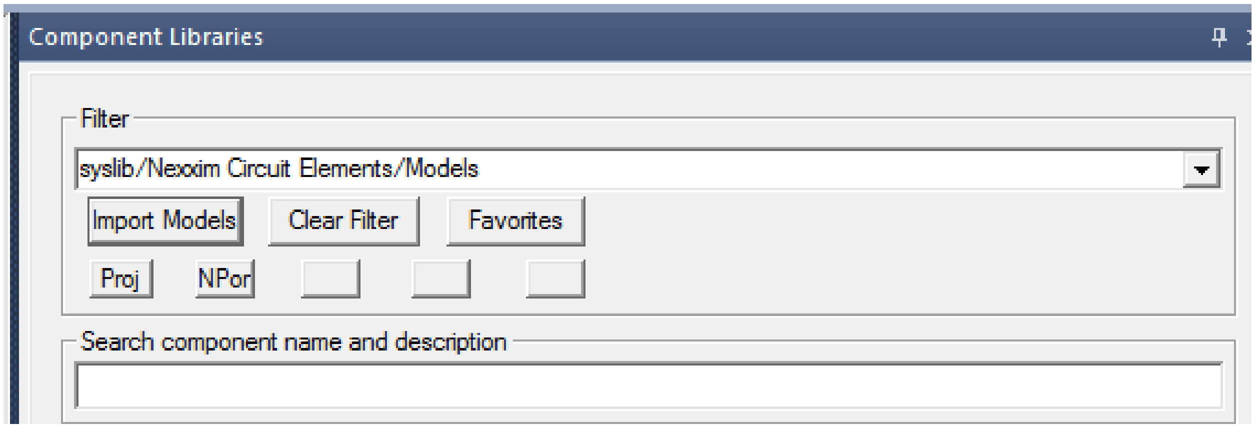
At simulation time, the top-level circuit receives solution data from its subcircuit in order to perform its own simulation. The top-level circuit uses a callback mechanism to wait for subcircuit data while simulating. Since co-simulation circuits are from different products, the type of data being sent back and forth varies. In co-simulation, voltages and currents are usually sent back and forth.

Selecting a Dynamic Link Project

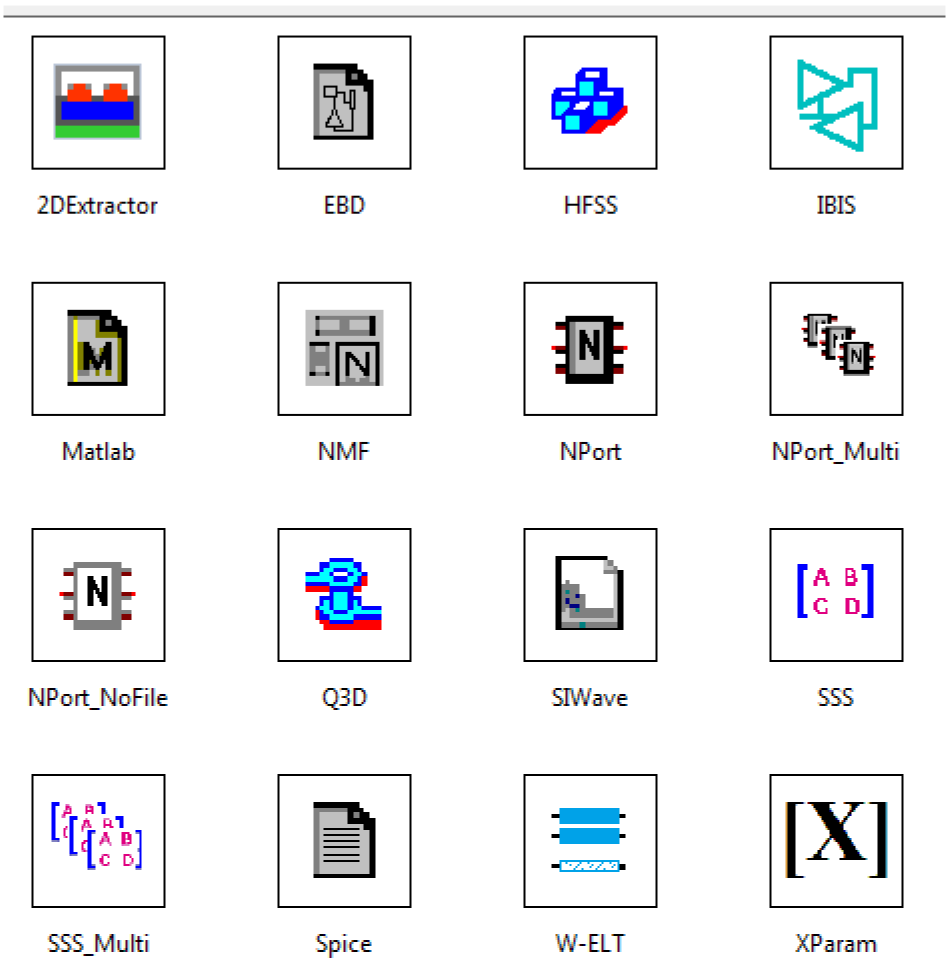
The Models Library is a library of templates to import and place components that represent dynamic links to HFSS, Q3D, 2D Extractor, and SIwave projects. There are several ways to create a dynamic link.

Dynamic Links on the Symbols Tab of the Component Libraries Panel

You can invoke the Models on the **Symbols** tab of the Component Libraries group box, by selecting **Import Models.** .



The **Symbols** group box displays the templates available for creating static and dynamic link models and components.



The dynamic link icons are **2DExtractor**, **HFSS**, **Q3D**, and **SIwave**. Clicking on one of these icons opens a window to choose the project file. When the file is open, if necessary, a second window opens to retrieve additional information. Once you click **OK**, clicking in the schematic places the new component.

Dynamic Links on the Add Subcircuits Menu

To create a dynamic link on the **Project Manager** window, right click a project and select **Add Subcircuit** to open a submenu. Clicking on one of the **Add Link** selections opens a window that allows you to choose a project file. When the file is opened, if necessary, a second window opens to retrieve additional information. Once you click **OK**, clicking in the schematic places the new component.

Drag and Drop or Copy and Paste to Create a Dynamic Link

You can drag and drop, or copy and paste, a project from an open folder, or on the **Project Manager** window, into an open schematic project.

- Dragging and dropping an .aedt project opens the project.
- Dragging and dropping an HFSS, Q3D, 2D Extractor, or SIwave project in the same **Project Tree** creates a new dynamic link.

> **Note:**
>
> If you hold **Ctrl** down while dropping or pasting, the appropriate Dynamic Link window opens, which allows you to make choices such as using an HFSS dynamic link in Transmission Line Model mode.

Using the Model Filter

Once a component of a given type has been created, you can use the filter to display all the models of a given type. For example, after placing several NPort components, view the **Symbols** tab, and type "NP" into the **Filter** group box and choose "type: NPort". This filters to show only NPort components.

For more information, see Using the Component Libraries Window.

Adding a Dynamic Link from HFSS

To create a dynamic link from HFSS to Ansys Electronics Desktop, either import the HFSS data as an N-port device or drag and drop an HFSS design on the **Project Manager** window into an HFSS 3D Layout design to create a link, or use copy and paste on the **Project Manager** window to create a dynamic link.

Add an HFSS dynamic link in the same ways to a Nexxim design as to an HFSS 3D Layout design.

> **Note:**
> 1. When you add an HFSS subcircuit, **Electronics Desktop** opens **Electronics Desktop** version of the HFSS project. You must close the HFSS project and the Circuit project separately.
> 2. In HFSS, differential terminal data is transferred to **Electronics Desktop** as underlying terminal data — which is not differential. For this reason, differential data plotted in HFSS may not agree with raw terminal data plotted in Ansys Electronics Desktop.

Related Topics

Add an HFSS N-Port Model

1. Open the HFSS design you want to link.
2. Open the Circuit design.
3. In the **Component Libraries** group box, open the **Symbols** tab.
4. Click **Models**.
5. Click the **HFSS** tile. The **HFSS Dynamic Link** window opens:
   a. The **Name** fields displays the HFSS design name.
   b. Enter a **Description** if appropriate.
   c. Select a design on the **Project Manager** window. If the design you want is not present, click **Open Project** to find and open that design.
   d. Enable **Transmission Line Model** to use the dynamic link port impedance. The drop-down menu shows the port name of the HFSS design that is used for the transmission line. To change it, select a different port on the list.
   e. The **Link Information** group box at the bottom of the window shows the selected frequency points and number of pins.

> **Note:** The **HFSS Dynamic Link** window only opens if a project with an HFSS design is already opened. If no open projects have an HFSS design, an explorer window opens. In which case:
>
> 1. Browse to the directory that contains the HFSS project for the dynamic link.
> 2. Select the HFSS project file.
> 3. Click **Open**. **Electronics Desktop** pauses for a few seconds while HFSS starts up. The **HFSS Dynamic Link** window then opens then be displayed with this project displayed in the **Project Manager** window.

6. Click **OK**. An instance of the component is attached to the cursor. Drag and drop it into the schematic.

HFSS Component Menu

Right-click the instance to view the menu. These are the visible commands:

- **Edit Link Definition** – See the following description.
- **Refresh Dynamic Link** – Reads data from linked design (properties, solutions, ports, geometry) and updates the link component.
- **Recapture Link Image** – Reads the image on the linked design and updates the symbol in schematic to show the new image.
- **Convert to Parametric Snapshot** – Converts a dynamic link to a parametric snapshot which saves a static copy of the solution data and uses that instead of going back to the linked design for solution data.
- **Clear Solution Cache** – Deletes any solution data for the dynamic link; the next analysis requires obtaining new solution data on the linked design.

Edit HFSS Component Properties

Click the **Properties** item to open the **Properties** window for the element. Modify the values of any of the model parameters. For a description of the parameters see ***Edit HFSS Link Definition***. After modifying parameter values, click **OK** to close the **Properties** window.

Edit HFSS Definitions

1. To view or modify the component definition that was created for the linked project, right-click the component instance and click **Edit Component**. See Using the Component Editor for further information.
2. To view or modify the schematic symbol that was created for the linked project, right-click the component instance and click **Edit Symbol**. See Using the Symbol Editor for further information.
3. To view or modify the assignment of pins to signals on the device, right-click the component instance and click **Edit Symbol Bus Pins**. See Editing All N-Port Symbol Bus Pins for further information.

Edit HFSS Link Definition

1. To view the parameters of the dynamic link, right-click the component instance and select **Edit Link Definition** to open the **Properties** window.

The following controls are available:

- The **Name** field shows the dynamic link component name.
- The **File** field displays the HFSS design that is the source for the dynamic link.
- Enter a **Model Description** if appropriate.
- The **Design** field displays the name of the selected design.
- The **Solution** drop-down menu property lists the Solutions available on the linked design. The displayed menu choice is the Solution that supplies the matrix data during analysis. For HFSS linked designs more than one solution can be selected. In this case "-multiple-" is the menu choice. Choosing a different choice in the menu sets the active solution to the new choice.

- The **Transmission Line Port** property only shows for HFSS dynamic links that are set up as Transmission Line models. It shows the port name of the HFSS design that is being used for the transmission line.
- **Save project after simulate** specifies that the project is saved and closed once its data is accessed and the simulation is complete.
- **Unload project after use** specifies that the project is unloaded once its data is accessed and the window box is closed.
- **Simulation option** allows selection of the method for dealing with missing frequency points in the solution (Interpolate or Simulate).
- **Interpolation algorithm** field to select an interpolation algorithm option:

  Automatic – Linear interpolation is used if a full grid of solutions is present, otherwise Inverse Least Squares with Shadowing and Hyperplanes is used.

  Linear – When a full grid of solutions is available, the cube of solutions which surrounds the solution to be interpolated is located. The corners of this cube are linearly averaged to determine the interpolated solution. If a full grid is not available, an error is reported and the interpolation fails.

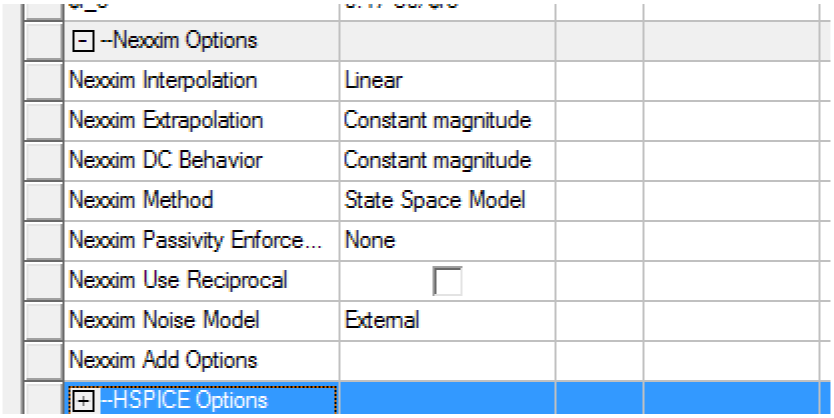
  Inverse Least Squares with Shadowing and Hyperplanes – Least squares interpolation with shadowing and hyperplanes.

  Inverse Least Squares with Shadowing, no Hyperplanes – Least squares interpolation with shadowing, but no hyperplanes.

> **Note:** The main difference between **Linear** interpolation and **Inverse Least Squares with Shadowing** (ILSS) is as follows:
> - Linear only works with a full grid of solutions — ILSS works with arbitrary data.
> - Linear only considers nearest data — ILSS considers data that could possibly be distant, although shadowing mitigates this effect.
> - Linear is not first-order continuous — ILSS is first-order continuous.

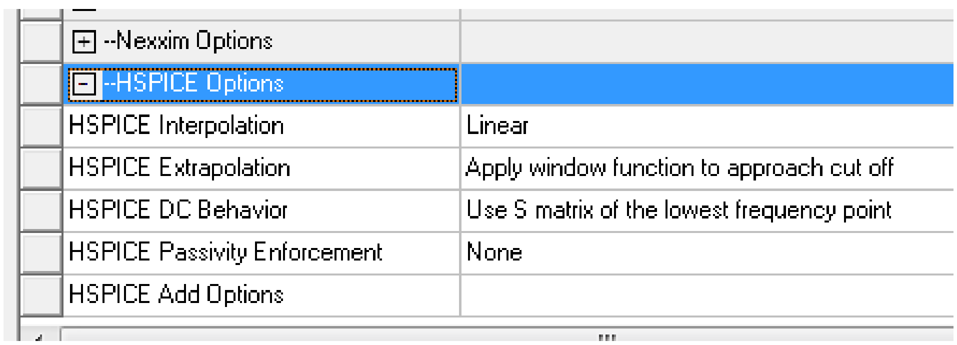
- **Interpolate Y matrix** uses the Y matrix as the basis for interpolation. When **Interpolate Y matrix** is not selected, the S matrix is used as the basis. **Interpolate Y matrix** is selected by default. Circuit particulars determine which basis (S or Y) yields better results, and it is not possible to decide beforehand which one works best.
- **Only use independent variables during interpolation** suppresses the calculation of dependent variable values during interpolation.
- **Renormalize** – If selected, the Scattering parameter Data (S-data) received on the dynamic link are re-normalized with a reference impedance value specified in **Renorm Impedance** before passing the original dynamic link S-data to the Nexxim Circuit solver. If not selected, re-normalization does not take place and the original dynamic link S-data is passed to the Nexxim Circuit solver.

- **Renorm Impedance** refers to the reference impedance used for renormalization of S-data. The default is 50 Ohms.
- **Nominal Values** displays the value of parameters on the linked project.
- **Nexxim Options** allows you to view or set the following Nexxim simulator options:



--Nexxim Options	
Nexxim Interpolation	Linear
Nexxim Extrapolation	Constant magnitude
Nexxim DC Behavior	Constant magnitude
Nexxim Method	State Space Model
Nexxim Passivity Enforce...	None
Nexxim Use Reciprocal	
Nexxim Noise Model	External
Nexxim Add Options	
--HSPICE Options	

- HSPICE options allows you to view or set the following HSPICE simulator options:



--Nexxim Options	
--HSPICE Options	
HSPICE Interpolation	Linear
HSPICE Extrapolation	Apply window function to approach cut off
HSPICE DC Behavior	Use S matrix of the lowest frequency point
HSPICE Passivity Enforcement	None
HSPICE Add Options	

Click **OK** to close the Link **Properties** window.

Refresh Dynamic Link

- **Refresh Dynamic Link** on the HFSS Component menu retrieves ALL data on the linked design, including changes to connectivity, solutions, and properties. Choose this command after making any edits to the linked design.

Recapture HFSS Link Image

- **Recapture Link Image** on the HFSS Component menu recaptures an image of the design to use in the schematic symbol. See Capturing or Recapturing Bitmap Symbols for Dynamic Link Models.

Convert HFSS Dynamic Link to Parametric Snapshot

- **Convert to Parametric Snapshot** on the HFSS Component menu changes the dynamic link to a static parametric snapshot. After this option has been selected, the menu item toggles to **Convert to Dynamic Link**, which allows you to restore the dynamic linkage.

Clear HFSS Solution Cache

- **Clear Solution Cache** on the HFSS Component menu when clicked immediately clears the solution cache memory.

HFSS Project Link Menu and Setup Menu

When the dynamic link project has been added to the design, click the subcircuit icon in the **Project Manager** window to display a brief menu. Use the menu items to create a copy of the dynamic link project, edit the link definition, recapture the link image, convert the dynamic link to a static parametric snapshot, clear the solution cache, or to display the simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

You can right-click any one of the dynamic link sweep icons to display a menu. Use the menu to select the sweep to be in the co-simulation, select ONLY this sweep, or display simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Renormalize Ports Based on HFSS Data

If there are HFSS designs within your Ansys Electronics Desktop design, renormalize the characteristic impedance of the circuit ports to the port impedance of one of the HFSS ports. To do so:

1. Double-click the port you want to renormalize in the schematic to open the **Port Definition** window.
2. Click **One port data** to enable a drop-down menu that displays the HFSS ports in the active design.
4. Select the appropriate HFSS design and port.
5. Select **Use characteristic impedance (Zo)** or **Use impedance matrix**.
6. Click **OK**.

The circuit ports are renormalized to the characteristic impedance of the HFSS port as a function of frequency and design parameters. For more information, see Editing an Interface Port Definition.

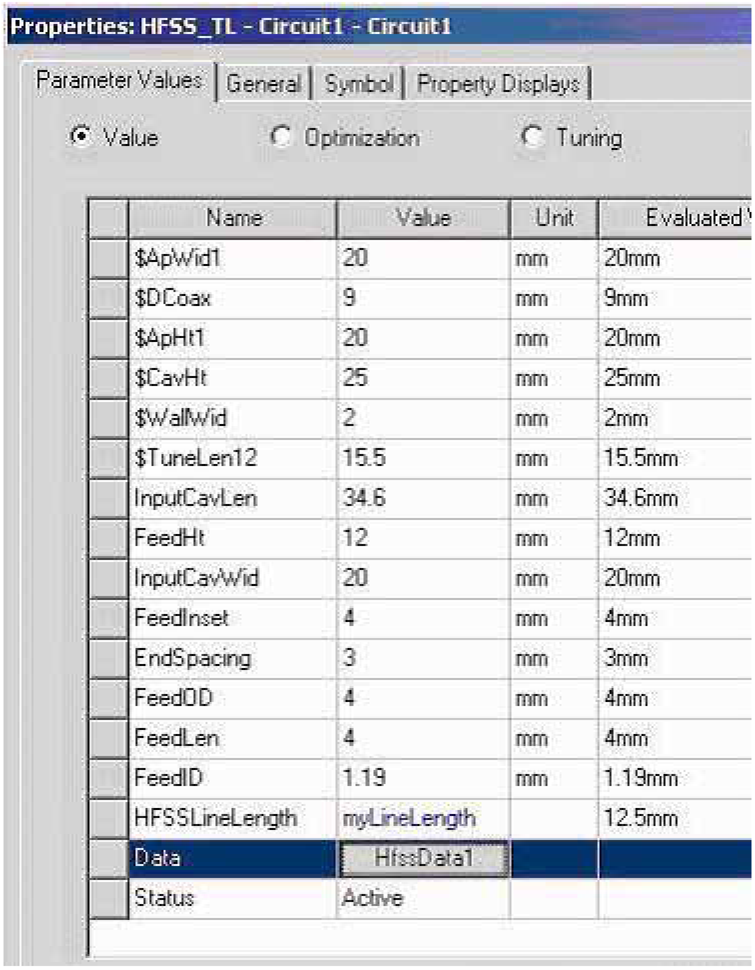
Transmission Lines Based on HFSS Data

A parameterized two port transmission line can be created in **Electronics Desktop** based on an HFSS port solution. The characteristic impedance and propagation constant from a port in an HFSS design is used along with a user input length to generate an S parameter matrix in **Electronics Desktop** for a single mode transmission line. The resulting transmission line can be used like any other Ansys Electronics Desktop component in the schematic editor.

> **Note:** In HFSS, differential terminal data is transferred to **Electronics Desktop** as underlying terminal data — which is not differential. For this reason, differential data plotted in HFSS may not agree with raw terminal data plotted in Ansys Electronics Desktop.

To create a W-element transmission line model instead of a standard N-port, check the **Transmission Line Model** check box on the HFSS Dynamic Link window. See Add an HFSS Dynamic Link Model.

The properties of this component includes all the properties on the HFSS model plus one called “TLineLength”:



The variable “myLineLength” was created in **Electronics Desktop** to vary the length of the transmission line.

Ansys Electronics Desktop creates a single mode transmission line using the characteristic impedance and propagation constant from the HFSS design. The length of the line is “TLineLength.” Port solution data from HFSS as a function of design parameters is either interpolated or solved automatically depending on the status of the “Interpolate missing solutions/Simulate missing solutions” button.

> **Note:** Calculations of Z_{IN} and Z_{OUT} depend on the correct terminations and propagation constants. These results may not match the results in HFSS.

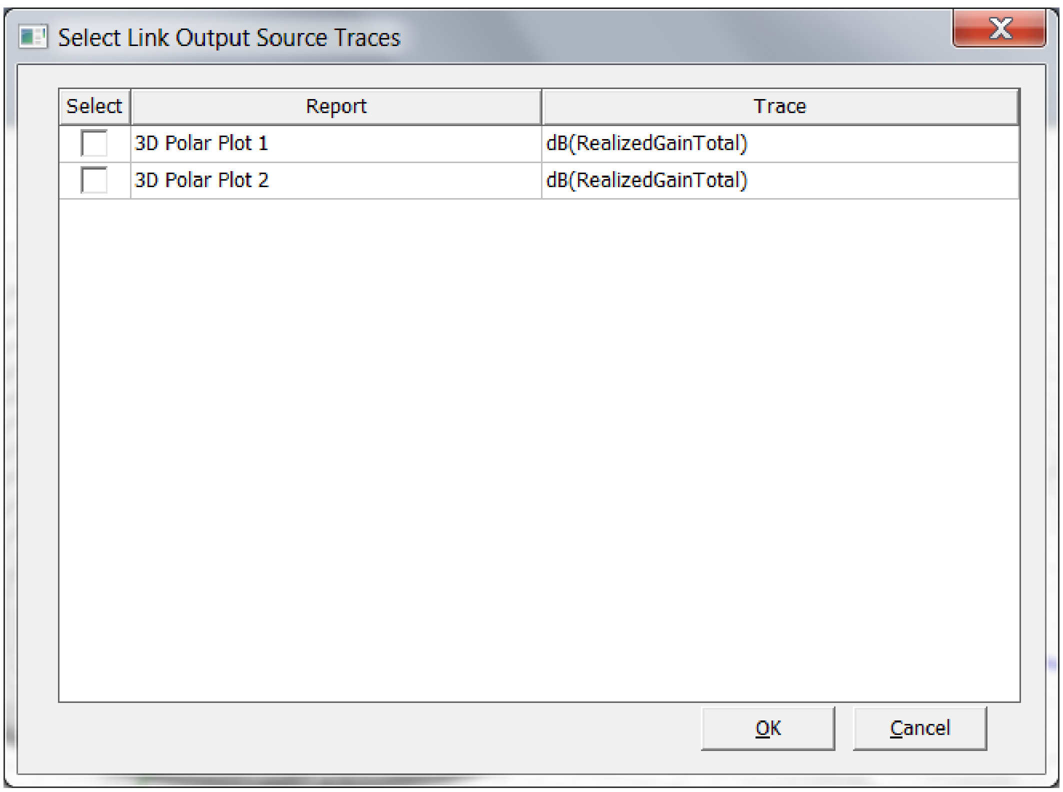
Access to HFSS Output Variables from Nexxim

HFSS dynamic link projects allow you to specify the plot traces to be made available for reporting in the upper-level Circuit project. You select the traces in the HFSS project, then use the outputs in a report in the Circuit Project.

Specifying Dynamic Link Output Traces

To select the traces to be included in dynamic link outputs:

1. Open the HFSS project and right-click the **Results** folder.
2. Select **Link Output ...** on the menu. The **Select Link Output Source Traces** window opens:

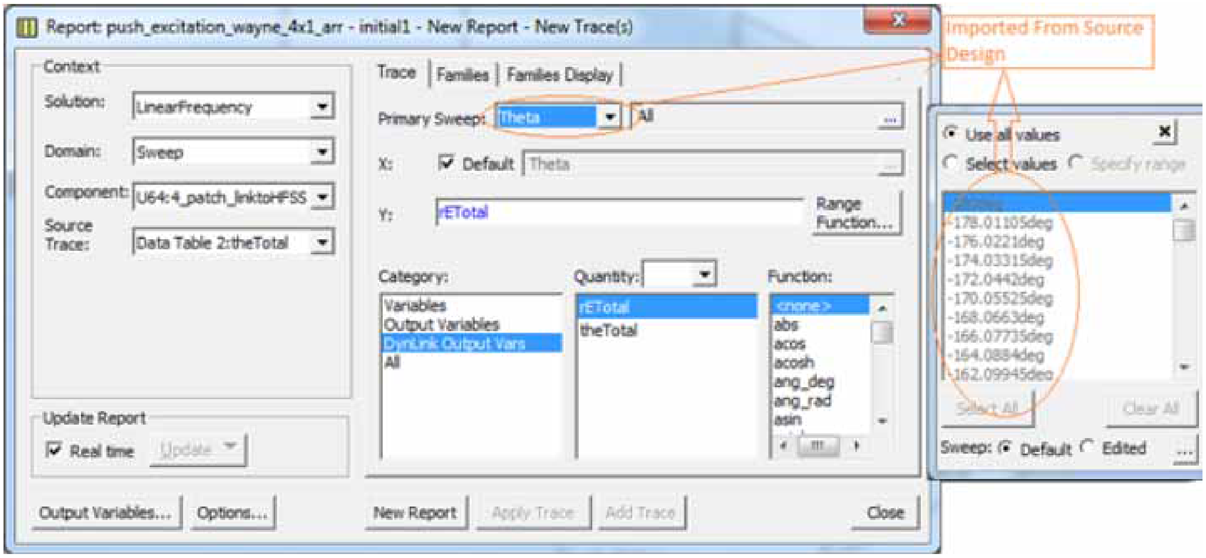


3. Select one or more traces with the check boxes. For each selected trace, the quantity and output variable used in its highest-dimension expression is exposed o the Circuit design.

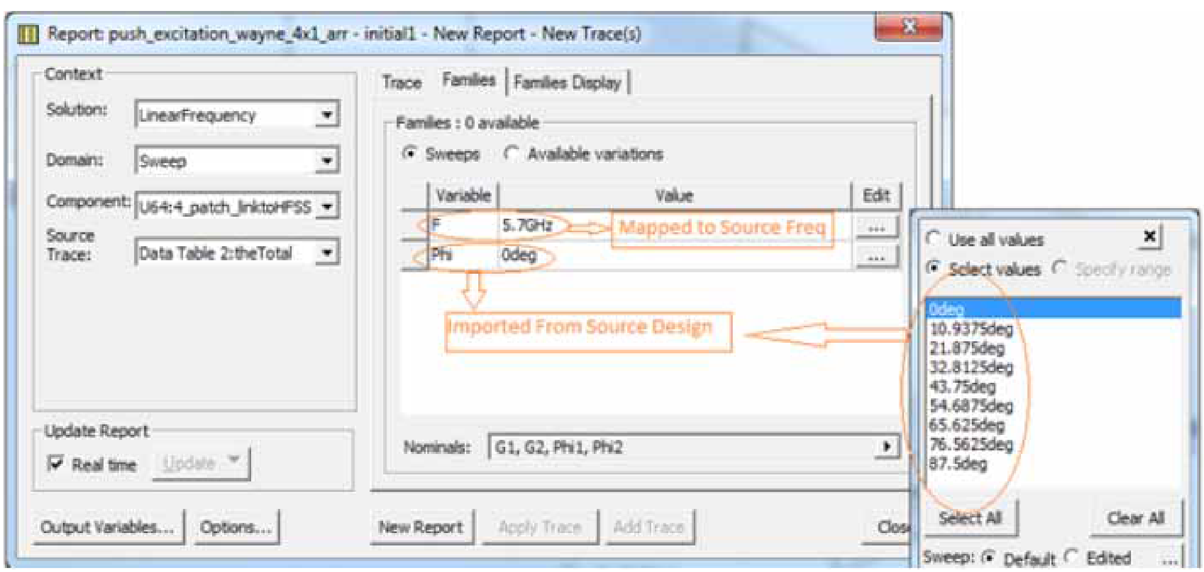
HFSS Dynamic Link Outputs in Circuit Reports

1. Select **Reports>Create Link Output Report,** then select a display type, such as **Rectangular Plot**, to open the **Create Report** window.
2. For **Solution**, select the Circuit solution. Select the Solution Domain.

3. The **Component** field lists the available HFSS dynamic link objects.
4. The **Source** trace field lists the traces selected to be exposed by HFSS.
5. From the **Trace** tab, select the HFSS sweep that is to be the primary sweep.



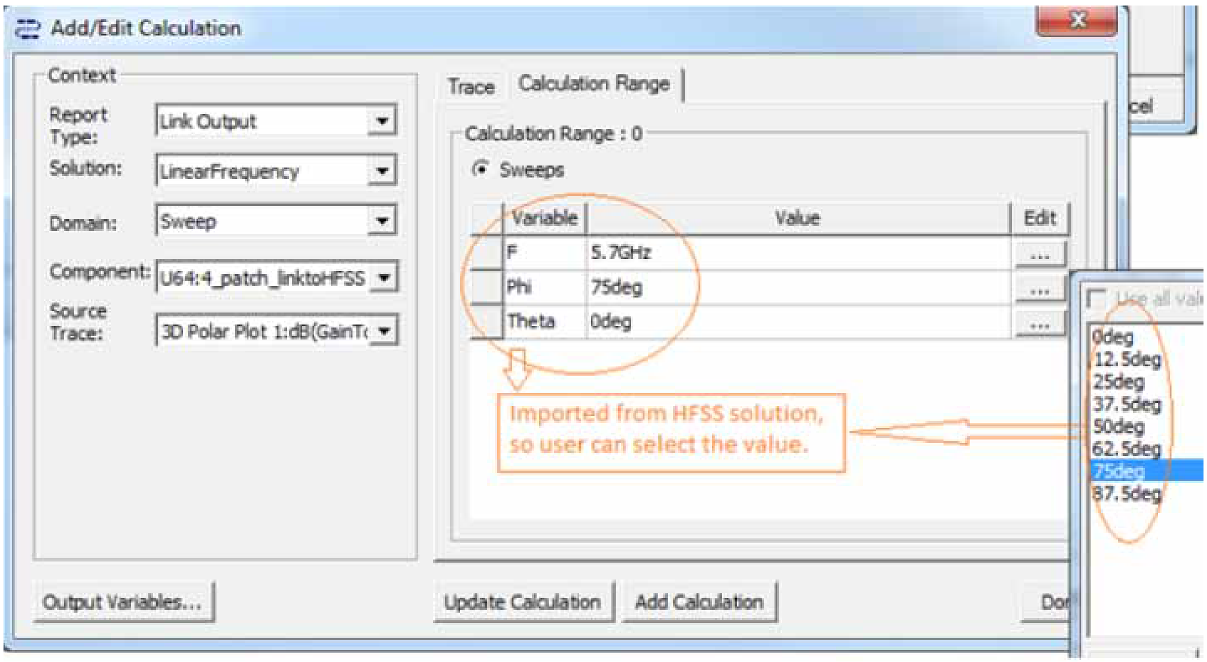
6. The **Category** is selected as **DynLink Output Var**.
7. The **Quantities** field lists the X, Y, or Z expression with its quantity and output variables. (1D trace: X expression; 2D trace: Y expression; 3D trace: Z expression.)
8. From the **Families** tab, there are a list of Circuit solution variables with HFSS variable mappings, and unmapped HFSS solution intrinsic variables.



9. The Circuit **F** intrinsic variable is automatically mapped to the HFSS **Freq** variable. Unmapped HFSS intrinsic variables values is imported on the HFSS solution. Unmapped Circuit variables get values on the Circuit solution only.

HFSS Variables in Optimetric Sweeps

Click **Setup Calculations** to open the **Add/Edit Calculation** window. Select **Link Output** as the **Report Type**.



The HFSS variables have been imported on the HFSS project.
nmj

Adding a Dynamic Link from Q3D

To create a dynamic link from Q3D to Ansys Electronics Desktop, you import the Q3D data as an N-port device.

> **Note:** When you add a Q3D subcircuit, **Electronics Desktop** opens the Q3D project. The Electronics Desktop version of Q3D is opened. You must close the Q3D project and the Circuit project separately.

Add a Q3D design as a dynamic-link subcircuit

1. Open the Circuit design to which you want to add a subcircuit.
2. In the **Component Libraries** group box, open the **Symbols** tab.

3. Click **Models**.
4. Click the **Q3D** tile. The **Q3D Dynamic Link** window opens:
   a. The **Name** fields displays the HFSS design name.
   b. Enter a **Description** if appropriate.
   c. Select a design on the **Project Manager** window. If the design you want is not present, click **Open Project** to find and open that design.
   d. **Reduce Matrix** sets the type of matrix reduction available in the project.
   e. The **Link Information** group box at the bottom of the window shows the selected frequency points and number of pins.

> **Note:** The **Q3D Dynamic Link** window only opens if a project with a Q3D design is already opened. If no open projects have a Q3D design, an explorer window opens. In which case:
>
> A. Browse to the directory that contains the Q3D project for the dynamic link.
> B. Select the Q3D project file.
> C. Click **Open**. **Electronics Desktop** pauses for a few seconds while Q3D starts up. The **Q3D Dynamic Link** window then opens then be displayed with this project displayed in the **Project Manager** window.

5. Click **OK**. A copy of the symbol is attached to the cursor. Drag and drop it into the schematic.

Q3D Component Menu

Right-click the instance to view the menu. These are the visible commands:

- **Edit Link Definition** – See the following description.
- **Refresh Dynamic Link** – Reads data from linked design (properties, solutions, ports, geometry) and updates the link component.
- **Recapture Link Image** – Reads the image on the linked design and updates the symbol in schematic to show the new image.
- **Convert to Parametric Snapshot** – Converts a dynamic link to a parametric snapshot which saves a static copy of the solution data and uses that instead of going back to the linked design for solution data.
- **Clear Solution Cache** – Deletes any solution data for the dynamic link; the next analysis requires obtaining new solution data on the linked design.

Edit Q3D Component Properties

- Click the **Properties** item to open the **Properties** window for the element. Modify the values of any of the model parameters. For a description of the parameters see ***Edit Q3D Link Definition***. After modifying parameter values, click **OK** to close the **Properties** window.

Edit Q3D Definitions

1. To view or modify the component definition that was created for the linked project, right-click the component instance and click **Edit Component**. See Using the Component Editor for further information.
2. To view or modify the schematic symbol that was created for the linked project, right-click the component instance and click **Edit Symbol**. See Using the Symbol Editor for further information.
3. To view or modify the assignment of pins to signals on the device, right-click the component instance and click **Edit Symbol Bus Pins**. See Editing All N-Port Symbol Bus Pins for further information.

Edit Q3D Link Definition

1. To view the parameters of the dynamic link, right-click the component instance and select **Edit Link Definition** to open the **Properties** window.

The following controls are available:

- The **Name** field shows the dynamic link component name.
- The **File** field displays the Q3D design that is the source for the dynamic link.
- Enter a **Model Description** if appropriate.
- The **Design** field displays the name of the selected design.
- The **Solution** drop-down menu property lists the Solutions available on the linked design. The displayed menu choice is the Solution that supplies the matrix data during analysis. Choosing a different choice in the menu sets the active solution to the new choice.
- **Reduce Matrix** selects the type of matrix reduction available in the project.
- **Save project after simulate** specifies that the project is saved and closed once its data is accessed and the simulation is complete.
- **Unload project after use** specifies that the project is unloaded once its data is accessed and the window box is closed.
- **Simulation option** allows selection of the method for dealing with missing frequency points in the solution (Interpolate or Simulate).
- **Interpolation algorithm** field to select an interpolation algorithm option:

  Automatic – Linear interpolation is used if a full grid of solutions is present, otherwise Inverse Least Squares with Shadowing and Hyperplanes is used.

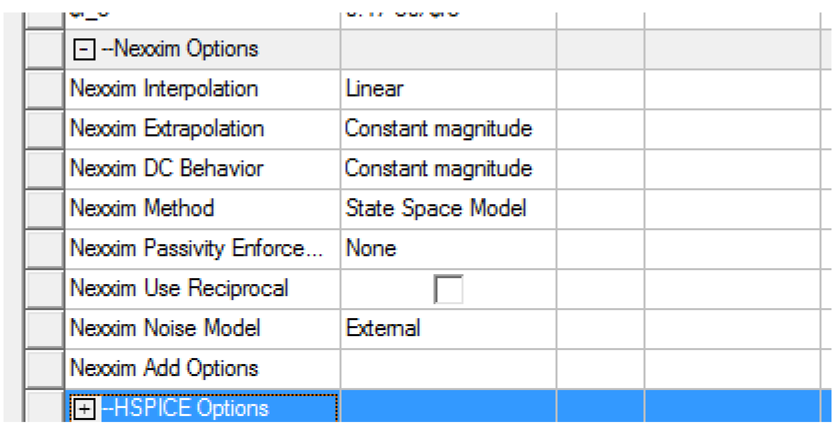
  Linear – When a full grid of solutions is available, the cube of solutions which surrounds the solution to be interpolated is located. The corners of this cube are linearly averaged to determine the interpolated solution. If a full grid is not available, an error is reported and the interpolation fails.

Inverse Least Squares with Shadowing and Hyperplanes – Least squares interpolation with shadowing and hyperplanes.

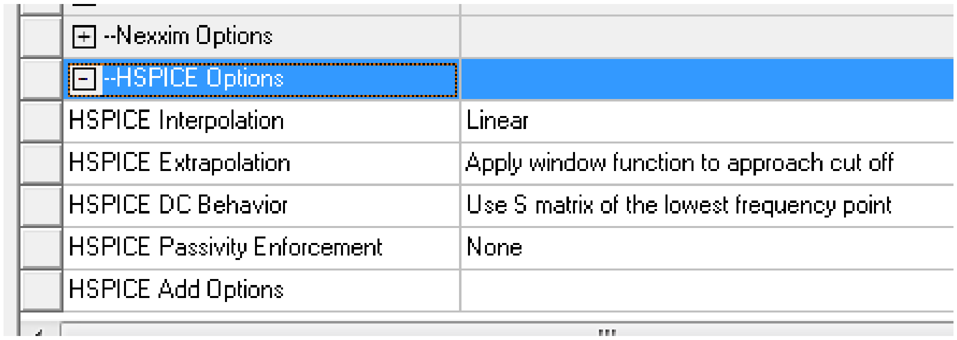
Inverse Least Squares with Shadowing, no Hyperplanes – Least squares interpolation with shadowing, but no hyperplanes.

> **Note:** The main difference between **Linear** interpolation and **Inverse Least Squares with Shadowing** (ILSS) is as follows:
>
> - Linear only works with a full grid of solutions — ILSS works with arbitrary data.
> - Linear only considers nearest data — ILSS considers data that could possibly be distant, although shadowing mitigates this effect.
> - Linear is not first-order continuous — ILSS is first-order continuous.

- **Interpolate Y matrix** uses the Y matrix as the basis for interpolation. When **Interpolate Y matrix** is not selected, the S matrix is used as the basis. **Interpolate Y matrix** is selected by default. Circuit particulars determine which basis (S or Y) yields better results, and it is not possible to decide beforehand which one works best.
- **Only use independent variables during interpolation** suppresses the calculation of dependent variable values during interpolation.
- **Renormalize** – If selected, the Scattering parameter Data (S-data) received on the dynamic link are re-normalized with a reference impedance value specified in **Renorm Impedance** before passing the original dynamic link S-data to the Nexxim Circuit solver. If not selected, re-normalization does not take place and the original dynamic link S-data is passed to the Nexxim Circuit solver.
- **Renorm Impedance** refers to the reference impedance used for renormalization of S-data. The default is 50 Ohms.
- **Nominal Values** displays the value of parameters on the linked project.
- **Nexxim Options** allows you to view or set the following Nexxim simulator options:



- HSPICE options allows you to view or set the following HSPICE simulator options:



Click **OK** to close the Link **Properties** window.

Refresh Dynamic Link

- **Refresh Dynamic Link** on the Q3D Component menu retrieves ALL data on the linked design, including changes to connectivity, solutions, and properties. Choose this command after making any edits to the linked design.

Recapture Q3D Link Image

- **Recapture Link Image** on the Q3D Component menu recaptures an image of the design to use in the schematic symbol. See Capturing or Recapturing Bitmap Symbols for Dynamic Link Models.

Clear Q3D Solution Cache

- **Clear Solution Cache** on the Q3D Component menu, when clicked, immediately clears the solution cache memory.

Q3D Project Link Menu and Setup Menu

When the dynamic link project has been added to the design, click the subcircuit icon in the **Project Manager** window to display a brief menu. Use the menu items to create a copy of the dynamic link project, edit the link definition, recapture the link image, clear the solution cache, or to display simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Click any one of the dynamic link sweep icons to display a menu. Use the menu to open the simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Adding a Dynamic Link from 2D Extractor

To create a dynamic link from 2D Extractor to Ansys Electronics Desktop, you import the 2D Extractor data as an N-port device.

> **Note:** When you add a 2D Extractor subcircuit, **Electronics Desktop** opens the 2D Extractor project. The Electronics Desktop version of 2D Extractor is opened. You must close the 2D Extractor project and the Circuit project separately.

Add a 2D Extractor n-port as a dynamic-link subcircuit to a design

1. Open the Circuit design to which you want to add a subcircuit.
2. In the **Component Libraries** group box, open the **Symbols** tab.
3. Click **Models**.
4. Click the **2DExtractor** tile. The **2DExtractor Dynamic Link** window opens:
   a. The **Name** fields displays the 2DExtractor design name.
   b. Enter a **Description** if appropriate.
   c. Select a design on the **Project Tree**. If the design you want is not present, click **Open Project** to find and open that design.
   d. **Reduce Matrix** selects the type of matrix reduction available in the project.
   e. The **Link Information** group box at the bottom of the window shows the selected frequency points and number of pins.

> **Note:** The **2DExtractor Dynamic Link** window only opens if a project with a 2DExtractor design is already opened. If no open projects have a 2DExtractor design, an explorer window opens. In which case:
> 1. Browse to the directory that contains the 2DExtractor project for the dynamic link.
> 2. Select the 2DExtractor project file.
> 3. Click **Open**. **Electronics Desktop** pauses for a few seconds while 2DExtractor starts up. The **2DExtractor Dynamic Link** window then opens then be displayed with this project displayed in the **Project Manager** window.

5. Click **OK**. A copy of the symbol is attached to the cursor. Drag and drop it into the schematic.

2D Extractor Component Menu

Right-click the instance to view the menu. These are the visible commands:

- **Edit Link Definition** – See the following description.
- **Refresh Dynamic Link** – Reads data from linked design (properties, solutions, ports, geometry) and updates the link component.
- **Recapture Link Image** – Reads the image on the linked design and updates the symbol in schematic to show the new image.

- **Convert to Parametric Snapshot** – Converts a dynamic link to a parametric snapshot which saves a static copy of the solution data and uses that instead of going back to the linked design for solution data.
- **Clear Solution Cache** – Deletes any solution data for the dynamic link; the next analysis requires obtaining new solution data on the linked design.

Edit 2D Extractor Component Properties

- Click the **Properties** item to open the **Properties** window for the element. Modify the values of any of the model parameters. For a description of the parameters see ***Edit 2D Extractor Link Definition***. After modifying parameter values, click **OK** to close the **Properties** window.

Edit 2D Extractor Definitions

1. To view or modify the component definition that was created for the linked project, right-click the component instance and click **Edit Component**. See Using the Component Editor for further information.
2. To view or modify the schematic symbol that was created for the linked project, right-click the component instance and click **Edit Symbol**. See Using the Symbol Editor for further information.
3. To view or modify the assignment of pins to signals on the device, right-click the component instance and click **Edit Symbol Bus Pins**. See Editing All N-Port Symbol Bus Pins for further information.

Edit 2D Extractor Link Definition

1. To view the parameters of the dynamic link, right-click the component instance and select **Edit Link Definition** to open the **Properties** window.

The following controls are available:

- The **Name** field shows the dynamic link component name.
- The **File** field displays the 2DExtractor design that is the source for the dynamic link.
- Enter a **Model Description** if appropriate.
- The **Design** field displays the name of the selected design.
- The **Solution** drop-down menu property lists the Solutions available on the linked design. The displayed menu choice is the Solution that supplies the matrix data during analysis. Choosing a different choice in the menu sets the active solution to the new choice.
- **Reduce Matrix** selects the type of matrix reduction available in the project.
- **Save project after simulate** specifies that the project is saved and closed once its data is accessed and the simulation is complete.
- **Unload project after use** specifies that the project is unloaded once its data is accessed and the window box is closed.

- **Simulation option** allows selection of the method for dealing with missing frequency points in the solution (Interpolate or Simulate).
- **Interpolation algorithm** field to select an interpolation algorithm option:

  Automatic – Linear interpolation is used if a full grid of solutions is present, otherwise Inverse Least Squares with Shadowing and Hyperplanes is used.

  Linear – When a full grid of solutions is available, the cube of solutions which surrounds the solution to be interpolated is located. The corners of this cube are linearly averaged to determine the interpolated solution. If a full grid is not available, an error is reported and the interpolation fails.

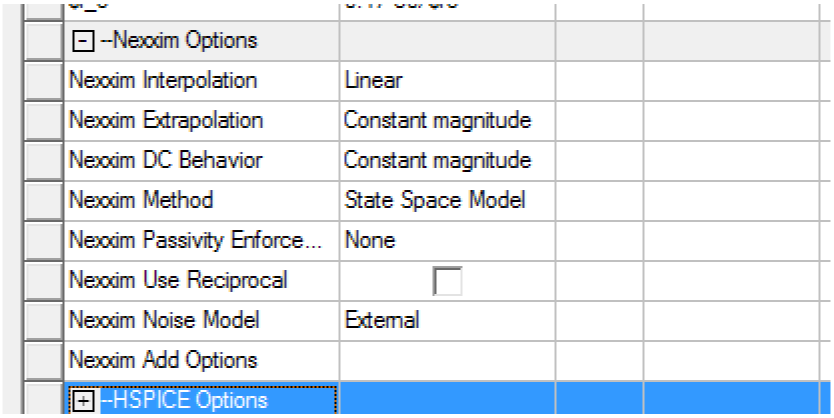
  Inverse Least Squares with Shadowing and Hyperplanes – Least squares interpolation with shadowing and hyperplanes.

  Inverse Least Squares with Shadowing, no Hyperplanes – Least squares interpolation with shadowing, but no hyperplanes.

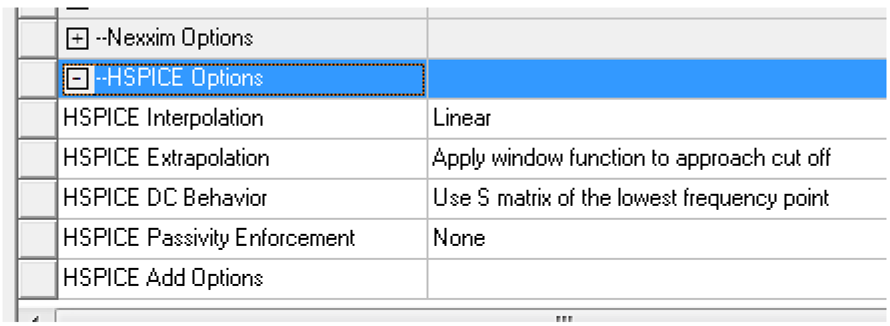
> **Note:**
>
> The main difference between **Linear** interpolation and **Inverse Least Squares with Shadowing** (ILSS) is as follows:
>
> - Linear only works with a full grid of solutions — ILSS works with arbitrary data.
> - Linear only considers nearest data — ILSS considers data that could possibly be distant, although shadowing mitigates this effect.
> - Linear is not first-order continuous — ILSS is first-order continuous.

- **Interpolate Y matrix** uses the Y matrix as the basis for interpolation. When **Interpolate Y matrix** is not selected, the S matrix is used as the basis. **Interpolate Y matrix** is selected by default. Circuit particulars determine which basis (S or Y) yields better results, and it is not possible to decide beforehand which one works best.
- **Only use independent variables during interpolation** suppresses the calculation of dependent variable values during interpolation.
- **Nominal Values** displays the value of parameters on the linked project.

- **Nexxim Options** allows you to view or set the following Nexxim simulator options.



- HSPICE options allows you to view or set the following HSPICE simulator options.



Click **OK** to close the Link **Properties** window.

Refresh Dynamic Link

- **Refresh Dynamic Link** on the 2D Component menu retrieves ALL data on the linked design, including changes to connectivity, solutions, and properties. Choose this command after making any edits to the linked design.

Recapture 2D Extractor Link Image

- **Recapture Link Image** on the 2D Extractor component menu recaptures an image of the design to use in the schematic symbol. See Capturing or Recapturing Bitmap Symbols for Dynamic Link Models.

Clear 2D Extractor Solution Cache

- **Clear Solution Cache** on the 2D Extractor component menu immediately clears the solution cache memory.

2D Extractor Project Link Menu and Setup Menu

When the dynamic link project has been added to the design, click the subcircuit icon in the **Project Manager** window to display a brief menu. Use the menu items to create a copy of the dynamic link project, edit the link definition, recapture the link image, clear the solution cache, or to open the simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Click any one of the dynamic link sweep icons to display a menu. Use the menu to select the sweep to be in the co-simulation.

Adding a Dynamic Link from SIwave

To create a dynamic link from SIwave to Ansys Electronics Desktop, you import the SIwave data as an N-port device.

> **Note:** When you add an SIwave subcircuit, **Electronics Desktop** opens the SIwave project. By default, the latest version of SIwave is opened. You must close the SIwave project and the Circuit project separately.

Add an SIwave design as a dynamic-link subcircuit

1. Open the Circuit design to which you want to add a subcircuit.
2. From the **Component Libraries** menu, open the **Symbols** tab then click **Models**. Select the **SIwave** icon on the group box of icons to open the **Import Components** window.
3. Browse to the directory containing the SIwave project for the dynamic link. Select the project and click **Open** to open the **SIwave Dynamic Link** menu.
4. Make any appropriate changes, then click **OK**. An instance of the component is attached to the cursor. Drag and drop in into the schematic

SIwave Component Menu

Right-click the instance to view the menu. These are the visible commands:

- **Edit Link Definition** – See description.
- **Refresh Dynamic Link** – Reads data from linked design (properties, solutions, ports, geometry) and updates the link component.
- **Recapture Link Image** – Reads the image on the linked design and updates the symbol in schematic to show the new image.
- **Convert to Parametric Snapshot** – Converts a dynamic link to a parametric snapshot which saves a static copy of the solution data and uses that instead of going back to the linked design for solution data.
- **Clear Solution Cache** – Deletes any solution data for the dynamic link; the next analysis requires obtaining new solution data on the linked design.

Edit SIwave Definitions

1. To view or modify the component definition that was created for the linked project, right-click the component instance and click **Edit Component**. See Using the Component Editor for further information.
2. To view or modify the schematic symbol that was created for the linked project, right-click the component instance and click **Edit Symbol**. See Using the Symbol Editor for further information.
3. To view or modify the assignment of pins to signals on the device, right-click the component instance and click **Edit Symbol Bus Pins**. See Editing All N-Port Symbol Bus Pins for further information.

Edit SIwave Link Definition

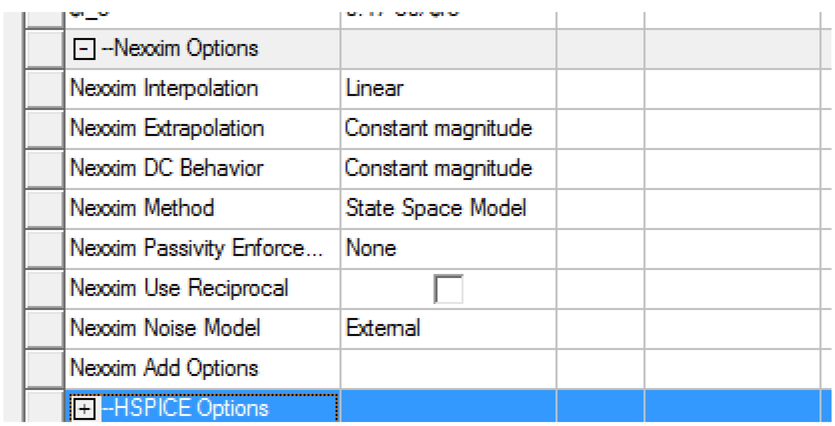
1. To view the parameters of the dynamic link, right-click the component instance and select **Edit Link Definition** to open the **Properties** window.

The following controls are available:

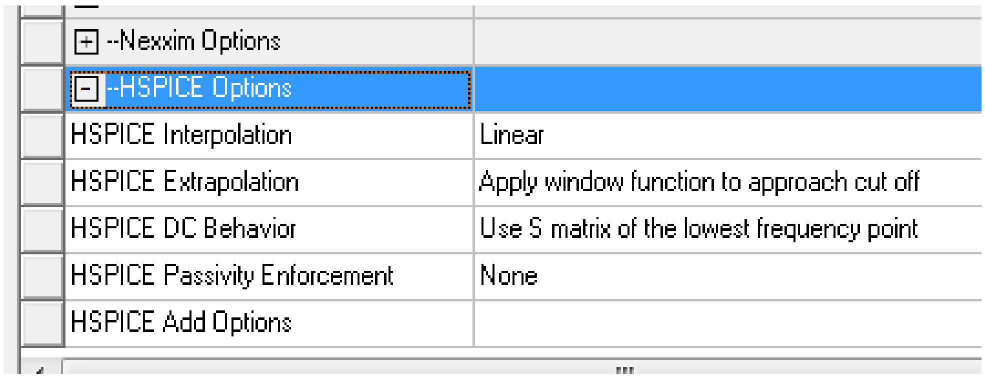
- The **Name** field shows the dynamic link component name
- The **File** field displays the SIwave design that is the source for the dynamic link.
- Enter a **Model Description** if appropriate.
- The **Design** field displays the name of the selected design.
- The **Solution** field displays the solution name.
- **Save project after simulate** specifies that the project is saved and closed once its data is accessed and the simulation is complete.
- **Unload project after use** specifies that the project is unloaded once its data is accessed and the window is closed.
- **Simulation option** allows selection of the dynamic link simulation method:
  - Cache solution (no dynamic simulation)
  - Dynamic simulation (reloads project and simulates as needed)
- **Interpolation algorithm** field to select an interpolation algorithm option:
  - **Automatic** – Linear interpolation is used if a full grid of solutions is present, otherwise Inverse Least Squares with Shadowing and Hyperplanes is used.
  - **Linear** – When a full grid of solutions is available, the cube of solutions which surrounds the solution to be interpolated is located. The corners of this cube are linearly averaged to determine the interpolated solution. If a full grid is not available, an error is reported and the interpolation fails.
  - **Inverse Least Squares with Shadowing and Hyperplanes** – Least squares interpolation with shadowing and hyperplanes.
  - **Inverse Least Squares with Shadowing, no Hyperplanes** – Least squares interpolation with shadowing, but no hyperplanes.

> **Note:** The main difference between **Linear** interpolation and **Inverse Least Squares with Shadowing** (ILSS) is as follows:
>
> - Linear only works with a full grid of solutions — ILSS works with arbitrary data.
> - Linear only considers nearest data — ILSS considers data that could possibly be distant, although shadowing mitigates this effect.
> - Linear is not first-order continuous — ILSS is first-order continuous.

- **Interpolate Y matrix** uses the Y matrix as the basis for interpolation. When **Interpolate Y matrix** is not selected, the S matrix is used as the basis. **Interpolate Y matrix** is selected by default. Circuit particulars determine which basis (S or Y) yields better results, and it is not possible to decide beforehand which one works best.
- **Only use independent variables during interpolation** suppresses the calculation of dependent variable values during interpolation.
- **Renormalize** – If selected, the Scattering parameter Data (S-data) received on the dynamic link are re-normalized with a reference impedance value specified in **Renorm Impedance** before passing the original dynamic link S-data to the Nexxim Circuit solver. If not selected, re-normalization does not take place and the original dynamic link S-data is passed to the Nexxim Circuit solver.
- **Renorm Impedance** refers to the reference impedance used for renormalization of S-data. The default is 50 Ohms.
- **Nominal Values** displays the value of parameters on the linked project.
- **Nexxim Options** allows you to view or set the following Nexxim simulator options:



- HSPICE options allows you to view or set the following HSPICE simulator options:



2. Click **OK** to close the Link **Properties** window.

Refresh Dynamic Link

- **Refresh Dynamic Link** on the SIwave Component menu retrieves ALL data on the linked design, including changes to connectivity, solutions, and properties. Choose this command after making any edits to the linked design.

Recapture SIwave Link Image

- **Recapture Link Image** on the SIwave Component menu recaptures an image of the design to use in the schematic symbol. See Capturing or Recapturing Bitmap Symbols for Dynamic Link Models.

Convert SIwave Dynamic Link to Parametric Snapshot

- **Convert to Parametric Snapshot** on the SIwave Component menu changes the dynamic link to a static parametric snapshot. After this option has been selected, the menu item toggles to **Convert to Dynamic Link**, which allows you to restore the dynamic linkage.

Clear SIwave Solution Cache

- **Clear Solution Cache** on the SIwave Component menu when clicked immediately clears the solution cache memory.

SIwave Project Link Menu

When the dynamic link project has been added to the design, click the subcircuit icon in the **Project Manager** window to display a brief menu. Use the menu items to create a copy of the dynamic link project, edit the link definition, recapture the link image, convert the dynamic link to a static parametric snapshot, clear the solution cache, or to display simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Running a Dynamic Link Co-simulation

To run a co-simulation involving a dynamic link subcircuit, right-click **Analysis** in the **Project Manager** window for the top-level circuit and select **Add Solution Setup**. Set up the top-level analysis as appropriate. An icon for the solution is added to the **Analysis** group box.

Right-click the top-level solution icon and select **Analyze**. The **Progress** window displays simulation updates for the top-level circuit and subcircuit simultaneously.

When simulation is successfully completed, view the results for the top-level circuit. See Generating Reports and Post Processing for further information.

> **Note:**
>
> You can create one or more solution setups local to a subcircuit by expanding the subcircuit icon, clicking on its local **Analysis** icon, and selecting **Add Solution Setup**. When you execute a local setup, it solves just the subcircuit.
>
> However, these local solution setups are not used in the co-simulation; any local setup information and options are ignored during co-simulation. The co-simulation parameters are set up as described in the topics on HFSS, Q3d, 2D Extractor and SIwave dynamic links.

Adding a Design Variable for Dynamic Links

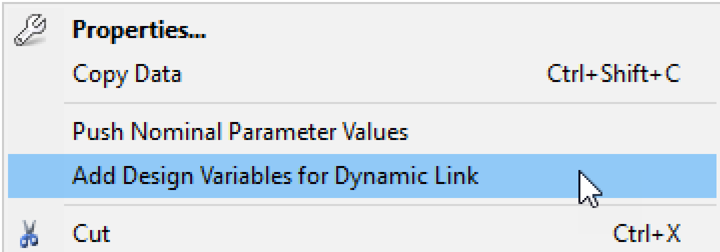
Design variables from an HFSS, Q3D, 2D Extractor, Icepak, or Mechanical design can be imported though a dynamic link to a Circuit or 3D Layout design. New design variables are created in the Circuit or 3D Layout design and correspond to the dynamic link variables. When the new design variables are created, 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.

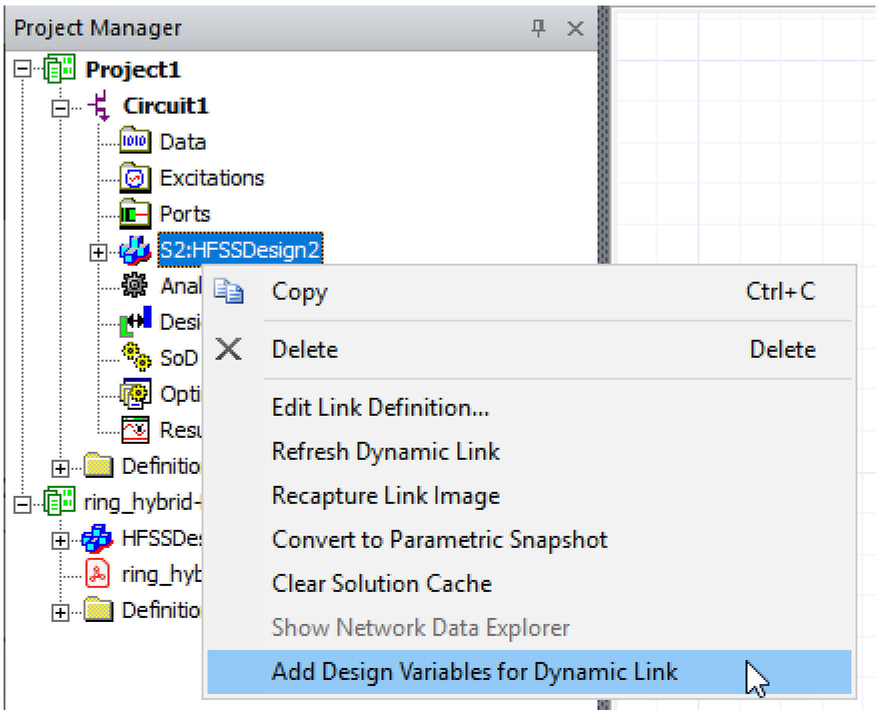
Design variables can be created from existing or newly added variables in the dynamic link design.

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**.



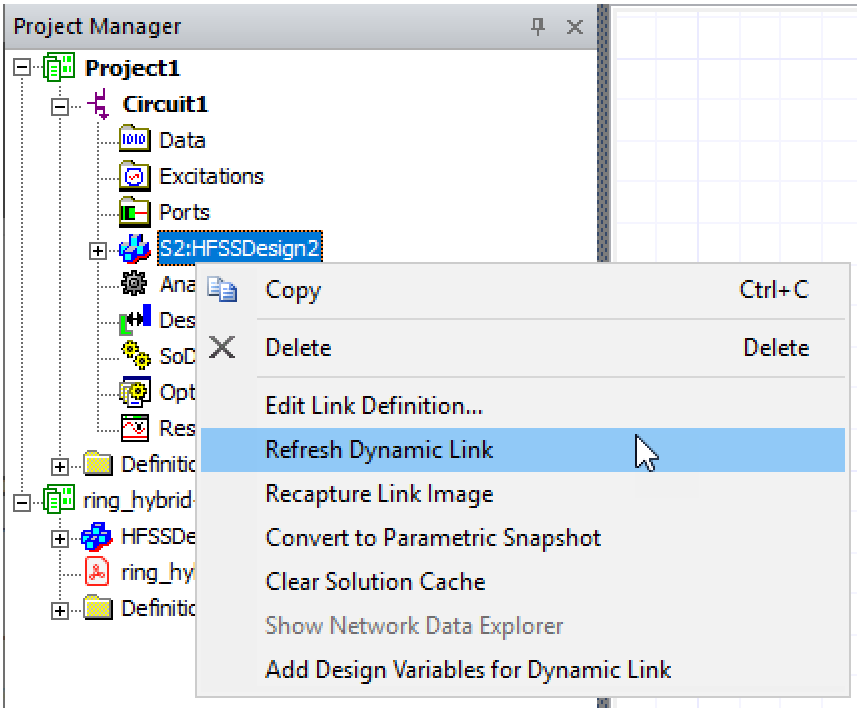
b. From the **Project Manager** window, right-click **Project Name** > **Design Name** > **Dynamic Link Design** and select **Add Design Variables for Dynamic Link**.



To create design variables based on variables added to a dynamic link design after it was linked:

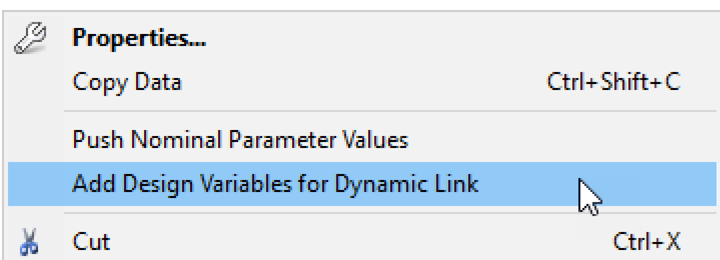
1. Create a new design variable in the dynamic link project.

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**.

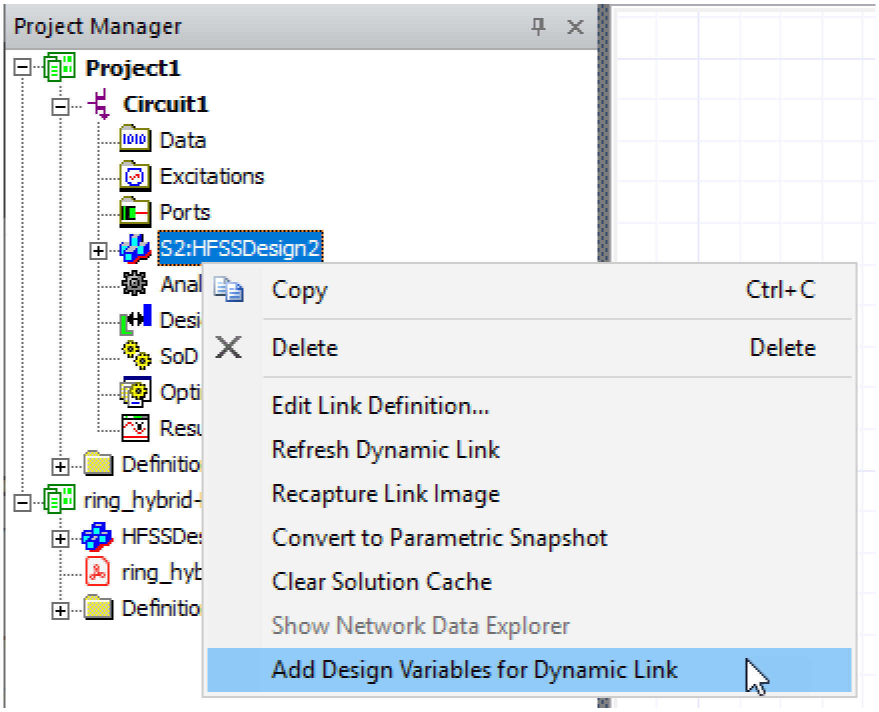


3. Either:

   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**.



Pushing Variable Values back to a Dynamic Link Design

If you have changed a parameter for a dynamic link component in the Ansys Electronics Desktop, right-clicking the component and choosing the **Push Nominal Parameter Values** command changes the corresponding design parameter in the linked design to have the changed value. This works for all link types except SIwave. Pushing variable values may be needed to enable the **Push Excitations** option to work correctly. See Pushing Excitations for details.

Managing Dynamic Link Simulation Setups

When the dynamic link project has been added to the design, click the subcircuit icon in the **Project Manager** window to display a brief menu. Use the menu items to create a copy of the dynamic link project, or to display simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Click any one of the dynamic link simulation setups to display a menu. Use the menu to select the sweep to be in the co-simulation, select ONLY this sweep, or display simulation results in Network Data Explorer. The **Show Network Data Explorer** item is activated only when there are simulation results to display.

Viewing Dynamic Link Geometry in the Layout Editor and a 3D View

Geometry from dynamically linked HFSS and Q3D models can be viewed in the **Electronics Desktop Layout Editor** and the 3D viewer. To do this, open the **Layout Editor** after inserting the HFSS or Q3D model into Ansys Electronics Desktop, double-click the model to open the **Properties** window and check the “Model Graphics” check box on the Footprint tab. When you close the **Properties** window Ansys Electronics Desktop copies a snapshot of the HFSS or Q3D model into Ansys Electronics Desktop that can be viewed in the layout or 3D editors. Note that if you change the model in HFSS or Q3D you must reload the model geometry into **Electronics Desktop** by clicking **Reload Graphics** on the Footprint tab of the Properties window.

Ansys Electronics Desktop respects the location and dimensions of the model and places it accordingly in Ansys Electronics Desktop’s layout. You can manipulate the view of the model in Ansys Electronics Desktop by setting certain properties in the **Properties** window: shift the model in X & Y with the Location property; rotate the model with the Angle property; flip the model with the Flipped check box; and scale the model with the Scaling property. Colors and transparency used to display the dynamically linked model are those set in HFSS or Q3D.

> **Note:** The top-down and 3D views of dynamically linked models is available only if the layout contains at least one stackup layer.

Capturing or Recapturing Bitmap Symbols for Dynamic Link Models

The automatically generated symbols for HFSS, Q3D, and 2D extractor models normally include a bitmap image of the model. The image may fail to render under certain circumstances, or may not show the appropriate view. You can update the bitmap image of the model. Right-click on the component symbol in the schematic and click **Recapture Link Image** to update the bitmap image associated with the model.

The symbols for components that have been placed in the schematic editor are not updated with the new bitmap view, however, new components that are generated by dragging and dropping the model onto the schematic window show the new bitmap image embedded in the symbol.

Pushing Excitations from Circuit Designs

You can set the excitations in dynamically linked field solver elements from a parent circuit so true electromagnetic field levels can be calculated by the electromagnetic solver (HFSS, SIwave, Planar EM, Q3D). This means the field values calculated by the electromagnetic solver element approximate the true field values when the circuit is active. These calculations are useful for active and passive antenna simulations, EMI/EMC simulations, and signal integrity applications. Ansys calls this "pushing excitations" from a circuit to a field solver.

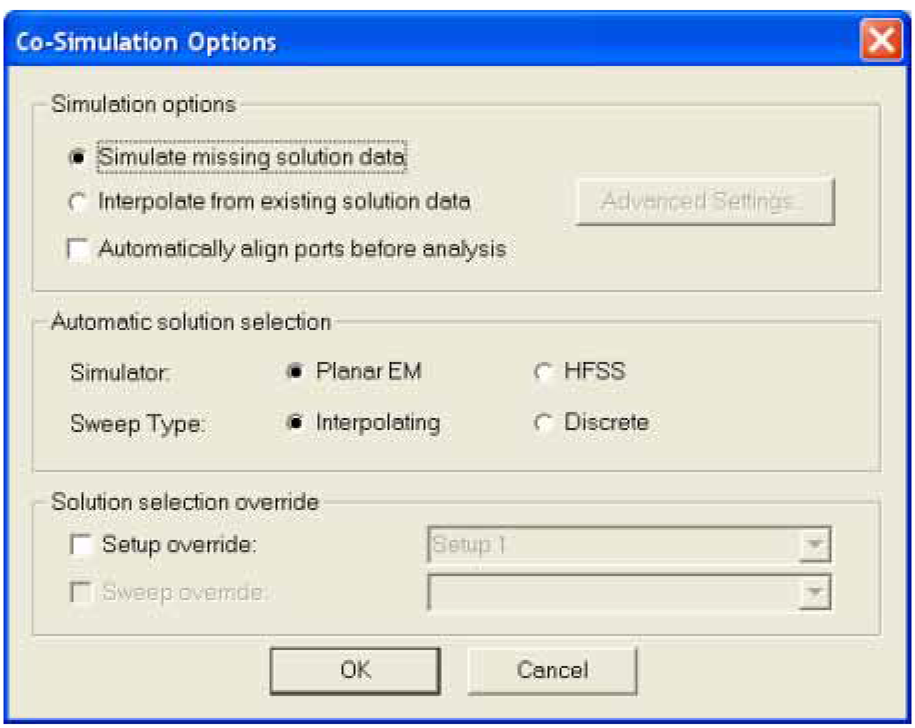
The procedure used to push excitations differs slightly depending on whether the electromagnetic simulator is in the Ansys Electronics Desktop, as is the case with the HFSS 3D Layout, Planar EM, HFSS, Q3D, and 2D Extractor simulators, or external to the Ansys Electronics Desktop, as is the case with SIwave.

To begin, insert a Planar EM, HFSS, Q3D, or SIwave model into a circuit using the techniques described in previous topics. While the circuit/field-solver combination can be simulated immediately and S-parameters can be extracted, some additional setup is required to view field values in the field solver element.

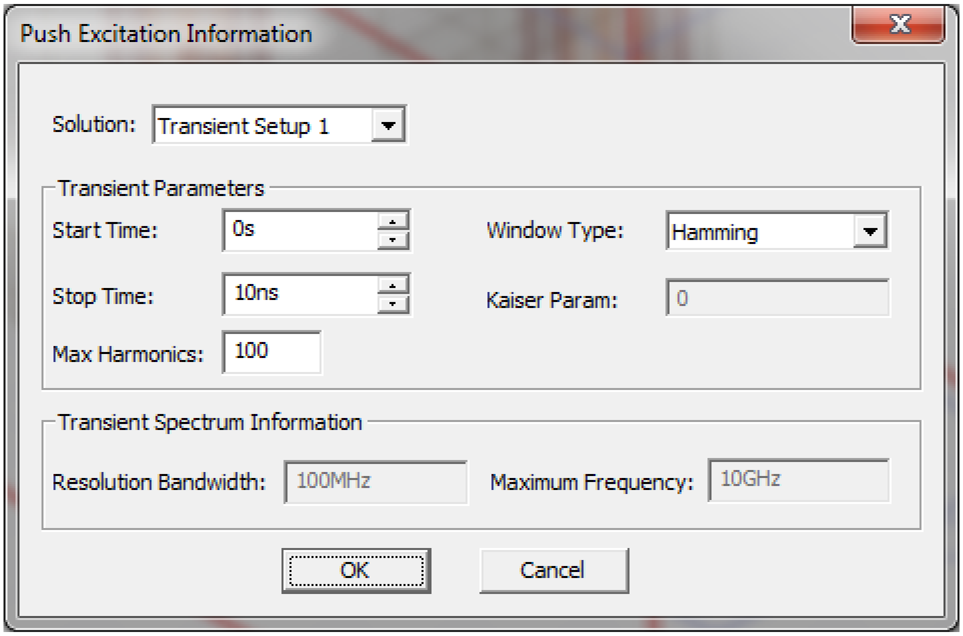
Pushing Excitations to an EM Design

In order to view fields and currents in an EM Design you must force the simulation engine to compute and save currents. To do this, navigate to the EM Design in the Ansys Electronics Desktop **Project Tree** and insert a simulation setup and discrete frequency sweep into the model. Be sure to select **Generate surface currents** in the EM Design model.

Override the default co-simulation options so the simulator uses this setup and sweep. Select the EM Design model in the **Project Manager** window, then click **EM Design > Solution Setup > Co-Simulation Options**. Then select the setup and sweep that save the EM Design currents.



When you simulate the top level circuit, the EM Design subcircuit is automatically simulated using the setup and sweep defined in the Co-Simulation Options window. After simulating the circuit/field-solver combination, right-click the field solver element in the **Schematic Editor** and choose which solution to use to populate the field solver excitation values.

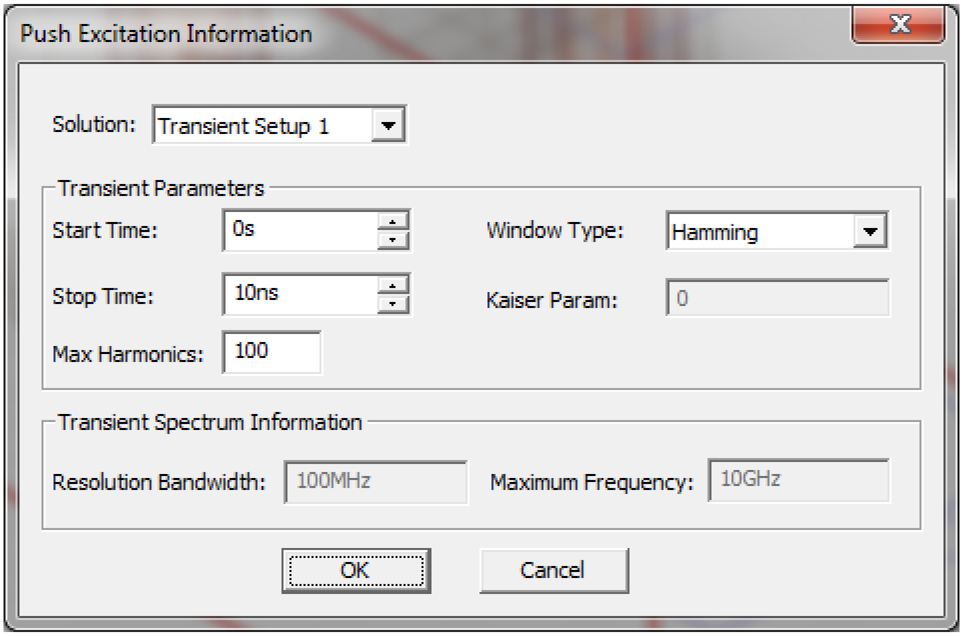


- The **ResolutionBandwidth** and **MaximumFrequency** of the data (that is sent or "pushed" to the dynamically linked project) are calculated based on the transient parameter settings: **StartTime**, **StopTime**, and **Max Harmonics**. The start and stop times must lie in the transient simulation time limits.
- Select the appropriate **Window Type**. When non-rectangular windows are used, the window automatically changes the power level of the signal. This is known as the "coherent gain" or "processing loss" of the window. The coherent gain is always adjusted to account for the processing loss of the window when pushing excitations to dynamically linked projects.

Pushing Excitations to an HFSS, SIwave, or Q3D Dynamic Link Model

The procedures for pushing excitations to dynamically linked models are all similar. You must first set up the field solver element so the fields and currents are saved during simulations (See the HFSS, SIwave, or Q3D documentation). After simulating the circuit/field-solver combination, right-click the field solver element in the **Schematic Editor**, select **Push Excitations** to launch the **Push Excitation Information** dialog, and choose which solution to use to populate the field solver excitation values.

> **Note:** When working with a Q3D Dynamic Link Model, the **Push Excitations** option will only be available if the beta option is enabled. Additionally, the **Push Excitations** option will only become available after simulating the circuit/field-solver combination.



- The **ResolutionBandwidth** and **MaximumFrequency** of the data (that is sent or "pushed" to the dynamically linked project) are calculated based on the transient parameter settings: **StartTime**, **StopTime**, and **Max Harmonics**. The start and stop times must lie in the transient simulation time limits.
- Select the appropriate **Window Type**. When non-rectangular windows are used, the window automatically changes the power level of the signal. This is known as the "coherent gain" or "processing loss" of the window. The coherent gain is always adjusted to account for the processing loss of the window when pushing excitations to dynamically linked projects.

> **Note:**
>
> - Current quantities, rather than voltage quantities, are pushed to SIwave or Q3D.
> - For SIwave or Q3D, push excitations is available for Transient solution only.
> - When pushing excitations to SIwave, Transient Parameters are disabled as FFT is not available from this dialog.

Note:

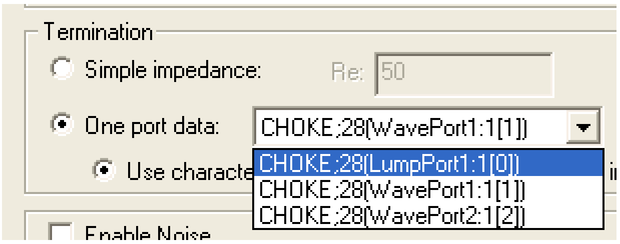
1. The most common reason Push Excitations fails is that one or more terminals of the dynamically linked model are left unconnected. Nexxim does not compute voltages on unconnected terminals so no excitations can be determined. Adding short unconnected wires to the unconnected terminals of the model causes Ansys Electronics Desktop to assign unique net names to the terminals and allow Nexxim to compute voltages. Once Nexxim has computed voltages excitations can normally be pushed to the dynamically linked design.
2. If there is no voltage source in the design, an error appears. You can either cancel the push or add a 1V voltage source, resimulate the design, and push the excitations.
3. Excitations cannot be pushed to HFSS or SIwave if parameter values are swept in the Ansys Electronics Desktop setup.
4. For SIwave simulations, when **Enforce Causality** is selected, if the frequency step is too big it can result in an inconclusive test for causality enforcement. In this case, the causality checker automatically retries the simulation using rational interpolation and analytical integration.
5. For HFSS dynamic links, Push Excitations also pushes available-power to the HFSS Design. The amount of the power pushed is the sum of the available powers calculated at each Circuit Design port connected to the dynamic link — directly or indirectly — with the following caveats:
   - The port has either an associated voltage or power source (current sources are not supported).
   - The solution data being pushed is for a Linear Frequency Network simulation (no restrictions on port impedance values).
   - The solution data is for an HB or Transient simulation AND the port impedance is a constant value (e.g., 50 ohms).

   The pushed available-power shows up in the linked HFSS Design in its **Edit post-process sources** window that is accessible on the Field Overlays or Excitations icon shortcut menu. In HFSS, this power is called System Power and is used only for calculating 'System Gain' in the reporter and 'Total Realized System Gain' in antenna calculations.

Dynamic Port Termination with a Dynamic Link Model

Nexxim can use port impedances set dynamically on the field solver cosimulator. The port impedances can be frequency-dependent. The Nexxim setup specifies the frequency range of interest to the Nexxim design, say f1 to f3. The field solver tool is set to run over a range of frequencies, f3 to f4, and can provide impedance data for that range. If the frequency range for which port impedance data is dynamically determined (f3, f4) is a subset of the frequency range of interest (f1, f2), then the non-overlapping frequencies have the same impedances as the end frequencies.

To specify a dynamic impedance, double-click the port and select **Edit Port**. In the Termination group box of the Edit Port window, select **One port data** and **Use characteristic impedance**. From the drop-down menu, select the field solver port which is to supply the impedance data for this Interface port.



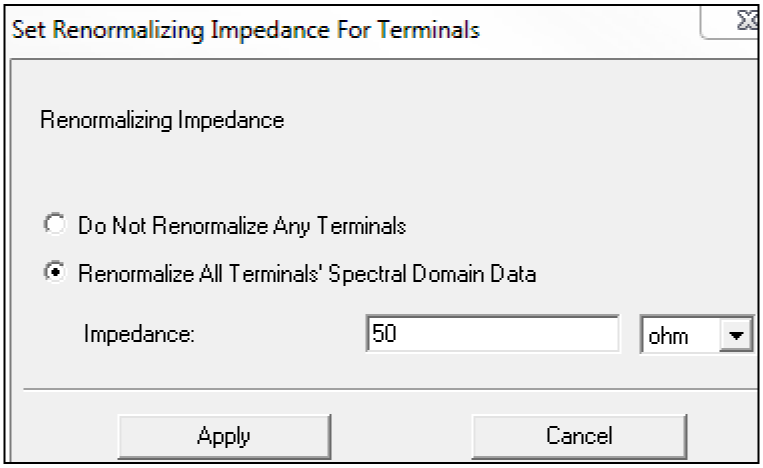
Users can specify a static list of frequencies and corresponding impedance in the following ways:

- In the **Termination** group box, select **One port data** and **Use impedance matrix**. Click **Create New** to open the **N-port data** window. Select a file with 1 port s-parameter data. Click **OK** to create a model. Back in the **Port Definition** window, click this model on the **One port data** menu.
- To enter data manually, first create a NPort_NoFile model with 1 port following these instructions. Place the component in the schematic for this model. In the **Termination** group box of the **Port Definition** window, click **One port data**, this model on the **One port data** menu, and **Use impedance matrix**.

The impedances are interpolated for frequencies intermediate between given points. Impedances for frequencies outside the range given take the end values.

Set Renormalizing Impedance for Terminals

The setup panel for each terminal includes its post processing renormalizing impedance. You can set this value either for all excitations, or for a specific port. If a design includes at least one wave port, the setup panel also includes the radio buttons **Do Not Renormalize Any Terminals**, or **Renormalize All Terminals' Spectral Domain Data** .



To set the renormalizing impedance for **all excitations**:

1. Right-click **Excitations** in the Project tree and select **Set Terminal Renormalizing Impedances**.

   The **Set Renormalizing Impedance for Terminals** dialog box appears.

2. In the field for **Impedance**, set the value, and select the units from the pull down.

   The format is "<real_part> + <complex_part>*i ohm", for example 50ohm + (-5ohm) * 1i

   > **Note:**
   >
   > The reference impedance is meant to represent the component modeled by the lumped port. You can assign a variable to these values, (for example " resistance + (reactance) * 1i"). This variable can be dependent on the frequency, which allows use of a dataset for frequency dependent impedance, (for example, pwl (ds1,freq) + (pwl(ds2,freq)) * 1i).

3. Click the **Apply** button to close the dialog and apply the change.

To Set the Reference for All terminals on a Port:

1. Right-click the port icon in the Project tree and click **Set Terminal Renormalizing Impedances**.

   The **Set Renormalizing Impedance for Terminals** dialog appears. It differs from the related command for all excitations by specifying that the **Renormalizing Impedance** is for terminals on the selected port.

2. In the field for **Impedance**, set the value, and select the units from the pull down.

   This value can be a variable. This variable can be dependent on the frequency, which allows use of a dataset for frequency dependent impedance.

3. Click the **Apply** button to assign the impedance value.

> **Note:**
>
> For more information, see Scaling a Source Magnitude and Phase.

You can also set the Terminal Reference Impedance on a port by selecting the port and editing the value in the **Properties** dialog.

In designs with at least one wave port, where you want to view un-renormalized Zo impedance and/or the corresponding S parameters in either the Matrix data or in a report, you can select the **Do Not Renormalize Any Terminals** radio button.

Solver On Demand

Solver-on-Demand allows you to specify the simulator model to use for each component in a design. The simulator model for a given component can be switched between simulations.

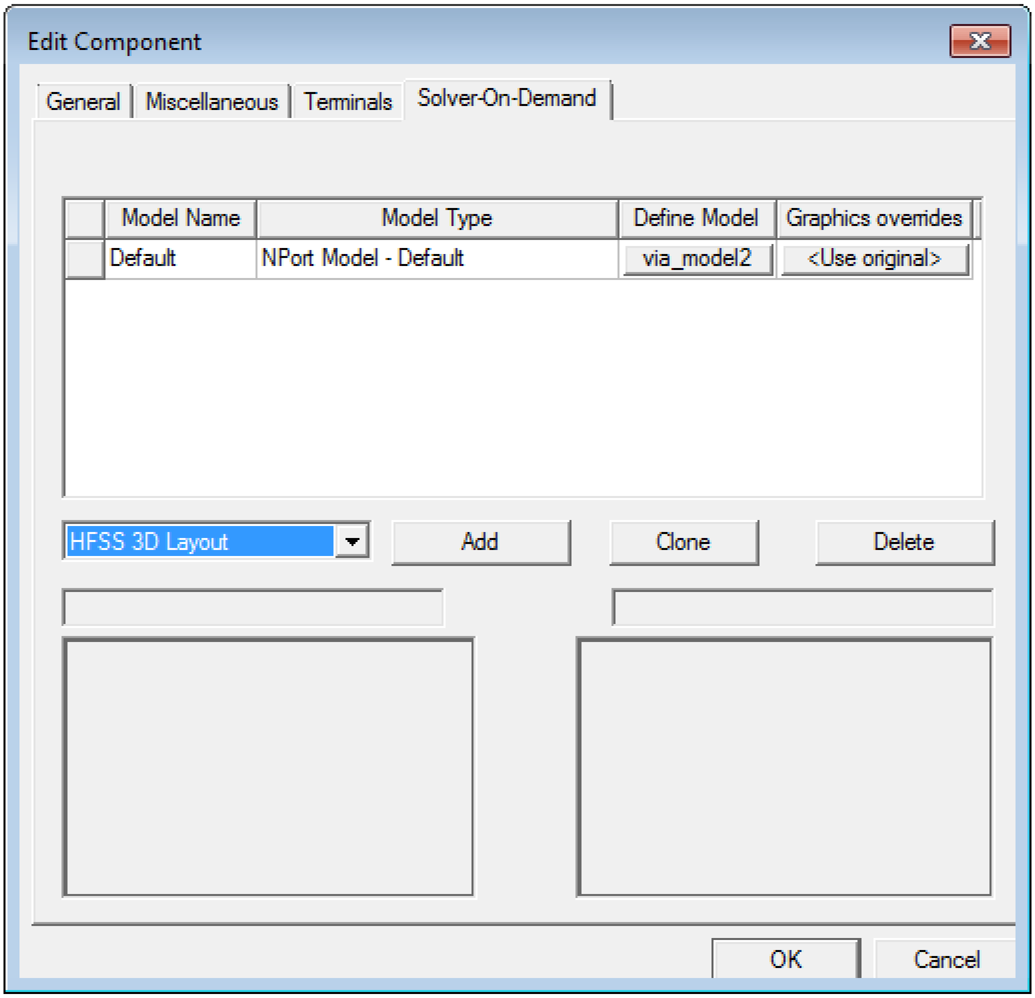
Setting Up Models for a Component Type

The first step is setting up for **Solver-on-Demand** is to define the models that is available for each component *type*. All instances of that component type inherits the defined set of Solver On Demand models.

1. Open the **Definitions** folder on the **Project Manager** window, then open the **Components** subfolder.
2. Double-click the component in the list to open the Edit Component window. This window sets parameters that affect all components of the given type.

> **Note:**
>
> Alternative ways to open the **Edit Component** window are:
>
> - Right-click a component instance that has already been placed in the schematic and choose **Edit Component**.
> - From the **Component Libraries** docking window, right-click the component and choose **Edit Component** in either the *Components* or **Symbols** tabs.
>
> See Using the Component Editor for further information.

3. Select the **Solver-on-Demand** tab:



To add a model for Solver On Demand:

1. Make a selection on the model type drop-down menu. Then click **Add**. A new model of the selected type is added to the list. The symbol and footprint preview windows at the bottom of the window are automatically updated to reflect whatever changes you make to a component's symbol/footprint definitions.
2. Click in the **ModelName** field for the new model and type the name you want to use for this model (or accept the default which already appears in the **ModelName** field). Note that when you highlight a **Model Name**, its symbol and footprint (if defined) are automatically previewed in the windows at the bottom of the window. These preview windows are automatically updated to reflect whatever changes you make to the component's symbol/footprint definitions.
3. Click the button in the appropriate **Define Model** field to open a **Cosim** or **Netlist** definition window that is specific to the model type.

4. Click the button in the appropriate **Graphics overrides** field to open an **Edit overrides** window to modify **Symbol/Footprint** settings or **Terminals** settings. For more information see General Tab, Terminals Tab, and Solver On Demand Symbol and Footprint Override.
5. Repeat steps 1 through 4 to create all the models you wish to use for this type of component.
6. Click **OK** when you have defined all the models for this component type.

Repeat the above procedure to specify models for all components that is simulated using Solver On Demand, Dynamic Links, or static models.

> **Note:**
>
> After a component type has been created, Edit the definition. Two common ways to edit a Component Type are:
>
> - Right-click a component instance that has already been placed in the schematic and choose **Edit Component**.
> - From the **Component Libraries** docking window, right-click the component and choose **Edit Component** in either the **Components** tab or the **Symbols** tab.
>
> See Using the Component Editor for further information.

Solver-on-Demand components are solved from internal model data: netlists for Nexxim, Nexsys, and HSPICE, or stackups (layout, substrate, footprint) for HFSS 3D Layout and Planar EM.

Select on the following to define a Solver On Demand component model type:

- HFSS 3D Layout
- MultiProduct Netlist
- Netlist

Dynamic Link components link to active HFSS, Q3D, 2D Extractor, or SIwave projects. Select on the following to define a Dynamic Link component model type:

- HFSS Model
- Q3D Model
- 2D Extractor Model
- SIwave Model

Static models contain a single set of solution data such as SYZ-parameters. Select on the following to define a static component model type:

- NPort Model
- Parametric NPort Model

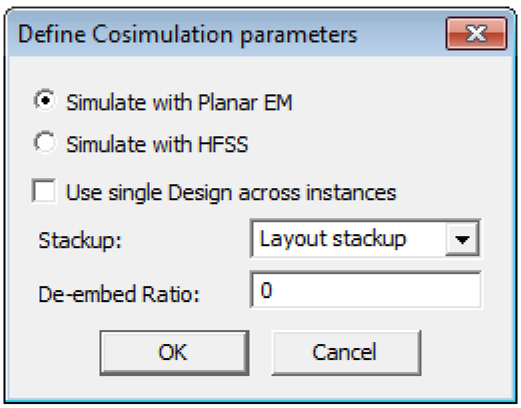
- Matlab Model
- State-Space Model

The Custom and Bypass model types are available for special needs:

- Custom
- Bypass

HFSS 3D Layout as Solver On Demand

HFSS 3D Layout and Planar EM are available field solvers that simulate from a layout, substrate, or footprint defined in the component itself. From the **Solver-on-Demand** tab of the **Edit Component** window, select **HFSS 3D Layout** on the drop-down menu and click **Add** to open the co-simulation definition window.



- Select **HFSS** or **Planar EM** as the solver for the co-simulation.
- **Stackup** – Set to one of three values (Layout, Substrate, or Footprint) to determine which stackup definition are used to model the component when it is simulated by the selected solver. For more information see Stackup Definitions.

Solver-on-Demand with the HFSS 3D Layout or Planar EM tool requires the component possess a physical definition, in the form of a footprint. Many components in the Ansys Electronics Desktop® library have footprints defined so they can be simulated using HFSS 3D Layout or Planar EM as the **Solver-on-Demand**. When you define a component, it can be simulated using **Solver-on-Demand** as long as the component possesses a physical definition. For more information see Physical Definition.

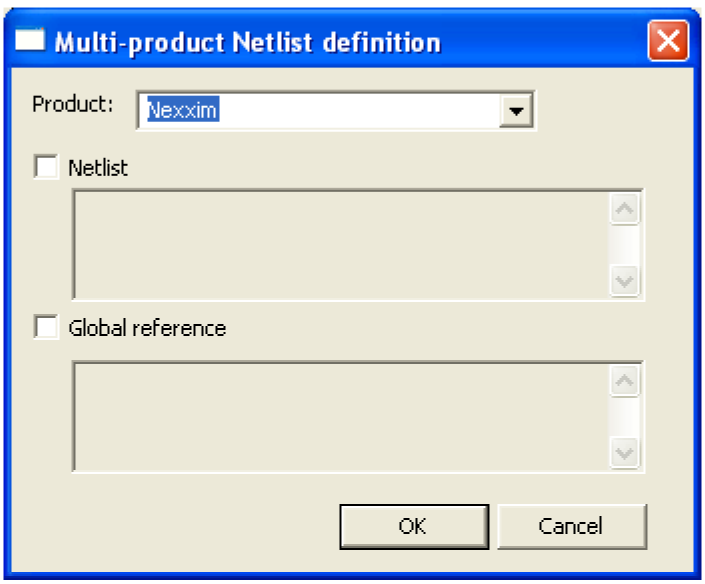
- **De-embed Ratio** – Determines the manner in which excitation signals are fed to the component via transmission lines during EM simulation. Short sections of transmission line are automatically added to the ports of the **Solver-on-Demand** component to allow discontinuities near the ports to decay in the EM simulation. The length of these transmission lines is equal to the De-embed Ratio times the width of the **Solver-on-Demand** port. That is, Length = De-embed Ratio * W (port connection). The Planar EM solution data is automatically de-embedded back to the initial **Solver-on-Demand** component geometry to remove the effects of the additional transmission lines.

- **Use single Design across instances** enables the same design to be used for all instances of the component type, eliminating the need to specify the **Solver-on-Demand** model individually.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Multiproduct Netlist in Solver On Demand

Nexxim/Nexsys and HSPICE are available circuit solvers that simulate from netlists defined in the component itself. In the **Solver-on-Demand** tab of the **Edit Component** window, when choose **MultiProduct Netlist** on the model drop-down menu and click **Add** to open the netlist definition window.

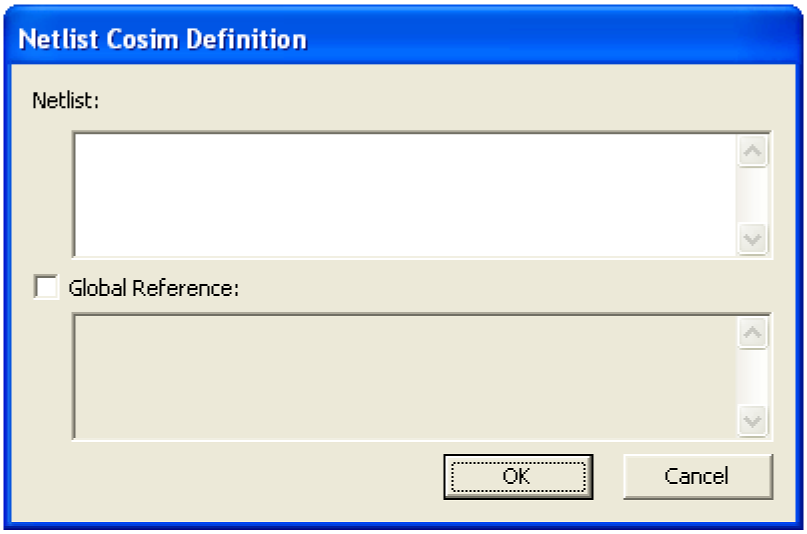


- Click the drop-down menu **Product** menu to choose which **Electronics Desktop** product netlist to define. The choices are Nexxim (which includes Nexsys) and HSPICE.
- Click in the **Netlist** field to define a new netlist.
- Click **Global reference** to define a global reference. The Global Reference group box is for any text that is to appear in the netlist outside of the circuit block, including substrate definitions, model definitions, references to external files, parameters, and comments to document the design.

Click **OK** to implement your changes, close the window, and return to the **Edit Component** window.

Netlist in Solver On Demand

The simple Netlist model is invoked for whatever circuit-level simulator is running. In the **Solver-on-Demand** tab of the **Edit Component** window, when choose **Netlist** from the model drop-down menu and click **Add** to open the co-simulation definition window. (The same window opens when you click **Edit** with a previously added **Netlist** model name selected).



- Click in the **Netlist** field to define a new netlist or make changes to an existing netlist
- Click **Global reference** to define a global reference. The Global Reference group box is for any text that is to appear in the netlist outside of the circuit block, including substrate definitions, model definitions, references to external files, parameters, and comments to document the design.
- Click **OK** to implement your changes, close the window, and return to the **Edit Component** window

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

HFSS Dynamic Link Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a dynamic link to an HFSS project.

1. From the **Solver-on-Demand** tab of the **Edit Component** window, select **HFSS Model** > **Add** to open an explorer window.
2. Browse (...) to select an HFSS project for the dynamic link.
3. Click **Open**. Ansys Electronics Desktop pauses for a few seconds while HFSS starts up. The **HFSS Dynamic Link** window are displayed

See Adding a Dynamic Link from HFSS for further information.

Click **OK** to save your changes and close the Edit Component window, or click **Cancel** to close the window without saving any changes.

Q3D Dynamic Link Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a dynamic link to a Q3D project.

1. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **Q3D Model** on the model drop-down menu and click **Add** to open an explorer window.
2. Browse (...) to select a Q3D project for the dynamic link.
3. Click **Open**. Ansys Electronics Desktop pauses for a few seconds while Q3D starts up. The **Q3D Dynamic Link** window are displayed

See Adding a Dynamic Link from Q3D for further information.

Click **OK** to save your changes and close the Edit Component window, or click **Cancel** to close the window without saving any changes.

2D Extractor Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a dynamic link to a 2D Extractor project.

1. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **2D Extractor Model** from the model drop-down menu and click **Add** to open the **File Open** window.
2. Browse (...) to select a Q3D project for the dynamic link.
3. Click **Open**. Ansys Electronics Desktop pauses for a few seconds while Q3D starts up. The **2D Extractor Dynamic Link** window then appears

See Adding a Dynamic Link from 2DExtractor for further information.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

SIwave Dynamic Link Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a dynamic link to an SIwave project.

1. From the **Solver-on-Demand** tab of the **Edit Component** window, select **SIwave Model** on the model drop-down menu and click **Add** to open an explorer window.
2. Browse (...) to select an SIwave project for the dynamic link.
3. Click **Open**. Ansys Electronics Desktop pauses for a few seconds while SIwave starts up.

See Adding a Dynamic Link from SIwave for further information.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

NPort Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a link to a file with static solution data such as SYZ-parameters.

1. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **NPort Model** on the model drop-down menu and click **Add** to open an explorer window.
2. Browse (...) to select an appropriate solution data file for the component.
3. Click **Open**.

See Creating an N-Port Model for further information.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Parametric NPort Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a link to a file with static parametric solution data.

1. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **Parametric NPort Model** from the model drop-down menu and click **Add** to open an explorer window.
2. Browse (...) to select an appropriate solution data file for the component.
3. Click **Open**.

See Creating a Parametric N-Port Model for further information.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Matlab Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a static link to a file with static model data such as a Nexsys Matlab model.

1. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **Matlab Model** on the model drop-down menu and click **Add** to open an explorer window.
2. Browse (...) to select an appropriate Matlab model file for the component.
3. Click **Open**.

See Creating a MATLAB UDM Schematic Component for further information.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

State-Space Model in Solver On Demand

Click the **Add** Model menu in Solver On Demand to set up a static link to a file with static solution data such as SYZ-parameters.

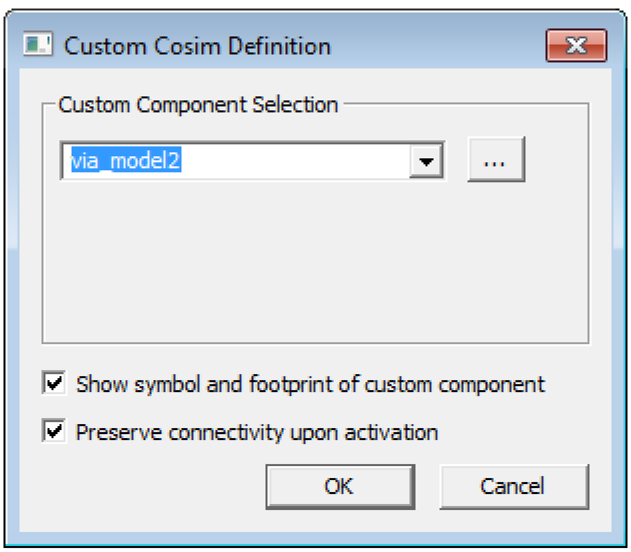
1. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **State-space Model** from the model drop-down menu and click **Add** to open an explorer window.
2. Browse (...) to select an appropriate solution data file for the component.
3. Click **Open**.

See State-Space Model Support for further information.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Custom Model in Solver On Demand

Add a new component that references an existing component definition. In the **Solver-on-Demand** tab of the **Edit Component** window, choose **Custom** on the model drop-down menu and click **Add** to open the co-simulation definition window.



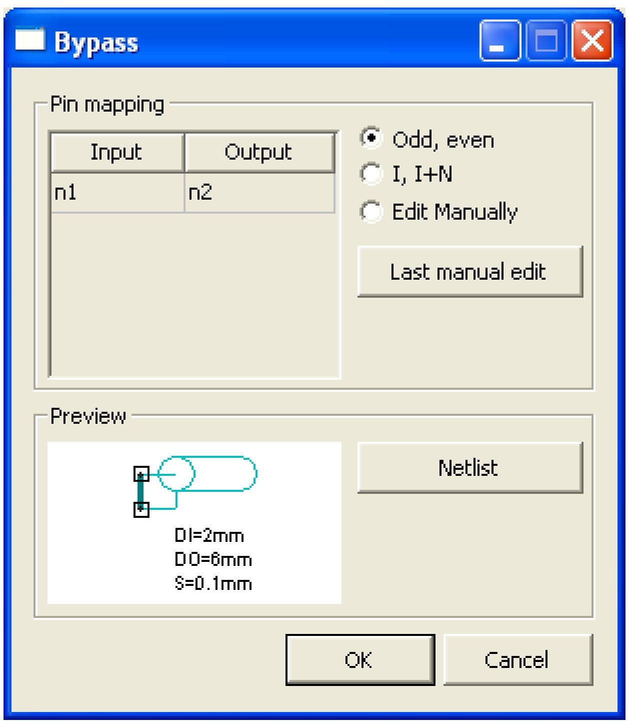
- Click the drop-down menu to alter the choice of **Custom Component Selection**, or browse for the selection (...). The menu lists the components already placed in the design that have the same number of terminals as the Custom component.
- If appropriate, uncheck **Show symbol and footprint of custom component** (on by default).
- If appropriate, uncheck **Preserve connectivity upon activation** (on by default).

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Bypass in Solver On Demand

The **Bypass** Solver-on-Demand model allows you to auto-generate netlists to bypass a component through .CONNECT expressions. Using the **Bypass** definition window, customize the pin connections of a component. For two-port components, a **Bypass** model is created automatically. For components with more than two ports, it is necessary to define the **Bypass** pin mapping.

In the **Solver-on-Demand** tab of the **Edit Component** window, choose **Bypass** on the model drop-down menu and click **Add** to open the **Bypass** window. Alternately, If a **Bypass** model is already listed in the **Edit Component** window, click in the **Define Model Value** field to open the **Bypass** window.



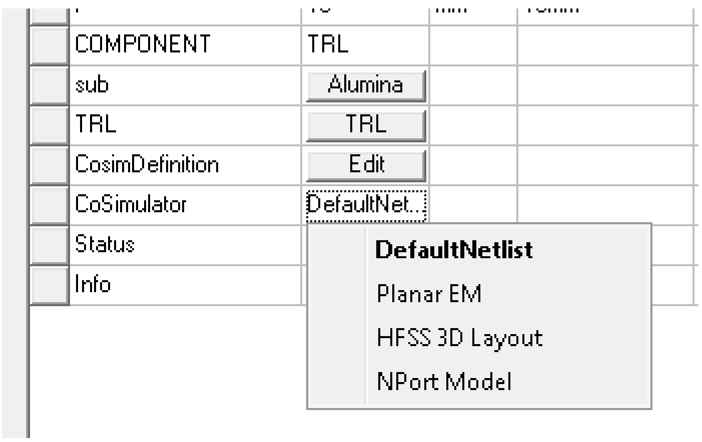
- The **Bypass** window displays the current pin mappings.
- Select **Odd, even** or **I, I+N** to use a predefined pin combination.
- Click **Edit** to manually define connections. To define a connection, in the **pin mapping** grid, first select on the **Input** drop-down menu, then select from the **Output** drop-down menu.
- Select **Last manual edit** to cancel any changes you have made and return to the last saved pin combination.

- Select **Reset** to reset the pin combinations to their preset defaults.
- Click **Netlist** to see the **Bypass** netlist.
- Once the pin combination is defined, a preview of the component's design is displayed in the **Preview** window.

Click **OK** to save your changes and close the window, or click **Cancel** to close the window without saving any changes.

Selecting a Model for an Individual Component

The **Solver-on-Demand** model type can be defined for each individual component in the schematic. Initially, the default solver, **DefaultNetlist**, applies to all components. To specify a different solver for a component, select the component in the Schematic Editor. The properties for that component are then displayed in the Property window. Scroll down to the CoSimulator property:

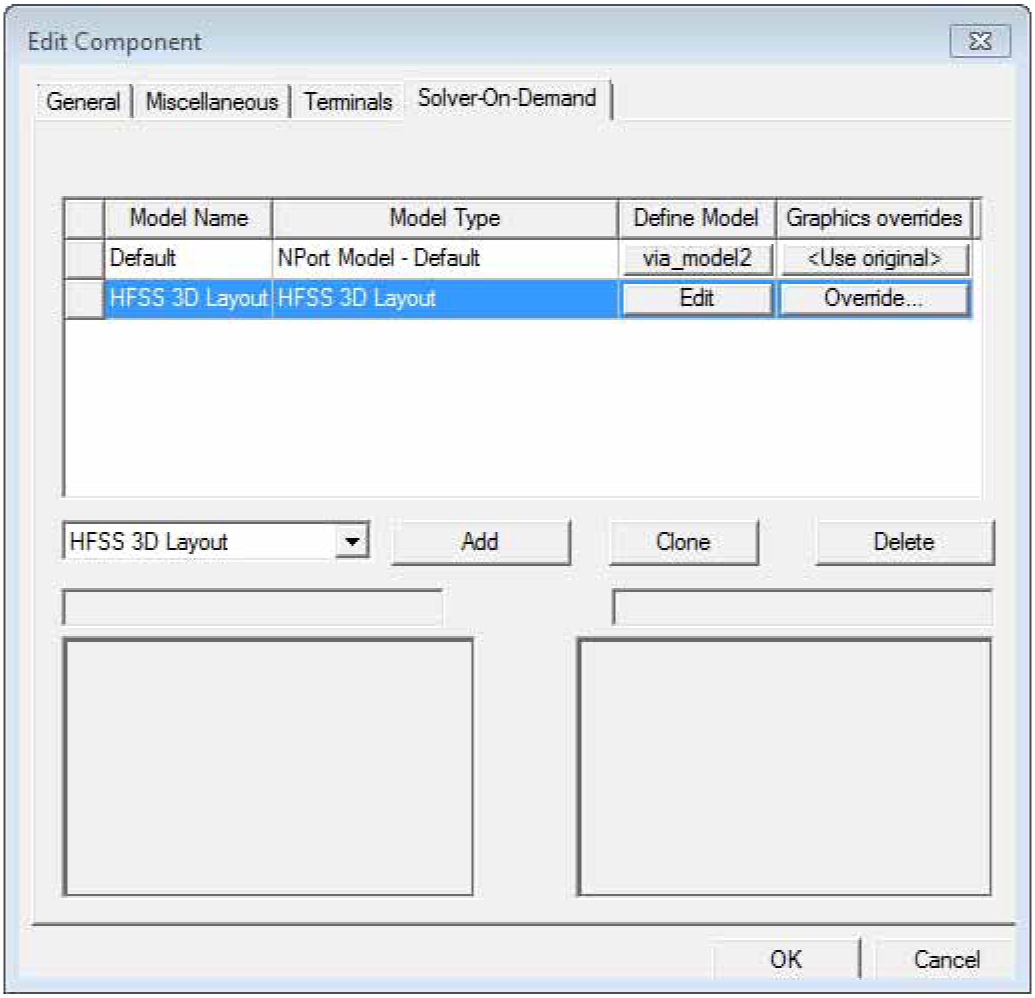


Click in the **Value** field to open the dropdown menu. The drop-down menu shows all the models that have been defined for this type of component. Click a model to select it and close the drop-down. Repeat this process for all components that uses **Solver On Demand** models other than the **DefaultNetlist**.

For more information about selecting and editing component properties, see Using the Component Editor.

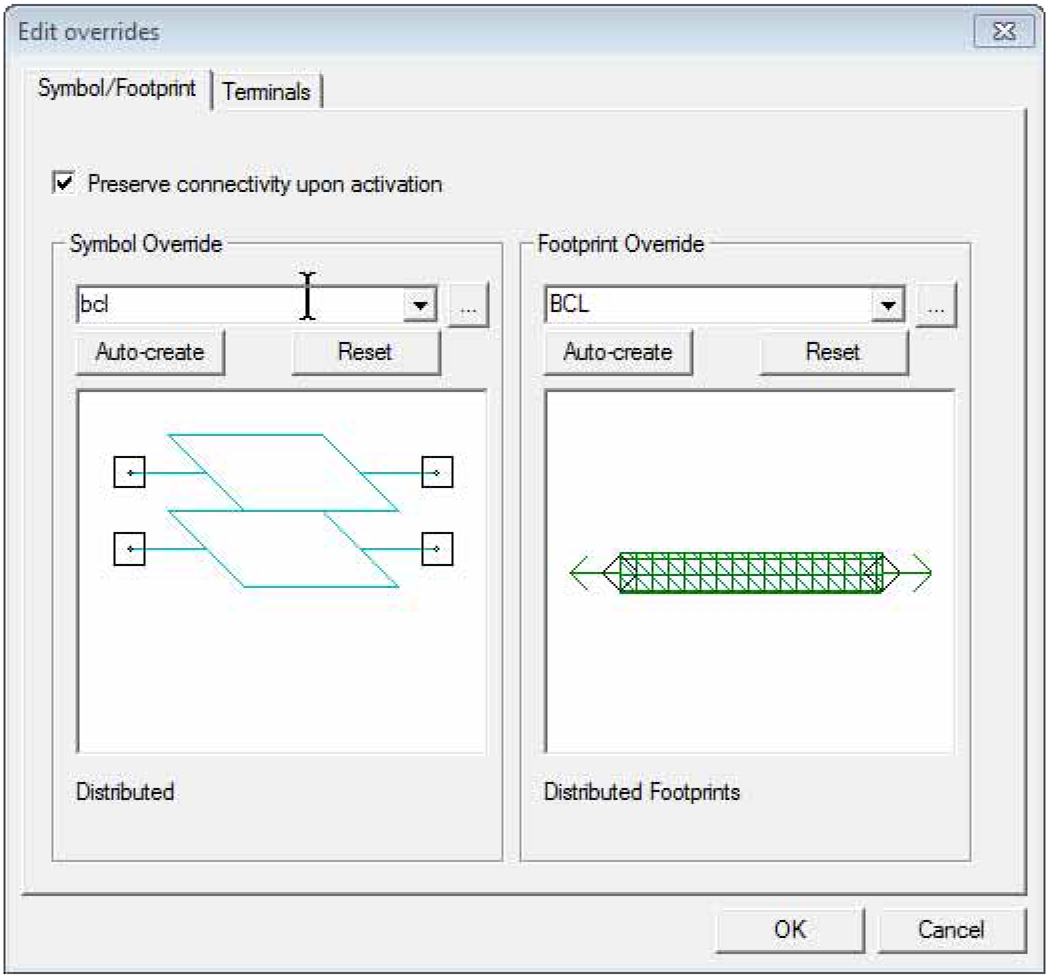
Solver on Demand Symbol and Footprint Override

The **Solver-on-Demand** tab in the Edit Component window has symbol and footprint image previews which show the symbol/footprint that is displayed in the Schematic/Layout. The symbol and footprint images can be overridden when you define each model **Cosim Definition**.



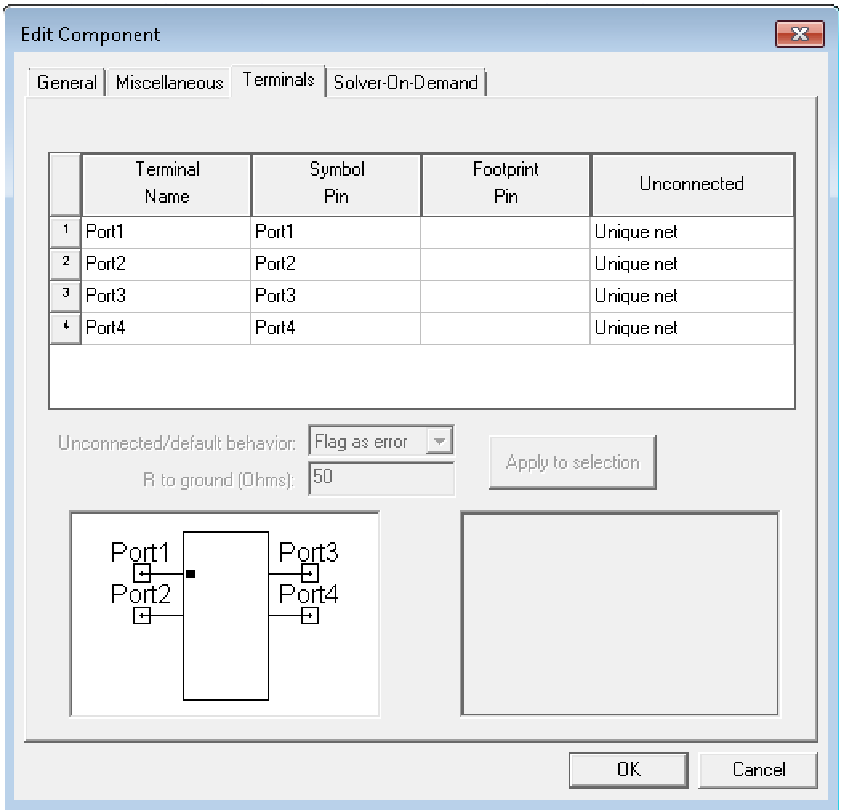
Click **Override** in the **Graphics overrides** column to open the **Edit overrides** window. You can then override the symbol/footprint using the **Symbol** and **Footprint** drop-down menus of the window.

In the following figure, the override symbol and footprint (in this case, **bcle** and **BCL**) are displayed in the symbol/footprint preview windows at the bottom of the window. These are the same symbol/footprint that is displayed in Schematic/Layout when the component is selected.



- If appropriate, click **Preserve connectivity upon activation.**
- **Auto-create** a symbol, or instead **Reset** the symbol/footprint.
- Or **Browse** to choose a symbol/footprint from an existing library.
- Once finished, click **OK** to commit changes.

The chosen symbol and footprint are displayed in Schematic/Layout when the component is selected. You can modify terminal assignments to the symbol or footprint by clicking the **Terminals** tab and editing its contents.



For more information see Components Terminals Tab.

23 - Ansys Workbench Integration Overview

Ansys Workbench combines the strength of its core product solvers with the project management tools necessary to manage project workflow. In Ansys Workbench, analyses are built as *systems*, which can then be combined into a *project*. The project is driven by a schematic workflow that manages the connections between the systems.

From the schematic, you can interact with applications (called workspaces) that are native to Ansys Workbench and that display within the Ansys Workbench interface. Native workspaces include: Project Schematic, Engineering Data, and Design Exploration (Parameters and Design Points).

You can also launch applications that are data-integrated with Ansys Workbench, meaning the application's interface remains separate, but the data from the application communicates with the native Ansys Workbench data. Thus, data can be passed back and forth between any Ansys Electromagnetics product on a Workbench Project Schematic and any supported Ansys or Ansys Electromagnetics desktop product. Depending on the application, data integration can include basic actions such as saving projects, as well as more complex actions such as the coupling of Ansys Electromagnetics product variables to Workbench Design Exploration parameters.

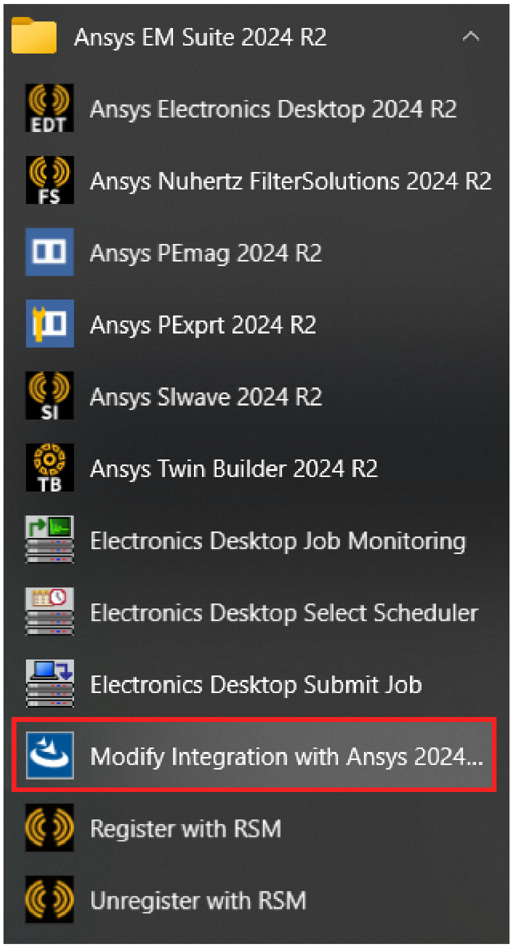
Data-integrated applications include the following Ansys Electromagnetics products: Circuit, HFSS, Maxwell/RMxprt, Q3D Extractor, and Twin Builder.

> **Note:**
>
> For detailed information on working with Ansys Workbench, please refer to the Workbench documentation.

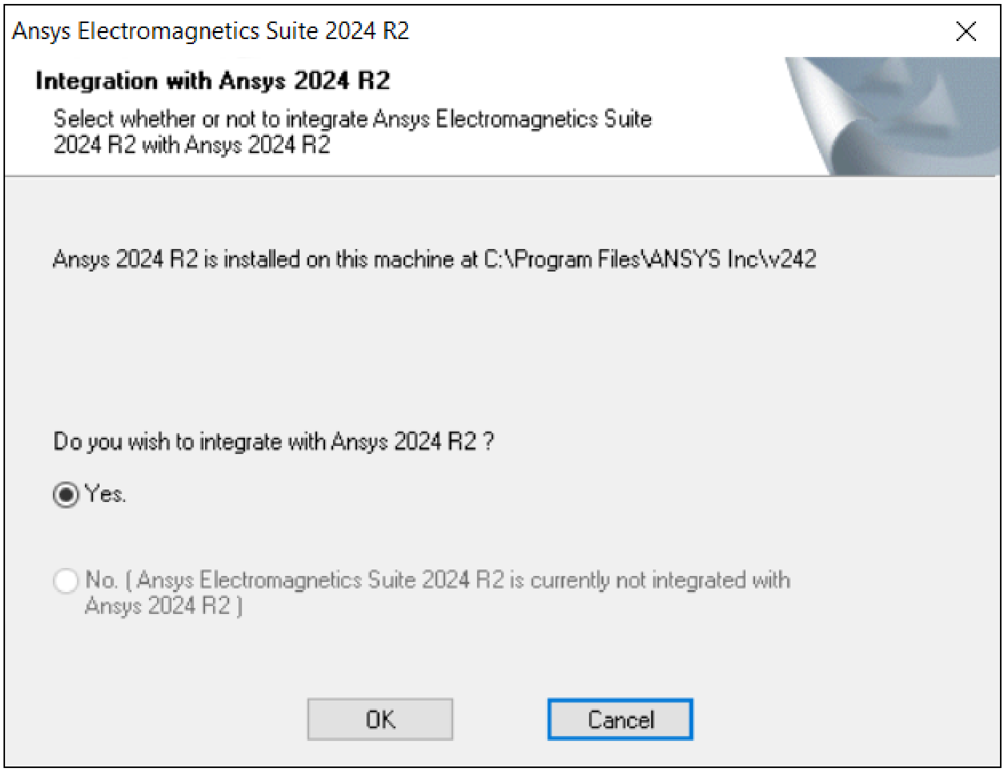
Integrating Ansys Electromagnetics Products with Ansys Workbench

After installation, you can integrate Ansys Electromagnetics products with Ansys Workbench:

1. From the **Start** menu, select **Ansys EM Suite 2024 R2** > **Modify Integration with Ansys...**

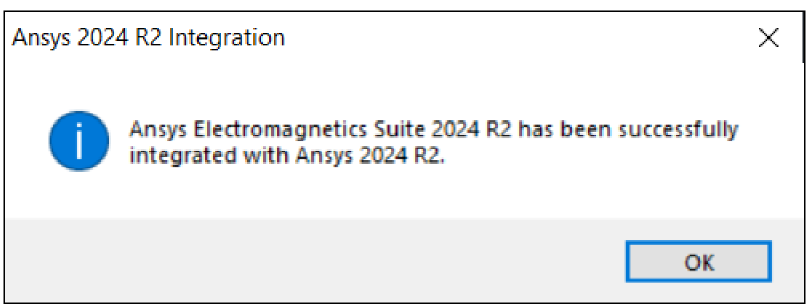


2. When the **Integration with Ansys** step displays, ensure that the **Yes** radio button is selected.



3. Click **OK** to complete the integration process.

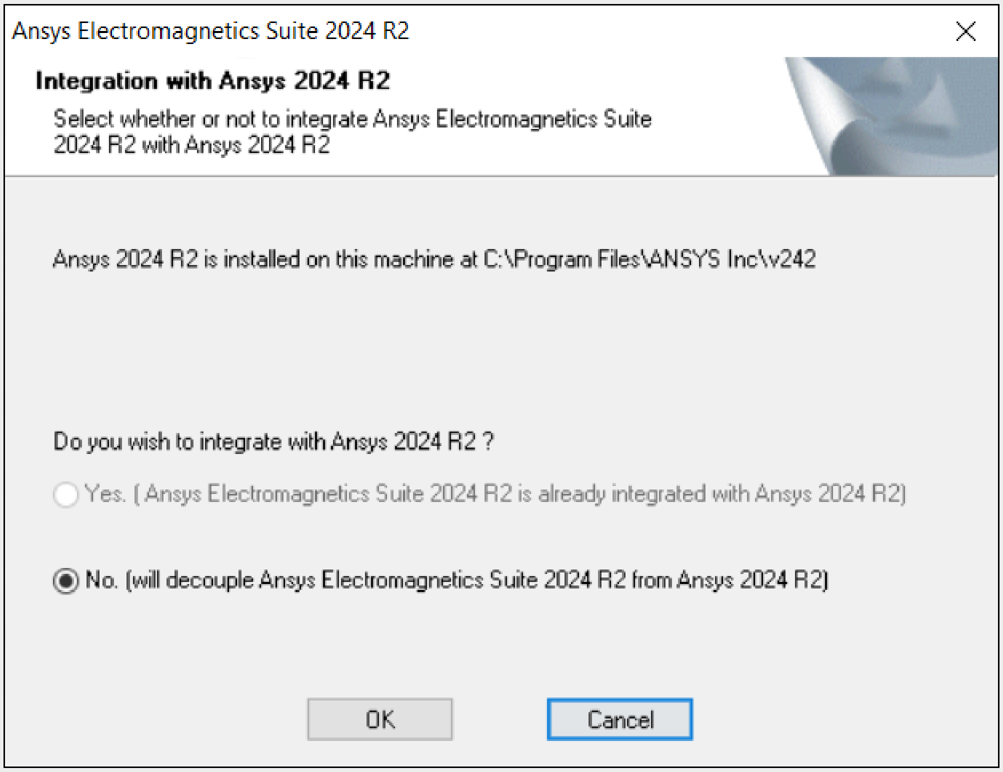
   An alert appears, indicating success.



You can confirm that Ansys 2024 R2 is aware of Ansys Electromagnetics via Ansys Workbench. From Workbench, click **Tools** > **Options** to open the Options window. If Ansys EM is detected, the **Electromagnetics** tab shows the path to the integrated Ansys EM application. From this tab, you can also specify Workbench's default journal recording language; changes applied here apply to any new Ansys Electronics Desktop launch after the change. Any open instance will continue recording in the language that was in effect when it was launched.

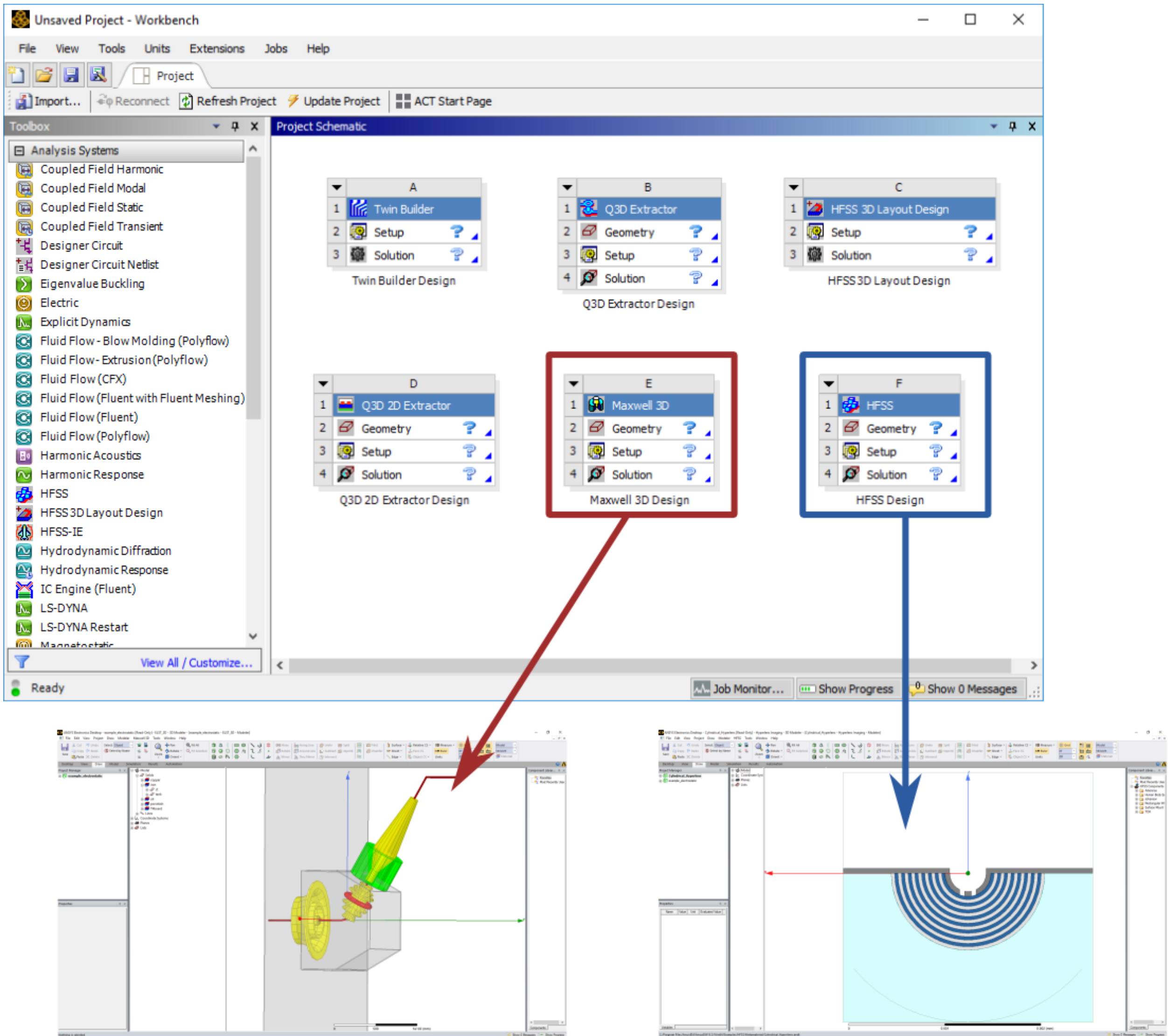
Note:

To decouple Ansys EM Suite from Ansys 2024 R2, follow the same steps and select **No**.



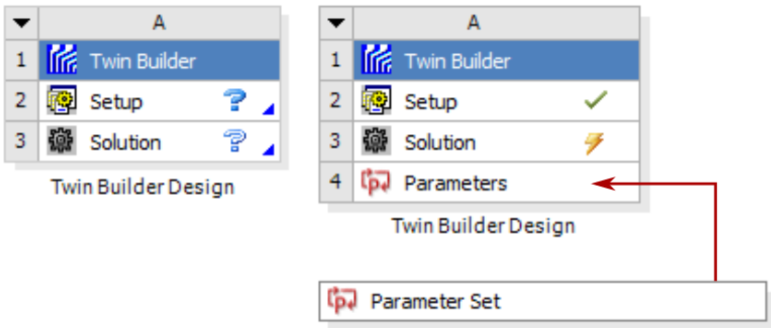
Workbench Data Integration Overview

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*. RMxprt, Circuit, HFSS 3D Layout, and Twin Builder appear on Workbench Project Schematics as systems with two cells: **Setup** and **Solution**. HFSS, Maxwell, Q3D Extractor, Icepak, and Mechanical add an additional **Geometry** cell.

If you invoke Ansys **DesignXplorer** to use variables for refining a design, a **Parameters** cell is added, with a link to the associated Workbench **Parameter Set**. Refer to the Ansys Workbench help for details on working with systems, cells, and parameter sets.



All Ansys Electromagnetics desktop products integrate with Workbench commands, services, and DesignXplorer in a similar manner. Some capabilities include:

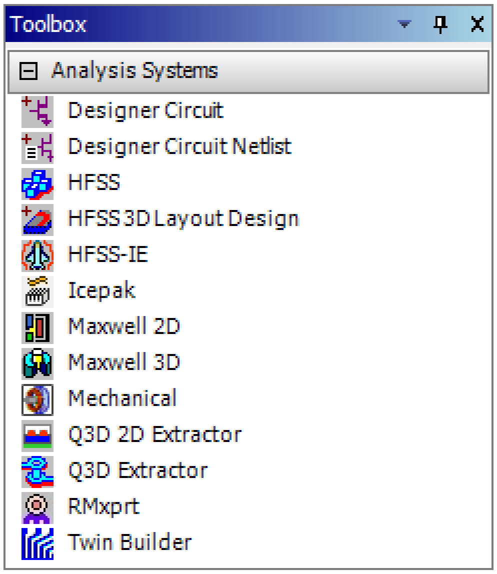
- Adding new analysis systems
- Importing existing desktop projects
- Editing models
- Analyzing models
- Performing parameter studies
- Scripting

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 Electromagnetics projects are also displayed in Workbench.

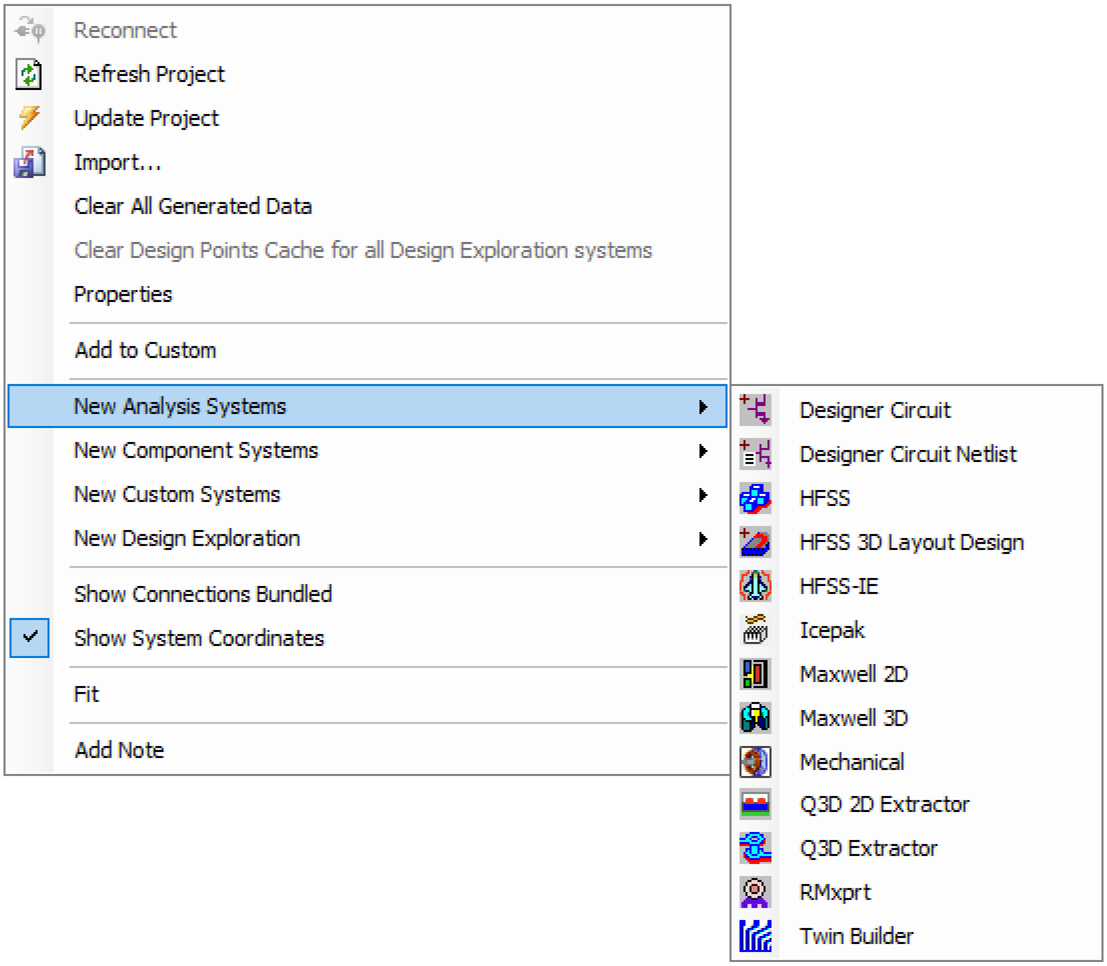
> **Note:**
>
> Detailed information for using these operations can be found in the Ansys Workbench help.

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:


Toolbox
Analysis Systems
Designer Circuit
Designer Circuit Netlist
HFSS
HFSS 3D Layout Design
HFSS-IE
Icepak
Maxwell 2D
Maxwell 3D
Mechanical
Q3D 2D Extractor
Q3D Extractor
RMxprt
Twin Builder

Alternately, you can add a system by right-clicking in the Workbench Project Schematic area and selecting **New Analysis Systems > [System Type]**.



Related Topics

Workbench Data Integration Overview

Importing existing desktop projects

Editing models

Analyzing models

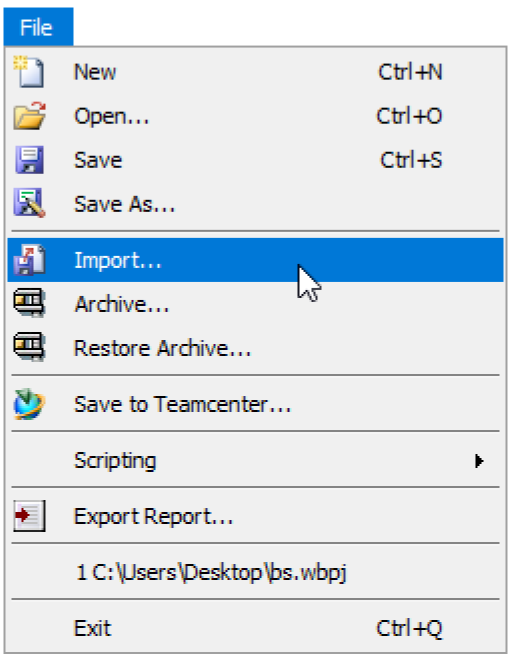
Performing parameter studies

Scripting

Importing Ansys EM Projects into Ansys Workbench

You can import existing Ansys Electromagnetics desktop projects to 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.

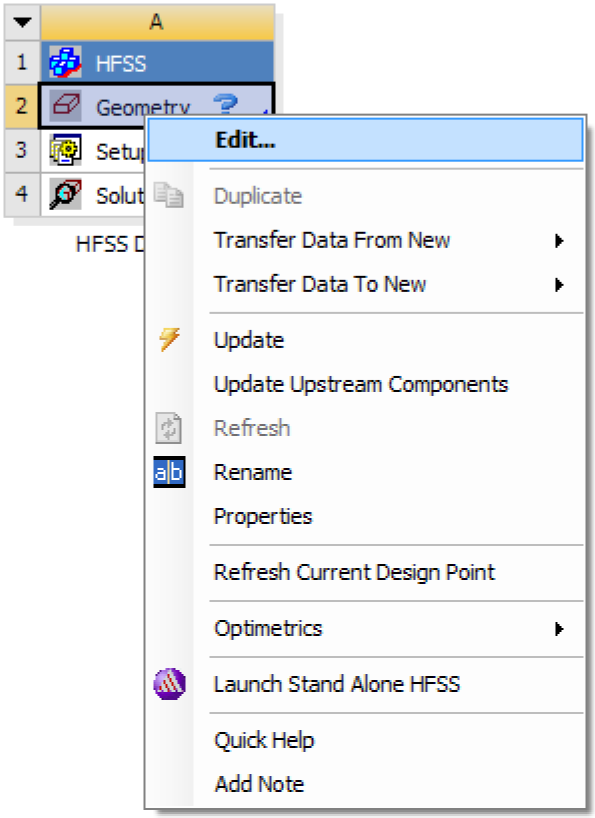
> **Note:**
>
> Object, material, and parameter names with non-ASCII characters are not allowed for data transfer. Such transfers fail and produce an error message.



Editing Ansys EM Models in Workbench

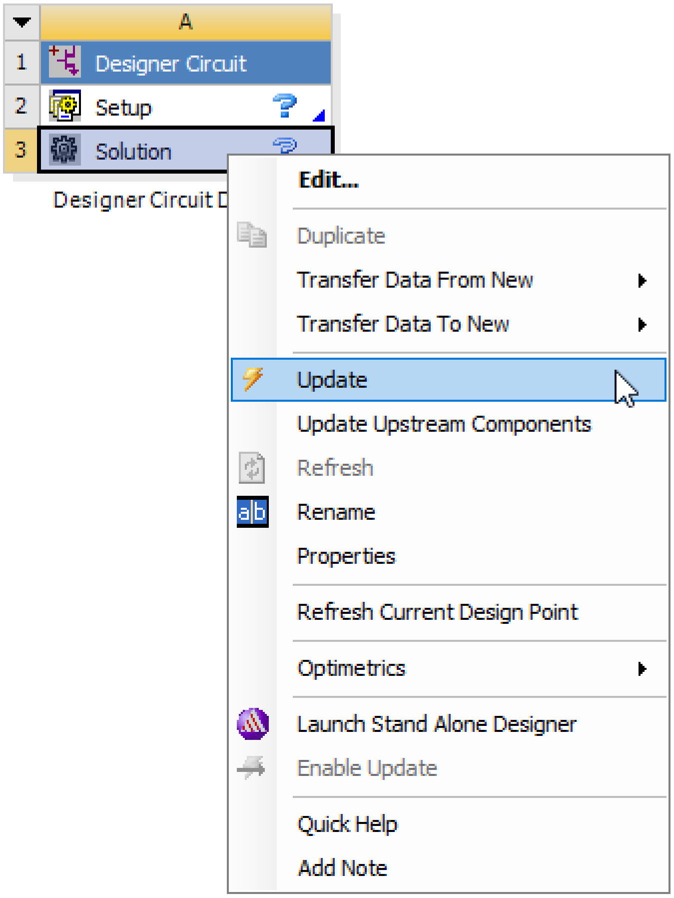
You can edit various properties and parameters of the Ansys Electromagnetics project (geometry, setup, solution, etc.) either by right-clicking the project in Workbench and selecting **Edit**, or by double-clicking the project.

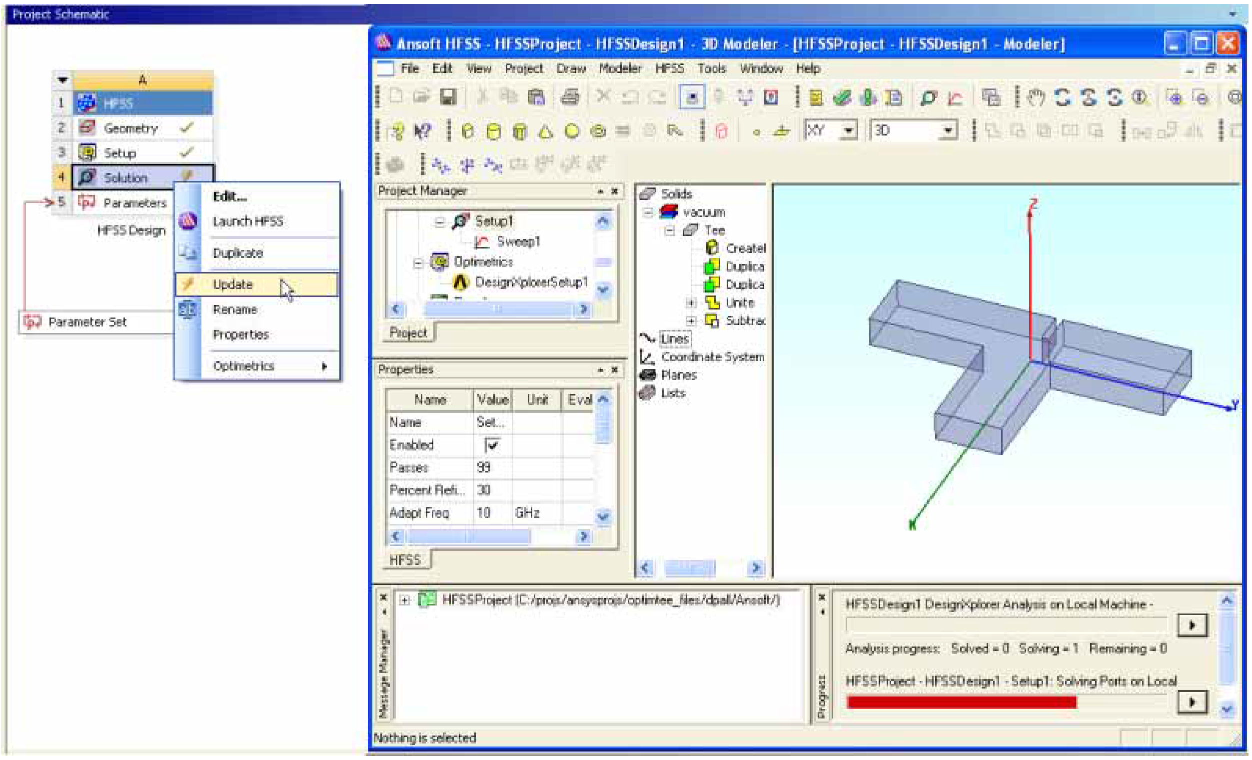
Doing so launches the Ansys Electromagnetics desktop application and loads the project so that you can set up your project in a familiar desktop environment. Changes made to the Ansys Electromagnetics project are saved to the project instance in the Workbench project folder.



Analyzing Ansys EM Models in Workbench

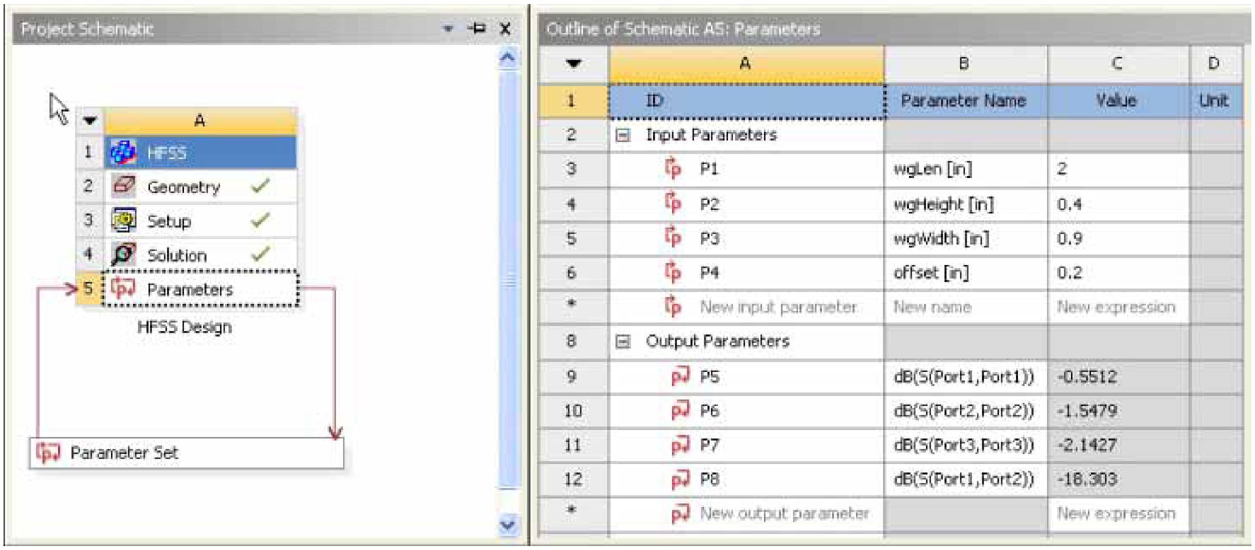
You can use Workbench's **Update** command to run analyses in the integrated Ansys Electromagnetics project. Progress information is also shown in Workbench.


A
1 Designer Circuit
2 Setup
3 Solution
Designer Circuit D
Edit...
Duplicate
Transfer Data From New
Transfer Data To New
Update
Update Upstream Components
Refresh
Rename
Properties
Refresh Current Design Point
Optimetrics
Launch Stand Alone Designer
Enable Update
Quick Help
Add Note

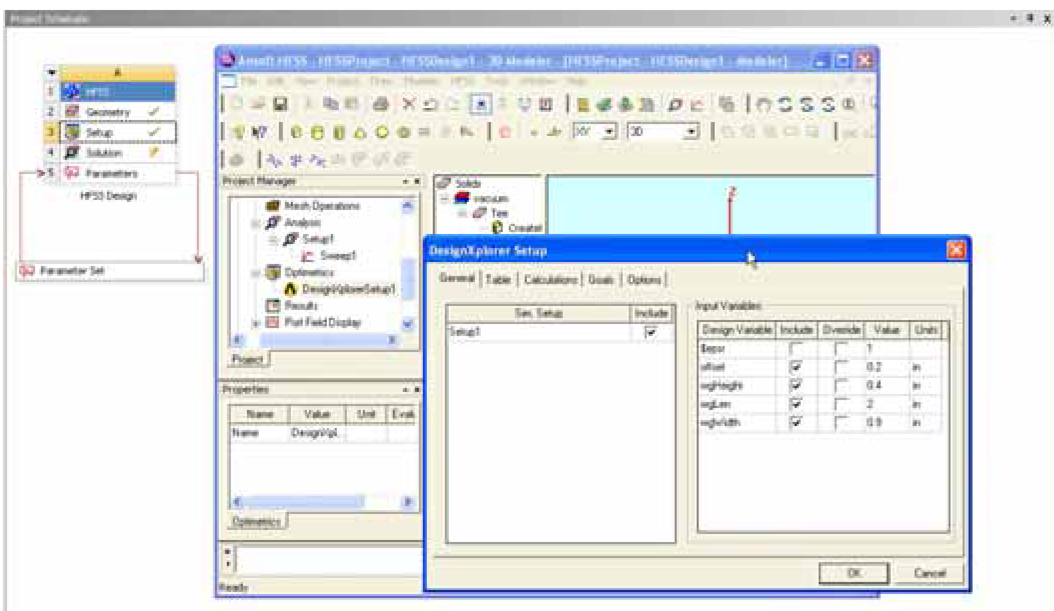


Performing Parameter Studies in Workbench

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.



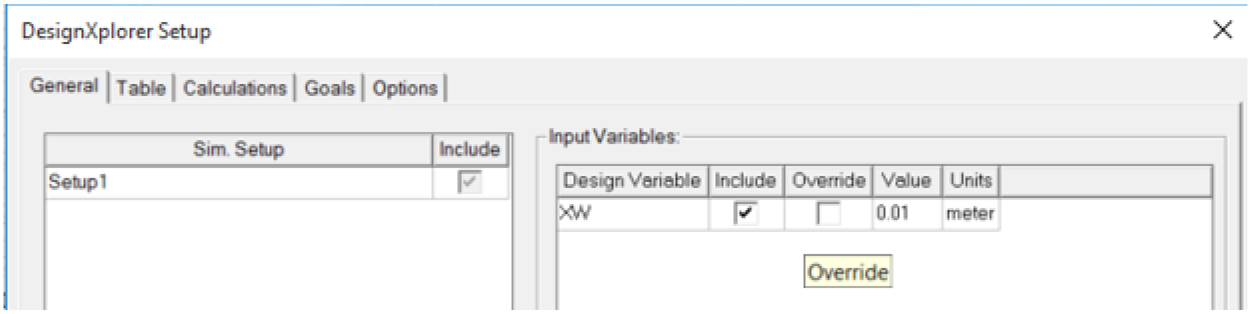
Parameters from the Ansys Electromagnetics project are exposed to Workbench through the **DesignXplorer** setup. The Ansys Electromagnetics system's cell status on the Workbench project is updated as changes are made in the Ansys Electromagnetics application desktop.



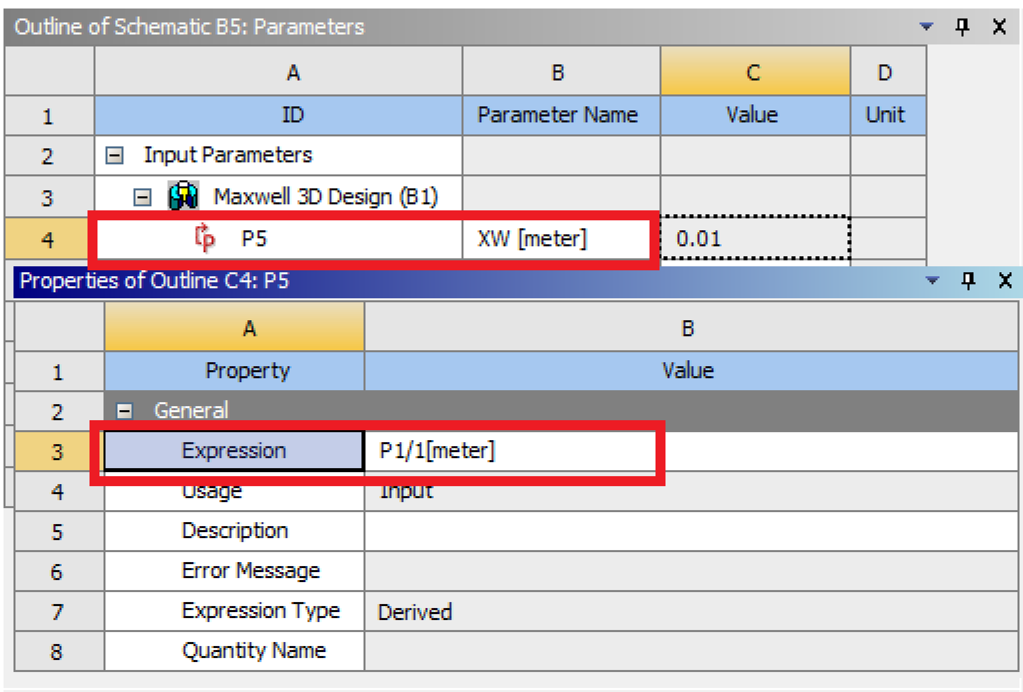
The workflow for using Electronics Desktop systems with the Workbench to take advantage of Distributed Analysis is as follows.

1. In the Ansys Electronics Desktop, specify a variable an as input parameter. The following figure shows a DesignXplorer Setup that includes a geometry variable called XW This

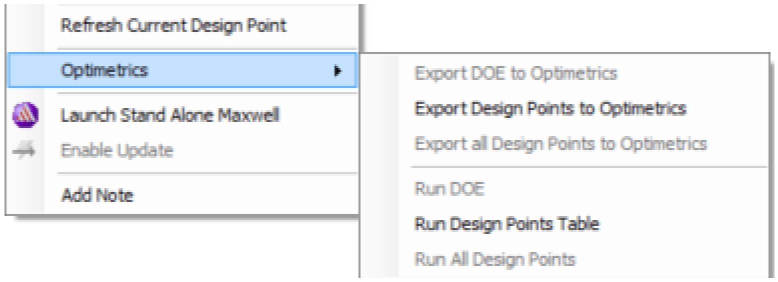
variable is mapped to a variable in the corresponding Workbench Design, in this example, P5.



2. In the Workbench Design, change the P5 expression to a new value (in this example figure, the P5 parameter is assigned and expression of P1/1[meter] to tie it to the geometry system's variable.



In the Workbench, the **Export Design Points to Optimetrics** and **Run Design Points Table** commands are enabled.

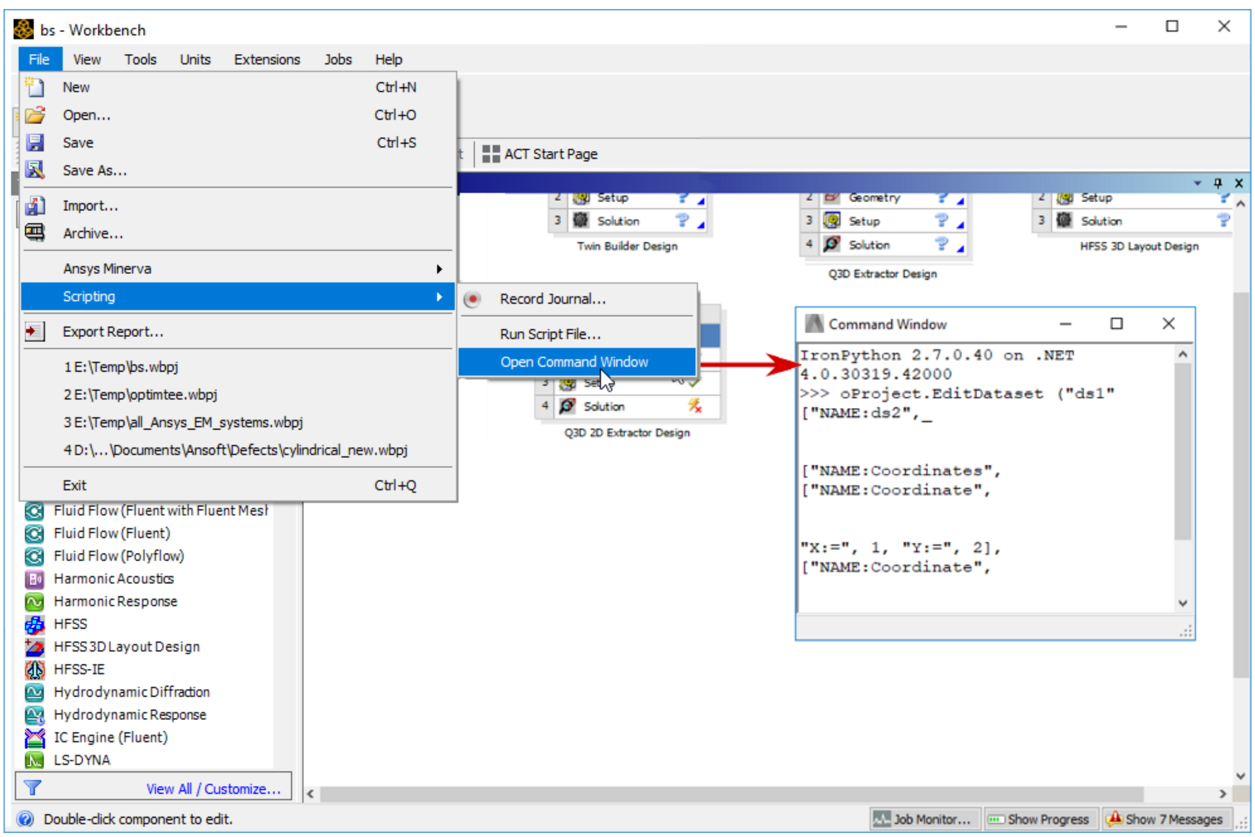


3. In the Electronics Desktop, you can click on the default DesignXplorer Setup, and choose **Generate Variation Data**.

   If you have configured your Analysis Configuration for multiple machines, the Project is ready for Distributed Analysis. After analysis, the Workbench Parameter table should be updated correctly for all design points.

Scripting in Workbench

Scripts that include Ansys Electromagnetics projects can be recorded and run in Workbench.



When a WB journal is run from the command line, Electronics Desktop provides a beta feature that allows the Ansys EDT portion to also run in batch mode (that is, when using RunWB2 -B).

To use this beta feature, set the `NON_GRAPHICAL_COMMAND_EXECUTION` environment variable before running the script.

From a Windows Command prompt, for example:

```
Set ANSYSEM_FEATURE_SF6694_NON_GRAPHICAL_COMMAND_EXECUTION_
ENABLE=1
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

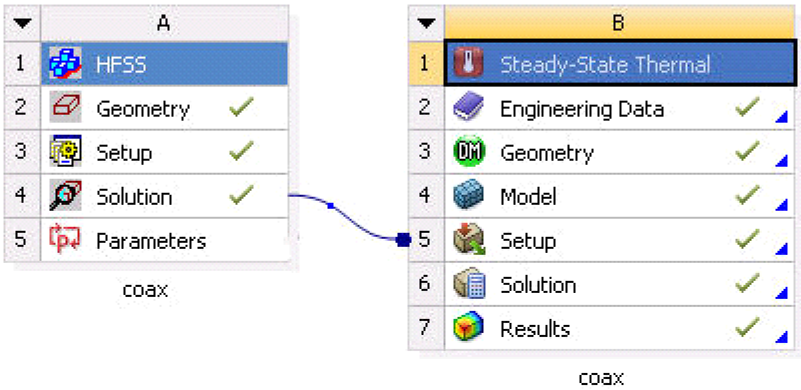
Multiphysics Coupling on Workbench with Ansys Structural

Multiphysics Coupling between Ansys EM Field Systems on Workbench

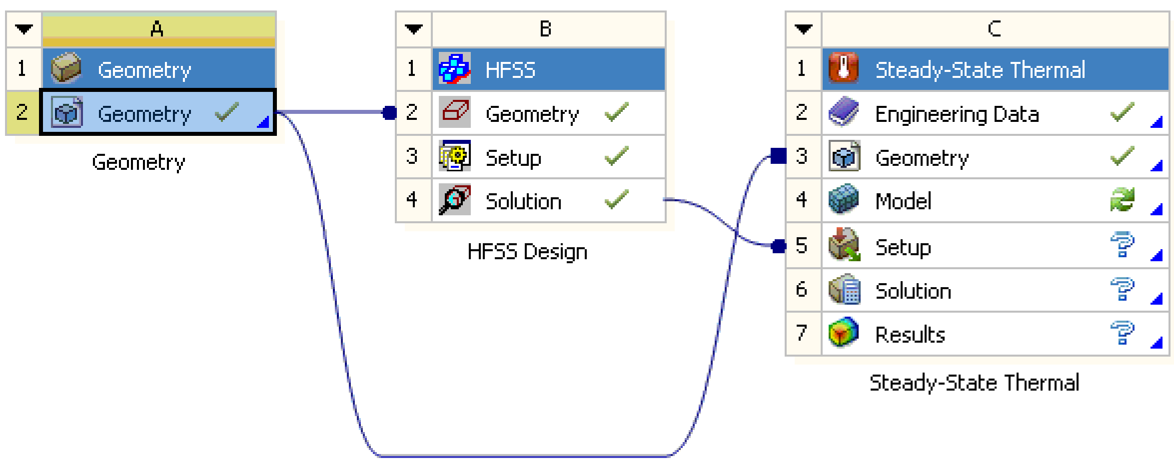
Multiphysics Coupling on Workbench with Ansys Thermal

Using data integration, HFSS, Maxwell, and Q3D Extractor provide heat losses (heat generation and heat flux) to Ansys Thermal. 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.

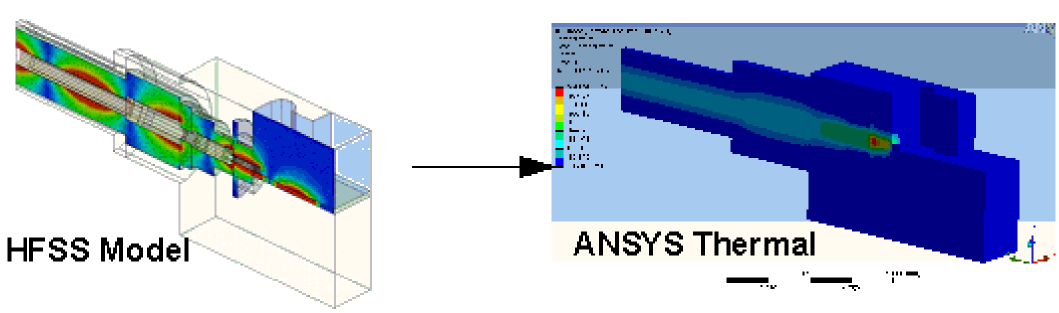
Note how the HFSS design is linked visually to Ansys Thermal on the Workbench project schematic.



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.



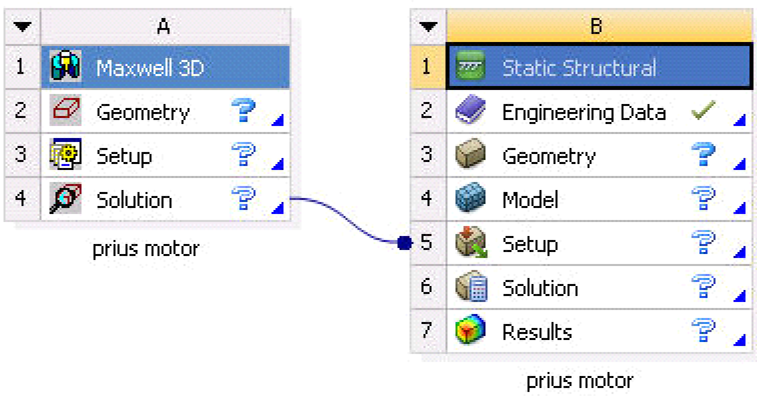
In this example, HFSS coax model Solution provides heat loss data as a thermal load to the Ansys Thermal Setup. The resulting analysis shows a thermal "hotspot", providing the user with the information needed to adjust the design's material to fix the problem.



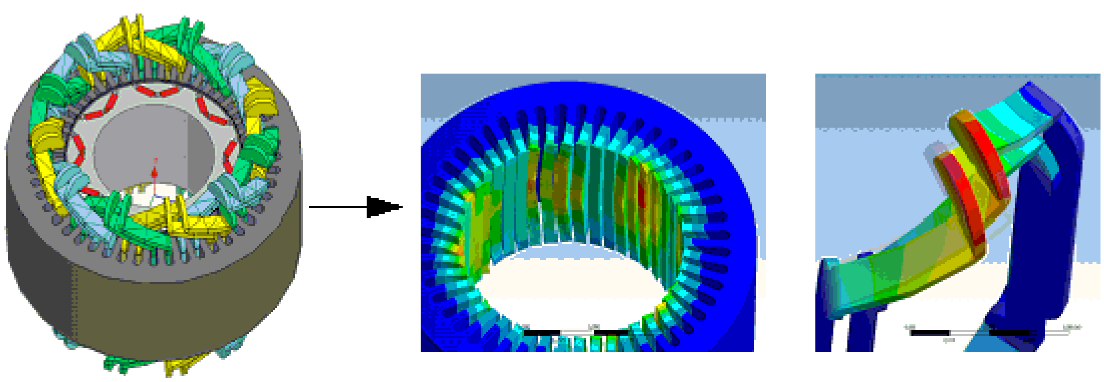
You enable this feature by checking Enable Feedback in the dialog for Setting the Temperature of Objects.

Multiphysics Coupling on Workbench with Ansys Structural

Using data integration, Maxwell 2D and Maxwell 3Dcan provide forces to Ansys Structural.



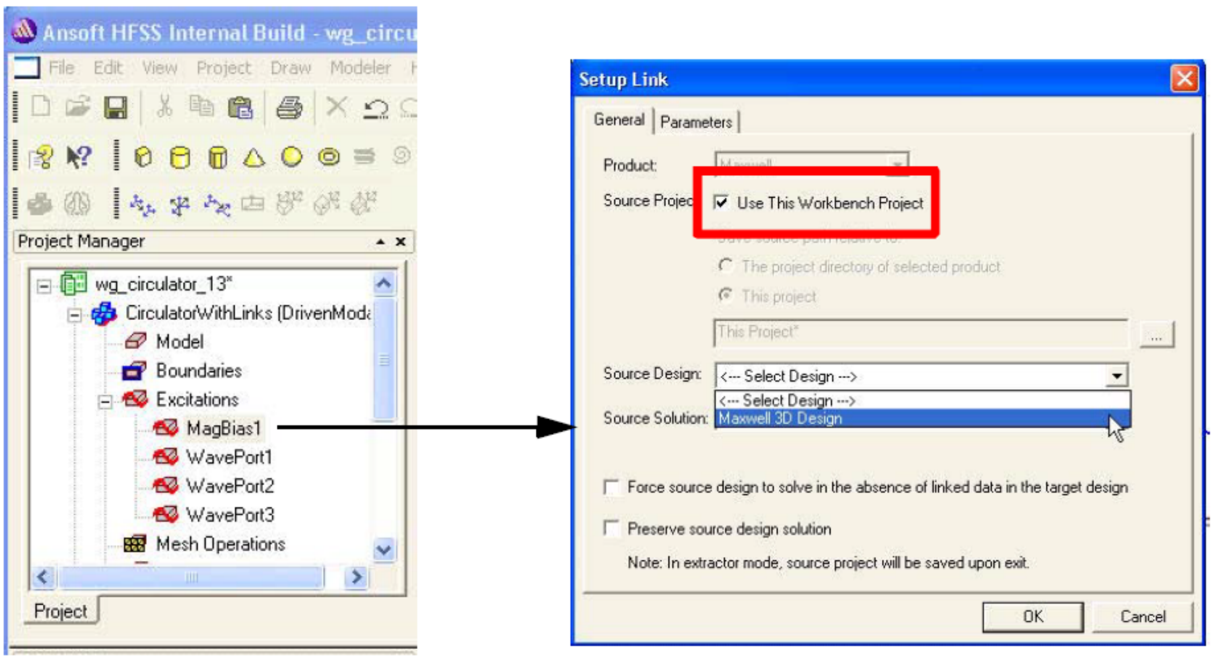
In this example, the Maxwell 3D electromagnetic force density **Solution** is used as the load in Ansys Structural to determine how these forces deform the motor's stator and coils.



Maxwell 3D Model Deformation of the Stator Coil Deformation

Multiphysics Coupling between Ansys EM Field Systems on Workbench

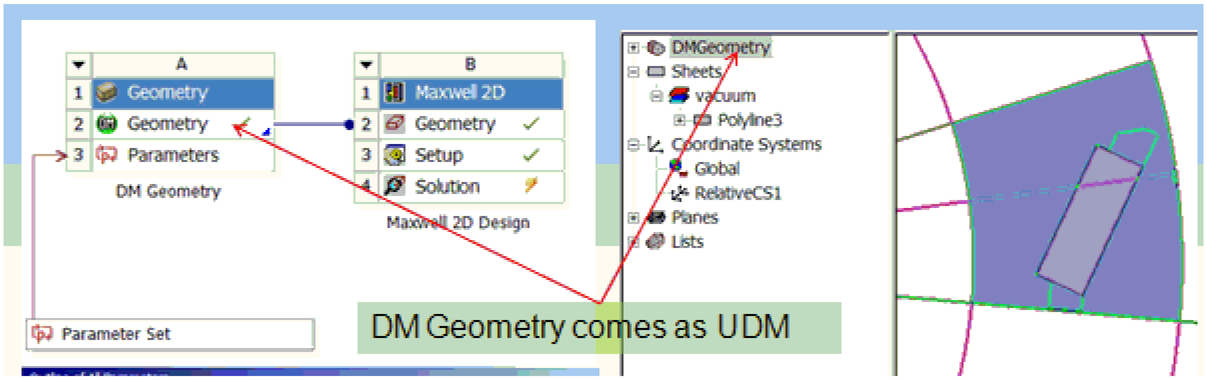
You can set up links between Ansys Electromagnetics field systems that reside on a Workbench project schematic. Linking is set up in the Ansys Electromagnetics application as shown in the following example.



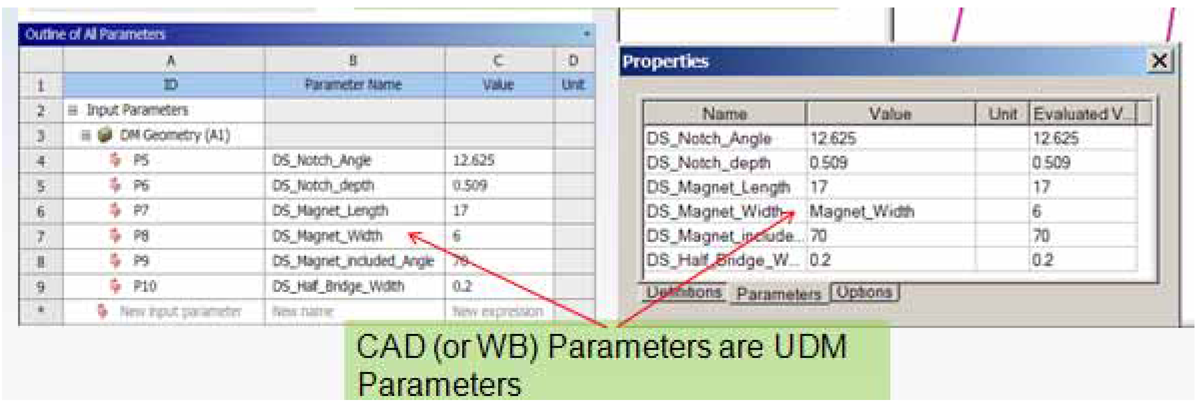
Ansys EM CAD Integration through Workbench

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.

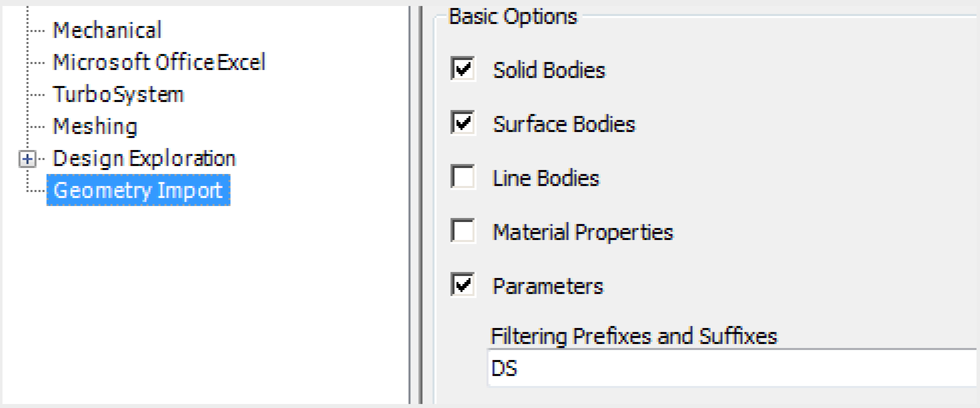


The UDM format makes it possible to exchange parameters. Ansys online help contains further description of the UDM feature and function.



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)
- Ansys SpaceClaim Direct Modeler (SCDM)

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](#)

[Bi-Directional CAD Integration](#)

[CAD Integration Model Edits](#)

[Multiple Geometry Links for CAD Integration](#)

[Creating Dynamic Links to EDT Designs](#)

[Adding a Design Variable for Dynamic Links](#)

[CAD Integration Functionality](#)

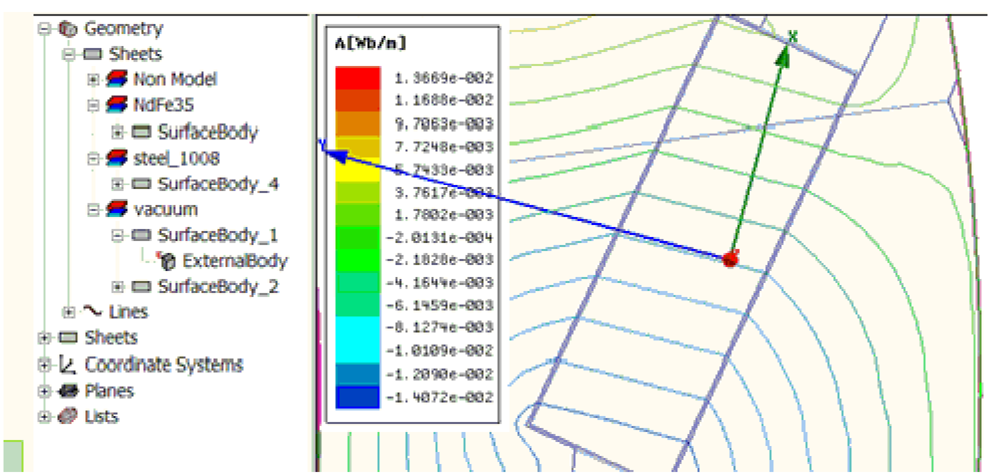
[Healing with CAD Integration](#)

[Important Geometry Options for CAD Integration](#)

CAD Integration Functionality

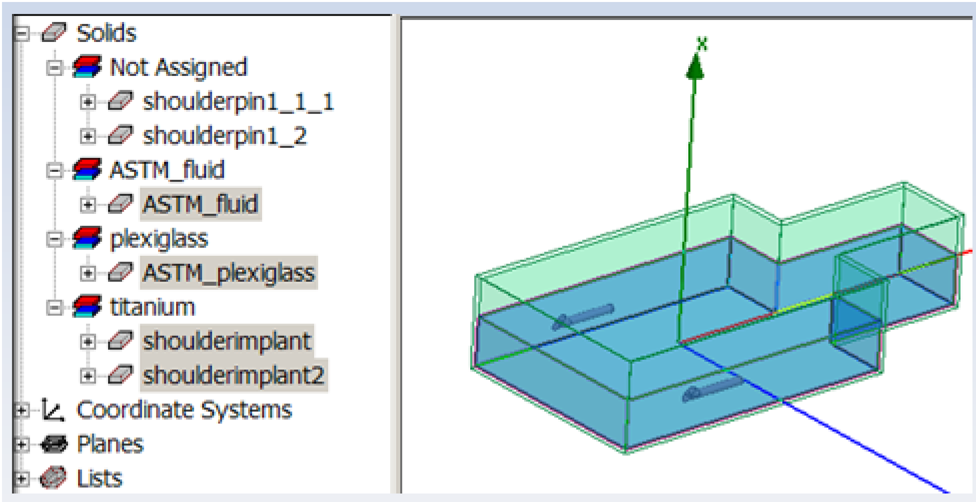
CAD Integration includes the following functionality:

- DX analysis
- WB Update Project
- WB Update All Design Points
  - Parametric analysis with DSO
  - Animation
  - Geometry
  - Field plots

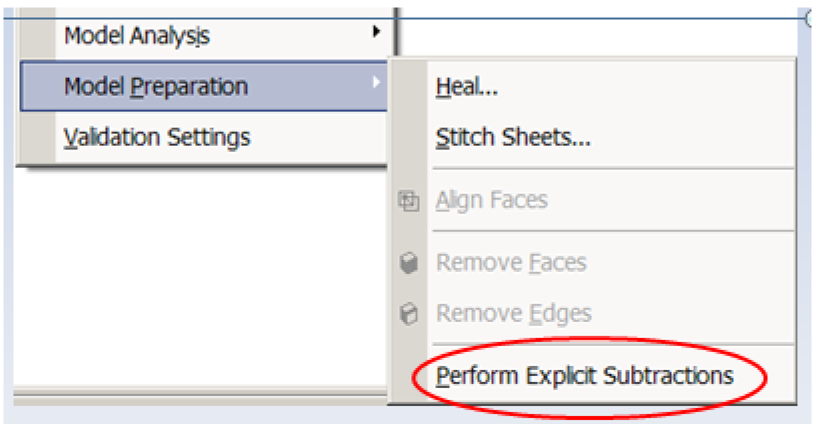


Ansys Mechanical does not do implicit subtraction. Ansys Electromagnetics products can do explicit subtractions before sending geometry to Ansys using **Modeler > Model Preparation > Perform Explicit Subtraction**.

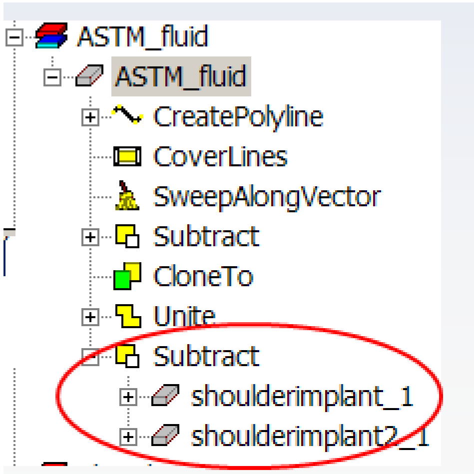
For example, consider the following model.



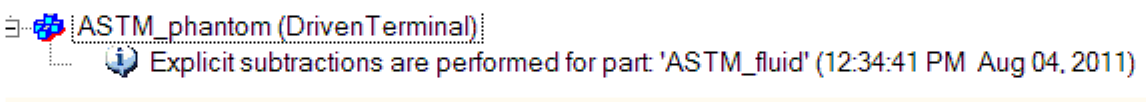
Perform Explicit Subtractions can be performed.



Results appear in the History tree, as shown:



The **Messages** window reports this action.

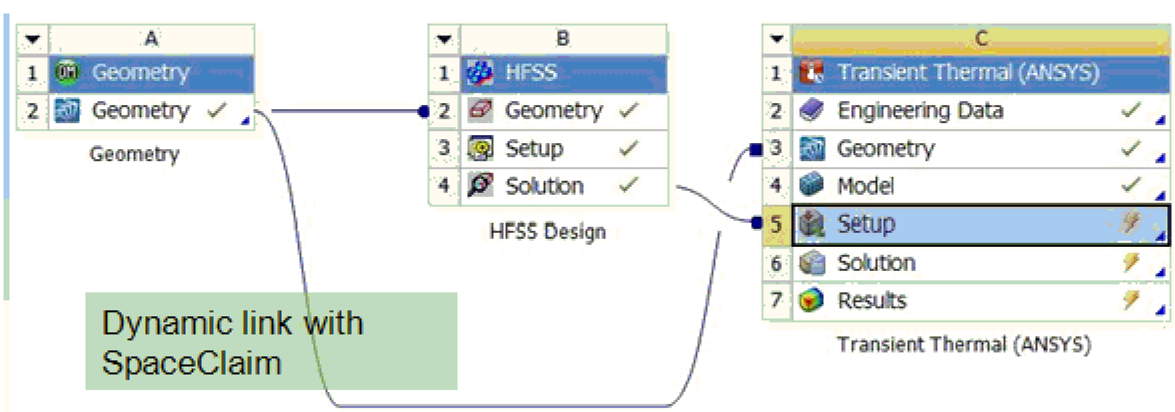


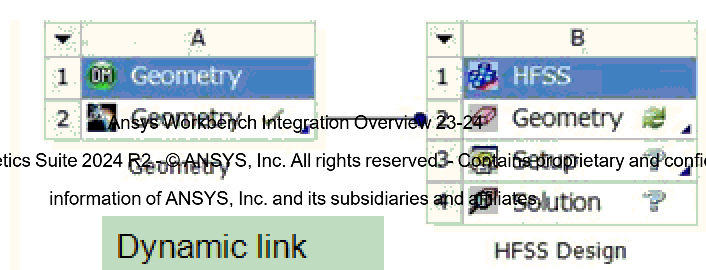
CAD Integration and Geometry Sharing

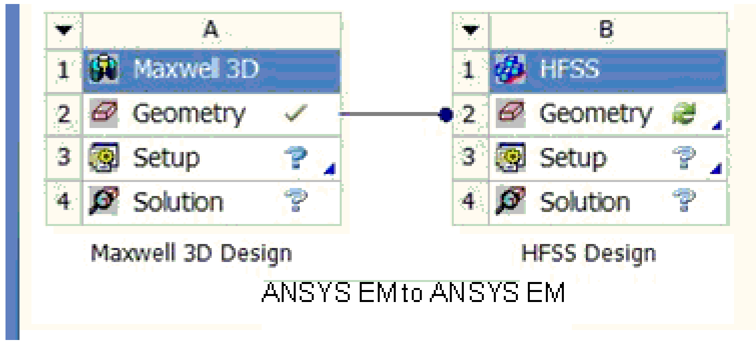
A CAD model comes into Ansys Electromagnetics as a User Defined Model (UDM).

The inputs to Ansys Electromagnetics from CAD are:

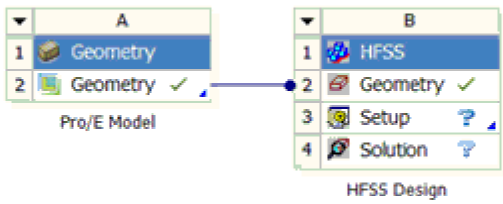
- Geometry/Topology with persistent IDs
- CAD parameters
- Material assignment
- Attributes, such as name and color



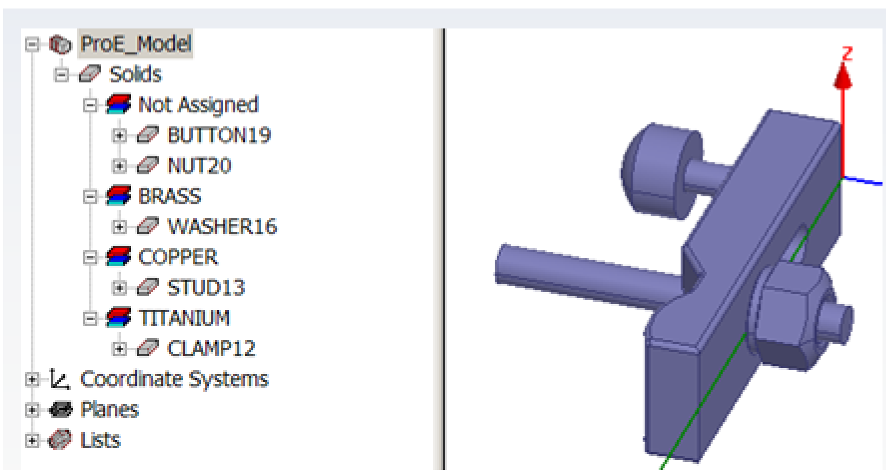




For example, in Workbench, a Pro/E Model can be linked to Ansys.



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_Wdith	0.2	
*	New input parameter	New name	New expression	

Through the Ansys Electromagnetics CAD integration, the linked UDM includes the same parameters.

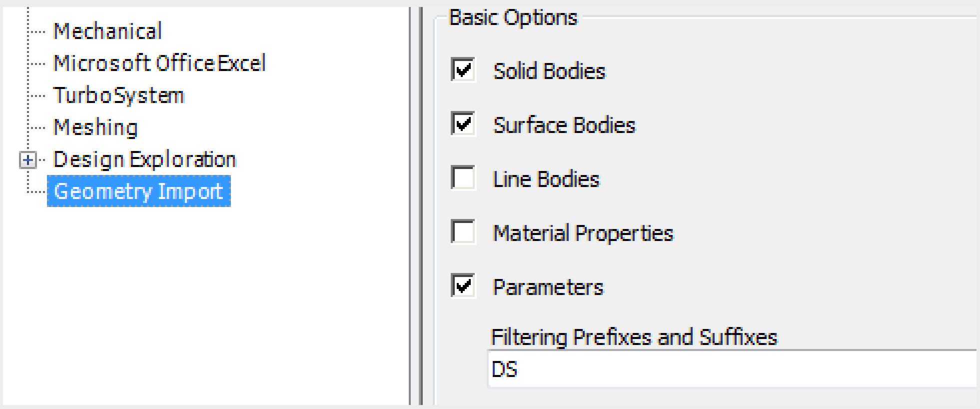
Properties

Name	Value	Unit	Evaluated V...
DS_Notch_Angle	12.625		12.625
DS_Notch_depth	0.509		0.509
DS_Magnet_Length	17		17
DS_Magnet_Width	Magnet_Width		6
DS_Magnet_include...	70		70
DS_Half_Bridge_W...	0.2		0.2

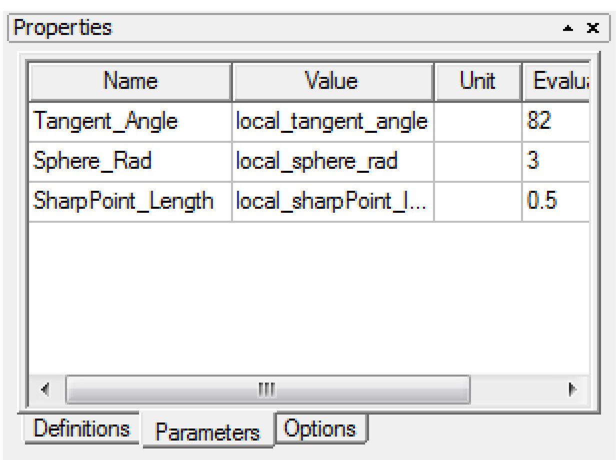
Definitions | Parameters | Options

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.

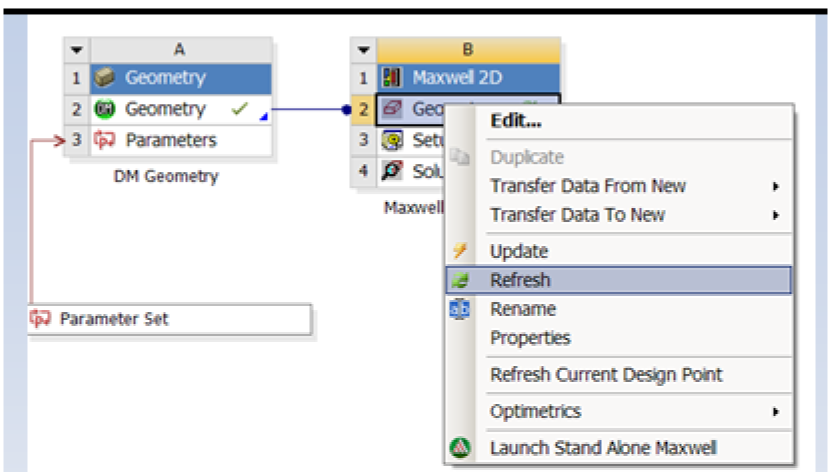
Table of Schematic D2: Design of Experiments (Custom)

	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.

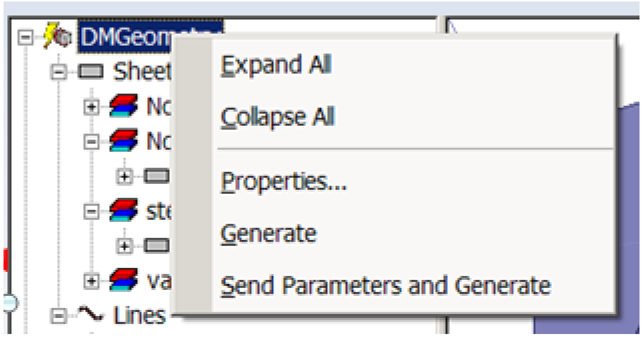
You make an edit in a CAD application and run **Refresh** on an Ansys Electromagnetics Geometry Cell.



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.

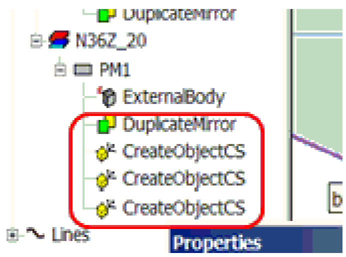
If you edit UDM (CAD) parameters in the Ansys Electromagnetics modeler window you can run the **Send Parameters and Generate** command.



The command passes the edited parameters to the linked CAD application and then pulls corresponding CAD geometry.

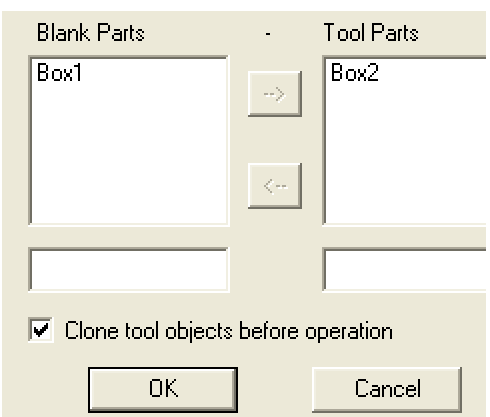
CAD Integration Model Edits

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**.



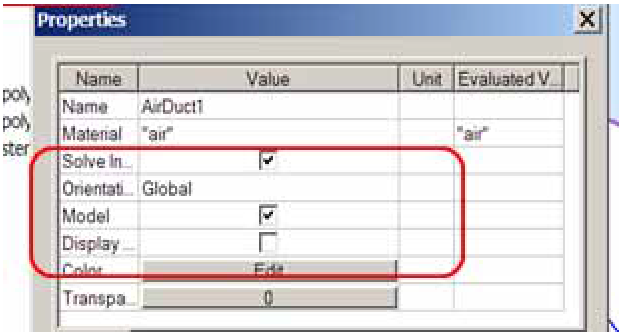
The following operations are not allowed:

- 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

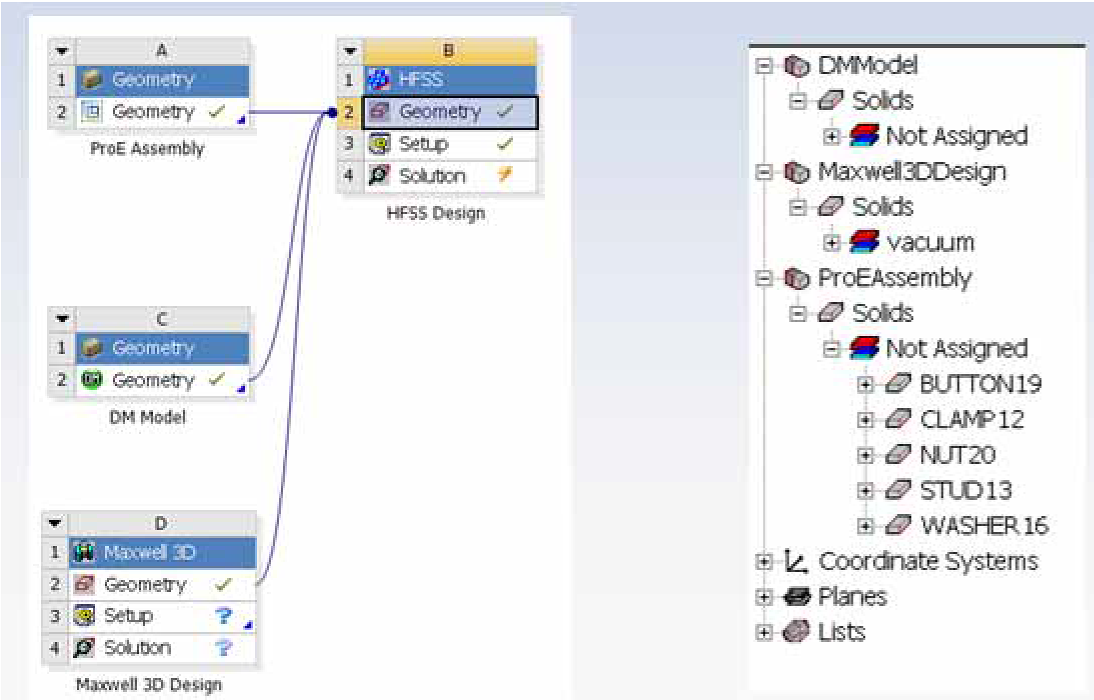


It is not possible to delete individual parts of UDM.

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.

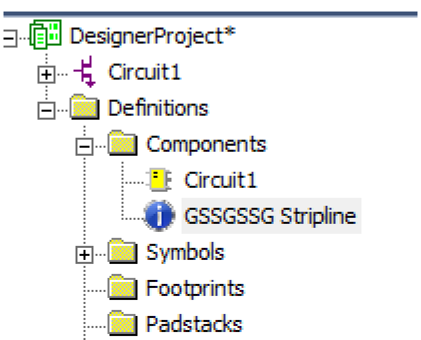
For example, the following figure shows a DM Model, a Maxwell model, and ProE model linked to HFSS in Workbench, and displayed in the HFSS History tree as three UDMs.



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.

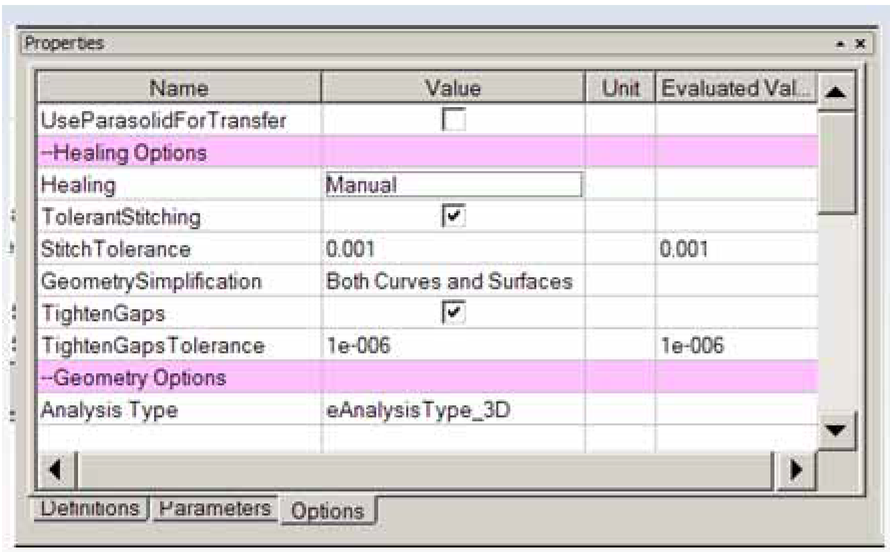
Update the Circuit Design to open the Electronics Desktop. In the menu bar, select **Workbench > Update Dynamic Link Components**. This command will create components in the project tree for each design in the Workbench project.



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.

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.

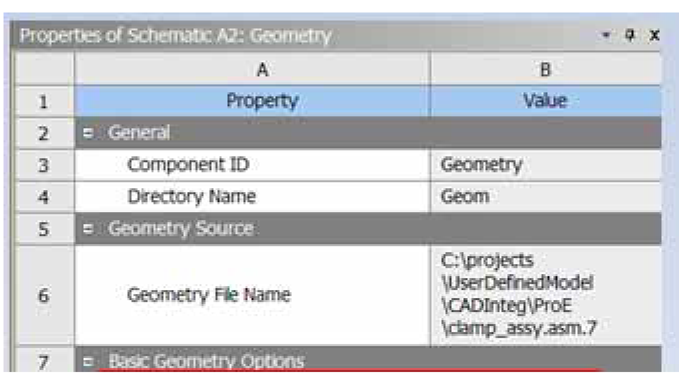


Name	Value	Unit	Evaluated Val...
UseParasolidForTransfer	☐		
--Healing Options			
Healing	Manual		
TolerantStitching	☑		
StitchTolerance	0.001		0.001
GeometrySimplification	Both Curves and Surfaces		
TightenGaps	☑		
TightenGapsTolerance	1e-006		1e-006
--Geometry Options			
Analysis Type	eAnalysisType_3D		

Healing options are None, Auto and Manual. By default, healing is off (None) and should be turned on only if required.

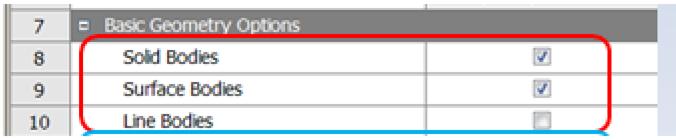
Important Geometry Options for CAD Integration

Select a Geometry Cell in Workbench to see options in the Properties window.

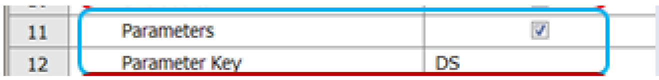


	A	B
1	Property	Value
2	General	
3	Component ID	Geometry
4	Directory Name	Geom
5	Geometry Source	
6	Geometry File Name	C:\projects \UserDefinedModel \CADInteg\ProE \clamp_assy.asm.7
7	Basic Geometry Options	

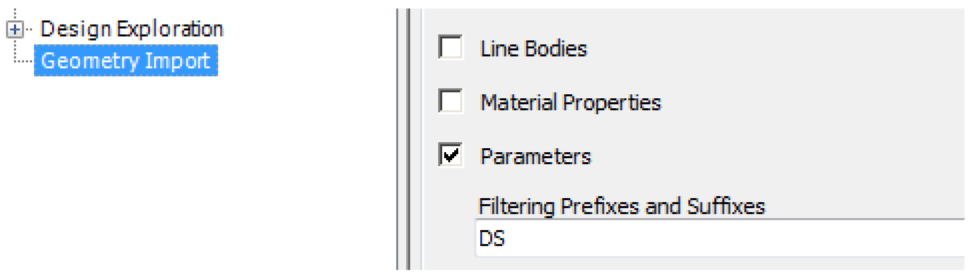
Control dimension of bodies coming from CAD



Make sure **Parameters** is checked and that the parameter key (filter) is appropriate for CAD parameters.



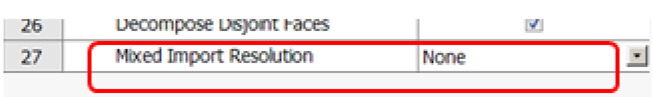
The default parameter key filtering means that only parameters whose names start with DS will come through. This is the default for the Workbench **Tools>Options** for **Geometry Import**. To import parameters with different prefixes or names, assign an appropriate prefix or clear the **Filtering Prefixes 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.

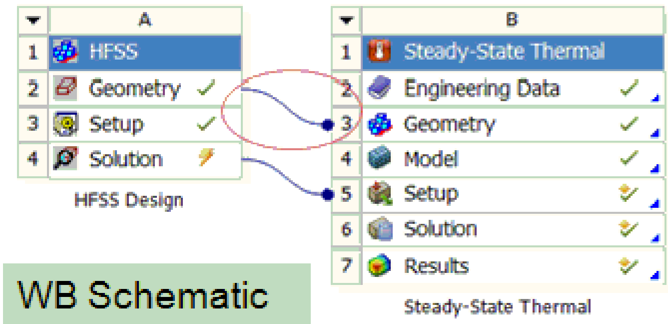
The **Mixed Import Resolution** option is used to resolve parts with mixed dimension (typically from Pro/E)



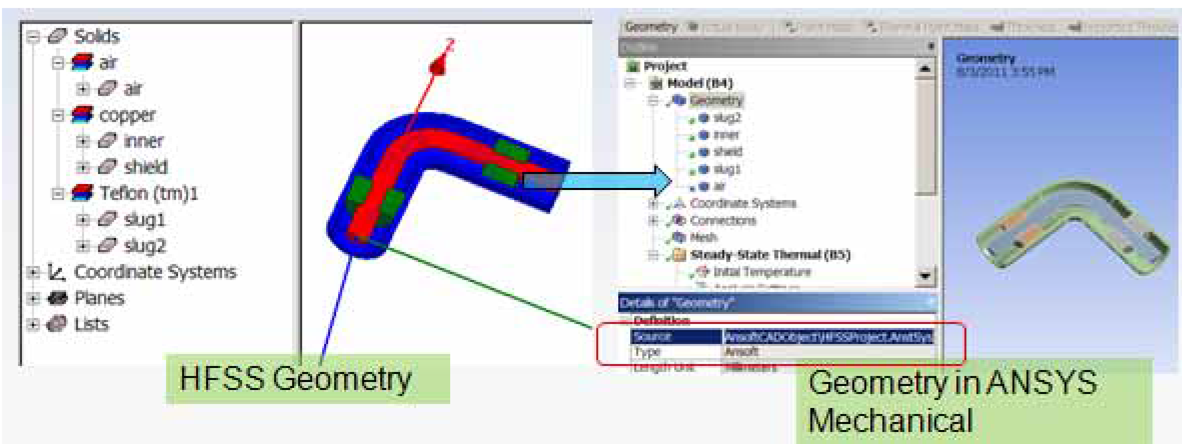
Refer to Ansys Help for details.

Ansys EM to Ansys Geometry Transfer

Ansys Electromagnetics CAD Integration transfers model information based on Workbench links.

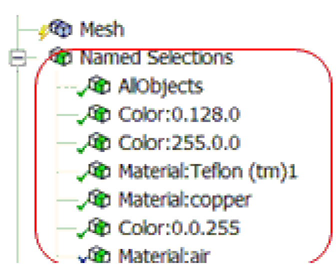


The following figure shows how the information is transferred between simulators.

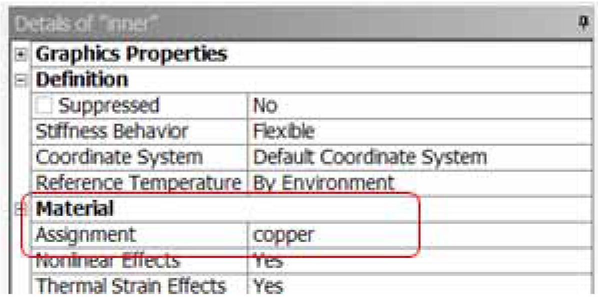


Information transferred includes:

- 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.

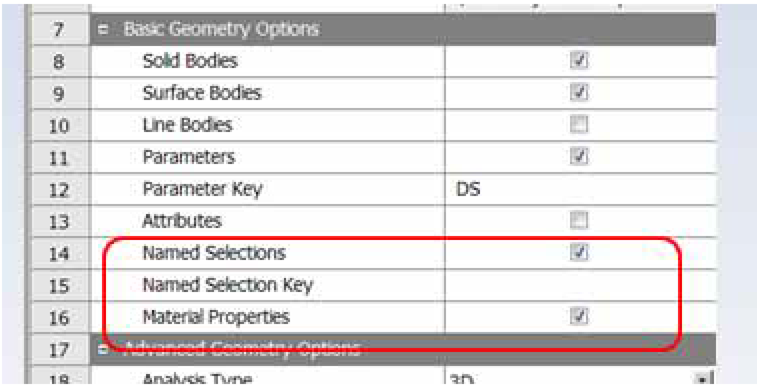
Material Assignment Transfer

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**.

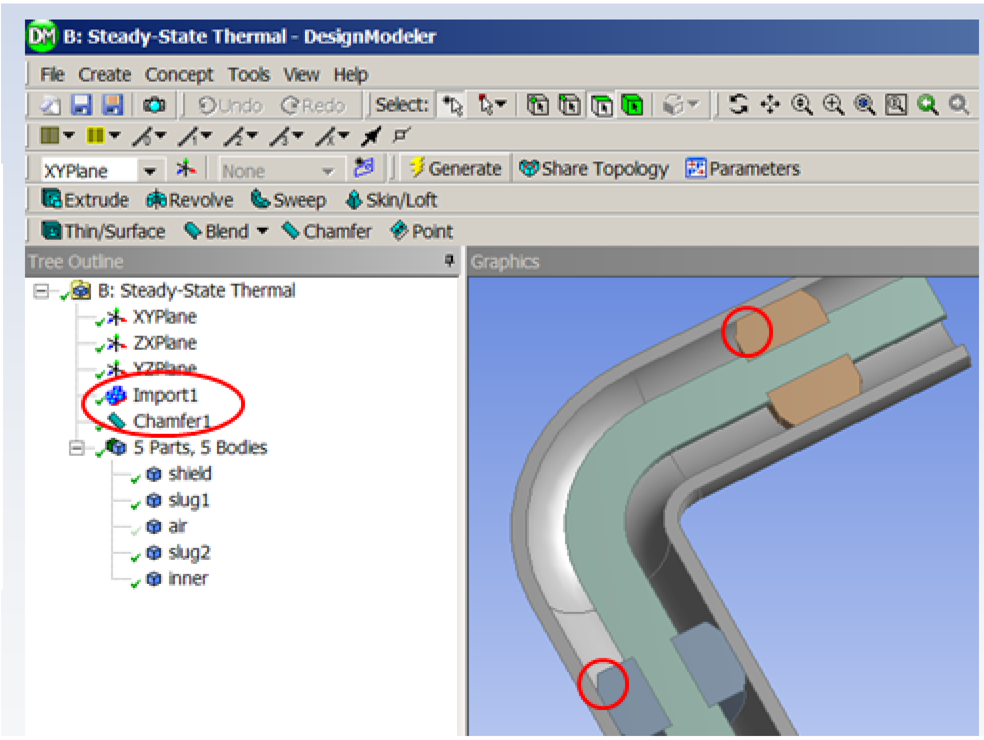
The Named Selection Key should either be empty or contain "Material."



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.

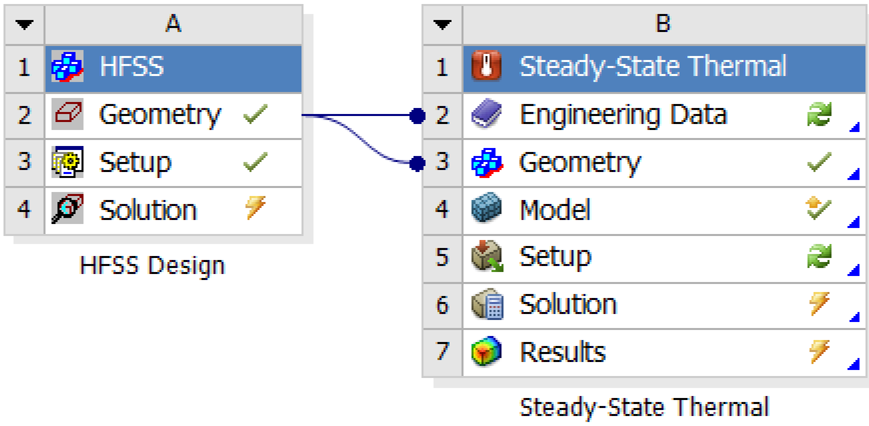
Consider an HFSS model linked to DM through the Workbench. In the following example, a chamfer operation on geometries is being imported.



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.

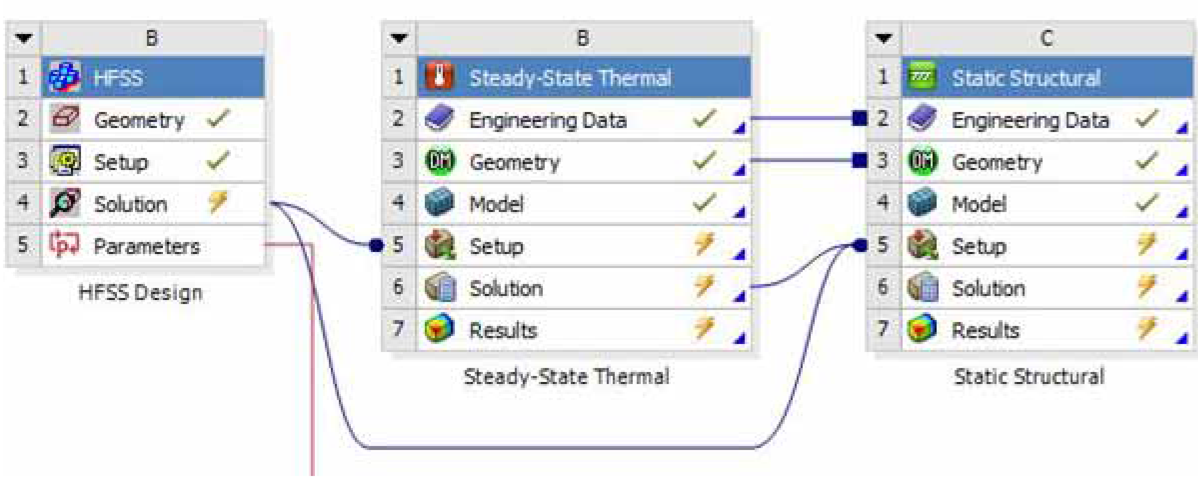
When the downstream model cell is refreshed or updated, material properties from the ED in the same system are used in the model's materials, based on material assignments set during the geometry cell refresh. Note that if material assignment fails, the Workbench default material (Structural Steel) is 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.

Stress Feedback to HFSS using Workbench

Including stress feedback to HFSS involves coupling an HFSS system with both thermal and stress systems. The thermal system serves as an upstream system to the stress system and simulates the effect of heat induced displacement. And through the Workbench link between the Solution cell of HFSS and the setup cell of Mechanical, the displacement can feedback into HFSS. The HFSS project can be Driven or Eigenmode.



You use a dialog to select one or more objects (geometry) in HFSS upon which displacement data from Mechanical applies. This feature lets you avoid applying displacement to the whole model where only certain parts are sensitive to stress. You must select the same objects in the thermal and stress system to apply the respective imported loads.

The results of the process include deformation feedback applied to selected objects. The Solution Profile includes a report of Solve with thermal/displacement feedback, including information Maximum Delta T and Maximum Delta Displacement. You can create field plots with based on Displacement and Mag_Displacement, and you can scale the deformation displayed. The Fields Calculator also includes these input quantities.

Interface Changes for Stress Feedback from Mechanical to HFSS

Process Flow for Stress Feedback to HFSS

Interface Changes for Stress Feedback from Ansys Mechanical to HFSS

When used through the Workbench for stress feedback projects, the HFSS interface contains some commands and information that do not appear in standalone HFSS.

Deformation of Objects dialog

Solution Profile information for Solve with Displacement Feedback

Plot Fields Menu with Displacement and Mag_Displacement

Deformation Scale tab for Modify Plot Attributes

Revert to Zero Displacement command under the **Solve Setup** menus

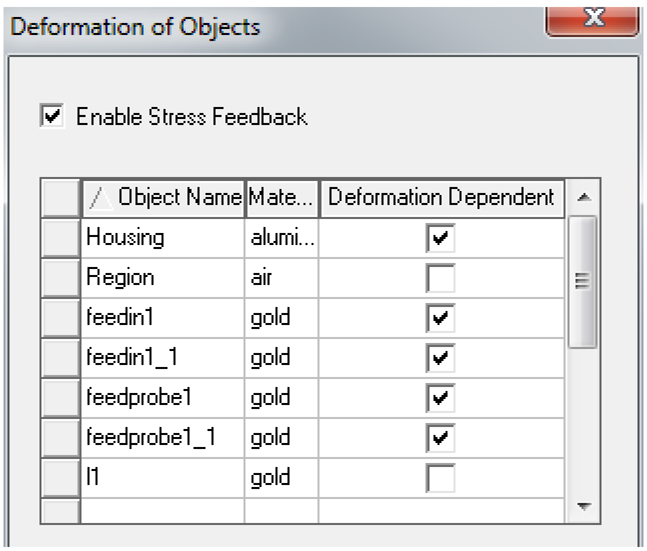
Fields Calculator Field Quantities for Stress Feedback Projects

Plot Fields Context Menu for Stress Feedback Projects

Modify Plot Attributes Dialog for Stress Feedback Projects

Deformation of Objects Dialog

When working with HFSS through Ansys Workbench, the HFSS menu for Driven and Eigenmode projects includes a Deformation of Objects command that opens this dialog for specifying which objects shall be deformation dependent.



The Enable Stress Feedback check box enables object selection. All selections are disabled when this is unchecked. Changing the setting invalidates any existing solutions.

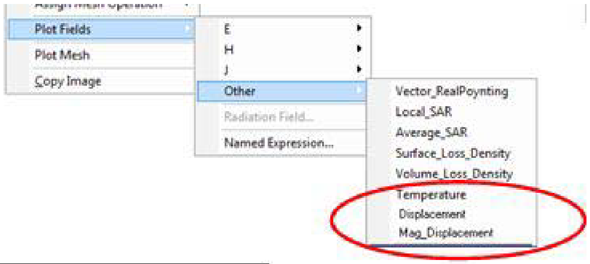
The objects listed do not include objects without materials assigned, non-model objects, or non-3D objects. Objects assigned materials such as air or vacuum are not deformed and don't apply to stress feedback.

Solution Profile with Displacement Feedback

The Solution Profile includes a report of Solve with thermal/displacement feedback, including information Maximum Delta T and Maximum Delta Displacement.

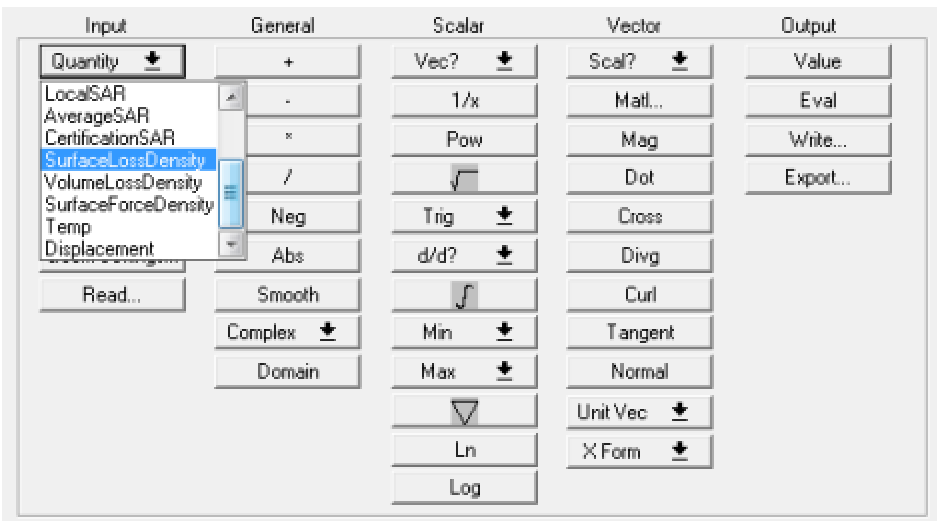
Plot Fields Menu with Displacement and Mag_Displacement

The Plot Fields menu includes entries for displacement and mag_displacement.



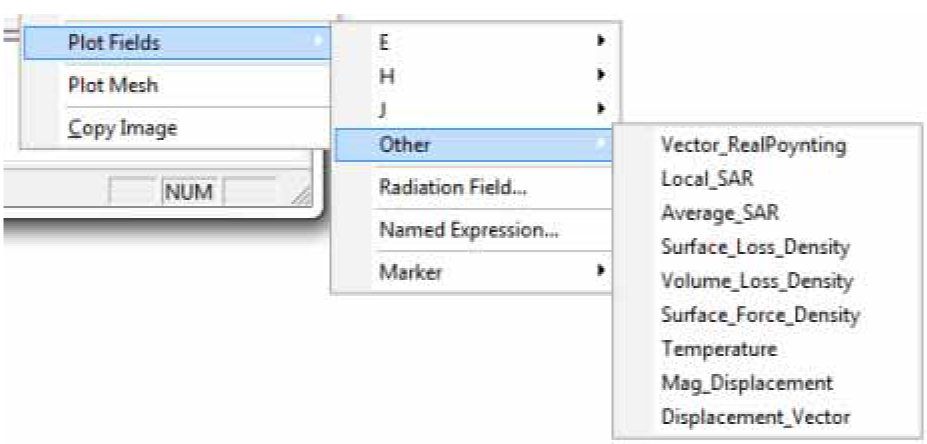
Fields Calculator Field Quantities for Stress Feedback Projects

The Quantity menu for the Fields Calculator lists Displacement for Stress Feedback projects.



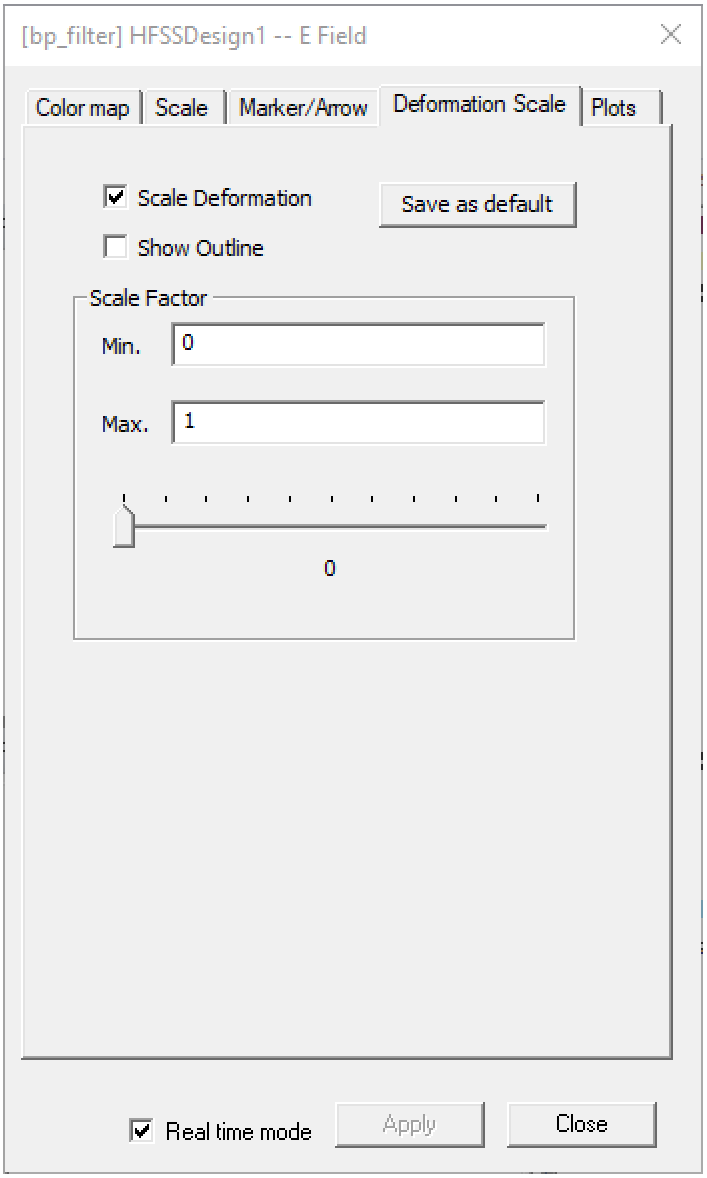
Plot Fields Context Menu for Stress Feedback Projects

The **Plot Fields>Other** menu includes Displacement_vector and Mag_Displacement, a scalar. A field plot remains empty until the displacement data applies to the HFSS solution.



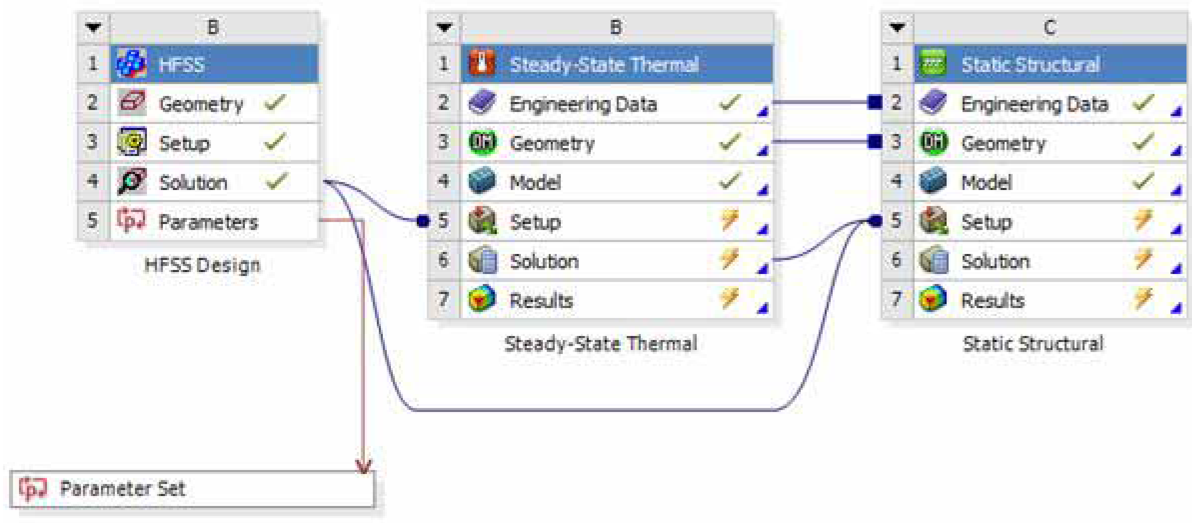
Modify Plot Attributes Dialog for Stress Feedback Projects

When an HFSS plot includes deformation, the **HFSS>Fields>Modify Plot Attributes** dialog includes a tab for scaling the deformation. If you check Scale Deformation, you can also select Show Outline.

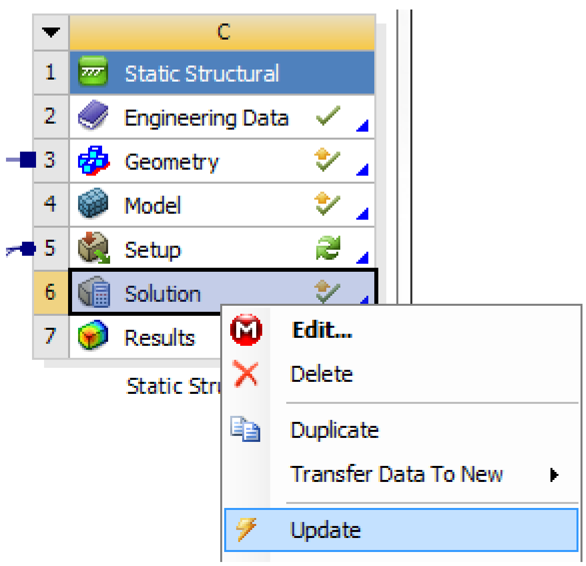


Process Flow for HFSS Workbench for Stress Feedback

to set up the coupling, drag the "Solution" cell of the HFSS system and drop it at the "Setup" cell of a Thermal and Stress system. You also need to also couple the Thermal and Stress system to capture the effect of thermal force. The following image illustrates a coupling setup where both the Thermal and Stress system are ready to be "Updated". The HFSS adaptive solution has converged in 3 passes. The HFSS system has both thermal and stress feedback enabled.

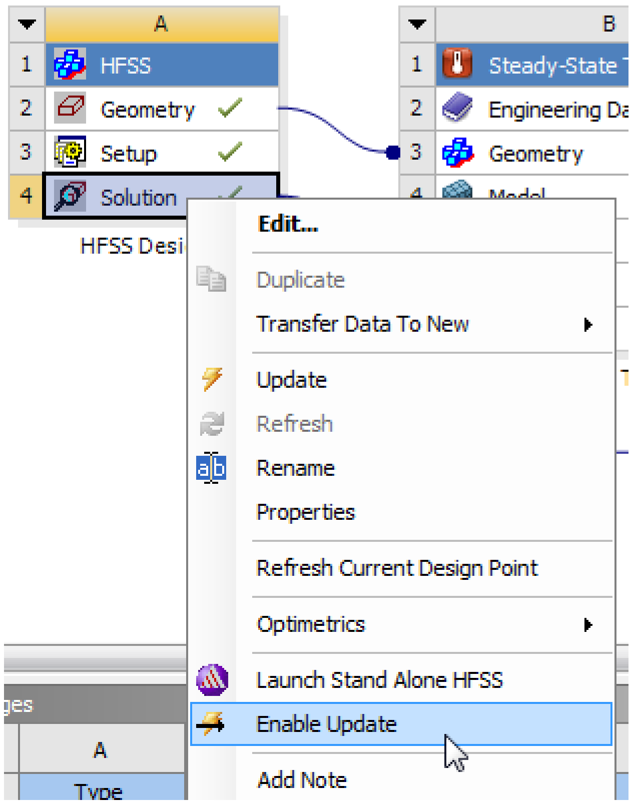


1. Right-click the Solution cell of Stress and select "Update"



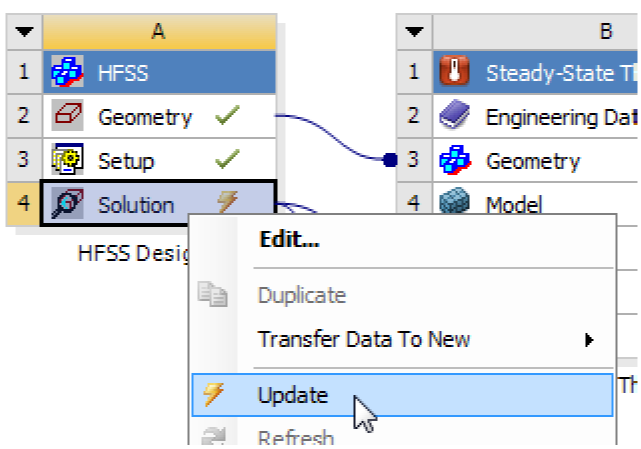
   a. The Setup of Thermal will be "Updated" with em loss from HFSS
   b. The Solution of Thermal will be "Updated" and temperature will be exported to HFSS
   c. The Setup of Stress will be "Updated" with thermal force from Thermal and force density (just zeros) from HFSS
   d. Displacement will be exported to HFSS after Stress finishes simulation

2. Right-click the Solution cell of HFSS and select "Enable Update"



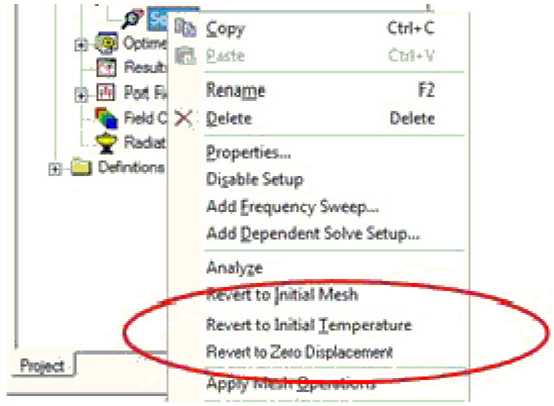
The Solution cell icon changes from a green checkmark to the Update lighting icon.

3. Right-click the Solution cell of HFSS and select "Update"



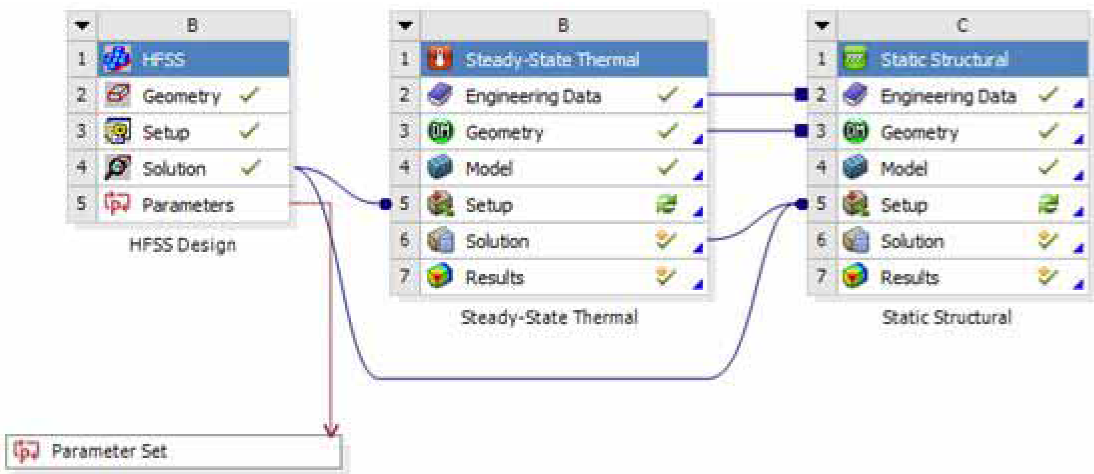
   a. HFSS re-simulate the 3rd pass with its mesh and the temperature/displacement feedback
   b. The profile will show information about the feedback

c. Both "Revert" menus will be available at the right mouse click menu of the solve setup item.

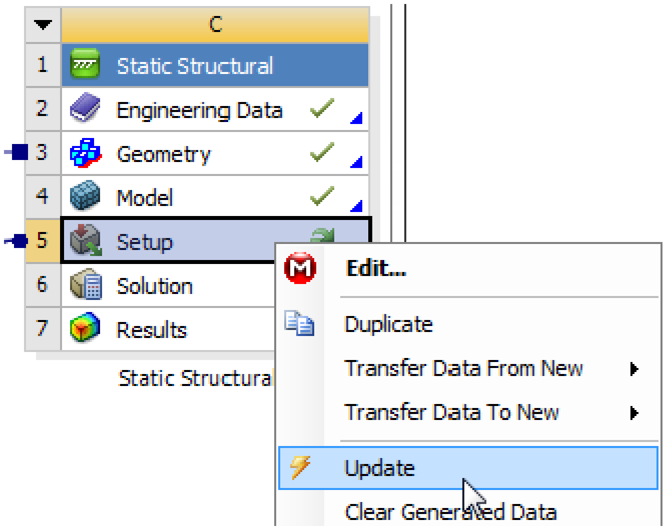


Manual Iteration After First Iteration Completes

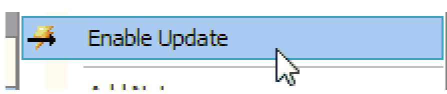
The following image illustrates the WB schematic after the previous situation, after the 1st iteration has completed. Notice the changed icons for the status for Setup for Thermal and Structural.



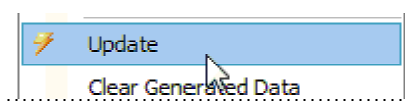
1. Right-click the Solution cell of Stress and select "Update"



   a. The Setup of Thermal will be "Refreshed" and "Updated" with EM loss from HFSS
   b. The Solution of Thermal will be "Updated" and temperature will be exported to HFSS
   c. The Setup of Stress will be "Refreshed" and "Updated" with thermal force from Thermal and force density from HFSS
   d. Displacement will be exported to HFSS after Stress finishes simulation
2. Right-click the Solution cell of HFSS and select "Enable Update"



3. Right-click the Solution cell of HFSS and select "Update"



   a. HFSS re-simulates the 3rd pass with its mesh and NEW temperature/displacement feedback
   b. The profile will show information about the feedback

Note that the delta temperature and displacement is being reported in the profile

Revert the HFSS Solution For Temperature and Displacement

You can revert the temperature and displacement in HFSS solution separately.

1. Select "Revert to Initial Temperature" in HFSS.

A warning message notifies you about the invalidation of solution

2. Right-click HFSS solve setup and select "Analyze"

HFSS re-simulates the 3rd pass with its mesh and displacement that was previously exported from Mechanical, but without temperature.

3. Select "Revert to Zero Displacement" in HFSS, and followed by "Analyze"

HFSS re-simulates the 3rd pass with its mesh without neither temperature nor displacement

Only Stress Feedback

The coupling setup is the same as in Use Case 1, with either one of the following differences

- The HFSS system is not enabled for thermal feedback. Since HFSS is not enabled to support feedback, the "Export Result" properties will be missing in the thermal system's property window.
- The HFSS system is enabled for thermal feedback, but users select not to "Export Result" in the thermal system. This is to disable the "automatic" feedback after Mechanical finishes its simulation.

1. Right-click the Solution cell of Stress and select "Update"
   a. The Setup of Thermal will be "Updated" with em loss from HFSS
   b. The Solution of Thermal will be "Updated"
   c. The Setup of Stress will be "Updated" with thermal force from Thermal and force density from HFSS
   d. Displacement will be exported to HFSS after Stress finishes simulation
2. Right-click the Solution cell of HFSS and select "Enable Update"
3. Right-click the Solution cell of HFSS and select "Update"
   a. HFSS re-simulate the 3rd pass with its mesh and the displacement feedback
   b. The profile will show information about the feedback and "Revert to Zero Displacement" will be enabled at the RMC menu of the solve setup item

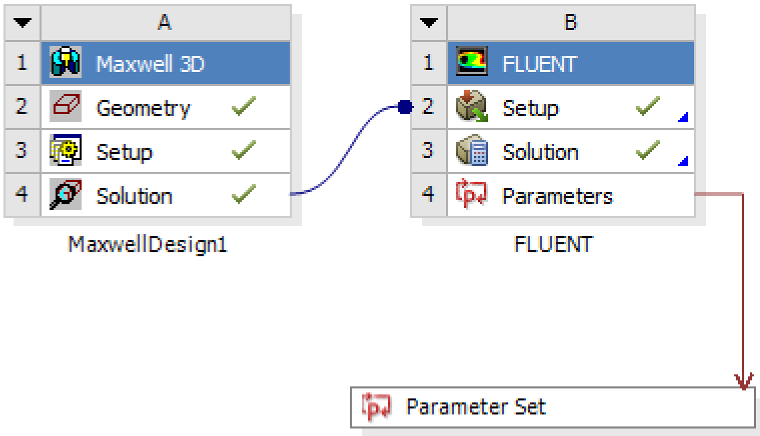
Feedback Iterator

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.

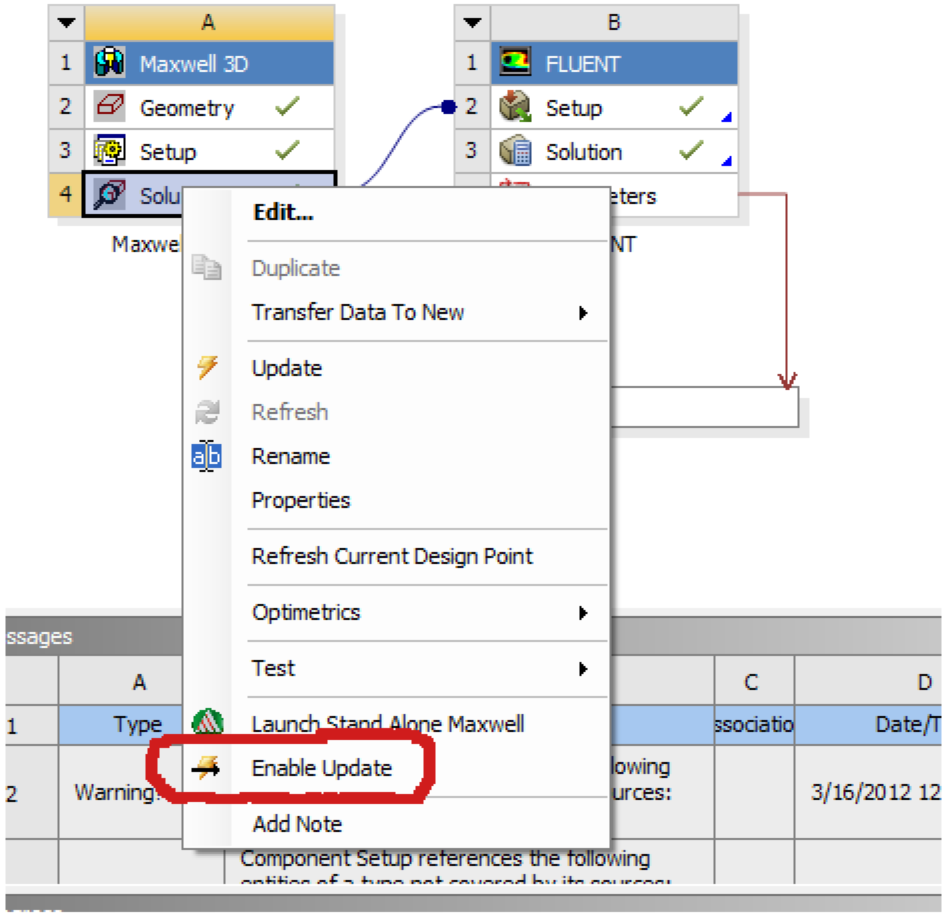
The one way portion (Upstream source component to Downstream target component) is handled via existing workbench data/transfer connection mechanism.



The round trip is handled by a separate protocol, agreed upon by the participating systems, whereby the downstream system exports a set of file to a location specified by the upstream system via its one way transfer. This exported data is then incorporated by the upstream system in its next update.

To run the next coupled solve iteration, invoke the Enable Update Gui 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

Feedback Iterator in Use

Feedback Iterator Component Properties

Feedback Iterator GUI Operations

Resetting the Feedback Iterator

Callback Interface

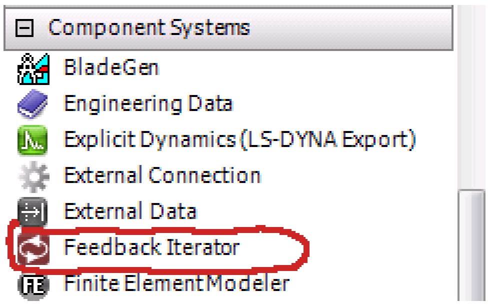
Example Scenarios for Feedback Iterator

The Feedback Iterator System

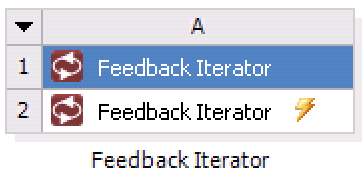
Ansys Electromagnetics Suite provides the Feedback Iterator system for automating the manual steps needed for driving a feedback utilizing system-pair to convergence. In addition to automating feedback incorporation over a user-specified number of iterations, the Feedback

Iterator also allows 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**.



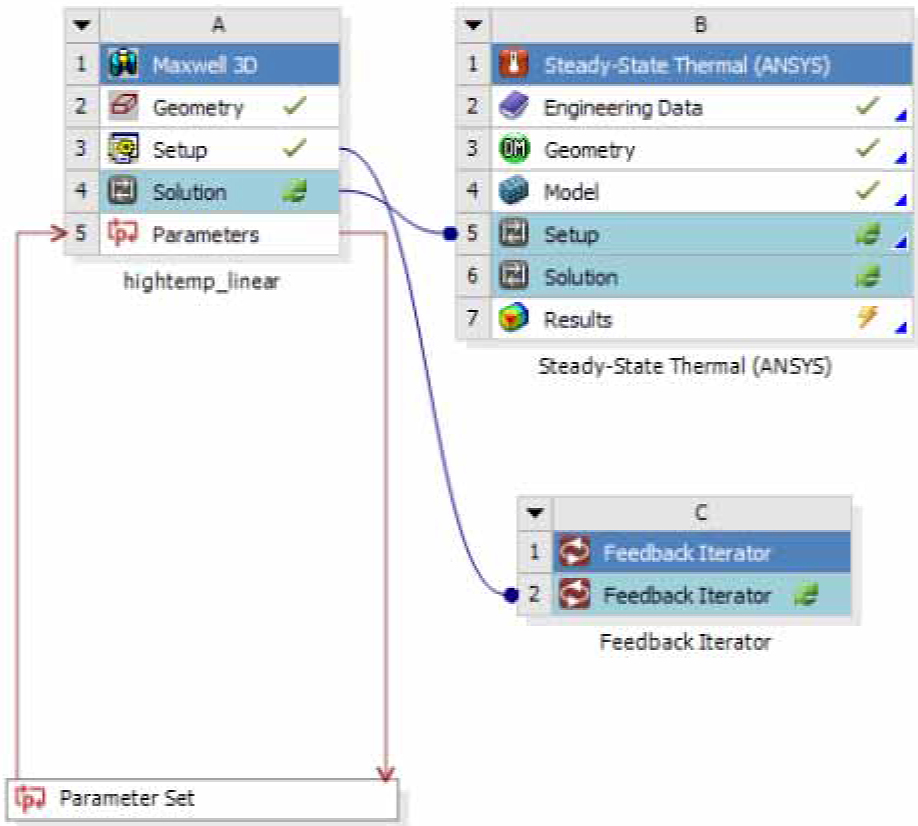
If you drag and drop an instance of the Feedback Iterator into Workbench Schematic, the default system name is Feedback Iterator.



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.

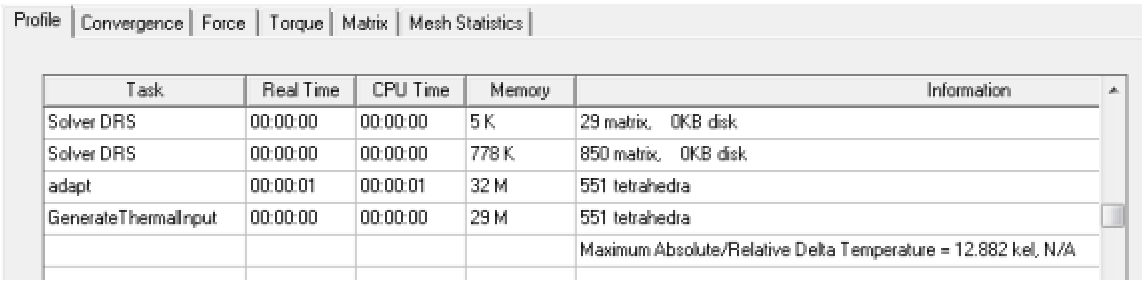
	A	B
1	Property	Value
2	⊟ General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	⊞ Notes	
7	⊞ Used Licenses	
9	⊟ Iterations	
10	Iterations Completed	0
11	Max Iterations	100
12	⊞ Callback	
14	⊟ Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	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)



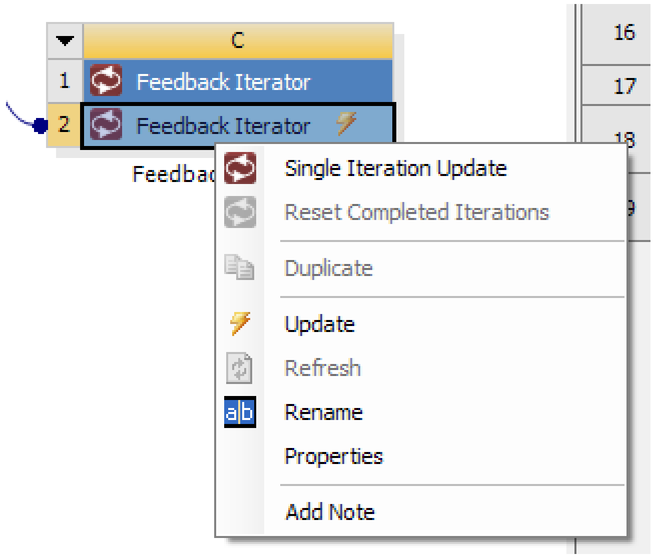
Task	Real Time	CPU Time	Memory	Information
Solver DRS	00:00:00	00:00:00	5 K	29 matrix, 0KB disk
Solver DRS	00:00:00	00:00:00	778 K	850 matrix, 0KB disk
adapt	00:00:01	00:00:01	32 M	551 tetrahedra
GenerateThermalInput	00:00:00	00:00:00	29 M	551 tetrahedra
				Maximum Absolute/Relative Delta Temperature = 12.882 kel, N/A

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

Related Topics

Callback and State API

Callback and State API

State is managed somewhat simply but in a limited fashion. All API methods take a final dictionary argument. This dictionary is limited to using **string keys** and **number or string values**. Within this limitation, simply add new keys, read old keys, clear the dictionary, etc. and it will be persisted across functions calls and 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 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, identify them.
Container	A single container that is being processed. This is a DataContainerReference.
State	A read/write python dictionary 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 use other files to send commands to various containers (vb, js, apdl, python, etc.), all of those files are best saved in the user_files directory. This allows you to use 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, 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 and an "OK" button.

Example Scenarios for Feedback Iterator

This section describes several scenarios for using the Feedback Iterator:

Setting up Iteration with Feedback Iterator

Breaking Iteration Control

Starting an Iterative Update

Running a Single Iteration

Interrupting an Iterative Loop

Resuming an Interrupted Iterative Loop

Modifying any of the Systems Involved in Iteration (coupled clients)

Iterating to Convergence

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 User clicks on the Iterator component and sets the desired Feedback Iterator properties.	Electromagnetics product setup component is connected to the Iterator. Electromagnetics product solution and downstream setup and solution components are coupled as clients to the iterator. The iterator component is the coupled master and the rest are coupled clients. Any coupling changes schematic visuals: • Coupled cells including coupling master are colored differently from normal cells • Coupled cell icons change to reflect the icon of the coupling master • Coupled client cells no longer display the **Update** context menu item and cannot be updated via script commands • The cell states of all coupled cells (master and clients) are synchronized to be the most pessimistic state among any of them (e.g., after an update, any modification will set all of them to the "Modified" state).
Electromagnetics 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	same as above	No changes occur beyond what is expected when you create a new connection (no

Scenario	Steps to execute	Outcome
cell does not have any downstream targets		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.

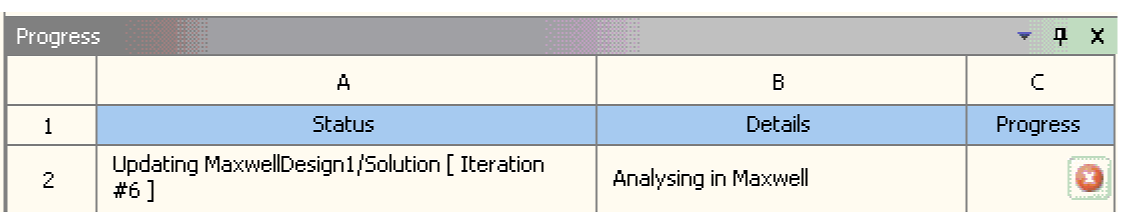
Breaking Iteration Control

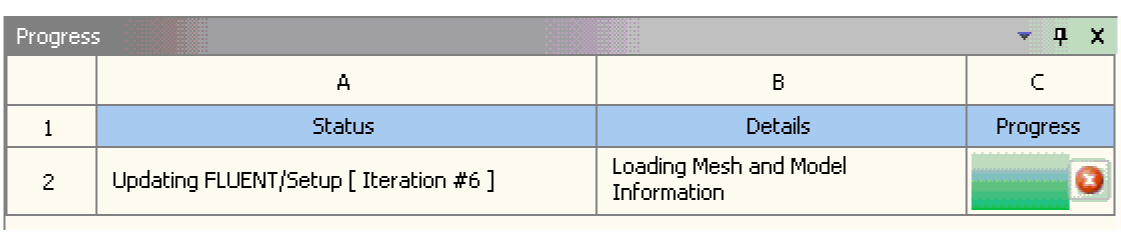
Scenario	Steps to execute	Outcome
Iterator coupled to Electromagnetics product solution cell and downstream setup/solution cells	• Break Electromagnetics product solution cell's provides link OR • Break Electromagnetics product setup link to iterator OR • Delete Electromagnetics product solution consumer system OR • Delete Iterator system	In addition to what is normally expected from the user actions, all coupled clients are de-coupled (their icons and colors on UI are restored)
Electromagnetics product setup cell is connected to 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

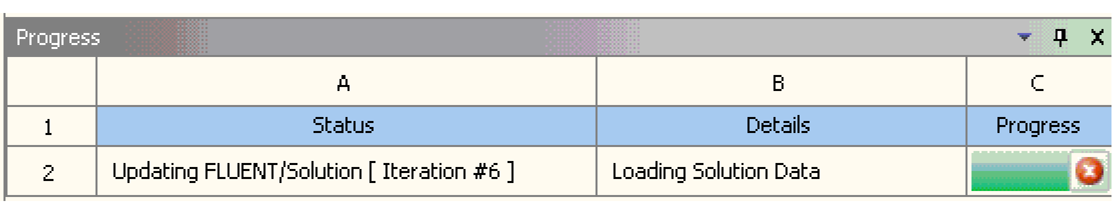
Scenario	Steps to execute	Outcome expected from the user actions

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.
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.

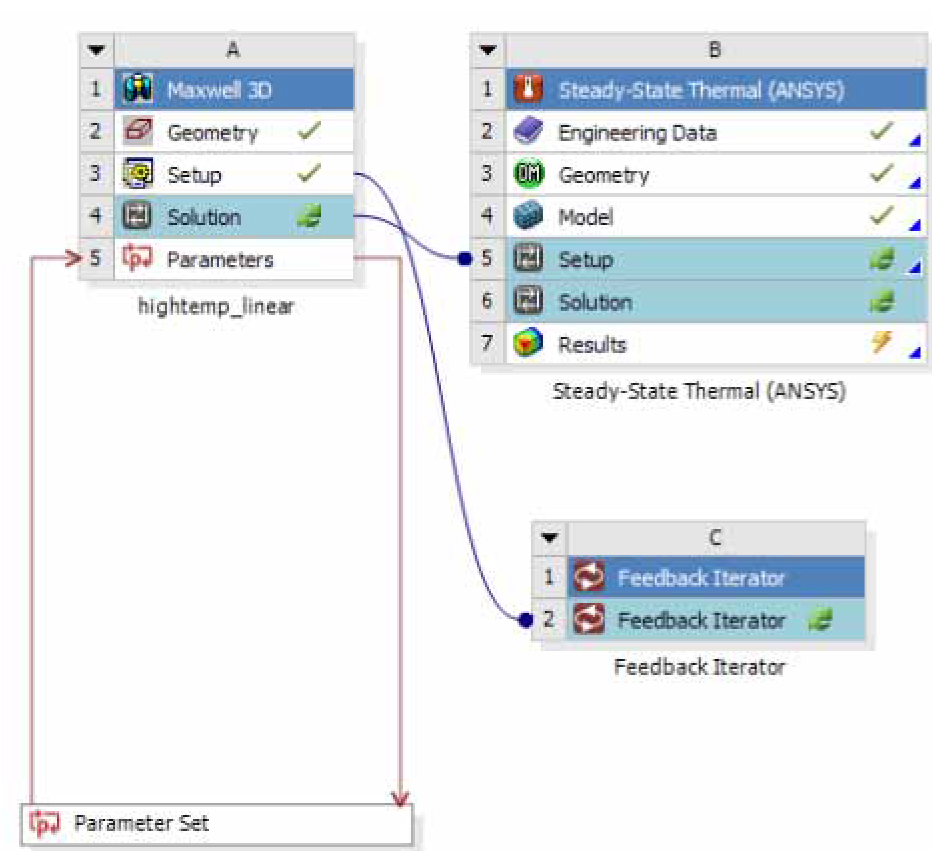
Modifying any of the Systems Involved in Iteration (Coupled Clients)

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)

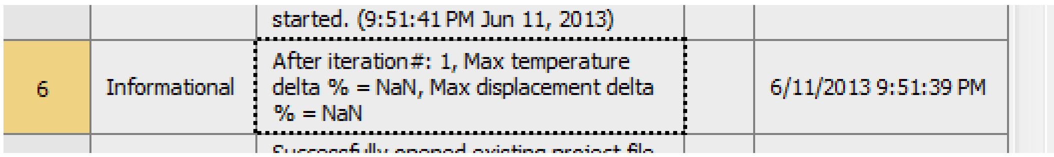
The following temperature convergence example uses Feedback Iterator with a Maxwell design, coupled with a Steady-State Thermal component. Iterating to convergence operates similarly for HFSS projects.



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

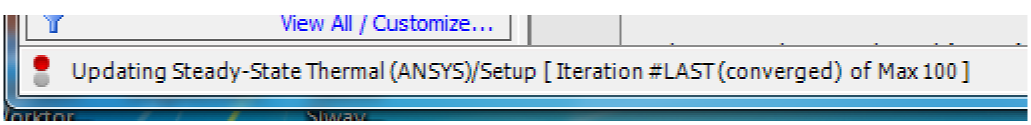
The first iteration, since it has no previous iteration, cannot return a meaningful delta value.

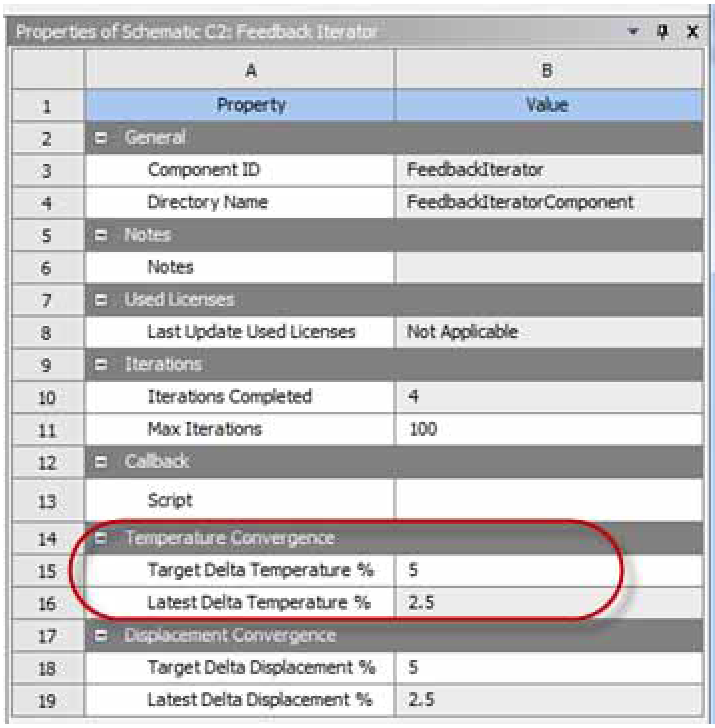


At the end of this iteration, the Latest Delta value is listed as Not Available.

Convergence Achieved

In this example, convergence has been achieved by the third iteration.



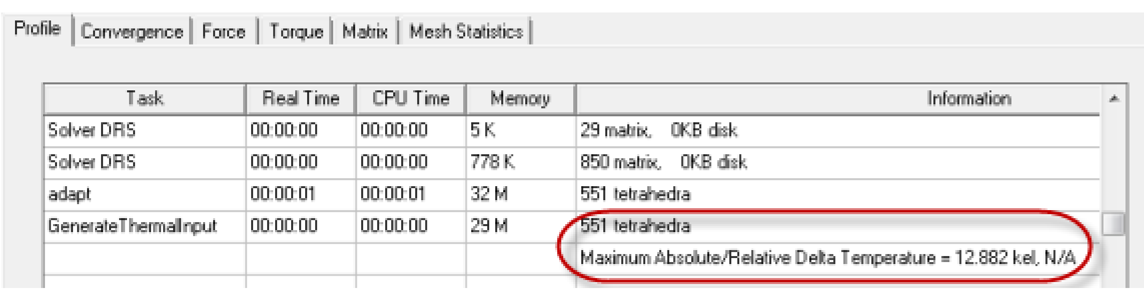


Properties of Schematic C2: Feedback Iterator

	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	Not Applicable
9	Iterations	
10	Iterations Completed	4
11	Max Iterations	100
12	Callback	
13	Script	
14	Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	2.5
17	Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	2.5

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 dialog box 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.



Profile | Convergence | Force | Torque | Matrix | Mesh Statistics

Task	Real Time	CPU Time	Memory	Information
Solver DRS	00:00:00	00:00:00	5 K	29 matrix, 0KB disk
Solver DRS	00:00:00	00:00:00	778 K	850 matrix, 0KB disk
adapt	00:00:01	00:00:01	32 M	551 tetrahedra
GenerateThermalInput	00:00:00	00:00:00	29 M	551 tetrahedra
				Maximum Absolute/Relative Delta Temperature = 12.882 kel, N/A

Surface Force Density in HFSS

HFSS can calculate surface force density for Work Bench coupling and for post processing within HFSS. The whole approach resembles Maxwell except that HFSS must calculate the surface force density at the post processing stage while solver provide the surface force density in Maxwell.

The surface force density formulation is provided in a technical note here. Surface forces may exist on surfaces where one side is conductor but the other is not, or finite conductivity and layered impedance boundary. The computation of surface force density will be performed at field geometry instance level.

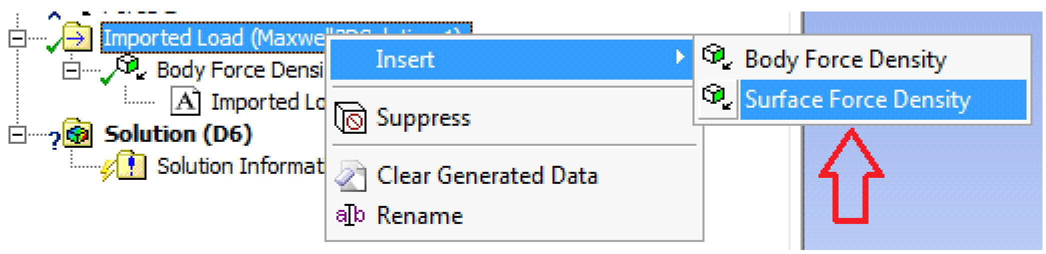
This feature is mainly for the purpose of mapping surface force density in HFSS to Work Bench Mechanical. We will also provide it for post processing within HFSS. This feature should work regardless of HFSS solution types as long as surface forces exist. Surface forces may exist when one side is conductor but the other is not, or finite conductivity and layered impedance boundary.

If an old coupling project happens to request surface force and is solved again, then the surface forces (likely nonzero) will be passed to mechanical side.

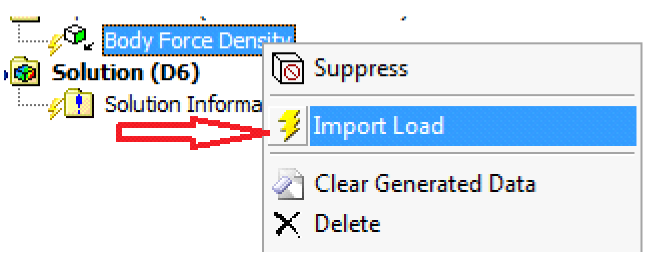
Mapping surface force density from HFSS to Work Bench Mechanical

The behavior is the same as all current couplings.

1. From the Mechanical side set up a surface force coupling to an HFSS design similar to Maxwell.



2. Import the load.

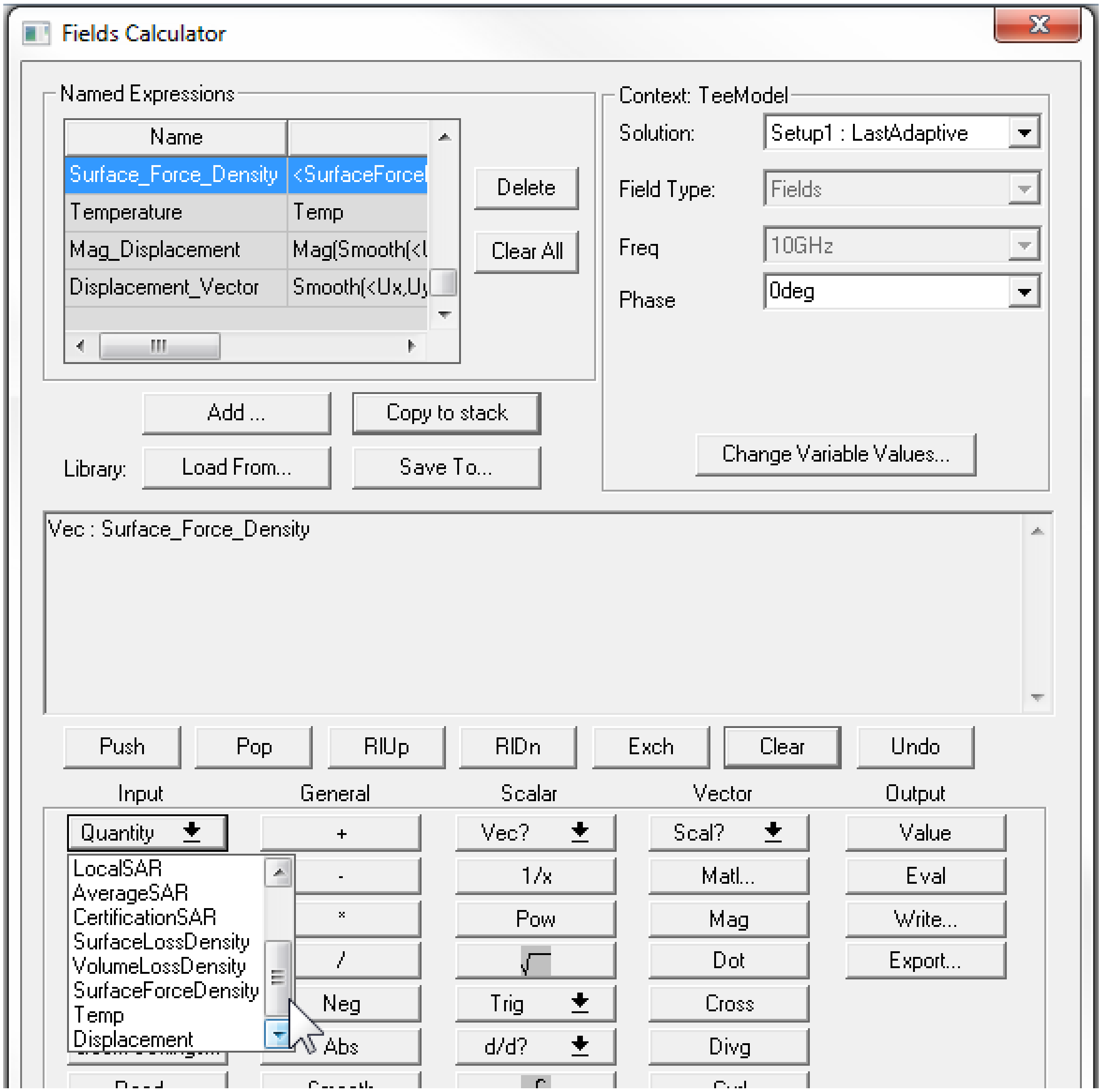


Behind the scenes Mechanical will pass necessary information to HFSS, points on surface to request surface forces and auxiliary files, HFSS will then calculate the surface force densities and pass them back to Mechanical.

3. Post process surface density within HFSS over any surfaces where surface forces exist.

Surface Force Density will appear in the context menu in the Other category under Fields menu.

In the fields calculator Surface Force Density appears in the Named Expression list and the Quantity drop-down menu. So the user can use it the same way as other vector quantities.



24 - 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.

25 - Circuit Design Technical Notes

The Circuit design technical notes provide operational details for the Nexxim analyses that help explain the available options and controls available. The technical notes also contain reference information about formats, options, layer colors, and data types. The topics discussed are:

State-Space Fitting Guide

State-space fitting is a technique for generating time-domain models of systems on their frequency-domain responses at a set of frequencies. The term state-space refers to the following representation of a linear system,

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

in which *u* is a vector of inputs, and y is a vector of outputs. The input/output relation is mediated by a *state vector, x*. All of these vectors depend on time, and in particular, the time-derivative of *x*, denoted $\dot{x}$ is governed by the first of these equations. The matrices A, B, C, and D are constant, and hence the first of these equations is a system of linear, first-order differential equations in the state vector. The second equation gives the output as a linear function of the state vector and the input. This is a general representation of a linear system, and one which is very convenient for integration in a time-domain solver, such as a circuit simulator. Moreover, in most cases, this representation is more compact than say, a representation by the impulse response, and this compactness leads to faster time-domain simulations.

The task of state-space fitting involves the determination of A, B, C, and D, such that the frequency-domain response of state-space model matches some frequency-domain data, typically given at a set of frequencies. The frequency-domain transfer function matrix associated with a state-space model can be obtained using the Laplace transform. The result is,

$$H(s) = C(sI - A)^{-1}B + D$$

where I is the identity matrix, and *s* is the Laplace transform variable, which is related to frequency f by

$$s = 2\pi jf$$

with

$$j = \sqrt{-1}$$

So, given a set of frequency-domain data, say

$$S(s_i) \text{ for } i = 1, 2, \ldots, N$$

the aim of a fitting algorithm is to find A, B, C, D, such that

$$H(s_i) \approx S(s_i)$$

for all *i*. One possible mathematical formulation for the state-space fitting problem is as a minimax problem,

$$\min_{A,B,C,D} \max_i |C(s_i I - A)^{-1}B + D - S(s_i)|$$

It is easy to show that H(s) is a rational function in s, so essentially, this is a problem of fitting a rational function to a set of data. There are several well-established algorithms for fitting rational functions, and the theory behind them is the foundation upon which Ansys's state-space fitting algorithms are built. However, in order for the models to be useful in a time-domain simulation, the condition

$$H(s_i) \approx S(s_i)$$

is not enough. The state-space model must satisfy several other conditions and this is what complicates the algorithms, and indeed even their use.

To be useful in a time-domain simulation, a state-space model must usually satisfy the following conditions:

- Stability
- Passivity

- Smoothness and low-order
- Good fit at DC

Let us explain what each of these mean.

Stability

The relevant notion of stability is Bounded-Input Bounded-Output (BIBO) stability. In general, for any set of frequency-domain data, it is possible to generate both stable and unstable state-space models which fit the data perfectly.

$$H(s_i) = S(s_i)$$

but unstable models are useless in a time-domain simulation because they result in diverging time-domain signals. In the frequency domain, the condition of stability for a state-space model is that the poles of the system, which are the eigenvalues of A, must have negative real parts. All state-space models generated by Ansys's tools are stable.

Passivity

A device is passive if it does not generate energy. This is a stronger condition than stability: not all stable models are passive. In the context of electromagnetic/circuit simulations, the quantities typically modeled are scattering (or S) parameters. These parameters are arranged in a square matrix that maps the incident wave amplitudes to the scattered wave amplitudes. For a passive device, the total power scattered by the device must be no greater than the power incident on it. Mathematically, this condition implies that the 2-norm, or equivalently, the maximum singular value, of the S-parameter matrix must be ≤1, for all frequencies.

$$\| H(s) \|_2 \leq 1 \; for \; all \; s = 2\pi jf$$

Note that a model must be passive even outside the band of frequencies for which data is available. When a non-passive device is introduced into a circuit simulation, the simulation results may diverge for a bounded input, even if the device is stable. A typical case in which this occurs is when the non-passive device is part of a feedback loop. When modeling passive devices, it is important to ensure that the model is passive. Enforcing the passivity constraint is done in a separate step, after a state-space model has been generated. If the model is found to be non-passive, it is perturbed so as to make it passive. This is a complicated and quite computationally demanding step. Ansys has developed several algorithms for enforcing passivity. At the time of writing, passivity enforcement is not on by default. In many cases, a passive fit is obtained even if passivity is not enforced. Moreover, even if the fit is slightly non-passive, the time-domain simulation may still run fine. So the current logic is that, by default, the fits are initially done without passivity enforcement, the time-domain simulation is run, and if the results diverge, then passivity is enforced and the simulation is run again.

Smoothness and Low-order

The order of a state-space model is the number of poles, which equals the size of the matrix *A*. All other things being equal, a low-order model is to be preferred over a higher-order one. Not only would the time-domain simulation run faster with a lower-order model, but a higher-order might match the data at the sample frequencies but behave erratically in between the sample points, or outside the band of given frequency points. Such erratic behaviour can lead to significant artifacts in the time-domain simulation. Keeping the order as low as possible while still meeting the other criteria *usually* leads to models that are smooth between sampling frequencies, and this is relied upon in common fitting algorithms. In the latest version of Ansys's state-space fitter, FastFit, the smoothness between sampling points and out of band are explicitly enforced. In the Tsuk-White Algorithm (TWA), the smoothness is controlled by searching for the lowest order that meets the tolerance, and by limiting the Q-factor of the poles.

Good Fit at DC

The quality of the fit at DC is particularly important, not least because it determines the operating point in a circuit simulation. In fact, the accuracy of the fit to the S-parameters often needs to be extremely high, especially in signal integrity applications. Consider, for example, a signal bus. The elements of the impedance matrix (the Z-parameters) of a bus tend to have a 1/f dependence, diverging at DC. This happens when the dielectric separating the signal and ground planes is assumed to have zero conductivity, which is often a valid assumption. Or, if the dielectric is not considered perfect, the DC impedances are still very large. However, the S-parameters must remain ≤1. This means that the mapping between S and Z,

$$Z = z_0\ (I + S)(I - S)^{-1}$$

where z_0 is the characteristic impedance, amplifies the tiniest errors in S to large errors in Z (following figure). So, a state-space model must either match the DC point data with extremely high accuracy, or, if there is no DC point, the fit to S must be such that the corresponding Z diverges at DC. Otherwise, there is a small leakage current between the signal and ground planes. The FastFit algorithm ensures that Z matches the DC point if one is available. If not, it checks whether it is possible to make Z diverge without affecting the quality of the fit at non-zero frequencies, and if so perturbs the model slightly so as to ensure this.

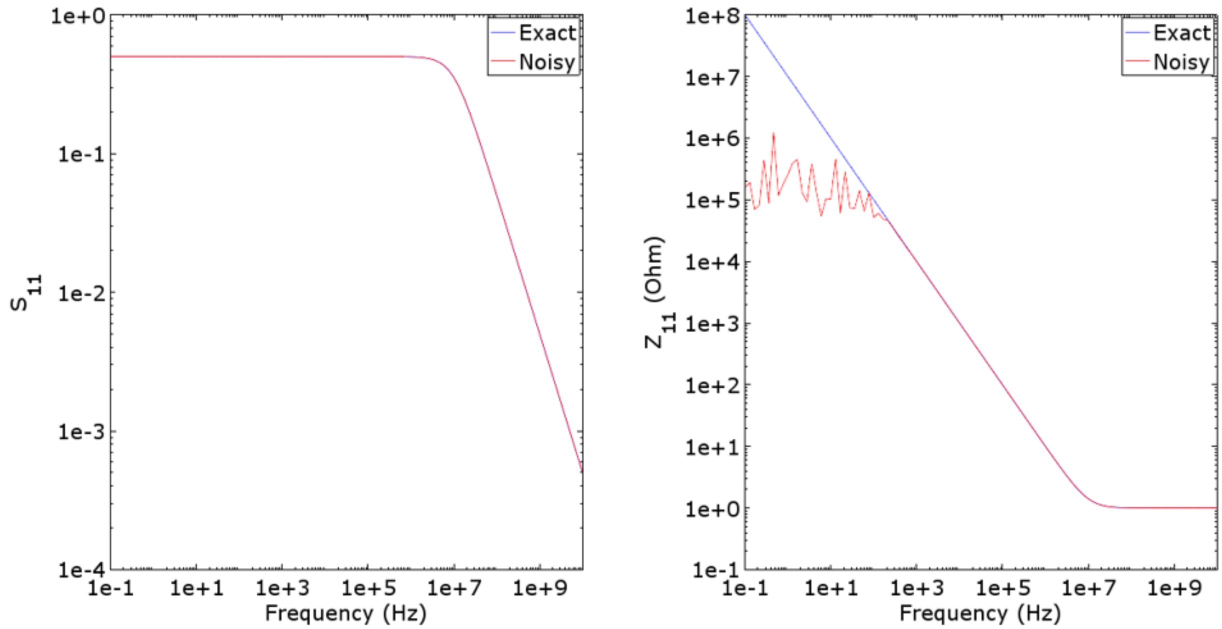


Figure 25-1 S_{11} and Z_{11} of a simple T-network, exact and noisy. Very small noise has been added to the exact S- matrix. This noise is amplified when the noisy S is converted to Z

TWA also tries to ensure a good fit a lower frequencies, though it does not match the DC point, nor ensure divergence if there is no DC point.

When passivity is enforced, the fit is perturbed and this may lead to a poor low-frequency fit. This problem has been recently resolved, however, with a fitting method called IFPVLF.

The various conditions mentioned above make state-space fitting a difficult computational problem, for which there is no known closed-form mathematical solution. Ansys has developed increasingly sophisticated algorithms for state-space fitting over the years. For the sake of backward compatibility, and because an older algorithm might, in rare cases, yield better results than the latest algorithm, nearly all algorithms and their options are still accessible. The main goal of this technical note is to guide the user through this maze.

Algorithms

There are three algorithms available for state-space fitting in Ansys' tools.

- Iterated rational fitting
- TWA
- FastFit

Iterated rational fitting is a fairly old method, which is not recommended. The TWA is a more recent algorithm, which generally yields good results, but it is somewhat slow. FastFit is the latest and fastest algorithm. It explicitly incorporates all the fitting conditions described above, and generally yields results similar to those of TWA. FastFit it is the default algorithm.

These algorithms try to avoid non-passive fits, but passivity is not enforced nor guaranteed. Passivity enforcement is a separate step, in which a non-passive model is perturbed so as to make it passive.

There are four algorithms for enforcing passivity:

- Convex optimization algorithm
- Passivity-by-perturbation algorithm
- Iterated fitting of passivity violations (IFPV)
- Iterated fitting of passivity violations, low frequency (IFPVLF)

The first two of these are quite old, and though theoretically superior in some respects, they tend to be extremely slow and are generally not recommended. IFPV is the default method, and it usually yields good results, except that it tends to perturb the fit at DC. When the accuracy of the fit at DC is important, the next algorithm IFPVLF is recommended.

Data Quality

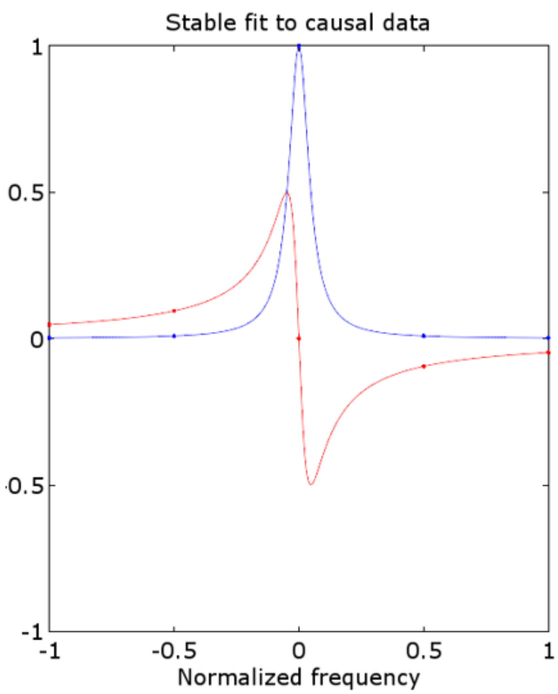
The default algorithms and option settings generally yield good results, with the possible exception of IFPV destroying the accuracy near DC. When results are not satisfactory, check the data before trying to change any of the default options. Examples of several common problems with frequency-domain data follow.

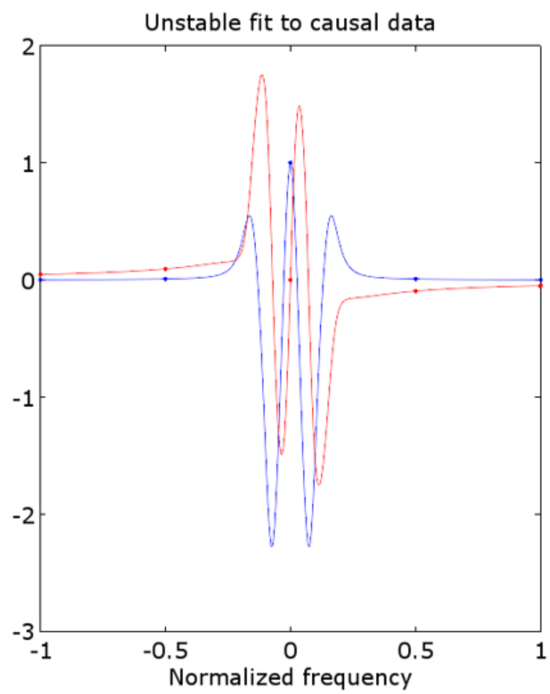
- Data causality
- Data passivity
- Guidelines for sampling frequency responses
- DC data should be real

Data Causality

All physical devices should have causal transfer functions. If the frequency-domain transfer function is Fourier-transformed, the result, which is the impulse response, should be zero for t < 0. Causality, which can be ascertained by inspection in the time-domain, is much more difficult to ascertain in the frequency-domain. The real and imaginary parts of a causal transfer function must be related by the Kramers-Kronig relations, which are difficult to use in practice, because they require knowledge of the frequency domain data for all frequencies from zero to infinity. The frequency-domain data is only given at a finite set of frequencies, and therefore it is impossible to determine whether they are samples of a purely causal system. Indeed, for any such dataset, it is possible to find both causal and non-causal systems from which they could have been sampled (following figure). Nonetheless, if the data come from a causal system, the non-causal fit is bound to be of high order, non-smooth and non-passive. So, although in a sense there is no such thing as causal or non-causal data. A good fit to it can be found that is of low-order (compared to the number of samples), smooth, and passive.

Non-causal data can occur in many ways. If the data come from measurements, then incorrect de-embedding can make them non-causal. If they come from a simulation, then numerical inaccuracies, or non-causal modeling of physical properties such as permittivity can result in non-causal data. Lastly, any manipulation of the causal data which does not take care to preserve causality is bound to destroy it.





All state-space models are causal, so fitting fails if the data is non-causal. Indeed, the fact that the fitting fails is a strong indication that the data may be non-causal. However, Ansys also has a causality checker, which attempts to ascertain the causality of the data, and which should be used whenever the fitting has failed.

Data Passivity

Ascertaining passivity of the data is much more straightforward than ascertaining causality. For each frequency in the dataset, the 2-norm of the S-parameter matrix should be computed, and it should be verified that

$$\| S \|_2 \leq 1$$

If this is not the case and passivity is enforced, fitting fails. Of course, if passivity is not enforced, a non-passive model is practically certain to be obtained.

A matrix whose 2-norm is greater than 1 can be perturbed so its 2-norm is one, and the minimal perturbation that achieves this can be computed using the Singular Value Decomposition (SVD). However unless the original norms are extremely close to 1, doing this manipulation is bound to make the data non-causal and should therefore be avoided.

Guidelines for Sampling Frequency Responses

State-space fitting can deal with quite significant gaps between samples, especially if each of the given samples can be predicted very accurately by a low-order (compared to the number of samples) model. But if the samples are somewhat noisy or are sampled from a system that has a large number of poles, a fitting algorithm might not produce an accurate, passive, and smooth fit. Ideally, samples should be such that no resonance is missed, and the data should vary fairly smoothly from one sample to the next. One rule of thumb is that the difference between a spline interpolation of the data and a linear interpolation of the data should be fairly small, say no more than 0.1-0.2 for S-parameters.

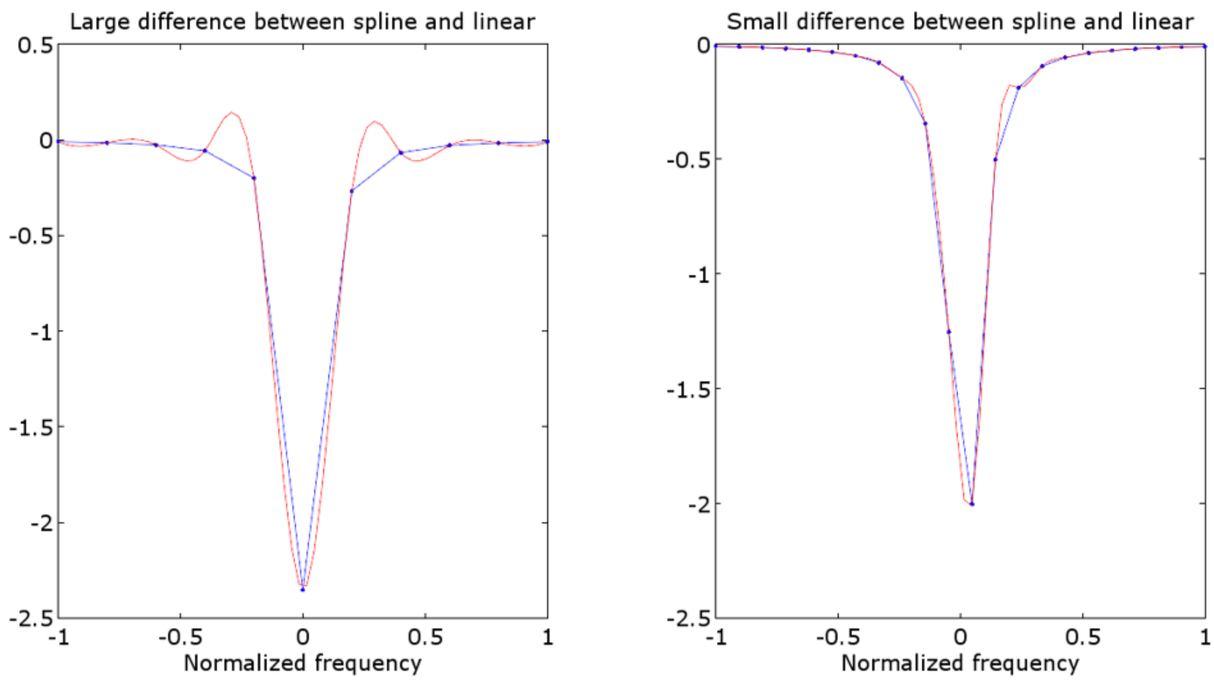


Figure 25-2 Adequate and inadequate sampling of a resonance, judged by the difference between linear and spline interpolations.

The sample frequencies should start as low as possible, preferably at DC, and end only once high-frequency losses become apparent, so the 2-norm of the S-parameter matrix begins to taper off. For systems with very small losses, this might be too high for practical simulations or measurements. The fitter should be able to handle such cases, in which the losses are very small throughout the sampling band, but they are challenging. The reason these cases are difficult is that the state-space model can in principle assume any values in these gaps or out of band. While the FastFit algorithm controls the magnitude of the fit in these gaps (and out of band), significant passivity violations may still occur. Then, when passivity is enforced, removing these passivity violations entails a significant perturbation of the fit, which results in significant errors w.r.t to the data. If it is not practical to sample up to a frequency in which losses become apparent, then at the very least the highest frequency should be above the highest frequency of the excitations that is used in the time-domain simulation.

DC Data Should Be Real

In order for the impulse response to be real, the transfer function must have a zero imaginary part at DC. Perhaps through the result of some manipulation of the data or numerical errors, complex DC data may occur. A state-space fitter cannot fit such data.

Options and Troubleshooting

If the data appears to be causal, passive, and adequately sampled, there are a handful of algorithm options that are worth experimenting with.

- If fitting fails
- If fitting succeeds, but time-domain simulation does not
- If fitting takes very long
- Fitting sensitivity to small changes

If Fitting Fails

By default, a fit with an error > 0.1 is considered to have failed. If this occurs, try:

- Turning off passivity enforcement. If the error is reduced significantly, the data is not quite passive, there are large gaps between frequency samples, or the highest frequency sample is not high enough.
- Experiment doubling and halving the appropriate fitting error. All fitting algorithms use complicated heuristics for determining the best tradeoff between accuracy, passivity, and order. In very rare cases, these heuristics may fail and changing the error tolerance might yield a better tradeoff.
- Try TWA instead of FastFit. TWA has different heuristics than FastFit. In particular, if it cannot find a fit that yields a good fit at low-frequencies, it defaults to finding a fit that ignores the quality of the low-frequency fit.

If all these experiments fail to yield a fit with error < 0.1, then most probably the data is non-causal. When running a circuit simulation, the solver halts if the error > 0.1. If the data cannot be improved upon, the solver's behavior can be deactivated with the option s_element.errorif = 0, so the simulation proceeds. Of course, the accuracy of the results cannot be trusted, but they might be good enough for ball-park estimates.

If Fitting Succeeds, But Time-domain Simulation Does Not

If the error of the fit is reasonable, say < 0.01, problems might still be encountered in the time-domain simulation. If this happens and the state-space model is suspect, try:

- Turning on passivity enforcement, especially if the results seem to diverge in the time-domain. Note that by default, if the transient engine fails to converge and passivity was not enforced, the fit is redone automatically with passivity enforced. This behavior is controlled with the option auto_enforce_passivity.
- Assuming passivity was on, use IFPVLF, especially when the results do not diverge but seem wrong. This can result from an incorrect fit to DC.

If Fitting Takes Very Long

In rare situations or very large problems, fitting may take a long time. This can be because:

- The port-count is very large (several hundreds or more), and passivity is enforced. Enforcing passivity for large port-count problems can be a difficult computational task. At the moment, IFPV is the fastest passivity enforcer Ansys has and, to our knowledge, it is the fastest algorithm there is at the time of writing.
- The data is sampled from a device with a large delay. Delays are not easily represented with a state-space models (they require numerous poles). If the device has a single delay, such as in a single cable, it is possible to extract the delay and fit only what is left over after the delay has been factored out of the transfer function. This has to be done manually by the user, at the time of writing. Another option is to use a convolution-based model, using the option s_element.convolution=1. Except for data representing delays, this option should be avoided, particularly because there is no control over the errors it may introduce.
- The data is very noisy, there are many samples, and the noise is large compared to the fitting tolerance. TWA attempts to recognize such situations and lowers the error tolerance, so it might be faster than FastFit in such a case. Of course, the tolerance can be reduced by the user, in which case FastFit usually is much faster than TWA.

Fitting Sensitivity to Small Changes

In mathematics, well-posed problems are defined as ones that always have a solution, which is unique and which depends continuously on the parameters defining the problem. The state-space fitting problem can be viewed as an inverse problem, of deducing a hidden model from some measurements and like other inverse problems, it is not well-posed. For example, if the data is not passive, there is no solution. If there is a solution, it need not be unique. Since the data is only defined in a finite frequency band, there may be several solutions that differ only outside of the frequency band. Lastly, the solution does not depend continuously on the data.

Consider the following scenario: a fit has been found that meets all requirements and the error tolerance. Now add a small delay to the data. As long as the delay is small enough, the new data can be fit to tolerance without changing the number of poles. If the tolerance begins to increase, there comes a point where new poles must be added to meet the error tolerance. At once, all the matrices of the state-space model must change. So do not expect that a small change to the data always leads to small changes in the model. This sensitivity is a characteristic of the fitting problem, not of any algorithm that is used to solve the problem. Hence, one should not be surprised if a small change in the data, such as roundoff error introduced in generating the data, should have a significant effect on the model.

Lastly, the S-to-Z parameter mapping amplifies small differences immensely. A difference in the 7th digit in the S-parameters can translate to MegaOhms of difference in the Z-parameters. As a result, very small changes in the data can lead to different results in the time-domain. When passivity is not enforced, the DC behaviour should always match the DC-point data (or be an open-circuit at DC if there is none).

When passivity is enforced with IFPVLF, this match should be preserved, although in rare cases, a small deviation still exists. In such cases, the DC behavior may change as a result of small changes in the data.

Summary of State-Space Fitting

State-space models are the preferred models for integrating linear systems in circuit solvers. Algorithms for their generation from frequency-domain data is quite mature and most of the time, require no user intervention. However, state-space fitting is a complicated mathematical problem. In particular, user intervention may be required when the data is inaccurate, non-causal, non-passive, or when the transfer function has not been adequately sampled. In such cases, different algorithms may select different trade-offs and obtain quite different results, so some experimentation with different methods is warranted. However when fitting fails, it is best to carefully examine the data because more often than not, that is where the problem lies.

Circuit S-Parameter Technical Notes

S-parameter elements or N-ports are generic components whose characteristics are defined by user-supplied network-parameter data. They are *black box* elements because no information regarding their function is provided except for their network parameters. S-parameter elements can introduce measured data into the simulation for comparison or modeling, or represent a part of a circuit for which modeling is difficult or undesired.

The network parameters are frequency-dependent and must be specified at many frequency points. The network parameters may be supplied as complex scattering parameters (S parameters), complex admittances (Y parameters), complex impedances (Z parameters), complex propagation constants (gamma parameters), or complex terminal characteristic impedances (Z_0 parameters). The data may be given in either polar form or in rectangular form.

Network parameter data may be entered in spreadsheet form in the N-port import window, but is more commonly read from external files. This data then may be extracted on the data files and used in the simulation. Several data file formats are supported:

- **Touchstone/EESof Data** — This format uses the **.s*p, .y*p**, or **.z*p** filename extension. Only one dataset may be present in each **.s*p, .y*p**, or **.z*p** data file. See "Touchstone Data Format" on page 25-132.
- **Planar EM/HFSS V6.0+ (Z_0-Gamma)** – This format uses the **.szg** file extension, and contains gamma or Z_0 parameter data.
- **Circuit (Compact FLP)** – This format uses the **.flp** filename extension, and may contain several data groups, each of which is preceded by a header. The data group header identifies the group by means of a label, which is used as a reference name. See "Compact FLP Data File" on page 25-132.
- **CITIfile** – This format uses the **.cit** filename extension, and contains S-parameter data in a proprietary format. See "CITIfile Format" on page 25-139.
- **SSS file** – This format uses the **.sss** filename extension, and contains data in a proprietary state-space format. See "State-Space Method " on page 25-18.

The same set of methods for handling frequency-dependent data is applied throughout Circuit designs. Most data is processed using a state-space method. Convolution is also available.

Since network data may come from any source, they may contain violations of causality and passivity that prevent simulation from succeeding. Topics in the following list describe the methods for identifying violations and correcting for them.

Links to S-Parameter Topics

Troubleshooting S-Parameter Issues

S-Parameter Basic Definitions

S-Parameter General References

State-Space Method

Causality Checking and Enforcement

Passivity Checking and Enforcement

Causality, Passivity, and Fitting Errors

Convolution Method

Reference Nodes on S-Parameter Elements

Noise in S-Parameter Elements

Interpolation and Extrapolation of Frequency-Dependent Data

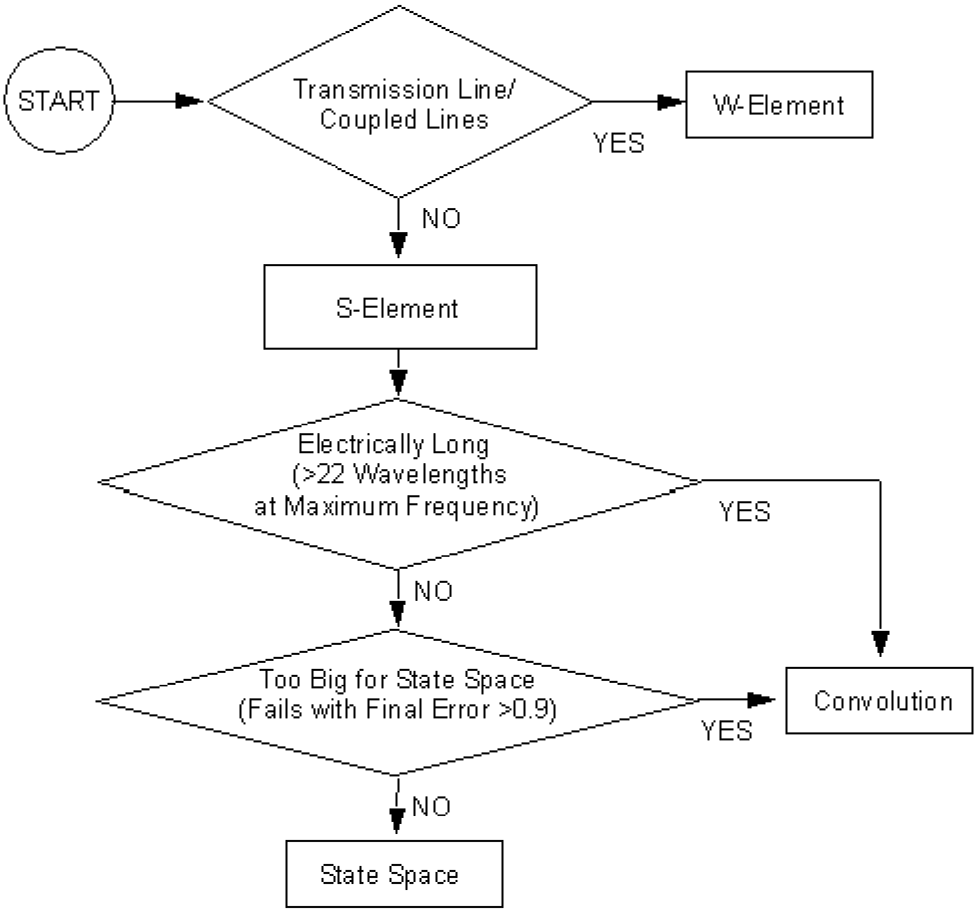
Traveling Wave and Power Wave Formulas

Troubleshooting S-Parameter Issues

These topics show how to select the correct S-parameter representation for your design and how to identify passivity and causality issues with state-space models of frequency-dependent data.

Selecting the Correct Element and Method

The following flow chart outlines the process of selecting the correct implementation for an element defined by S-parameters. The two main choices are between the S-element and the W-element, then, for the S-element, between the default state-space representation and the convolution method.



Selecting Implementation and Method for S-Parameter Elements

Selecting W-Element or S-Element

The first decision is whether to use a W-Element or an S-Element. W-Element modeling exhibits fewer stability issues than S-element modeling.

- When the S-parameter component to be simulated is a transmission line or a set of coupled lines, use a W-Element that points to an S-model. The S-parameter data must be well de-embedded. The S-model computes the unit RLGC or TABLE model for the W-element from the S-parameters. See *Transmission Lines* in the Nexxim Components help for details on the W-Element.
- When the element has many ports, has a short electrical length, or is not a transmission line despite a long electrical length, use an S-Element to model the component in Nexxim.

For processing S-elements, Nexxim uses one of two methods—state space or convolution.

Selecting State-Space or Convolution Method

Nexxim must convert the frequency-dependent scattering parameters in the data to a numerical model that is suitable for time-domain simulation. The default representation, a state-space model calculated by fitting methods, is the best method for most applications. The alternative convolution method calculates the impulse response for the element and convolves the impulse response with the data during transient simulation. Convolution should be selected when:

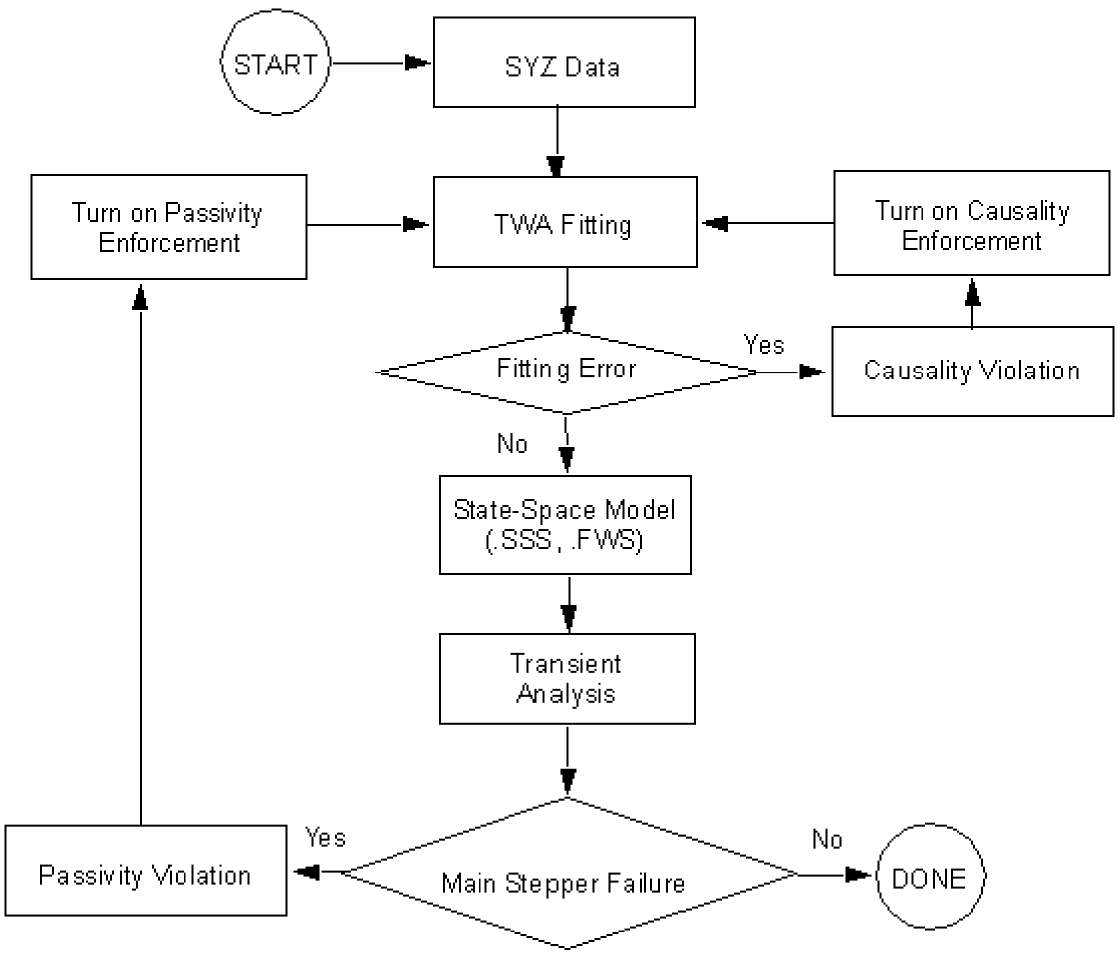
- The circuit is electrically long (greater than 22 wavelengths at the maximum frequency).
- The circuit is too big for the state-space representation. State-space fitting fails with a message such as "Final Error ε, unable to achieve acceptable error." The final error ε, is larger than 0.9 (ε can exceed 1.0 in rare cases, but can never exceed 2.0).

See Convolution Method for details on setting the convolution options.

See the State-Space Method for more on the State-Space representation.

Troubleshooting State-Space Issues

State-space models can exhibit passivity violations and causality violations. These stability issues may be present in the original S-parameter data or introduced by the numerical fitting methods. The following figure illustrates the process for diagnosing these two sources of error.



Identifying Causality and Passivity Issues

Identifying Causality Problems

Causality violations can lead to errors during the calculation of the rational fit to the S-parameter data. The final error indicates the magnitude of the fitting error.

Before starting transient simulation, Nexxim calculates a causal macromodel that approximates the original data in the frequency domain. If the original data is noncausal, the approximation by the macromodel may not be accurate. The more severe the noncausality, the higher is the fitting error of the approximation. If the fitting error is greater than 10%, transient analysis does not even begin. Even when transient runs, a fitting error greater than 1% can give results that are inaccurate, often to a degree that is unacceptable.

See Causality Checking and Enforcement for further information.

Identifying Passivity Problems

Passivity violations can be detected in the original S-parameter data and rejected if they are serious. Mild passivity errors in the data may persist in the state-space realization, and more serious passivity errors may be introduced by the fitting algorithms. State-space passivity violations may result in errors during Transient analysis such as exponential growth of the voltages at the ports of the S-parameter block or timesteps below minimum.

See "Passivity Checking and Enforcement " on page 25-35for details

Passivity Issues with SPICE Format S-Parameter Macromodels

A SPICE format S-parameter macromodel can be generated by broadband export from HFSS or SIwave analysis data, or from another tool such as Twin Builder. The macromodel can be placed as an S-parameter element in the schematic. If transient simulation of the circuit fails to converge, while issuing one of the following errors:

```
Starting stepper failed. Unable to proceed.
```

```
Main stepper failed. Unable to proceed.
```

A likely reason for this failure is a passivity violation in the SPICE format macromodel. However, the solver cannot automatically enforce passivity using the SPICE format. To remedy the situation, you should re-export the SPICE model with passivity enforcement activated , then replace the SPICE format sub-circuit.

Generally, the Touchstone format is the recommended format for passing an S-element to the circuit solver.

See Passivity Checking and Enforcement for more details.

S-Parameter Basic Definitions

Scattering parameters (S-parameters) represent the element behavior in matrix form. The S matrix indicates what fraction of an incident voltage wave is transmitted or reflected at each port. The voltage wave is normalized with respect to the square root of the characteristic impedance of the port, for example, $a = V*Z_0^{-0.5}$. If the characteristic impedance is completely real, the elements of the S matrix are equivalent to the square root of the fraction of power transmitted or reflected. An S-matrix for a three-port structure is:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

where:

- All quantities are complex numbers.
- a_i represents the incident voltage wave at port i.
- b_i represents the transmitted or reflected voltage wave at port i.
- The phase of a_i and b_i is the phase of the incident and of the transmitted or reflected voltage wave at t = 0:

  Za_i is the phase angle of the excitation voltage wave on port i at t = 0 (by default, it is zero).

  Zb_i is the phase angle of the transmitted or reflected voltage wave with respect to the excitation voltage wave.

- S_{ij} is the S-parameter describing how much of the incident voltage wave at port j is reflected back or transmitted to port i. Thus, S_{31} is the amount of power on the port 1 excitation that is transmitted to port 3. The phase of S_{31} specifies the phase shift that occurs as the voltage wave travels from port 1 to port 3.

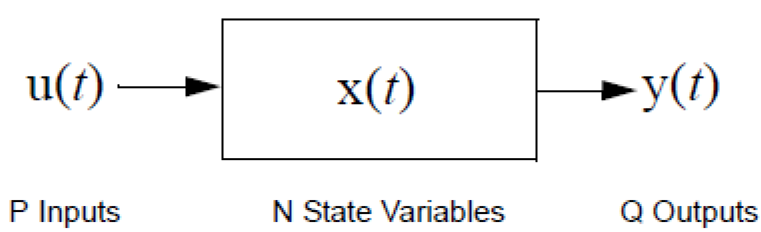
S-Parameter General References

[1] Gonzales, G. 1984. *Microwave Transistor Amplifiers: Analysis and Design*, Prentice-Hall, Inc., 2nd Edition, Appendix L.

[2] *TouchStone File Format Specification*, Rev. 1.1, EIA/IBIS Open Forum, 2002.

State-Space Method

S-parameters describe a system in terms of its frequency-dependent responses. The Nexxim state-space model assumes a linear time-invariant (LTI) system with a vector of inputs **u(*t*)** of length *P* and a vector of outputs **y(*t*)** of length *Q*:



Model of Linear Time-Invariant System

The state-space model represents the data as a vector of N state variables **x(*t*)** representing the responses of the system for all inputs in the specified range, and four state-space matrices, **A**, **B**, **C**, and **D**. See Reference [1] in State-Space References for an introduction to state-space modeling.

State Equations and Output Equations

The vector of state equations gives the rate of change of the state variables (responses), computed as a weighted sum of the current state vector and the input vector:

$$\dot{x}(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$

where:

$$\dot{x} = \frac{\mathrm{d}}{\mathrm{d}t}x(t)$$

Matrix **A**, the state matrix, contains the frequency-dependent poles of the system. Matrix **A** has dimension N by N. Matrix **B**, the input matrix, contains the weights applied to the input vector. Matrix **B** has dimension P by N.

From the time derivatives of the state variables *x(t)* one may calculate the values of $x(t + \Delta t)$ using the vector of output equations:

$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

The output vector is a weighted sum of the current state vector and the input vector. Matrix **C**, the output matrix, contains the weights applied to the state vector. Matrix **C** has dimension Q by N. Matrix **D**, the feed-forward matrix, contains weights applied to the input vector. Matrix **D** has dimensions Q by P.

State-Space Transfer Function

The transfer function is the ratio of the output to the input. Using the Laplace transform to convert from time domain to complex frequency domain *s*, the transfer function can be expressed in terms of the state-space matrices:

$$\boldsymbol{H}(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}$$

where *sI* is an N by N matrix with *s* on the diagonal and zeros elsewhere.

H(s) is a matrix of frequency-dependent transfer functions with dimension Q by P.

Impulse Response and Stability

The impulse response *h(t)* is the response of the system to the Dirac delta function, a pulse of vanishingly narrow width. The impulse response is the inverse Laplace transform of the transfer function, and can be expressed in terms of the poles of the system $\mathbf{p}_i$ (the entries of matrix **A**).

$$h(t) = \sum_{t>0} e^{p_i t} u(t)$$

Bounded-In, Bounded-Out (BIBO) stability specifies that as long as the input is bounded, the output must be bounded. For BIBO stability, the exponential factor e^{pt} must decay rather than increase. Thus, all the poles $\mathbf{p}_i$ must have negative real parts.

For BIBO stability, the norm of the impulse response must always be finite:

$$\|h(t)\| < \infty \quad \forall t$$

State-Space Fitting Methods

By default, Nexxim uses a proprietary fitting method — FastFit — for calculating the state-space matrices on the network data. The FastFit algorithm for state-space fitting is an alternative to the Tsuk-White algorithm (TWA) and Iterated Rational Fitting (IRF). FastFit is generally as accurate as TWA, but is significantly faster than both TWA and IRF. Also, FastFit aims to fit the lower frequencies with higher fidelity.

The Tsuk-White algorithm (TWA) is an alternative proprietary fitting method. TWA generates the poles of the state-space fit on the singular value decomposition of Loewner matrices derived on the input data [State-Space Reference 2]. Unlike rational fitting, TWA is not an iterative procedure. Thus, TWA can produce high-quality state-space fits in much less time than rational fitting.

The TWA algorithm may succeed where rational fitting produces a bad fit or gives a fit that is good but non-passive on a problem too big for passivity enforcement.

The option

```
s_element.disable_fastfit=0|1
```

can be used to toggle between the FastFit algorithm (0) and the method selected by the option s_element.twa (1). The default is 0 (FastFit is enabled).

When **s_element.disable_fastfit** =1, the option:

```
s_element.twa=1|0
```

can be used to toggle between the TWA algorithm (1) and the rational fitting method (0). The default is 1 (TWA is selected).

State-Space References

[1] Bjorn Gustavsen and Adam Semlyen, "Simulation of Transmission Line Transients Using Vector Fitting and Modal Decomposition," IEEE Transactions on Power Delivery, Vol 13, n. 2, pp. 605-614, April 1998.

[2] S. Lefteriu and A. C. Antoulas, "Modeling multi-port systems from frequency response data via tangential interpolation,' IEEE Workshop on Signal Propagation on Interconnects, May 2009.

Causality Checking and Enforcement

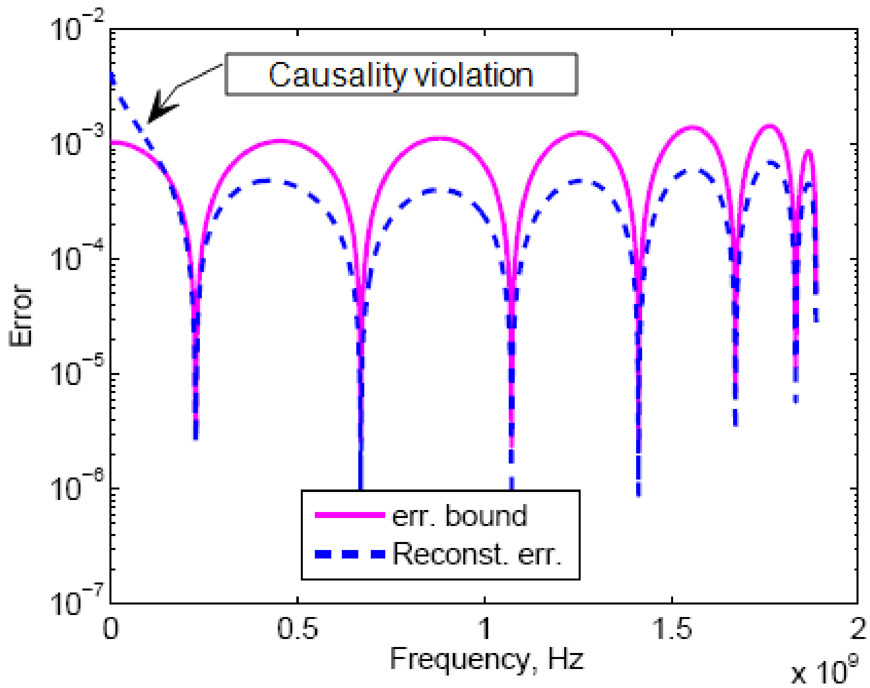
This topic explains how Nexxim handles causality checking and causality enforcement with S-parameter data.

Any physical, realizable system must be *causal*, that is, the time of its response cannot precede the excitation that produces the response. Although causality is a consideration in a simulation of any physical system, the focus for signal and power integrity simulations is on linear, time-invariant (LTI) systems, a category that includes transmission lines, interconnects, packages, connectors, and board-level systems. LTI systems are typically simulated from frequency response data, so ensuring the causality of frequency response data is a key issue.

Nexxim can check for causality violations and can correct causality problems. The basic strategy for causality checking is to compute a causal reconstruction of the data, calculate the difference between the reconstruction and the original data, and compare the reconstruction error to a known error bound or threshold for noncausality over all frequencies. The causal reconstruction can be used in place of the faulty original if necessary to correct the problem. Following are links to the causality checking and enforcement topics.

An Example of a Noncausal System

The following figure shows the internal results of the Nexxim causality checker with a transmission line.



The solid (magenta) line is the error bound for noncausality. The maximum value for the error bound has been set to 2x10^{-3}. The dashed (blue) line is the reconstruction error, the difference between the known-causal reconstructed data and the original. At the lowest frequencies, the reconstruction error exceeds the threshold or maximum error bound, indicating a causality violation in the original data.

Symptoms of Noncausality

Problems with causality of frequency response data shows up as errors in the accuracy and stability of the simulation.

Causality and Accuracy

Before starting transient simulation, Nexxim calculates a causal macromodel that approximates the original data in the frequency domain. If the original data is noncausal, the approximation by the macromodel may not be accurate. The more severe the noncausality, the higher is the fitting error of the approximation. If the fitting error is greater than 10%, transient analysis does not even begin. Even when transient runs, a fitting error greater than 1% can give results that are inaccurate, often to a degree that is unacceptable.

Causality and Stability

$[-\infty, \infty]$

Unstable results can be avoided if the macromodel is passive, or can be made passive. Thus, one possible solution for instability issues is to try passivity enforcement on the macromodel before running transient again. However, passivity enforcement increases the runtime, and any degree of noncausality adds to the problem. To obtain an acceptable fit with noncausal data, the macromodel may require a large model order (number of poles > 200). Model order has a direct effect on the passivity correction runtime.

A second, related issue is memory usage. Passivity correction encounters memory problems for systems with large numbers of I/O ports (>400). When the model order is also high due to noncausality, passivity enforcement is at high risk of running out of memory, which causes transient to terminate without simulating anything. Memory problems during passivity enforcement are common with package models for memory subsystems.

Passivity enforcement also affects the accuracy of the result. If the calculated passive macromodel has a fitting error (to the original data) of more than 10%, transient simulation does not begin. The higher the non-passivity (as measured by the maximum singular value), the greater chance that the macromodel is inaccurate. Hence, strong causality problems typically lead to poorly fitting passivity-enforced macromodels.

Sources of Noncausality

Touchstone data come from three different sources: from measurements, from field solvers (2-D, 3-D, or 2.5-D), and from circuit solvers. Any of these sources can introduce noncausality into data. This technical note analyzes causality issues in data from field solvers and circuit solvers.

Causality Issues with Field Solvers

There are many sources of noncausality in data from field solvers.

- Subsystems in field solvers can have noncausal models.
- Discontinuity of the field solutions can produce noncausal data. Field solvers typically use a special formulation for low-frequency solutions (near DC), which can be discontinuous with the solutions obtained at higher frequencies with the standard formulations.
- Noncausal interpolating functions during interpolation sweep are a frequent source of problems.
- Constant loss-tangent models for modeling frequency-dependent dielectric behavior often contributed to noncausality in the data. Fortunately, the Debye and Djordevic-Sarkar models for dielectric constants have eliminated this source of noncausality.

Causality Issues with Circuit Solvers

Circuit solvers can introduce noncausality into Touchstone data. Most of the noncausality comes from components meant to work at radio or microwave frequencies. A typical example of such a component is the coaxial cable transmission line model. The dielectric loss of the lines is still modeled by constant loss-tangent models (conductance is made to vary linearly with frequency, while capacitance is held constant), and the conductor loss is modeled by varying the square root of the frequency while making the inductance constant with frequency.

Testing for Causality

The basis for causality checking and enforcement is the *dispersion relation*:

A causal frequency response is equal to its own generalized Hilbert transform.

The following topics briefly show how this method for causality checking S-parameter data follows from constraints on the impulse response and on the frequency response of an LTI system.

Enforcing Causality of Touchstone Data

Nexxim can also perform causality enforcement on frequency response data that have been flagged as noncausal. The enforcement method employs fact that the real and imaginary parts of a causal frequency response are Hilbert transforms of each other [see equation (8)].

The reconstruction obtained by combining the real and imaginary parts is guaranteed causal. However, accuracy may suffer. Hence, causality enforcement is not recommended unless there are no alternative fixes.

Understanding the Causality Options

Three options control Nexxim causality checking and enforcement. Here are technical details for the causality options.

causality_check

Setting **.option s_element.causality_check=1** activates a causality test to check for non-causality in the (sampled) frequency data. Causality checking uses a formula based on the generalized Hilbert transform of equation (28).

A signal is declared causal if it meets the following two constraints:

$$\left| E_n^{tot}(\omega_k) \right| \le \delta_{nc} \quad \forall \omega_k$$

(39)

$$\left| \tilde{\Delta}_n(j\omega_k) \right| \le \delta_{nc} \quad \forall \omega_k$$

(40)

A signal is declared noncausal if any value exceeds the threshold:

$$\exists k : \left| \tilde{\Delta}_n (j\omega_k) \right| > \delta_{nc}$$

(41)

The causality check is inconclusive when:

$$\left| E_n^{tot}(\omega_k) \right| > \delta_{nc} \quad \exists \omega_k$$

By default, **s_element.causality_check=0**, and no causality checking is performed.

enforce_causality

Setting **.option s_element.enforce_causality=** 1 runs the causality check and attempts to compensate for any noncausality. Compensation replaces the original frequency data with the causal reconstruction estimate. Any cached S-parameter solution is updated with new corresponding causal estimates. Since causality enforcement affects accuracy, fixing the original non-causal data is always a better solution.

By default, **s_element.enforce_causality=0**, and no causality enforcement is performed.

causality_checker_tolerance

The **.option s_element.causality_checker_tolerance=***val* sets δ_{nc}, the maximum magnitude of error that still satisfies the causality checker. Nexxim sets the default causality tolerance to the overall tolerance for fitting errors. The default for the fitting error is 1% (0.001).

interpolation_typecc

The **.option s_element.interpolation_typecc=1** sets the causality checker to use cubic spline interpolations. 2 selects interpolation by rational functions. Cubic spline interpolation is the default.

integration_typecc

The **.option s_element.interpolation_typecc=1** sets the causality checker to use numerical integration. 2 selects analytical integration. Numerical integration is the default.

enable_cc_continuation

The **.option s_element.enable_cc_continuation=1** enables the causality checker to continue automatically when the default interpolation and integration types lead to an inconclusive causality check. A test is inconclusive when the causality checker cannot determine that the data is definitely causal or definitely noncausal. The causality checker starts with **interpolation_typecc=1** (cubic splines) and **integration_typecc=1** (numerical). If the test is inconclusive, the causality checker continues with **interpolation_typecc=2** (rational functions) and **integration_typecc=2** (analytical). The default is **enable_cc_continuation=1**, automatic continuation is enabled. To disable automatic continuation, set **enable_cc_continuation=0**.

Table 2 compares the performance of the four combinations of interpolation and integration types.

Table 2: Combinations of Interpolation and Integration Types

interpolation_typecc	integration_typecc	False Positives	Resolution	Accuracy	Runtime	Bounded Result
1	1	High	Low	Low	Good	Yes
1	2	High	Low	Low	Good	Not Always
2	1	Medium	Medium	Medium	Worst	Yes
2	2	Low	High	High	Bad	Yes

False positives: A false positive occurs when the checker reports a causality violation for data that are causal. Interpolation via cubic splines (**interpolation_typecc=1**) can yield a high percentage of false positives with either type of integration (first and second rows of Table 2). Interpolation via rational functions (**interpolation_typecc=2**) produces a lower incidence of false positives, especially in combination with analytical integration (**integration_typecc=2**) (third and fourth rows of Table 2).

Resolution: The higher the resolution of the causality checker, the smaller the true causality violations that can be detected reliably. Interpolation via cubic splines (**interpolation_typecc=1**) achieves only low resolution with either type of integration. Interpolation via rational functions (**interpolation_typecc=2**) achieves higher resolution, especially in combination with analytical integration (**integration_typecc=2**).

Accuracy: The accuracy of the computation is measured by comparing the reconstructed data to the original. A highly accurate causality check avoids false positives and inconclusive results. Spline interpolation yields low accuracy while rational function interpolation achieves high accuracy, especially with analytical integration.

Runtime: The runtime is the denominator in the equation for the efficiency of the causality check. Here, interpolation via cubic splines (**interpolation_typecc=1**) has a relatively short runtime with either type of integration (first and second rows of Table 2). Interpolation via rational functions (**interpolation_typecc=2**) uses more computer time in general. The combination of rational function interpolation with numerical integration (**integration_typecc=1**) yields the worst runtime of all the combinations.

Bounded Result: The combination of spline interpolation (**interpolation_typecc=1**) and analytical integration (**integration_typecc=2**) can lead to an unbounded result.

The automatic continuation strategy employed by **enable_cc_continuation=1** starts with low resolution/accuracy and fast runtime (first row in Table 2). If this setup does not yield a conclusive test, causality checking continues with high resolution/accuracy and slower runtime (fourth row in Table 2). The other combinations are not useful due to the runtime and stability issues noted above.

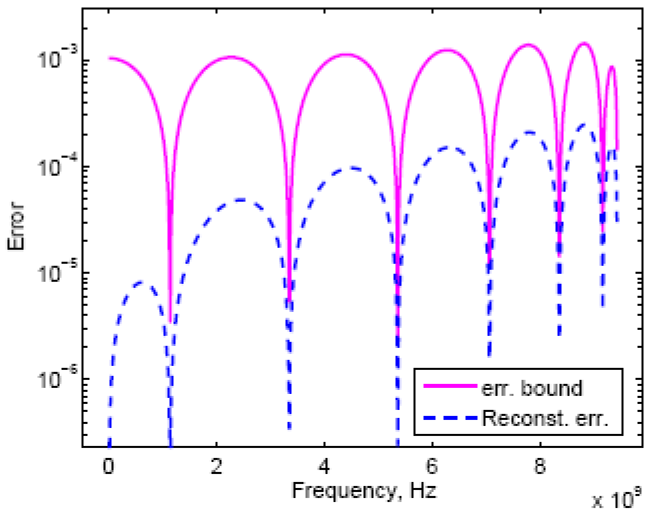
Examples of Causality Checking and Enforcement

This topic show the results of causality checking on a simple resistor, a lossy transmission line, and a known-causal W-element.

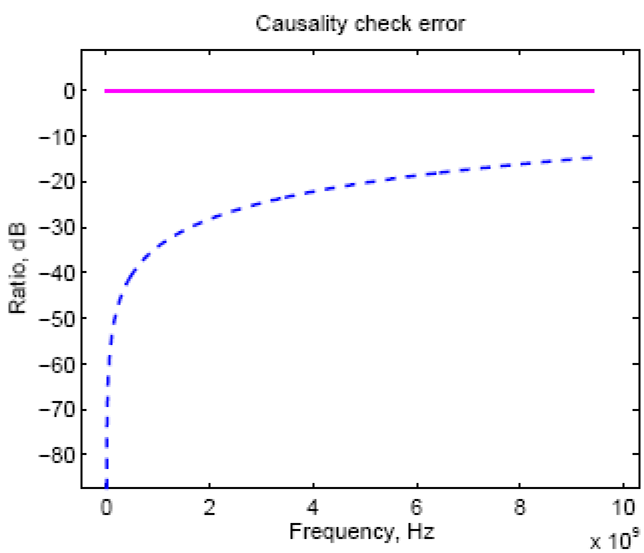
Causality Check of a Resistor

The first example is a resistor with resistance R (Figure 3a, 3b). Such a resistor is known to be causal: the impulse response is an impulse at *t*=0, hence causal. However, confirming this fact in the frequency domain with just the Hilbert transform is problematic due to the non-square-integrable nature of the resistance. Hence, the causality of a resistor is confirmed with the generalized Hilbert transform. Here, the resistance R is fixed as 75Ω and is converted to the corresponding one-port S-parameter. With a 50-Ω reference impedance, the S-parameter S(jω) is computed to be 0.2 at all frequencies. Tabulated values of this S-parameter are created for frequencies in the range [0, 10GHz] in steps of 10 MHz. The parameter ε is chosen as approximately 0.05. So from (19), $\Omega' \approx 2\pi[-9.5GHz, 9.5GHz]$. The causality checker tolerance δ_{nc} is chosen as 0.002. The quantity *n* was found to be 14 for $T_n(\omega)$ to satisfy (31). Because the S-parameter is constant with frequency, the bound $\tilde{D}_N(\omega) \approx 0$. Therefore, $E_N^{tot}(\omega)$ satisfies (33).

In Fig. 3, causality reports of the resistor are shown. In Fig. 3a, the computed $|\tilde{\Delta}_N(j\omega)|$ is compared against $E_N^{tot}(\omega)$. From the figure, it is clear that $|\tilde{\Delta}_N(j\omega)|$ is smaller than $E_N^{tot}(\omega)$ for all frequencies in Ω'except near the subtraction frequencies (minima in the plot). To get a clear picture even at the subtraction frequencies, the ratio of two quantities is computed. In Fig. 3(b), the ratio of the two quantities is shown as a function of frequency. If the ratio is more than 0 dB at any frequency, the frequency response has a possible noncausality at that frequency. From Fig. 3(b), it can be concluded that the frequency response satisfies (29) and, hence, is causal.



(a) Comparison of $|\tilde{\Delta}_n(j\omega)|$ ("Reconst. Err.") versus $E_n^{tot}(\omega)$ ("err. bound").



(b) Plot of the ratio $20\log\left(\frac{|\tilde{\Delta}_n(j\omega)|}{E_n^{tot}(\omega)}\right)$ versus frequency (dashed line). The solid line (0 dB) shows the maximum value the ratio can take for the frequency response to be declared causal.

Figure 3. Causality reports on the frequency response of a resistor, δ_{nc} = 0.002, *n*=14.

Causality Check of a Lossy Transmission Line

The S-parameters of a lossy transmission line are computed on the closed-form expressions of the frequency-dependent per-unit-length (p.u.l.) parameters R (resistance), L (inductance), G (conductance), and C (capacitance). See Figures 4, 5, 6, and 7.

Let R_{dc} and G_{dc} be the p.u.l. resistance and conductance, respectively, at DC. Further, let R_s and G_d be real-values constants such that $R(f) = R_{dc} + R_s\sqrt{f}$ and $G(f) = G_{dc} + G_d f$. Let the inductance
$L(f) = L_{ext}$, where L_{ext} is the p.u.l. external inductance at $f = \infty$, and let $C(f) = C_0$, where C_0 is the constant p.u.l. capacitance.

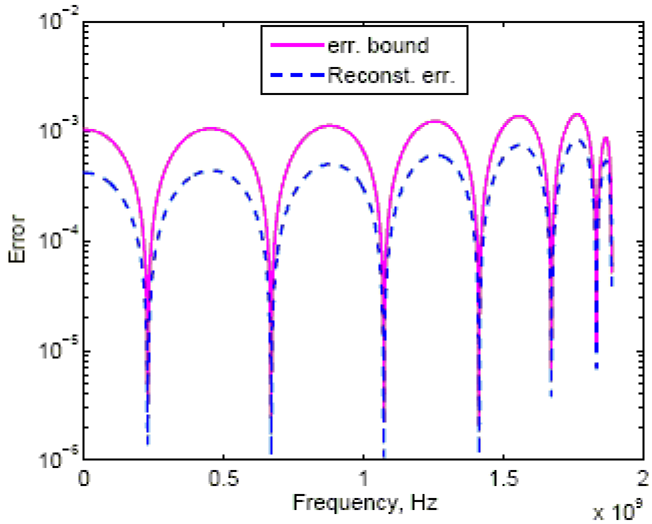
S-parameters computed from such p.u.l. parameters are expected to be noncausal, since the corresponding p.u.l. impedance ($R + j\omega L$) and admittance ($G + j\omega C$) are not causal frequency responses. However, when $R_s = 0$ and $G_d = 0$, the S-parameters are expected to be causal.

Cases 1 through 4 (Figures 4, 5, 6, and 7, respectively) illustrate how causality results depend on the values of R_s and G_d.

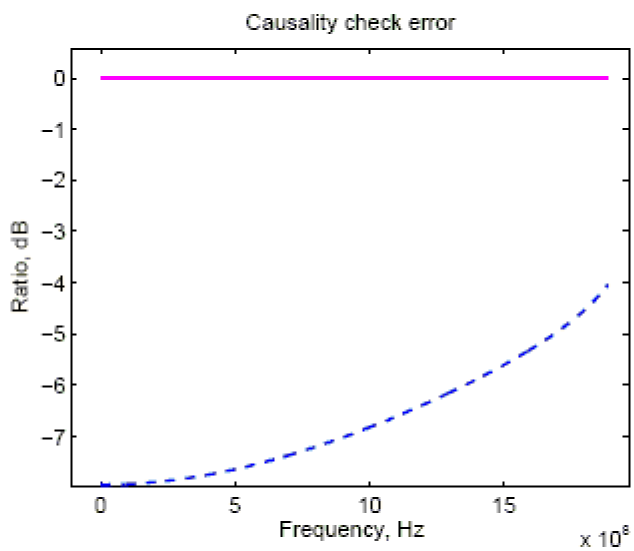
1. $R_s = 0$ and $G_d = 0$. See Figure 4. The transmission line still has loss due to the nonzero R_{dc} and G_{dc}. S-parameters are computed from tabulated frequencies in the range [0, 2GHz]. The causality checker tolerance δ_{nc} is 0.002 and n is 14, as in the resistor example. Figures 4a and 4b indicate that the transmission line is causal under these conditions.
2. $R_s = 0$ and $G_d \neq 0$. See Figure 5. The simulation parameters are the same as Case 1, except that G_d is made nonzero. Figures 5a and 5b show the resulting causality violation at low frequencies.
3. $R_s \neq 0$ and $G_d = 0$. See Figure 6. The parameters are the same as in Case 1, except that R_s is made nonzero. Figures 6a and 6b show the resulting causality violation over half the frequencies.
4. $R_s \neq 0$ and $G_d \neq 0$. See Figure 7. The parameters are the same as in Case 1, except that both G_d and R_s are made nonzero. Figures 7a and 7b show the resulting causality violation, similar to the result in Case 3.

Causality Check of a Causal Model W-Element Transmission Line

Compare the causality check results of the lossy transmission line S-parameters with those obtained using the Nexxim causal W-element RLGC model. In this example, the RLGC values are comparable to the ones used for the lossy transmission line. The causality reports are shown in Figures 8a and 8b. As can be seen on the figure, the new S-parameters are causal.

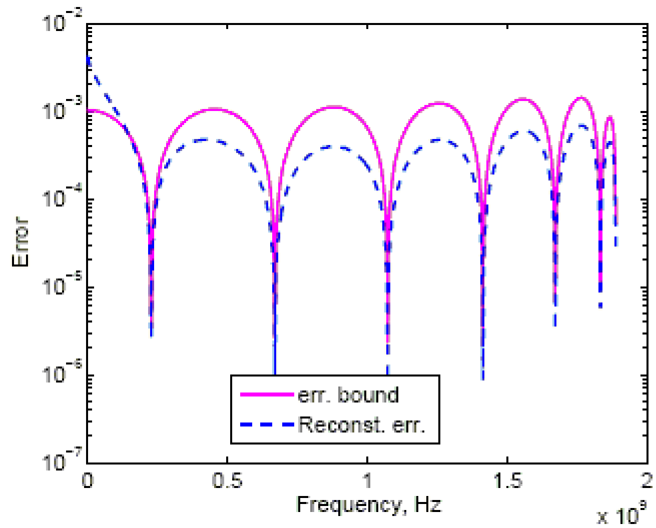


(a) Comparison of $|\widetilde{\Delta}_n(j\omega)|$ ("Reconst. Err.") versus $E_n^{tot}(\omega)$ ("err. bound").

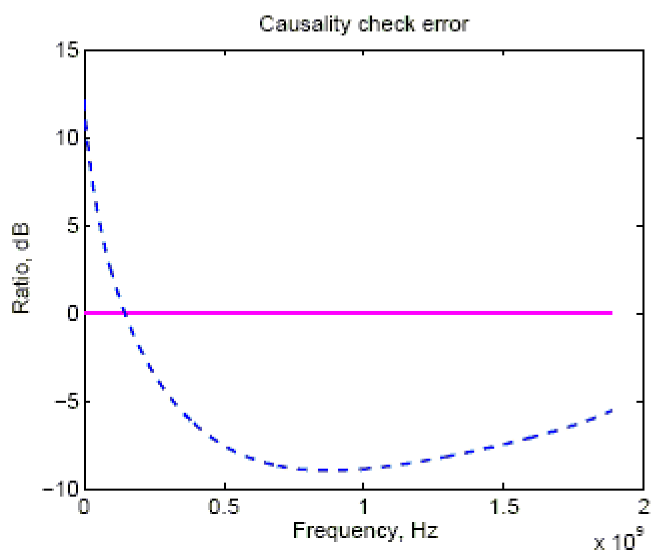


(b) Plot of the ratio $20 \log \left(\frac{|\widetilde{\Delta}_n(j\omega)|}{E_n^{tot}(\omega)} \right)$ versus frequency (dashed line). The solid line (0 dB) shows the maximum value the ratio can take for the frequency response to be declared causal.

Figure 4. Causality reports on the frequency response of a transmission line with R_s =0 and G_d = 0, δ_{nc} = 0.002, *n*=14.

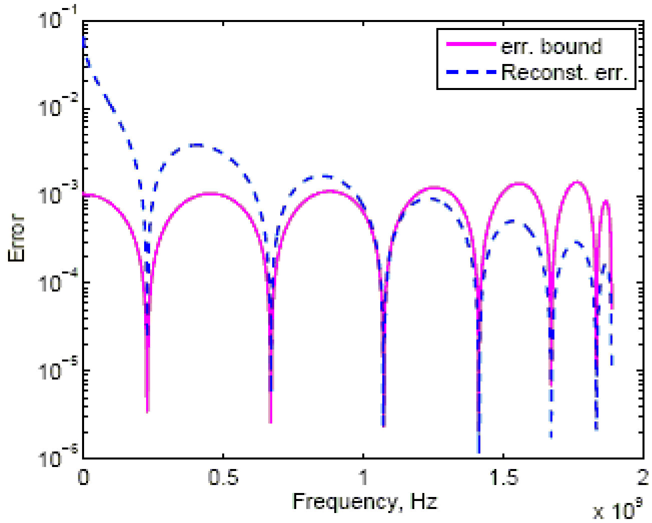


(a) Comparison of $|\widetilde{\Delta}_n(j\omega)|$ ("Reconst. Err.") versus $E_n^{tot}(\omega)$ ("err. bound").

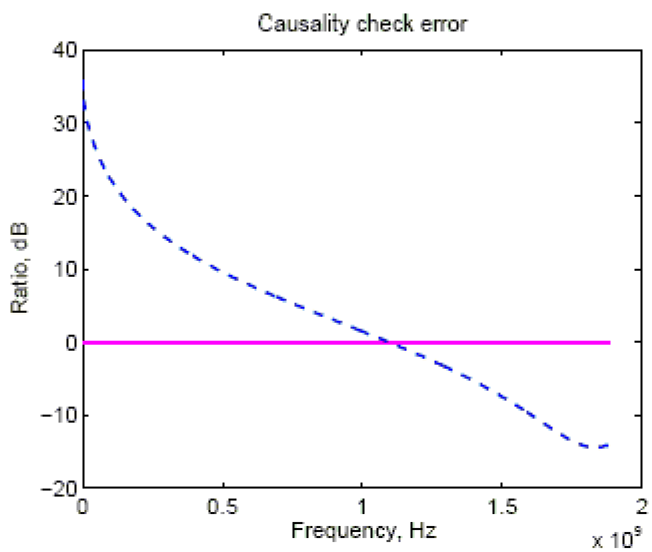


(b) Plot of the ratio $20\log\left(\frac{|\widetilde{\Delta}_n(j\omega)|}{E_n^{tot}(\omega)}\right)$ versus frequency (dashed line). The solid line (0 dB) shows the maximum value the ratio can take for the frequency response to be declared causal.

Figure 5. Causality reports on the frequency response of a transmission line with R_s =0 and G_d <> 0, δ_{nc} = 0.002, *n*=14.

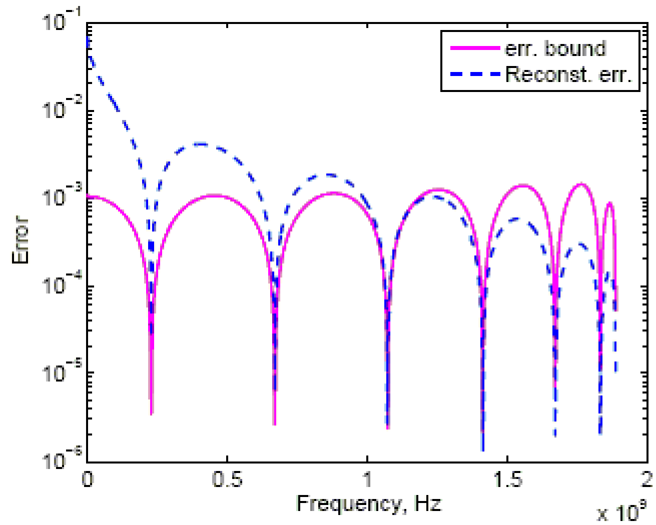


(a) Comparison of $|\tilde{\Delta}_n(j\omega)|$ ("Reconst. Err.") versus $E_n^{tot}(\omega)$ ("err. bound").

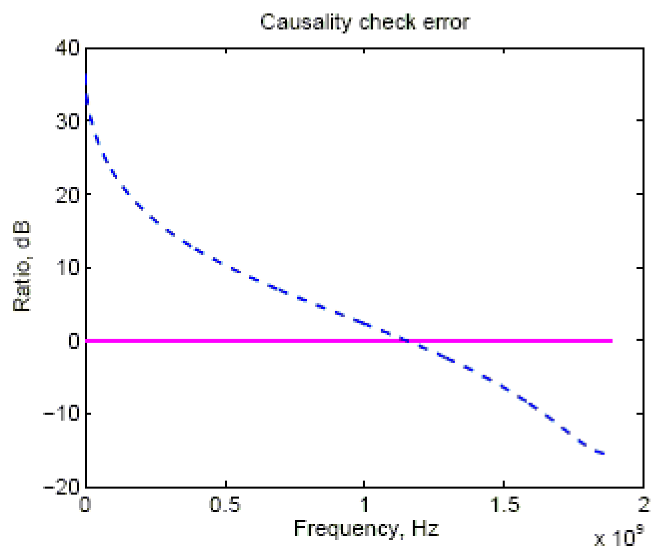


(b) Plot of the ratio $20\log\left(\frac{|\tilde{\Delta}_n(j\omega)|}{E_n^{tot}(\omega)}\right)$ versus frequency (dashed line). The solid line (0 dB) shows the maximum value the ratio can take for the frequency response to be declared causal.

Figure 6. Causality reports on the frequency response of a transmission line with R_s <> 0 and $G_d = 0$, $\delta_{nc} = 0.002$, *n*=14.

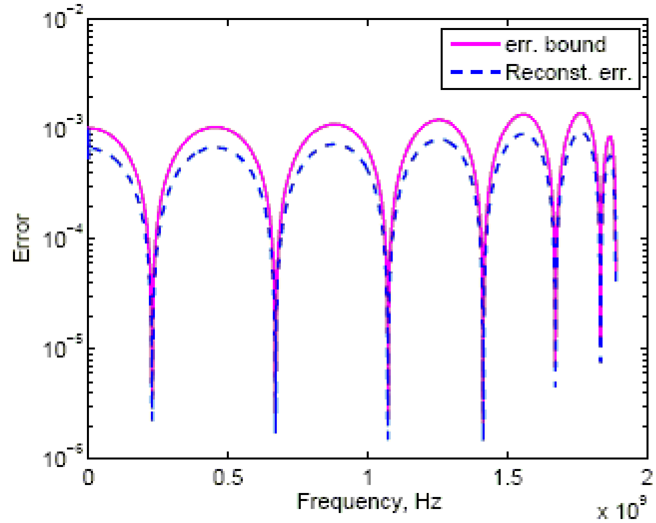


(a) Comparison of $|\tilde{\Delta}_n(j\omega)|$ ("Reconst. Err.") versus $E_n^{tot}(\omega)$ ("err. bound").

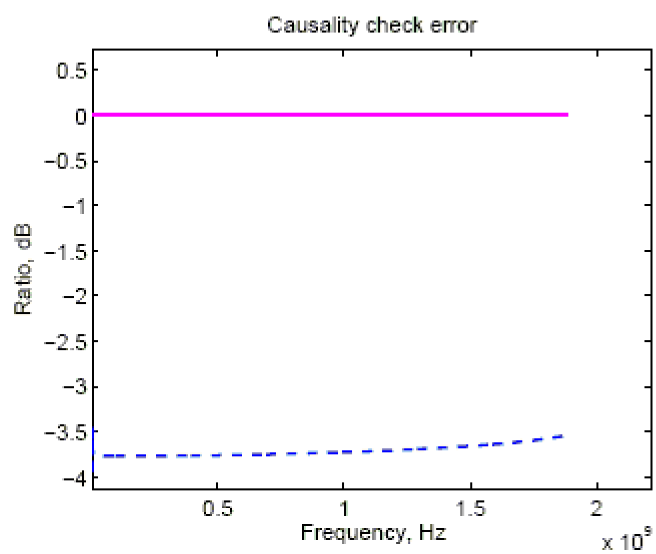


(b) Plot of the ratio $20 \log \left(\frac{|\tilde{\Delta}_n(j\omega)|}{E_n^{tot}(\omega)} \right)$ versus frequency (dashed line). The solid line (0 dB) shows the maximum value the ratio can take for the frequency response to be declared causal.

Figure 7. Causality reports on the frequency response of a transmission line with R_s <> 0 and G_d <> 0, δ_{nc} = 0.002, *n*=14.



(a) Comparison of $|\tilde{\Delta}_n(j\omega)|$ ("Reconst. Err.") versus $E_n^{tot}(\omega)$ ("err. bound").



(b) Plot of the ratio $20 \log \left(\frac{|\tilde{\Delta}_n(j\omega)|}{E_n^{tot}(\omega)} \right)$ versus frequency (dashed line). The solid line (0 dB) shows the maximum value the ratio can take for the frequency response to be declared causal.

Figure 8. Causality reports on S_{12} of the Nexxim causal W-element transmission line with R_s <> 0 and G_d <> 0, δ_{nc} = 0.002, *n*=14.

Causality References

[1] P. Triverio and S. Grivet-Talocia, "A Robust Causality Verification Tool for Tabulated Frequency Data," in 10th IEEE Workshop on Signal Propagation.

[2] P. Triverio and S. Grivet-Talocia, “On checking Causality of bandlimited sampled frequency responses,” in 2nd conference on PhD. Research in Microelectronics and Electronics (Prime), Otranto (LE), Italy, June 12-15, 2006, pp. 501-504.

[3] N. M. Nussenzveig, *Causality and Dispersion Relations*, Academic Press, 1972.

[4] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, “Stability, causality, and passivity in electrical interconnect models,” *IEEE Trans. Advanced Packaging*, Vol. 30, No. 4, Nov. 2007.

[5] P. Triverio and S. Grivet-Talocia, “Robust causality characterization via generalized dispersion relations,” *IEEE Trans. Advanced Packaging*, Vol. 31, No. 3, Aug. 2008.

[6] M. Abromowitz and I. A. Stegun, *Handbook of Mathematical Functions*, New York: Dover, 1968.

[7] S. Asgari, S. N. Lalgudi, and M. Tsuk, “Analytical integration-based causality checking of tabulated S-parameters,” *IEEE Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)"*., Oct 2010.

[8] B. Gustavson, “Rational approximation of frequency domain responses by vector fitting,” *IEEE Trans. on Power Delivery*, Vol. 14, No. 3, July, 1999.

[9] S. Lalgudi, S. Asgari, and M. Tsuk, “Improved procedure to test causality of tabulated S-parameters,” Submitted for review *IEEE Conference on Electrical Performance of Electronic Packaging (EPEPS)"*., Oct. 2012.

Passivity Checking and Enforcement

A passive system does not contain any energy sources. Frequency-dependent data may nevertheless exhibit non-passive behavior. Nexxim offers four methods for passivity checking and enforcement: an iterated fitting method, an iterated fitting method for better Z-fit at DC and low frequencies, a convex programming method, and a perturbation method.

Iterated Fitting Method of Passivity Violations

Iterated Fitting of Passivity Violations (IFPV) is the default passivity enforcement algorithm provided by Nexxim (**s_element.enforce_passivity=7**). IFPV requires relatively little memory to run, and runs significantly faster than either convex optimization or passivity by perturbation. However, IFPV does not guarantee goodness-of-fit as does the convex optimization method. The IFPV algorithm works by taking the results of the original state-space fit, selecting those areas that have passivity violations, and fitting just the excess values, with the original poles. The fitted residues are then subtracted on the original state-space fit. This step is iterated, balancing between goodness of fit and passivity, until a passive model is achieved.

Iterated Fitting of Passivity Violations Low Frequency

Iterated Fitting of Passivity Violations Low Frequency (IFPVLF) is the latest passivity enforcement algorithm (**s_element.enforce_passivity=8**). It builds upon the default passivity enforcement algorithm (Iterated Fitting of Passivity Violation) by providing a better fit to "Z" at DC and low frequencies.

In a dynamic link simulation, Circuit receives the calculated Scattering Parameter data (S-data) on the field solver. Circuit performs state-space fitting, which generates a macromodel that represents the input/output behavior of the structure modeled in the field solver, and performs time or frequency domain simulations. In many applications such as power delivery networks, Circuit must fit the S-data while also preserving its fit of the corresponding Z-data. This fitting process is called "Z-fitting".

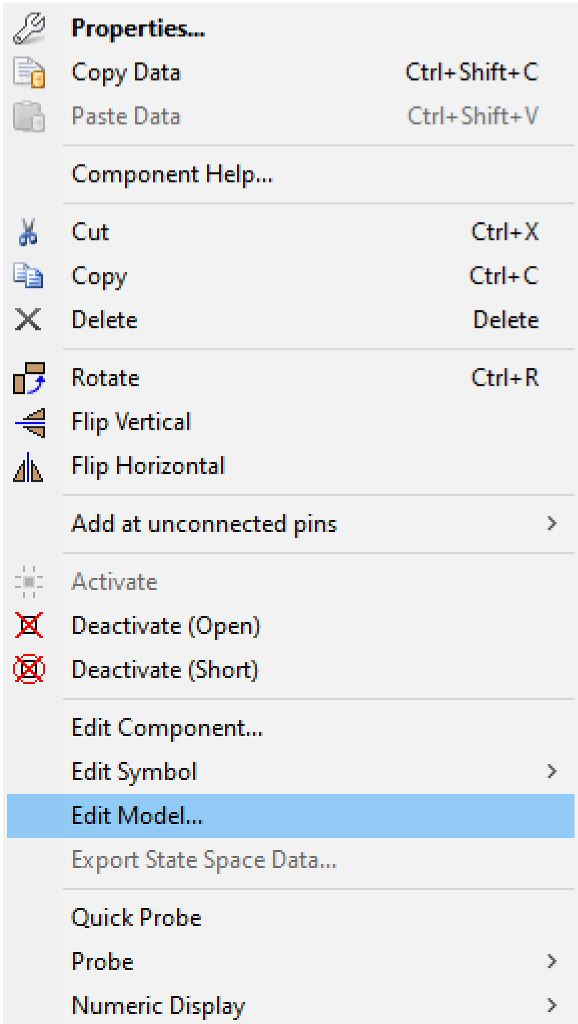
Often, the generated macromodel is not passive. When passivity enforcement is invoked using the default Iterated Fitting of Passivity Violation (IFPV) method, there is no guarantee that the Z-fit to DC and low frequencies are preserved. Preserving the fit to Z-data at DC and low-frequency can be very important. For example, due to inaccuracies of the fit to Z-data at DC and low frequencies, a DC-DC buck converter may not converge to a correct steady state voltage regulated value. To remedy this issue, Circuit offers a DC/low frequency Z-fitting passivity enforcement called Iterated Fitting of Passivity Violation at Low Frequency. IFPVLF provides advanced passivity protection to preserve Z-fit at the DC point, or a better fit to Z-data at DC and low frequencies, while remaining passive. Ideally, IFPVLF provides a significant improvement when the S-data is casual and passive, and the DC data point should be preserved. Use IFPVLF for passivity enforcement when a better Z-fit at DC and low frequencies are appropriate.

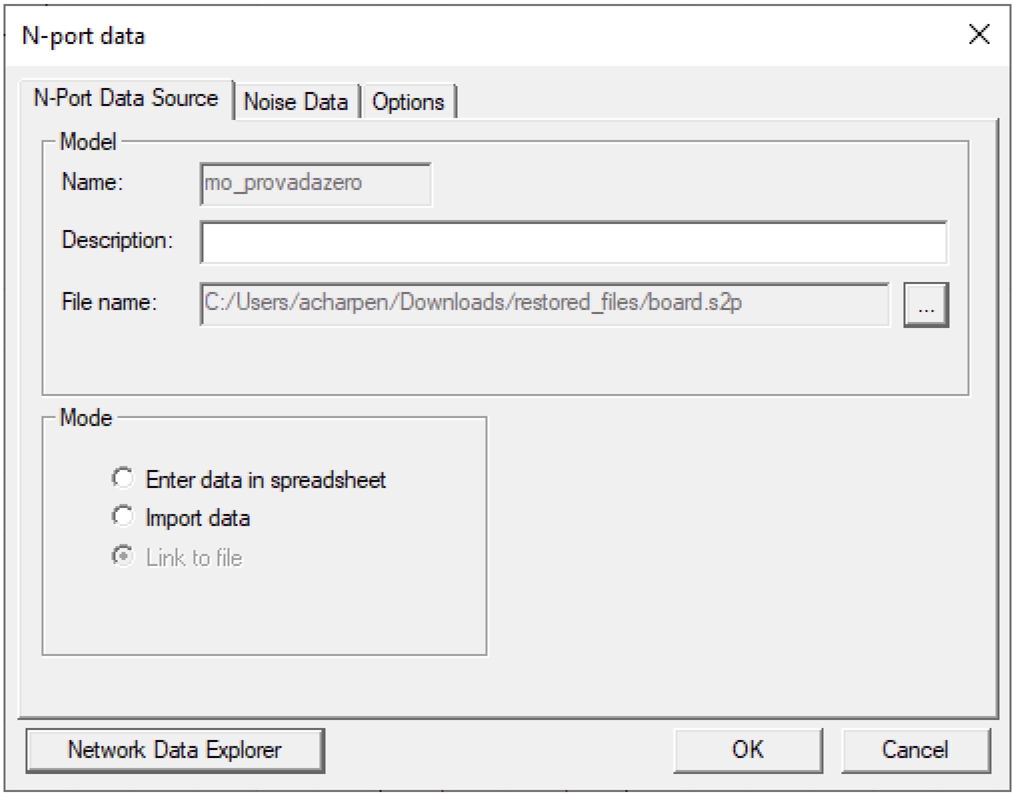
Selecting IFPVLF

To select the IFPVLF algorithm, complete the following steps for either a static or dynamic project.

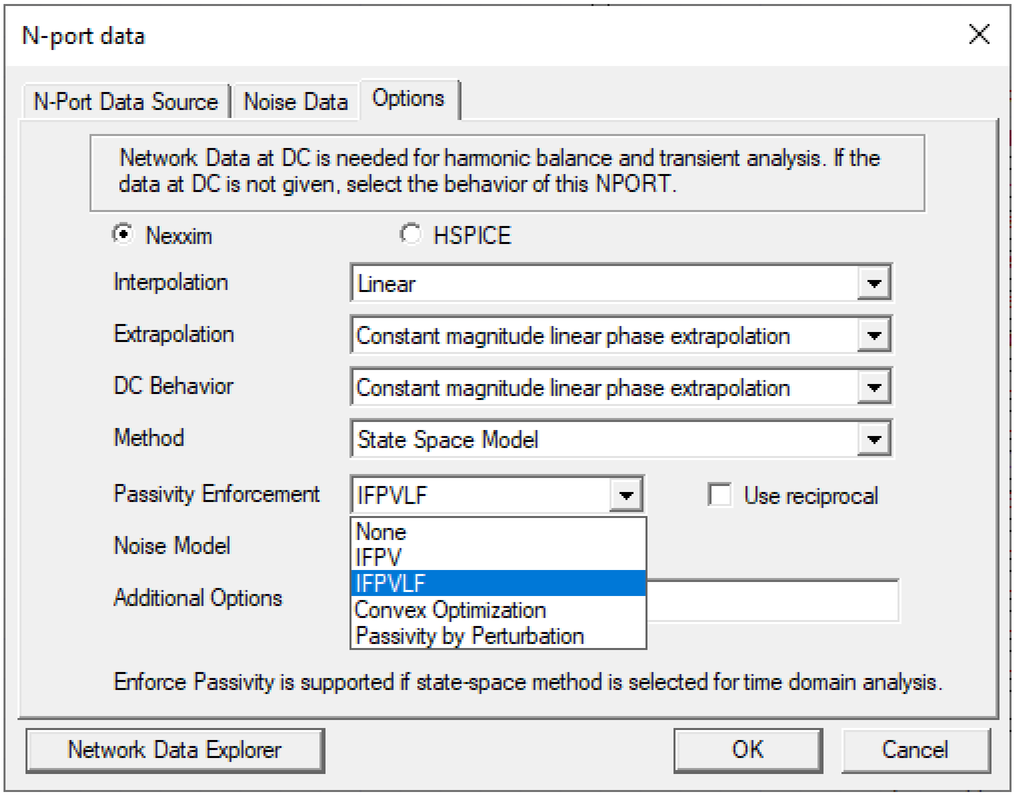
Static Project

1. From the **Schematic Editor**, right-click the model and select **Edit Model...** to open the **N-port data** window.


Properties...
Copy Data Ctrl+Shift+C
Paste Data Ctrl+Shift+V
Component Help...
Cut Ctrl+X
Copy Ctrl+C
Delete Delete
Rotate Ctrl+R
Flip Vertical
Flip Horizontal
Add at unconnected pins
Activate
Deactivate (Open)
Deactivate (Short)
Edit Component...
Edit Symbol
Edit Model...
Export State Space Data...
Quick Probe
Probe
Numeric Display



2. Click the **Options** tab.
3. From the **Passivity Enforcement** drop-down menu, select **IFPVLF**.

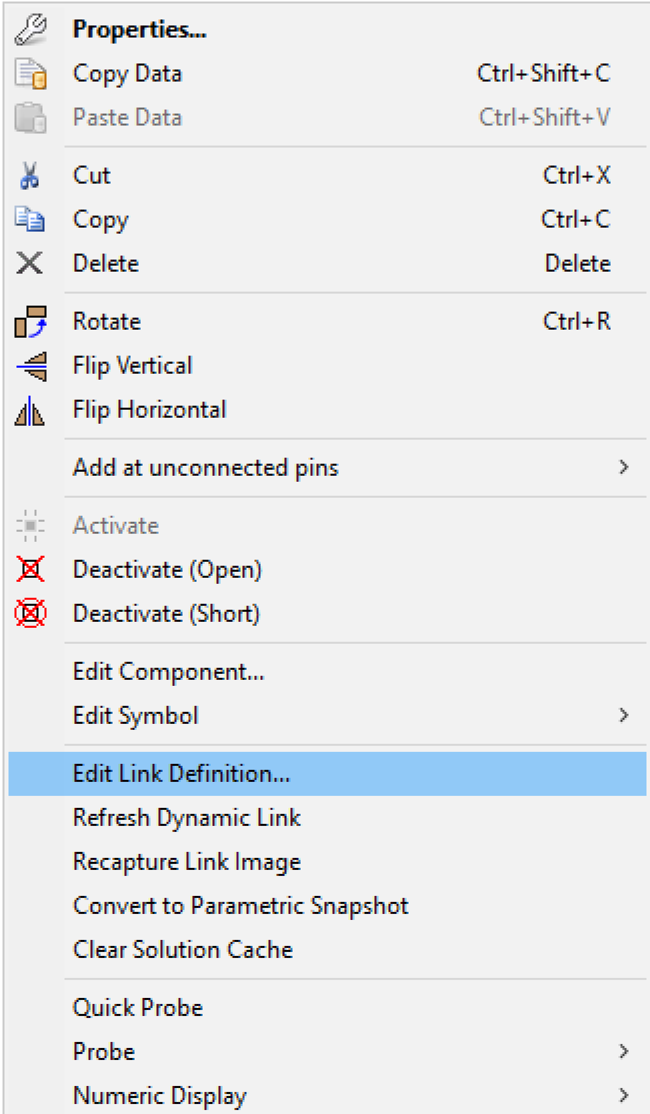


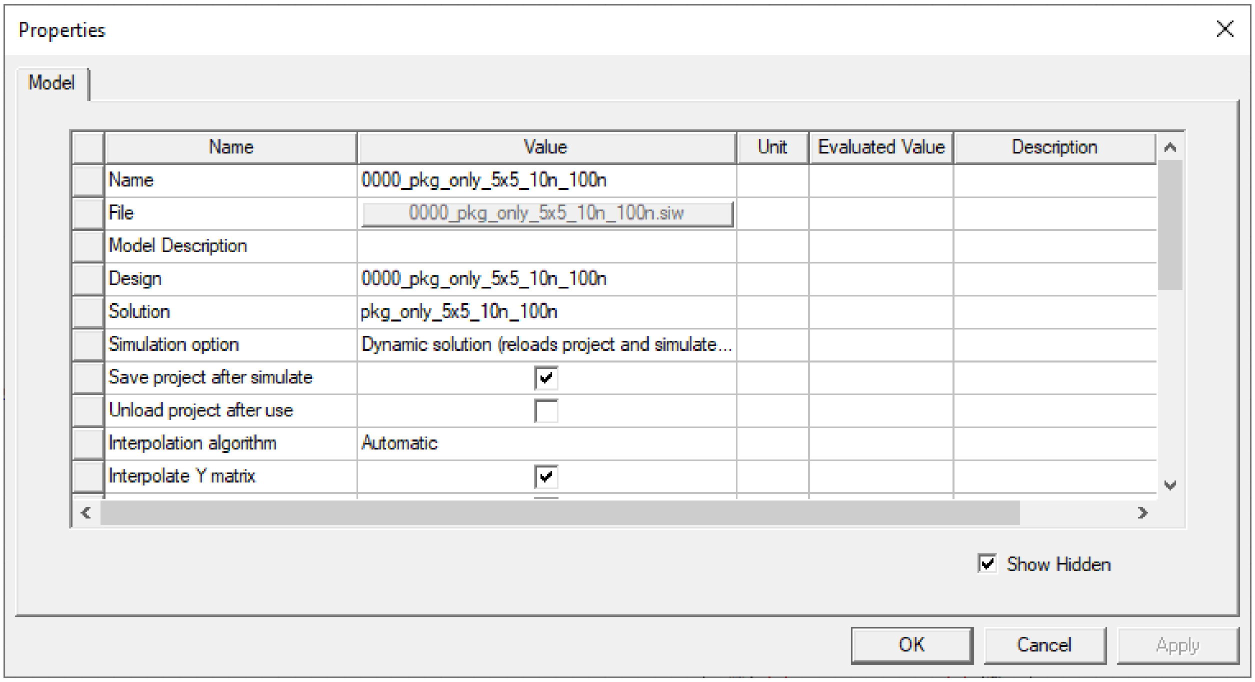
4. Click **OK**.

The procedure is complete.

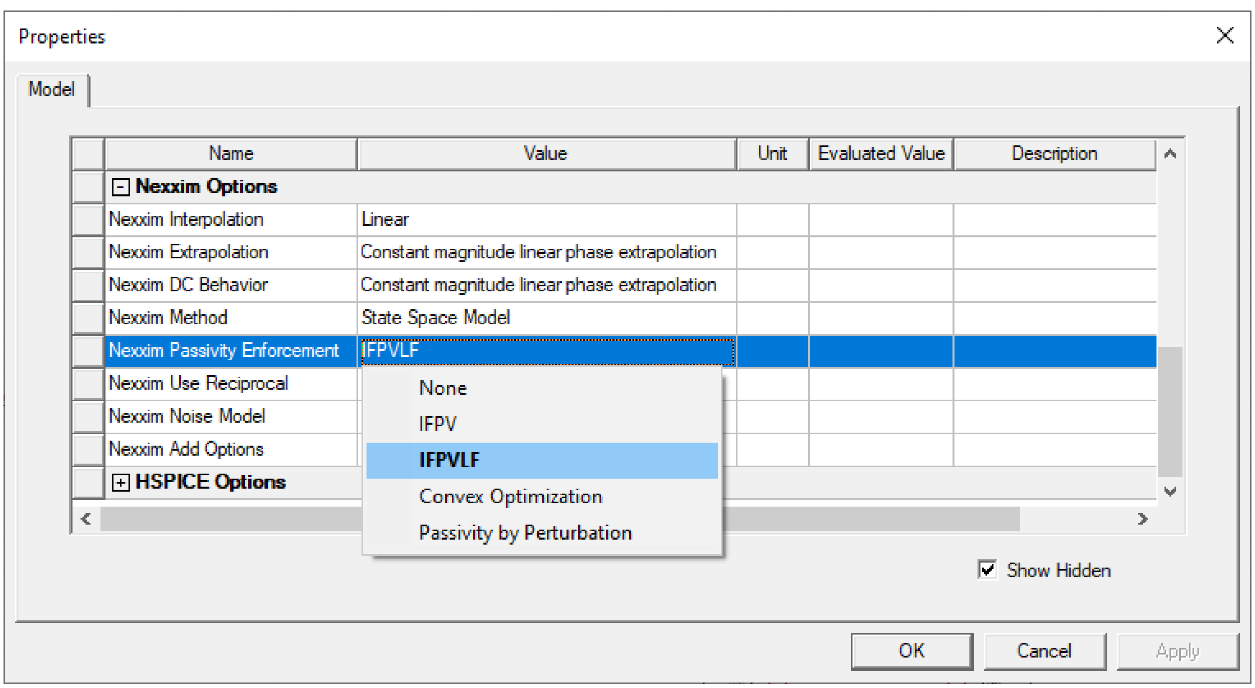
Dynamic Project

1. From the **Schematic Editor**, right-click in the model and select **Edit Link Definition...** to open the **Properties** window.


Properties...
Copy Data Ctrl+Shift+C
Paste Data Ctrl+Shift+V
Cut Ctrl+X
Copy Ctrl+C
Delete Delete
Rotate Ctrl+R
Flip Vertical
Flip Horizontal
Add at unconnected pins
Activate
Deactivate (Open)
Deactivate (Short)
Edit Component...
Edit Symbol
Edit Link Definition...
Refresh Dynamic Link
Recapture Link Image
Convert to Parametric Snapshot
Clear Solution Cache
Quick Probe
Probe
Numeric Display



2. Scroll down and expand **Nexxim Options**.
3. Click in the **Nexxim Passivity Enforcement Value** field and select **IFPVLF** on the drop-down menu.

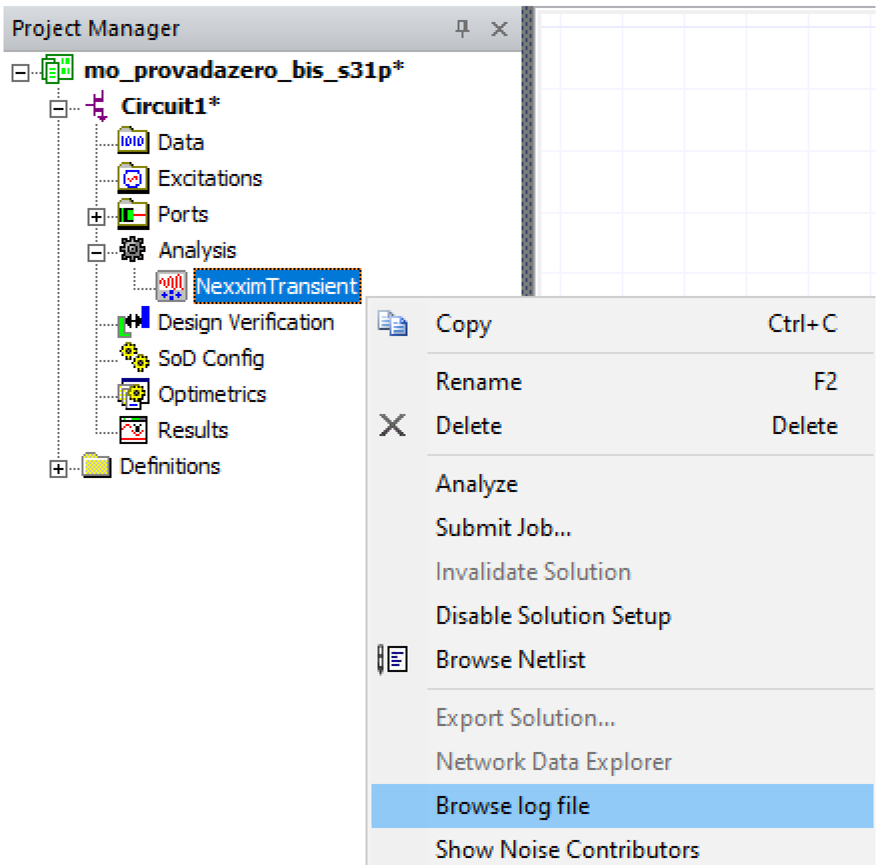


4. Click **OK**.

The procedure is complete.

Viewing IFPVLF Algorithm Status

Once analysis has begun, view the status of the advanced DC passivity protection on the log file. From the **Project Manager** window, expand the **Project Tree** > [*active design folder*] > **Analysis**. Then right-click the transient analysis icon and select **Browse log file** to open the log file in the operating system's default text editor.



```
vpulse_source                                  1
(status): License checkout took 5.502 seconds.
models:vpulse_source(warning): vvoltagepulsetime1_1 - period ( = 5e-09) i
models:s_element(status): s1 - State-space system file 'D:/OneDrive - ANS
models:s_element(status): s1 - Fitting state-space system using FastFit.
(status): DC fitting is enforced! Final DC fitting error:    6.0572e-14

models:s_element(status): s1 - Wrote state-space system 'D:/OneDrive - AN
models:s_element(status): s1 - Final error: 0.006041
models:s_element(status): s1 - Fit is passive.
models:s_element(status): s1 - Final model order: 1364
(info): Starting analysis type TRAN.
analysis(status): linear circuit detected.
analysis:dc(status): Total DC newton iterations = 4, DC conv = 1.
```

Advanced DC Passivity Protection in Action

Advanced DC Passivity Protection preserves DC fit in scenarios where the fit is destroyed during enforced passivity. A sequence of alternate projections is performed between the sets of all state-space fits. The accurate DC fits and passive fits finally converge to a fit that is both accurate and passive (i.e., fixed point). Advanced DC Passivity Protection utilizes the Anderson Acceleration (AA) Algorithm for fixed-point iterations to quickly reach the intersection of the best DC and passive fit. The resulting macromodel should possess an accurate DC fit that is several magnitudes better than the initial fit, and is also passive.

> **Note:** The AA Algorithm is used to speed up the iterative process of alternating projections between the sets of accurate DC fits and passive fits. The overall fit is not constrained during this process. In some rare situations, the overall fit worsens beyond tolerance. In such scenarios, IFPVLF automatically deactivates the algorithm and chooses the best fitting that meets the error tolerance before launching the DC fit preservation algorithm.

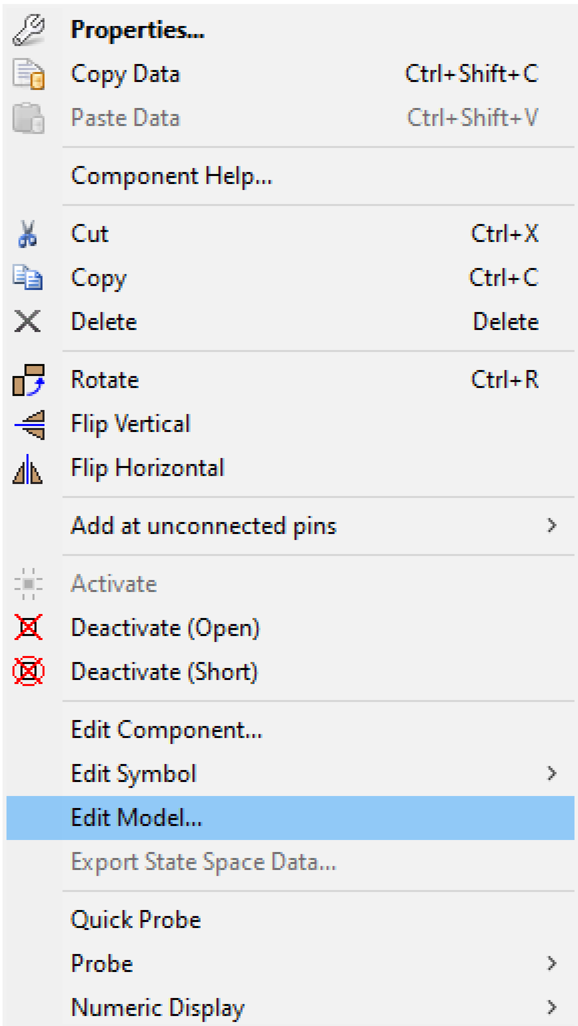
```
(status): License checkout took 0.887 seconds.
models:s_element(warning): s1 - Passivity enforcement requested for S-element during frequency-domain analysis. Switching to state-space modeling.
models:s_element(status): s1 - State-space system file 'D:/OneDrive - ANSYS, Inc/Worklog/features/FEATURE451710/QA testcase/mo_provadazero_ter_sweepstretto_s
models:s_element(status): s1 - Fitting state-space system using FastFit.
(status): DC fitting is enforced! Fitting error target could not be acheived! Choose best fitting of IFPVLF, Final DC fitting error:    0.0004

models:s_element(status): s1 - Wrote state-space system 'D:/OneDrive - ANSYS, Inc/Worklog/features/FEATURE451710/QA testcase/mo_provadazero_ter_sweepstretto_
models:s_element(status): s1 - Final error: 0.008434
models:s_element(warning): s1 - Fit is not passive (maximum passivity violation is 9.0922e-10). Results can be unstable or inaccurate. If results seem suspec
models:s_element(status): s1 - Final model order: 2077
(info): Starting analysis type LNA.
analysis:dc(status): Total DC newton iterations = 4, DC conv = 1.
(status): LNA cpu time = 0.46
(info): Simulation succeeded. Total simulator time: 0:01:20
```

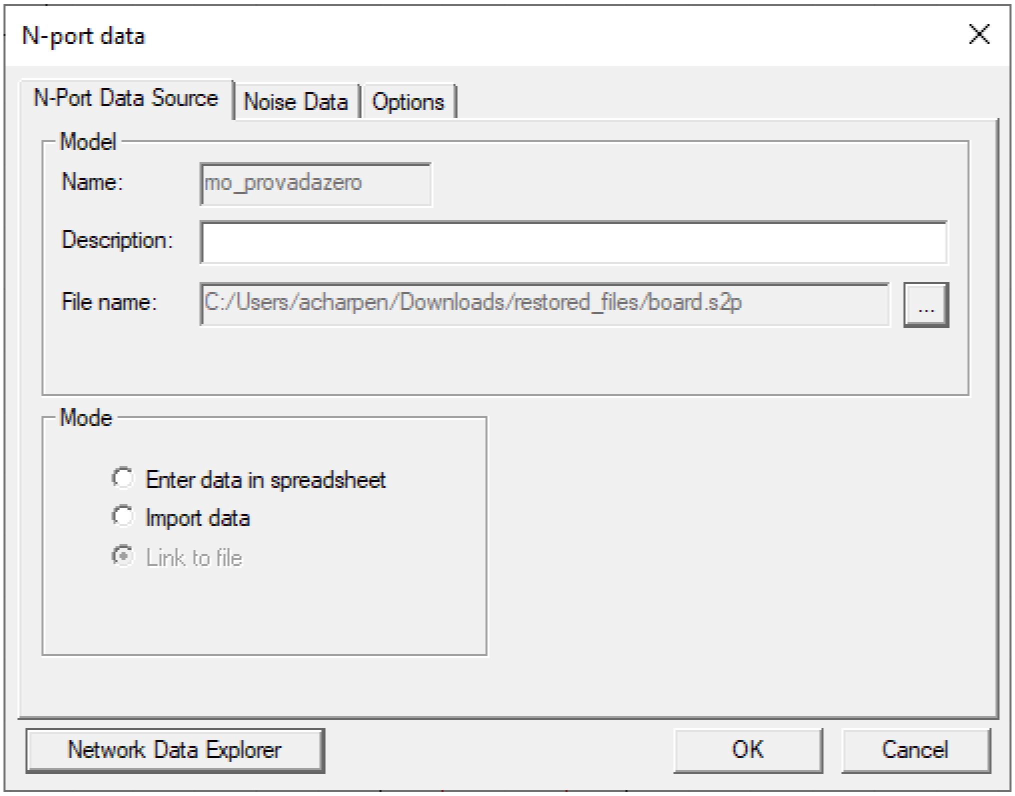
Deactivating Advanced DC Passivity Protection

To deactivate advanced DC passivity protection, complete the following steps for either a static or dynamic project.

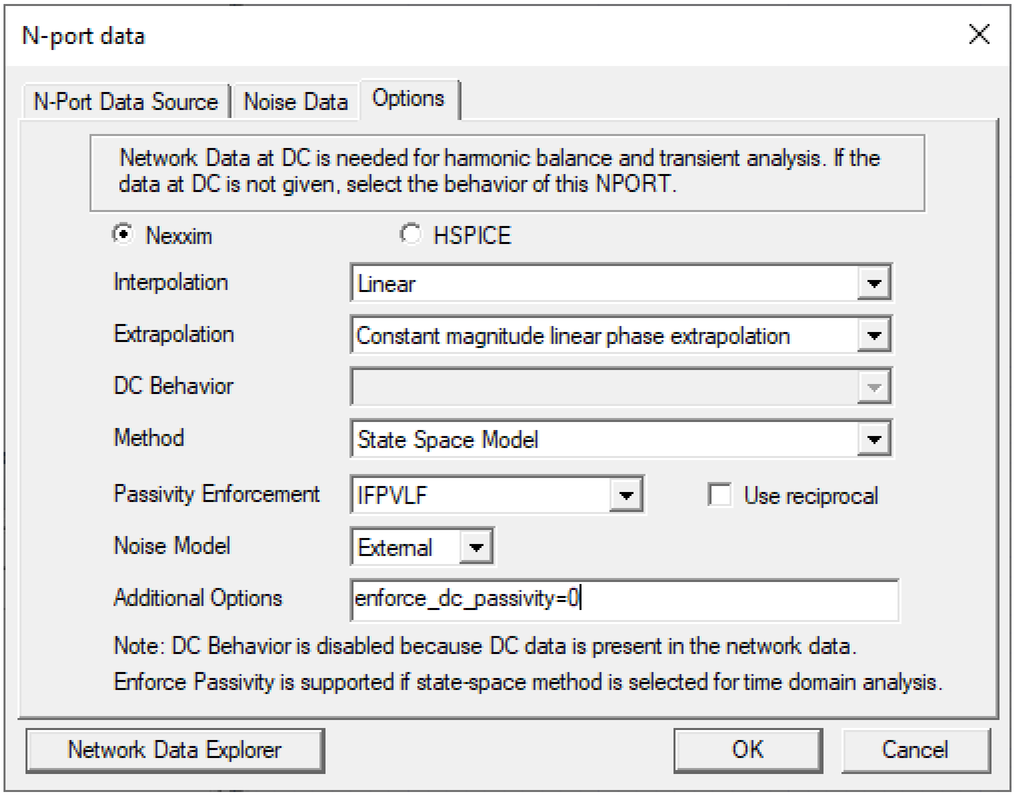
Static Project

1. From the **Schematic Editor**, right-click the model and select **Edit Model...** to open the **N-port data** window.





2. Click the **Options** tab.
3. Click in the **Additional Options** field and type [**enforce_dc_passivity=0**].

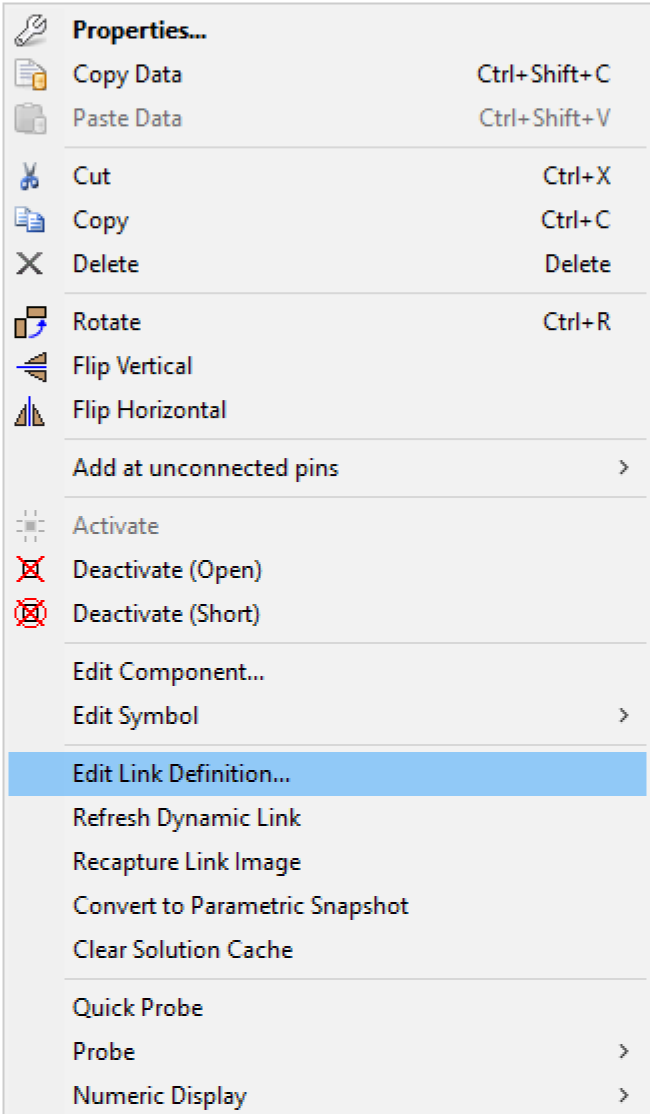


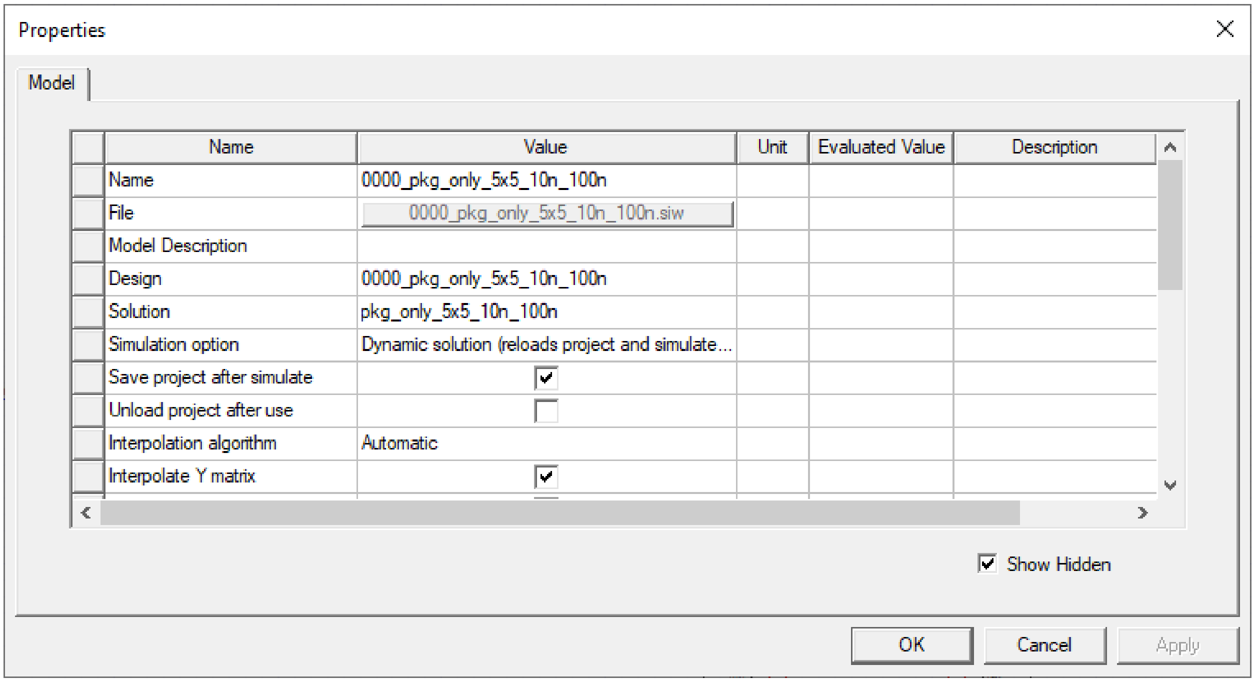
4. Click **OK**.

The procedure is complete.

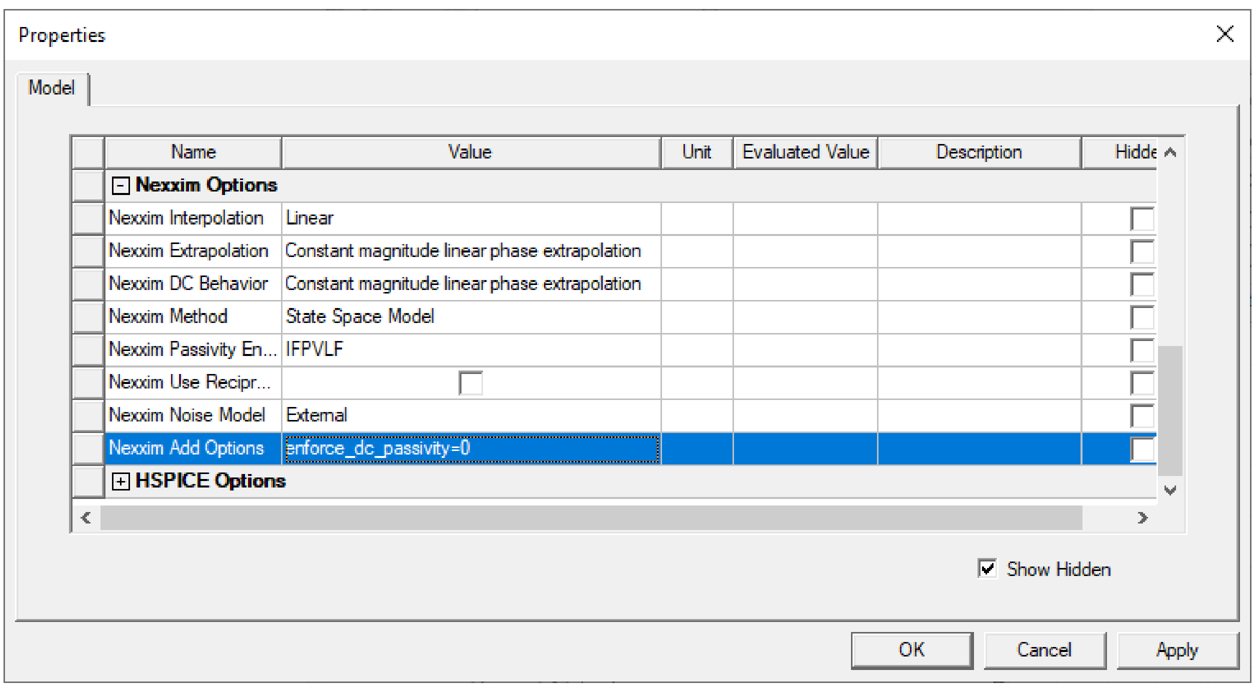
Dynamic Project

1. From the **Schematic Editor**, right-click in the model and select **Edit Link Definition...** to open the **Properties** window.


Properties...
Copy Data Ctrl+Shift+C
Paste Data Ctrl+Shift+V
Cut Ctrl+X
Copy Ctrl+C
Delete Delete
Rotate Ctrl+R
Flip Vertical
Flip Horizontal
Add at unconnected pins
Activate
Deactivate (Open)
Deactivate (Short)
Edit Component...
Edit Symbol
Edit Link Definition...
Refresh Dynamic Link
Recapture Link Image
Convert to Parametric Snapshot
Clear Solution Cache
Quick Probe
Probe
Numeric Display



2. Scroll down and expand **Nexxim Options**.
3. Click in the **Nexxim Add Options** field and type [**enforce_dc_passivity=0**].



4. Click **OK**.

The procedure is complete.

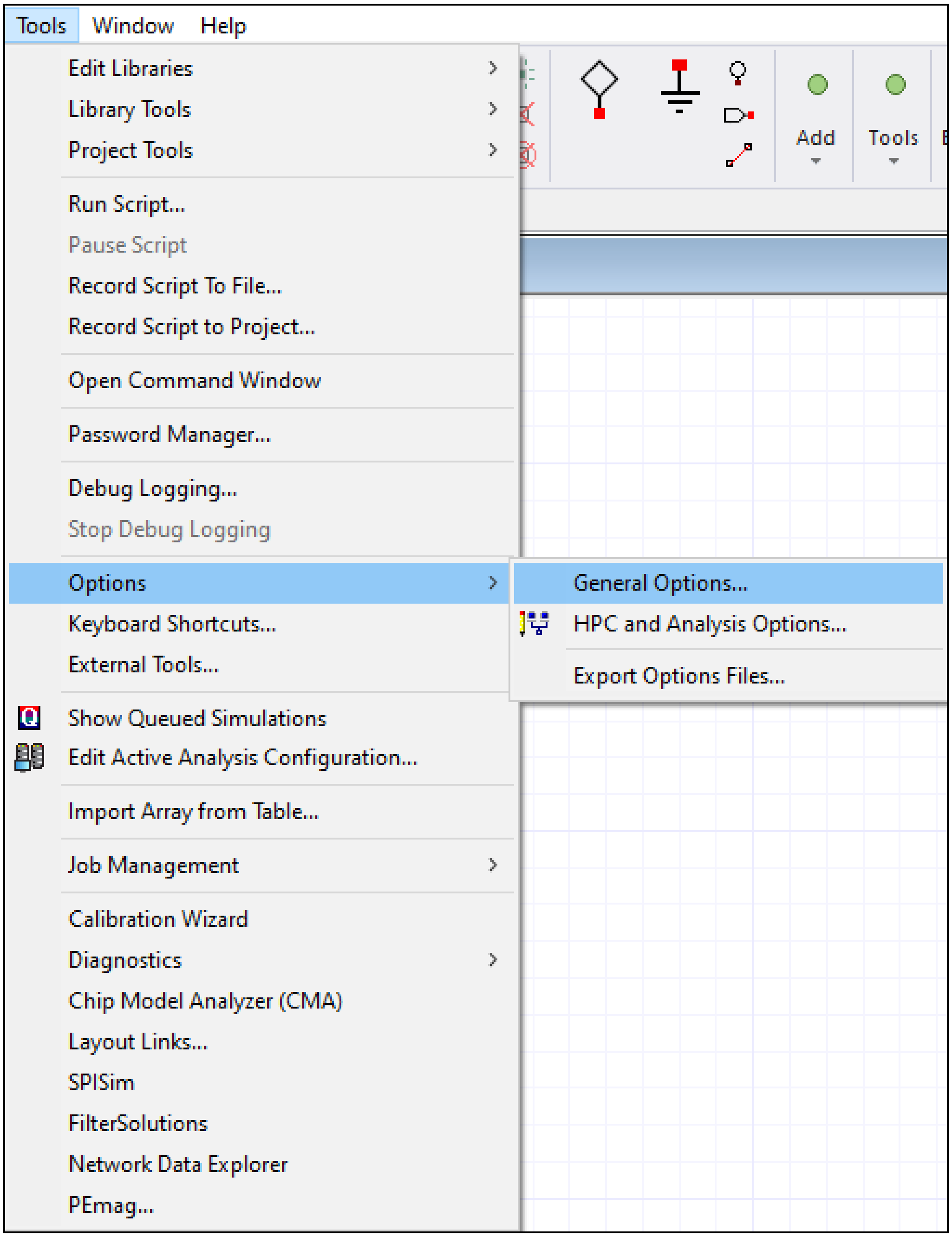
Augmented Data Passivity Enforcement (Beta Feature)

The **Augmented Data Passivity Enforcement (ADPE)** feature improves the fitting error of a passive macromodel. After performing state-space fitting on S-Parameter data (S-data), ADPE augments the original S-data with new causal and passive data to improve the fitting experience. If the fitting error is improved, the macromodel created from the augmented S-data is reported to the user. If not, **the fit to the** original S-data is reported instead. ADPE is only accessible if Nexxim's default state-space fitting method is utilized (i.e., **FastFit**) and either Iterated Fitting of Passivity Violations (IFPV) or Iterated Fitting of Passivity Violations Low Frequency (IFPVLF) is selected as the passivity enforcement algorithm.

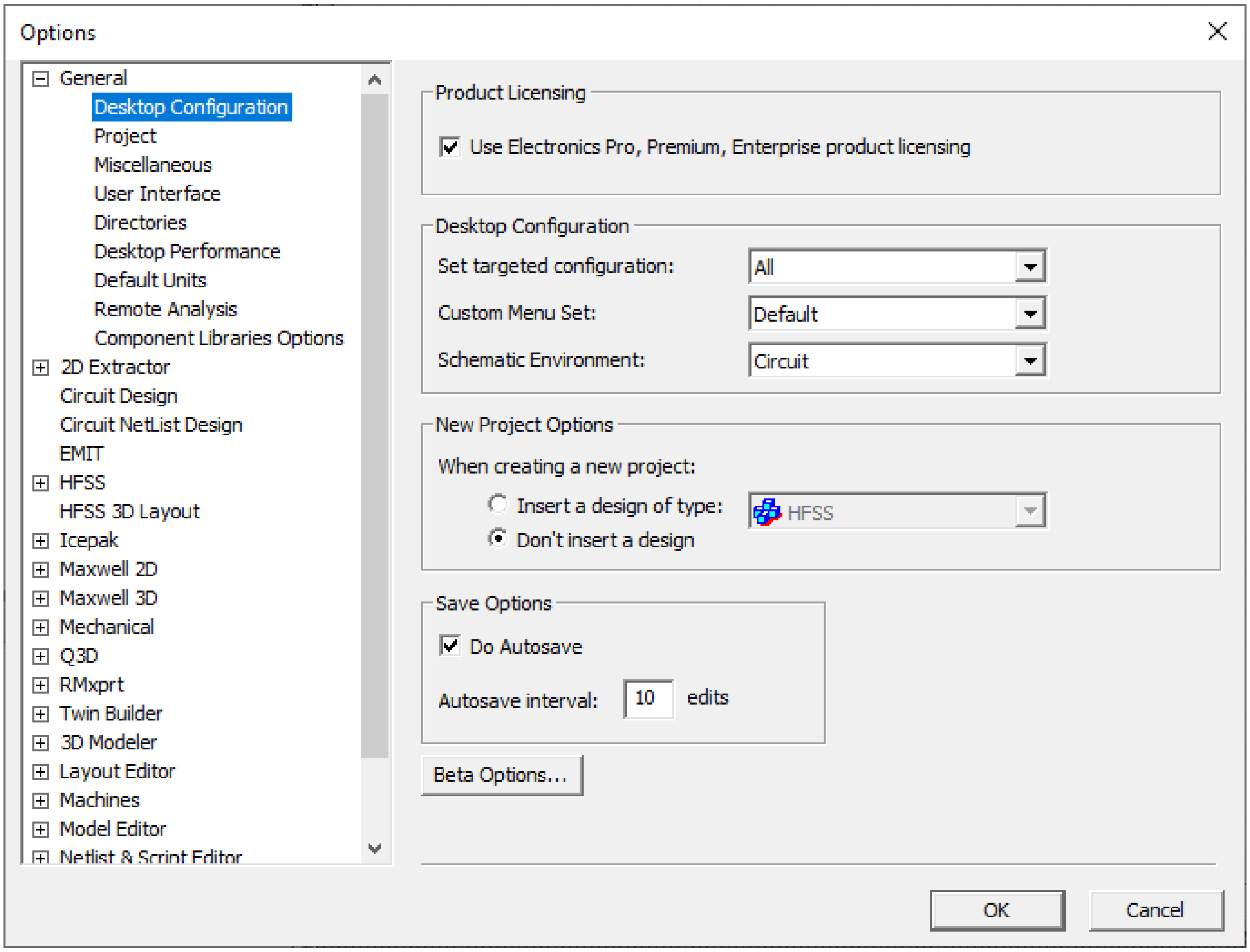
Activating the Augmented Data Passivity Enforcement Feature

Complete these steps to activate the **Augmented Data Passivity Enforcement** beta feature.

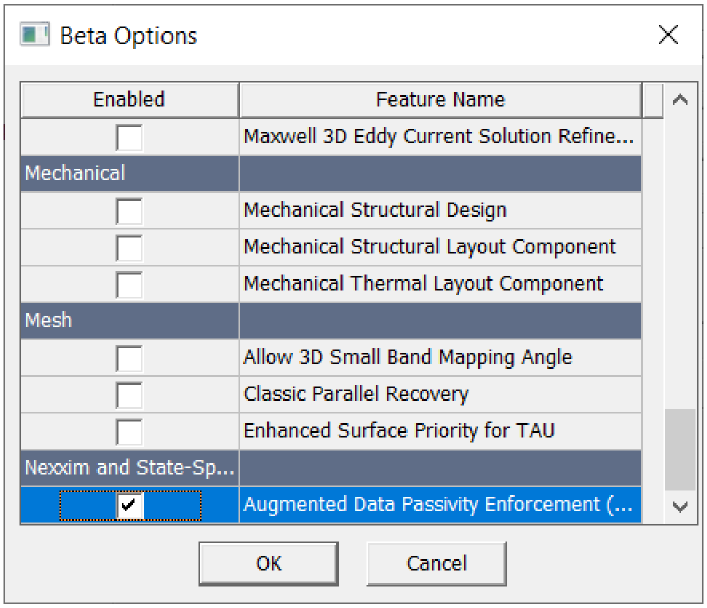
1. From **Tools**, select **Options** > **General Options...** to open the **Options** window.



2. From the **Options** window, select **General** > **Desktop Configuration** from the navigation tree. Then click **Beta Options...** to open the **Beta Options** window.



3. From the **Beta Options** window, check the box adjacent to **Augmented Data Passivity Enforcement**. Then select **OK** to close the **Beta Options** window and activate **Augmented Data Passivity Enforcement**.



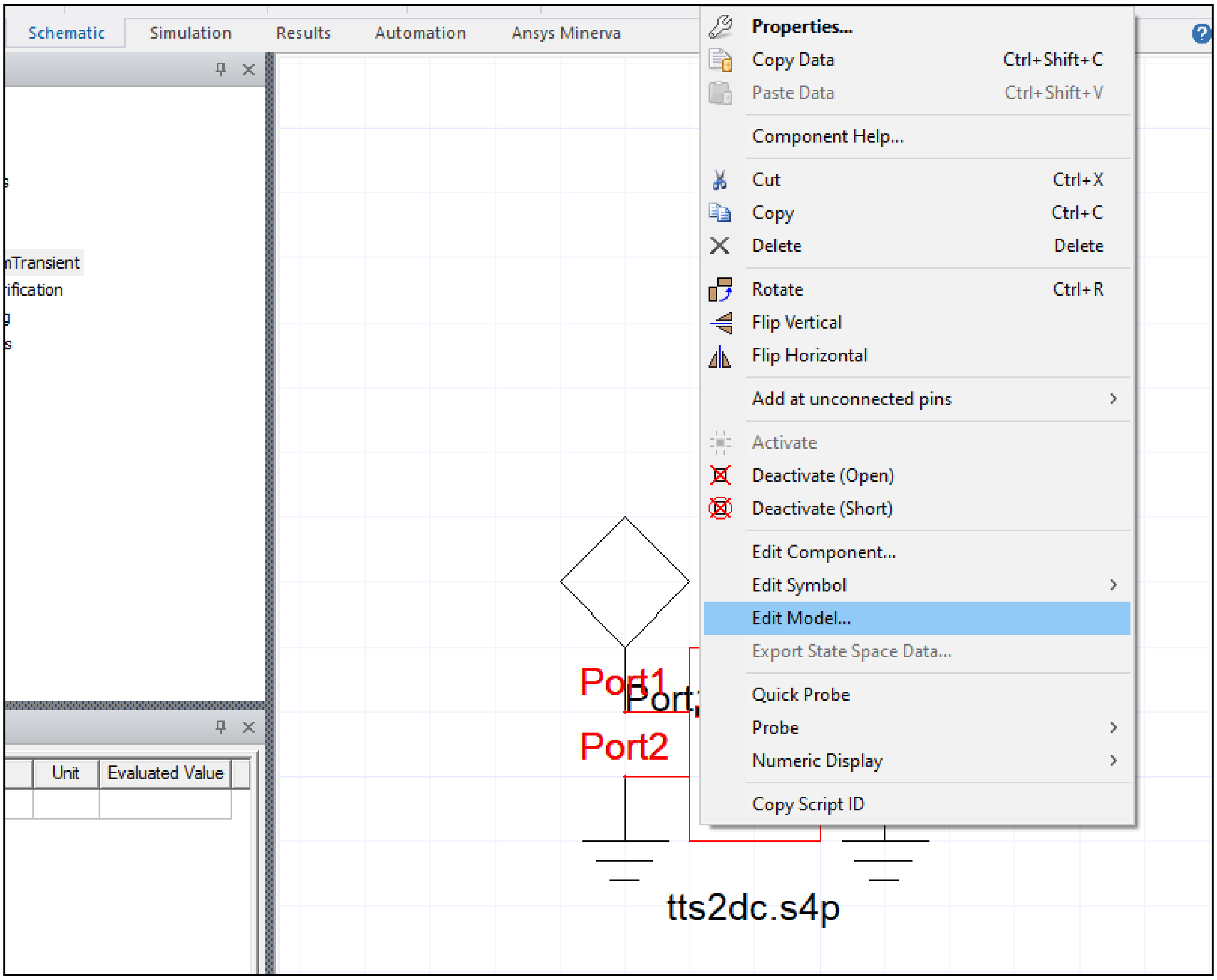
4. Select **OK** to close the **Options** window.

The procedure is complete.

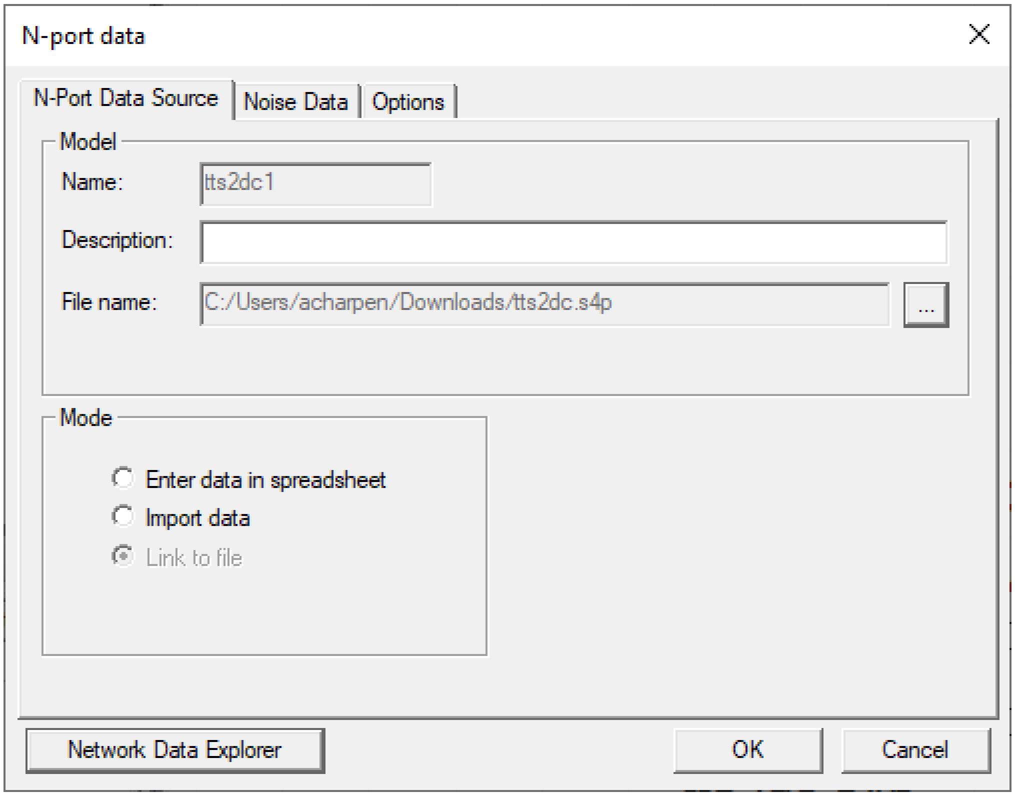
Configuring N-Port Data for ADPE

Complete these steps to configure **N-Port Data** settings so the **Augmented Data Passivity Enforcement** feature functions.

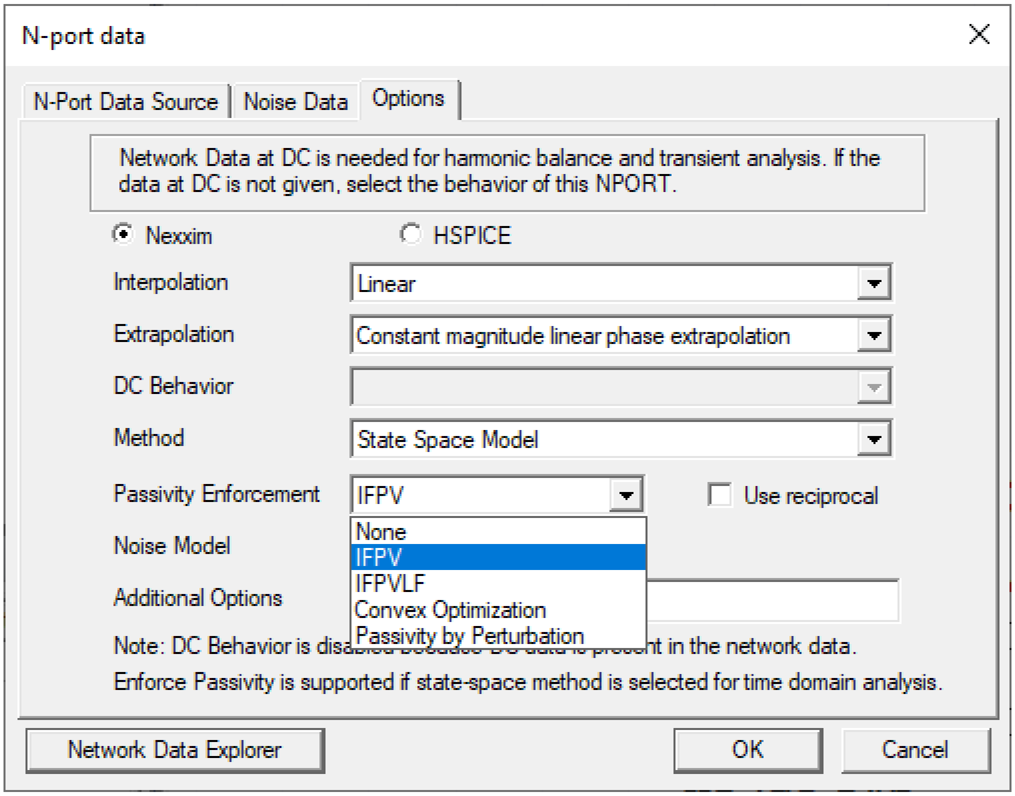
1. From the **Schematic** tab, right-click the model and select **Edit Model...** to open **N-port data** window.


Schematic
Simulation
Results
Automation
Ansys Minerva
nTransient
ification
Unit
Evaluated Value
Properties...
Copy Data Ctrl+Shift+C
Paste Data Ctrl+Shift+V
Component Help...
Cut Ctrl+X
Copy Ctrl+C
Delete Delete
Rotate Ctrl+R
Flip Vertical
Flip Horizontal
Add at unconnected pins
Activate
Deactivate (Open)
Deactivate (Short)
Edit Component...
Edit Symbol
Edit Model...
Export State Space Data...
Quick Probe
Probe
Numeric Display
Copy Script ID
Port1
Port2
tts2dc.s4p

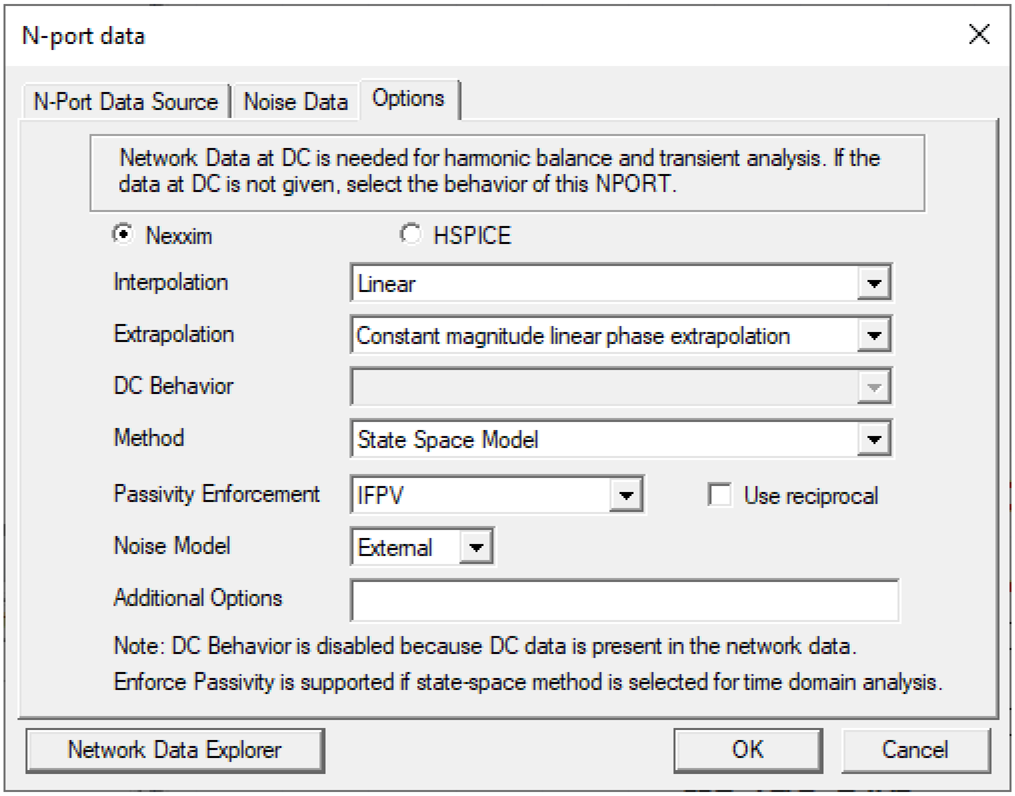
2. Navigate to the **Options** tab.



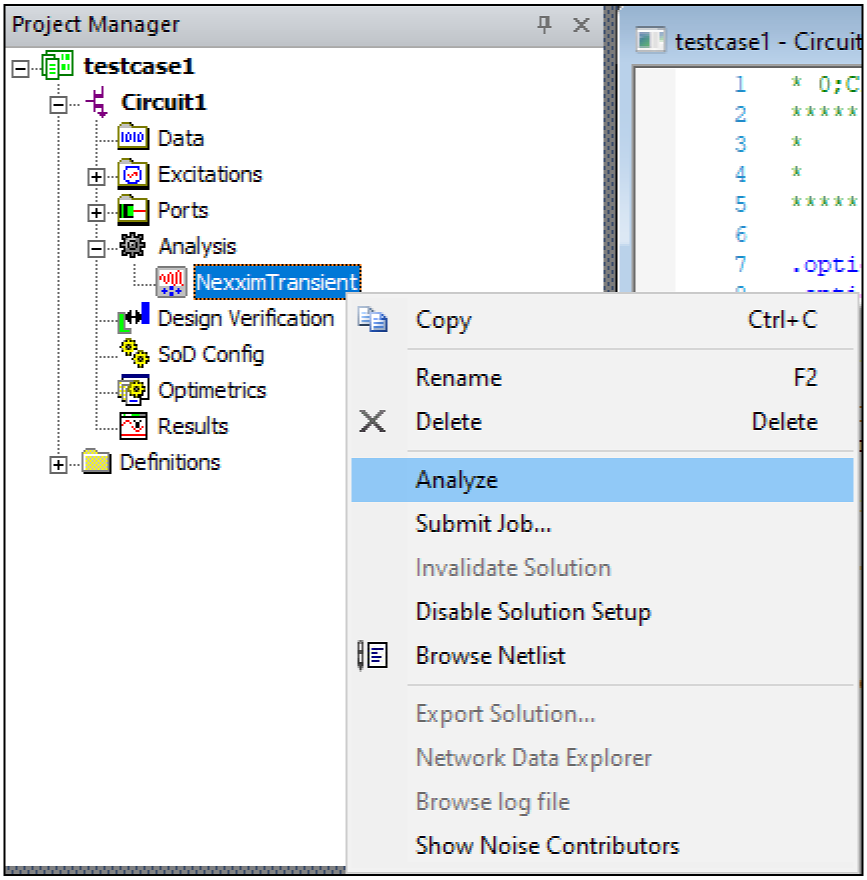
3. From the **Options** tab, select either **IFPV** (i.e., Iterated Fitting of Passivity Violations) or **IFPVLF** (i.e., Iterated Fitting of Passivity Violations Low Frequency) from the **Passivity Enforcement** drop-down menu. ADPE **only** functions with **IFPV** or **IFPVLF** selected.


N-port data
N-Port Data Source
Noise Data
Options
Network Data at DC is needed for harmonic balance and transient analysis. If the data at DC is not given, select the behavior of this NPORT.
Nexxim
HSPICE
Interpolation
Linear
Extrapolation
Constant magnitude linear phase extrapolation
DC Behavior
Method
State Space Model
Passivity Enforcement
IFPV
Use reciprocal
None
IFPV
IFPVLF
Convex Optimization
Passivity by Perturbation
Noise Model
Additional Options
Note: DC Behavior is dis
nt in the network data.
Enforce Passivity is supported if state-space method is selected for time domain analysis.
Network Data Explorer
OK
Cancel

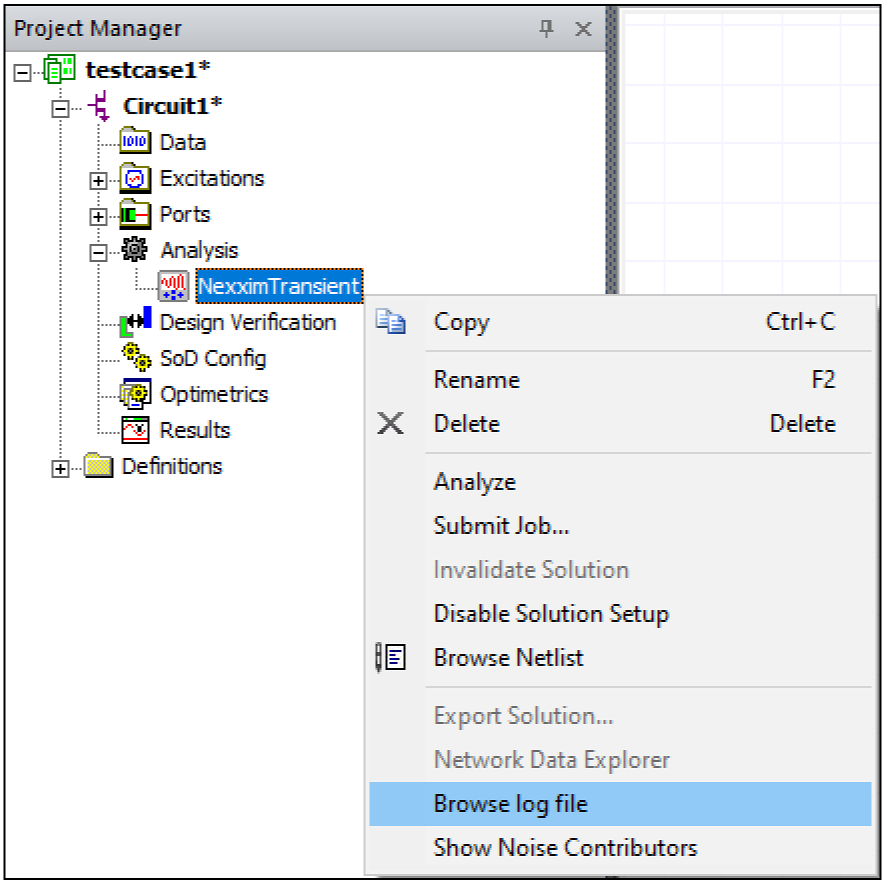
4. Click **OK** to close the **N-port data** window.



5. From the **Project Manager** window, right click the **NexximTransient analysis** and select **Analyze** to solve the analysis with the **Augmented Data Passivity Enforcement** feature assisting.



6. After analysis completes, right-click the **Nexxim Transient** analysis and select **Browse log file** to open the log file (e.g., **DV7_S5_V17.cir.log**).



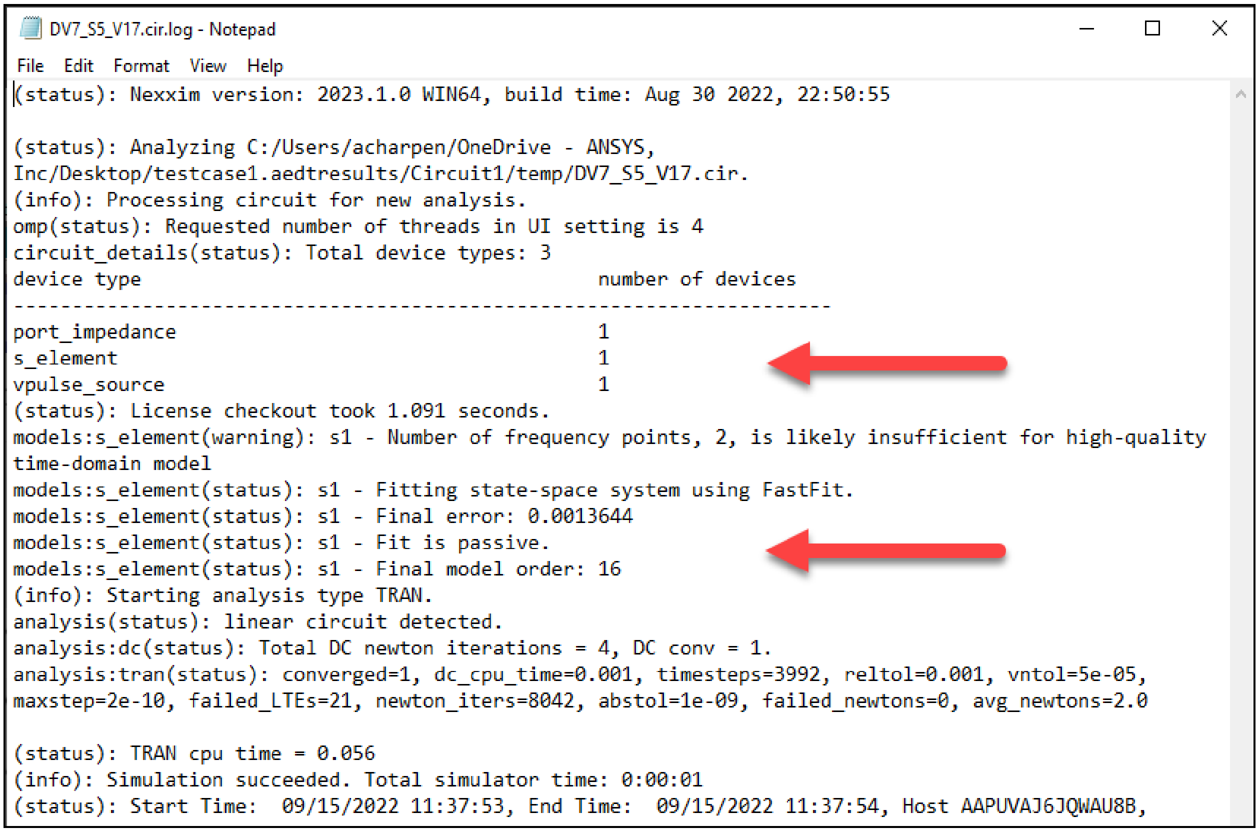
In the following example, the ADPE feature is inactive. As a result, the fitting error is unacceptable.

DV7_S5_V17.cir.log - Notepad

File Edit Format View Help

```
(status): Nexxim version: 2023.1.0 WIN64, build time: Aug 30 2022, 22:50:55

(status): Analyzing C:/Users/acharpen/OneDrive - ANSYS,
Inc/Desktop/testcase1.aedtresults/Circuit1/temp/DV7_S5_V17.cir.
(info): Processing circuit for new analysis.
omp(status): Requested number of threads in UI setting is 4
circuit_details(status): Total device types: 3
device type                                       number of devices
----------------------------------------------------------------------
port_impedance                                    1
s_element                                         1
vpulse_source                                     1
(status): License checkout took 0.643 seconds.
models:s_element(warning): s1 - Number of frequency points, 2, is likely insufficient for high-quality
time-domain model
models:s_element(status): s1 - Fitting state-space system using FastFit.
models:s_element(error): s1 - Final error: 0.57963; unable to achieve acceptable error.
(status): Reached the limit (1) for maximum number of warning messages in message manager.
(info): See simulation logfile for a complete list of warning messages.
models:s_element(warning): s1 - Number of frequency points, 2, is likely insufficient for high-quality
time-domain model
system_load_interface(error): device s1 (model tts2dc) rejected.
(error): Error found during loading: analysis aborted for C:/Users/acharpen/OneDrive - ANSYS,
Inc/Desktop/testcase1.aedtresults/Circuit1/temp/DV7_S5_V17.cir
(error): Some or all simulation failed for C:/Users/acharpen/OneDrive - ANSYS,
Inc/Desktop/testcase1.aedtresults/Circuit1/temp/DV7_S5_V17.cir. Total simulator time: 0:00:00
(status): Start Time:  09/15/2022 11:17:56, End Time:  09/15/2022 11:17:57, Host AAPUVAJ6JQWAU8B,
```

In the next example, the ADPE feature is active and the same design is analyzed using the same parameters. As a result of the ADPE feature augmenting the S-data, the fitting error is acceptable.



DV7_S5_V17.cir.log - Notepad

File Edit Format View Help

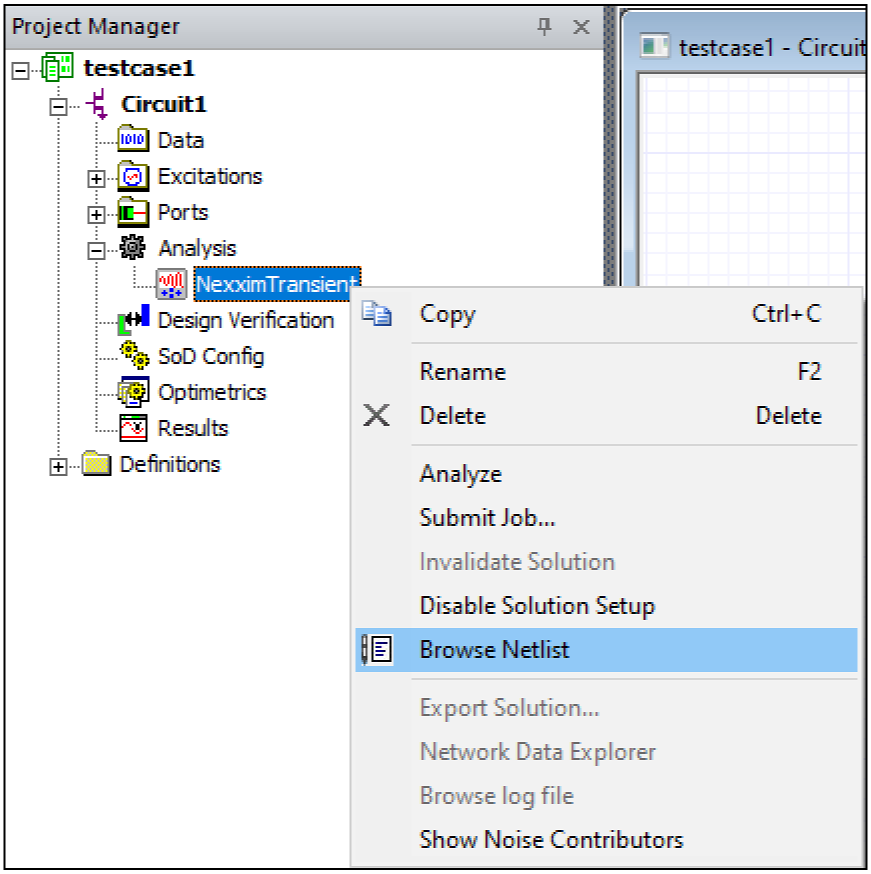
```
(status): Nexxim version: 2023.1.0 WIN64, build time: Aug 30 2022, 22:50:55

(status): Analyzing C:/Users/acharpen/OneDrive - ANSYS,
Inc/Desktop/testcase1.aedtresults/Circuit1/temp/DV7_S5_V17.cir.
(info): Processing circuit for new analysis.
omp(status): Requested number of threads in UI setting is 4
circuit_details(status): Total device types: 3
device type                                        number of devices
----------------------------------------------------------------------
port_impedance                                     1
s_element                                          1
vpulse_source                                      1
(status): License checkout took 1.091 seconds.
models:s_element(warning): s1 - Number of frequency points, 2, is likely insufficient for high-quality
time-domain model
models:s_element(status): s1 - Fitting state-space system using FastFit.
models:s_element(status): s1 - Final error: 0.0013644
models:s_element(status): s1 - Fit is passive.
models:s_element(status): s1 - Final model order: 16
(info): Starting analysis type TRAN.
analysis(status): linear circuit detected.
analysis:dc(status): Total DC newton iterations = 4, DC conv = 1.
analysis:tran(status): converged=1, dc_cpu_time=0.001, timesteps=3992, reltol=0.001, vntol=5e-05,
maxstep=2e-10, failed_LTEs=21, newton_iters=8042, abstol=1e-09, failed_newtons=0, avg_newtons=2.0

(status): TRAN cpu time = 0.056
(info): Simulation succeeded. Total simulator time: 0:00:01
(status): Start Time:  09/15/2022 11:37:53, End Time:  09/15/2022 11:37:54, Host AAPUVAJ6JQWAU8B,
```

The procedure is complete.

Confirming ADPE is Active

Complete these steps to confirm the **Augmented Data Passivity Enforcement** feature is active.

1. Assuming a **Nexxim Transient analysis** has been added to the design, from the **Project Manager** window, expand the **Project Tree** > [*active design folder*] > **Analysis**. Then right-click **NexximTransient** and select **Browse Netlist** to open the **Netlist** window.



2. From the **Netlist** window, search for **enforce_adpe=1**, which indicates that the ADPE feature is active.

```
* 0;Circuit1.sph
*********************************************************************************************************
*                                            Circuit Design netlist
*                              created by Ansys Electronics Desktop                            *
*********************************************************************************************************

.option PARHIER='local'
.option max_messages=1

* begin toplevel circuit
.model tts2dc S TSTONEFILE="D:/OneDrive - ANSYS, Inc/Worklog/features/F633769/Slides/tts2dc.s4p"
+ INTERPOLATION=LINEAR INTDATTYP=MA HIGHPASS=10 convolution=0 enforce_passivity=7 enforce_adpe=1\
Noisemodel=External

S1 Port1 0 0 0 FQMODEL="tts2dc"
RPort1 Port1 inet_100 PORTNUM=1 RZ=50 IZ=0
VVoltagePulseTime1_1 inet_100 0 PULSE(0 5 0 1e-10 1e-10 5e-09 1e-08)
.PORT Port1 0 1 RPort1 NSRC=1

* end toplevel circuit
.OPTIONS s_element.cache_state_space=0
.TRAN 1e-10 2e-08

.end
```

The procedure is complete.

Convex Programming Method for Passivity Enforcement

The convex programming method (**s_element.passivity_enforcement=1**) enforces passivity by constraining the transfer function to be positive real (Passivity Reference [1]).

By the Bounded Real Lemma, the state-space system ABCD is passive if a positive definite matrix **K** >0 exists such that:

$$\begin{bmatrix} -KA - A^T K & -KB & -C^T \\ -B^T K & I & -D^T \\ -C & -D & I \end{bmatrix} > 0$$

The convex programming method preserves the poles from the original state formulation (state matrix **A**). The formulation is such that the estimate of matrix **B** is not needed. For a fixed matrix **A**, the method recomputes matrices **C** and **D** to minimize errors while observing the Bounded Real Lemma.

The convex method of passivity enforcement is guaranteed to produce a passive state-space realization, but is very slow and memory-intensive. Convex optimization requires computing the Hessian or second derivative matrix, with dimension $(P+Q)^2$ by $(P+Q)^2$, where P+Q is the total number of input and output ports. The total memory requirement for the matrices is $(P+Q)^4$. The compute time to solve the matrices is $(P+Q)^6$. In practice, the exponential growth of the solution size limits the convex programming method to systems with the total number of ports (P+Q) less than or equal to 10.

Perturbation Method for Passivity Enforcement

The passivity by perturbation method is selected with (**s_element.passivity_enforcement=6**). With this method, passivity enforcement of scattering parameter data with larger numbers of ports is achieved by the perturbation of the singular values of the transfer function matrix.

The frequency-domain transfer function matrix **H** (jω):

$$\boldsymbol{H}(j\omega) = \boldsymbol{C}(j\omega\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}$$

is obtained on the transfer matrix fit to the S-parameter data *S(j*ω*).*

The singular values of the transfer matrix σ[**H** (jω)] are the square roots of the eigenvalues of [$\mathbf{H}^H$(jω)**H** (jω)], where $\mathbf{H}^H$(jω) denotes the complex conjugate transpose of **H** (jω).

For passivity, the singular values of the transfer matrix must be less than or equal to 1 for all frequencies.

$$max(\sigma_i[H(j\omega)]) \le 1 \qquad \forall\omega$$

The perturbation method is derived on the eigenvalue (modal) perturbation of [$\mathbf{H}^H$(jω)**H** (jω)], equivalent to the perturbation of the singular values of **H** (jω). Higher order terms are ignored. As a first step, the **D** matrix is passified to guarantee asymptotic passivity. Next, linear perturbation is applied iteratively to the **C** matrix, minimizing the norm of the perturbation and the transfer matrix deviation at each iteration.

Generally, a nonpassive singular value exceeds the threshold only over a range of frequencies. The method uses the slope of the curve at the points where the singular value crosses the threshold to determine the direction of the perturbation to be applied.

The perturbation method generates matrices of approximate size *k* * N* (P+Q), where N is the number of states and *k* is the number of singular values that are greater than 1. Thus, this method is applicable to systems with large numbers of ports.

Passivity Enforcement Actions for S-Elements and W-Elements

Special actions are taken for an S-element or a W-element when **enforce_passivity** is set to a nonzero value, in an instance statement, an **.option** statement, or a **.model** statement.

Actions for S-Elements

a. If **enforce_passivity** is set to a non-zero value in a frequency-domain analysis (LNA or HB analysis) and time domain modeling was not enabled a priori (i.e., there is no **.option** statement for **time_domain_s_model=1**), then Nexxim enables time-domain modeling and switches to state-space modeling. The following warning message is issued:

```
Passivity enforcement requested for one or more S-elements during a
frequency-domain analysis. Switching to state-space modeling.
```

b. If passivity is enforced (**enforce_passivity** is set to non-zero) but the **convolution** option is 1,2, or 3, then the request for passivity enforcement is ignored and convolution proceeds. The following warning message is issued:

```
Passivity enforcement not supported for convolution models.
Proceeding without passivity enforcement.
```

c. If **enforce_passivity** is zero and a frequency-domain analysis is run with **convolution** option 1, 2, or 3, the convolution options is ignored. Note that the option **time_domain_s_model=1** has no effect on the analysis in this case, since that option is valid only for state-space models, not for convolution models. The warning is:

```
Convolution modeling (convolution options 1, 2, and 3) ignored during
frequency-domain analysis.
```

Actions for W-Elements

a. If **enforce_passivity** is set to a non-zero value, and the default value of the **dc_only** option is not over-ridden, and a frequency-domain analysis (LNA or HB analysis) is chosen, then Nexxim enables time-domain modeling and switches to state-space modeling. The following warning message is displayed:

   ```
   Passivity enforcement requested for one or more W-elements
   during frequency-domain analysis. Enabling time-domain model.
   ```

b. If **enforce_passivity** is set to a non-zero value and the default value of the **dc_only** option is over-ridden, then Nexxim ignores the **enforce_passivity** option. The following warning message is displayed:

   ```
   Request for passivity enforcement for W-elements ignored when
   only DC solution is needed.
   ```

c. If passivity is not enforced, time-domain modeling is enabled, and the **dc_only** option is set to 1, then time-domain modeling is deactivated. The following warning message is displayed:

   ```
   Request for time-domain modeling ignored when only DC solution
   is needed.
   ```

Passivity References

[1] Carlos P. Coelho, Joel Philips, and Miguel Silveira, "A Convex Programming Approach for Generating Guaranteed Passive Approximations to Tabulated Frequency-Data," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, n. 4, pp. 293-301, February 2004.

[2] S. Grivet –Talocia, "Passivity Enforcement via Perturbation of Hamiltonian Matrices", *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 51 n. 9, pp. 1755-1769, September 2004.

[3] Carlos P. Coelho, Joel Philips, and Miguel Silveira, "Generating High Accuracy Simulation Models Using Problem-Tailored Orthogonal Polynomials Basis ," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 53 n. 12, pp. 2705-2714, December 2006.

[4] S. Grivet –Talocia, "An Adaptive Sampling technique for Passivity Characterization and Enforcement of Large Interconnect Macromodels", *IEEE Transactions on Advanced Packaging*, vol. 30 n. 2, pp. 226-237, May 2007.

[5] Dharmendra Saraswat, Ramachanda Achar, and Michel S. Nachla, "Global Passivity Enforcement Algorithm of Interconnect Subworks Characterized by Tabulated Data," *IEEE Transactions on Very Large Scale Integrated (VLSI) Systems*, Vol 13 n. 7, pp. 819-832, July 2005.

[6] D. Saraswat, R. Achar, and M. Nachla, "On Passivity Enforcement for Macromodels of S-Parameter Based Tabulated Subnetworks," *IEEE Transactions on Very Large Scale Integrated (VLSI) Systems*, Vol 15 n.. 1, pp. 3777-3780, January, 2007.

Causality, Passivity, and Fitting Errors, Technical Note

This technical note is to help understand the concepts of causality, passivity and fitting errors in the context of S – parameter extraction and its application or effect in the circuit time domain simulation.

Topics in this Technical Note:

- Definitions
- Introduction
- Impact of Type of Frequency Sweep
- How to Check Causality, Passivity & Fitting Error
- Why IFPVLF?
- Recommendations to Improve Causality, Passivity & Fitting Error
- PCB Example

Definitions

What is Causality?

Causality is a property of a system which ensures that output does not come before input (i.e., the output from the system at any instance depends on present and past inputs, not on future inputs).

What is Passivity?

Passivity is a property of a (causal) system which ensures that maximum singular value of each S - parameter is less than 1. Thus, the system is lossy and is not generating power on its own.

What is a fitting error?

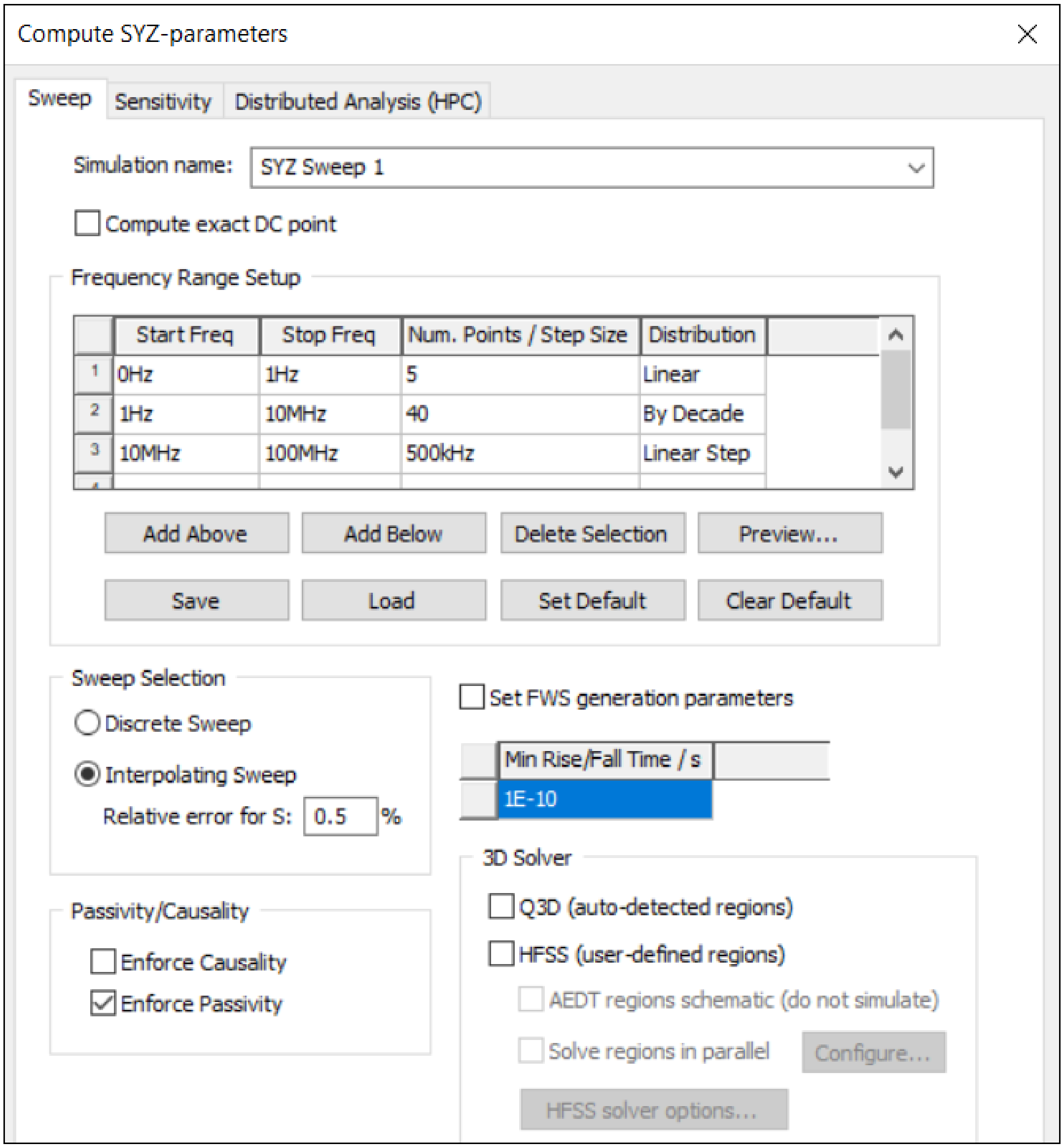
When performing transient simulations with S-parameter data extracted in frequency domain, the transient solver must convert the S-parameter data into a suitable quality macromodel (i.e., a Nexxim state space model). The resulting state space model is fitted to the extracted S-parameter data. The extent of error between the S-parameter data and the fitted state space model is called the fitting error.

Introduction

Passivity and Causality are two critical properties that decide the quality of an S-parameter data extracted from any field solver (e.g., SIwave, HFSS, Q3D etc.). They are necessary conditions (but might not be sufficient) for expecting a good fitting experience (i.e., low fitting error) when generating a broadband macromodel for a transient simulation. While the notion of causality and passivity is valid for the entire frequency range from -infinity to +infinity, the user will typically sample bandlimited data (e.g., from 0 Hz to 1 GHz, with a certain step size or distribution). Trying to make bandlimited data causal and passive can result in the fitting error worsening when trying to convert the data into a state space model for transient simulations. The following sections will explain some of the scenarios and fitting experiences the user may encounter.

Impact of Type of Frequency Sweep

Assume there is a PCB inside SIwave for which the user is trying to extract S -parameter data using SYZ analysis. Some of the settings inside SYZ analysis play a major role in the quality of the exported S-parameter data.

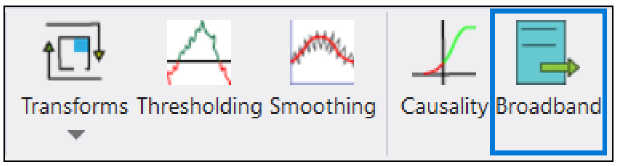


A discrete sweep solves the structure for the field solution at all the frequency sweep points given. An interpolating sweep solves selective points in the sweep based on an adaptive frequency sampling (AFS) algorithm. Then, based on the smaller number of solved points (i.e., basis points), the sweep interpolates the S-parameter data values of the remaining unsolved points in the sweep until the desired interpolation error is achieved. An Interpolating sweep is more suitable when the frequency range is wide, and the frequency response is smooth aside for a few resonances (i.e., most EMI/EMC applications). In those circumstances, an interpolating sweep will extract a reasonable matrix in a reasonable simulation time.

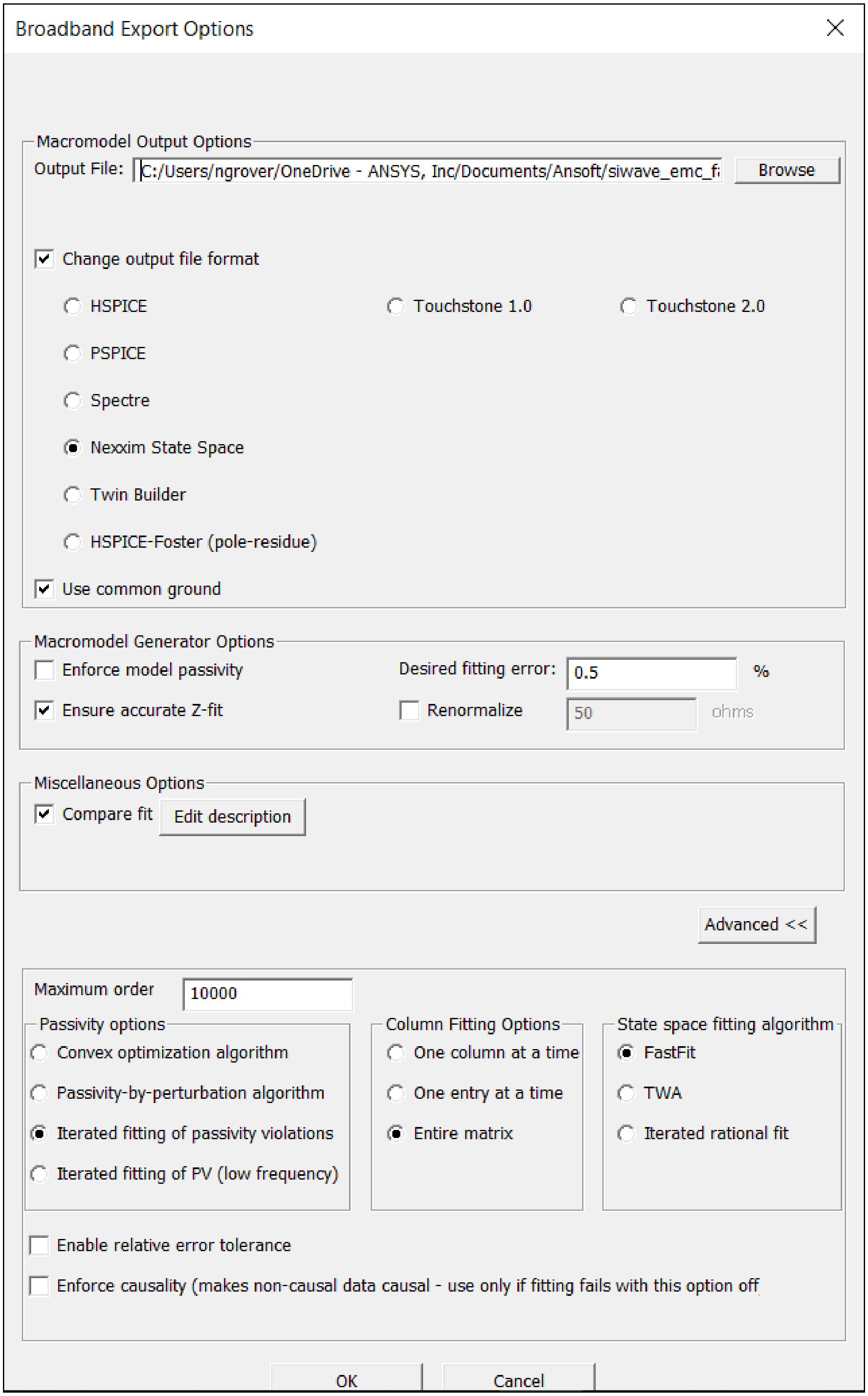
An interpolation sweep is performed in two ways: one that goes through every basis point and then interpolates out to the set of unsolved points (default, when the “Enforce Causality” checkbox is unchecked), and does not guarantee causality or passivity, and the other, which does a casual rational fit to the set of basis points and then interpolates (activate by checking the box for “Enforce Causality”), which also does not guarantee passivity. By checking the “Enforce Passivity” option within an interpolating sweep, the user is guaranteed that the field solver will generate causal and passive S-parameter data.

How to Check Causality, Passivity & Fitting Error

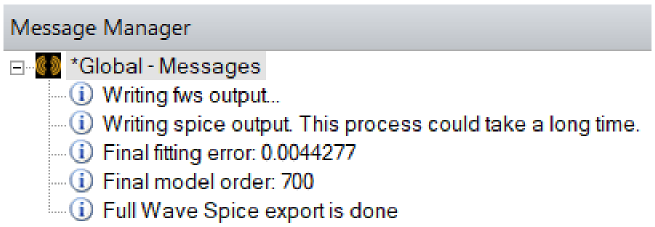
Use **Network Data Explorer** (NDE) to view the aspects described in Section III., such as the quality of data fitting. Upon reviewing the exported S-parameter data in NDE, review the causality of the data when it is exported into a macromodel (i.e., state space model) by clicking Broadband in the top ribbon of NDE (shown in the following screenshot).



Once the default settings and format are selected for a Nexxim state space model, generate the macromodel via the **Broadband Export Options** window.

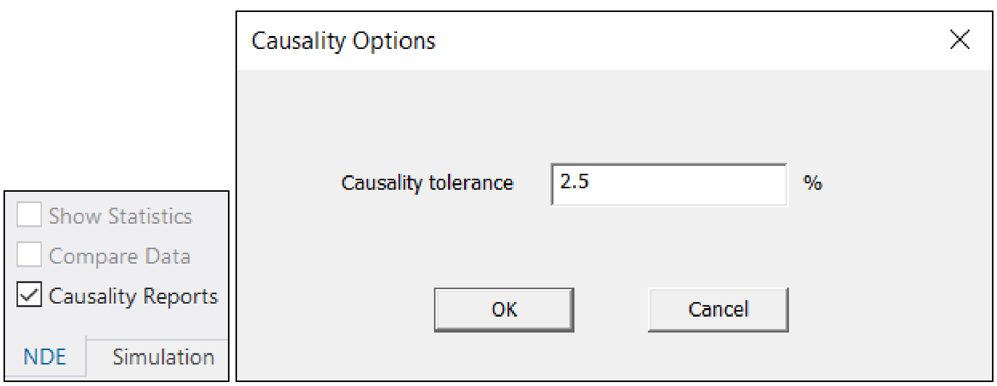

Broadband Export Options
Macromodel Output Options
Output File:
C:/Users/ngrover/OneDrive - ANSYS, Inc/Documents/Ansoft/siwave_emc_fi
Browse
Change output file format
HSPICE
Touchstone 1.0
Touchstone 2.0
PSPICE
Spectre
Nexxim State Space
Twin Builder
HSPICE-Foster (pole-residue)
Use common ground
Macromodel Generator Options
Enforce model passivity
Desired fitting error:
0.5
%
Ensure accurate Z-fit
Renormalize
50
ohms
Miscellaneous Options
Compare fit
Edit description
Advanced <<
Maximum order
10000
Passivity options
Convex optimization algorithm
Passivity-by-perturbation algorithm
Iterated fitting of passivity violations
Iterated fitting of PV (low frequency)
Column Fitting Options
One column at a time
One entry at a time
Entire matrix
State space fitting algorithm
FastFit
TWA
Iterated rational fit
Enable relative error tolerance
Enforce causality (makes non-causal data causal - use only if fitting fails with this option off
OK
Cancel

Once the macromodel is generated, the final fitting error will be displayed in the **Message Manager** window (shown in the following screenshot).

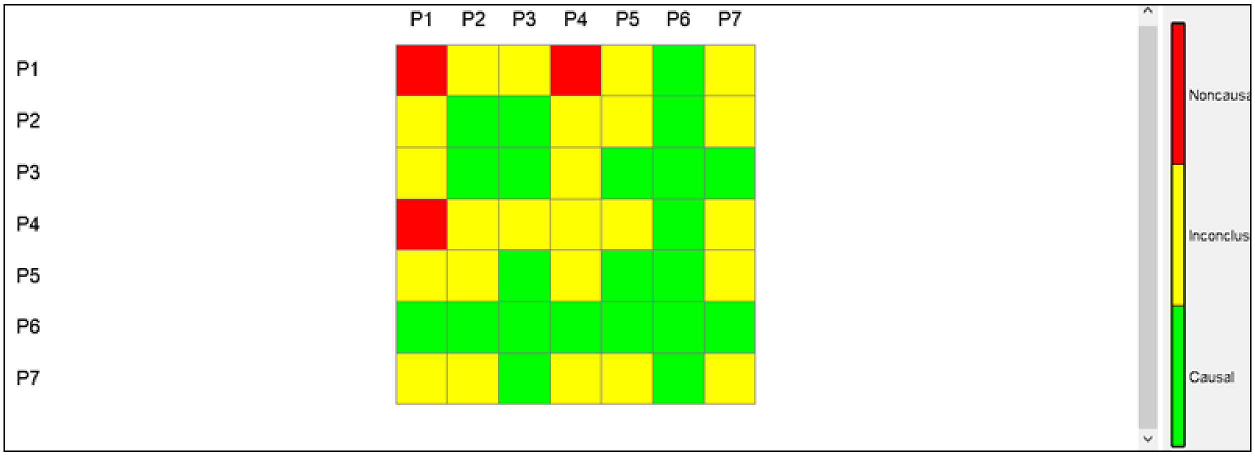


In this example, the final fitting error is 0.0044277 or approximately 0.44%. The macromodel is causal with respect to the fitting error tolerance of 0.5% set during export. The final fitting error could potentially exceed 0.5% which would mean, with a threshold of 0.5%, that the macromodel is not causal. The tolerance could also be altered (e.g., set to 0.1%, 1%, etc.) for double-checking the causality of a macromodel with tolerances lower and higher than the default 0.5%. Sometimes, tightening or loosening the tolerance gives a better fit. The latter works because a lower (i.e., tighter) fitting tolerance, the software may try to over fit the data and, in the process, destroy the fitting quality of the macromodel. This could result in, for example, many (out of band) passivity violations.

The following screenshot demonstrates one method for checking causality.

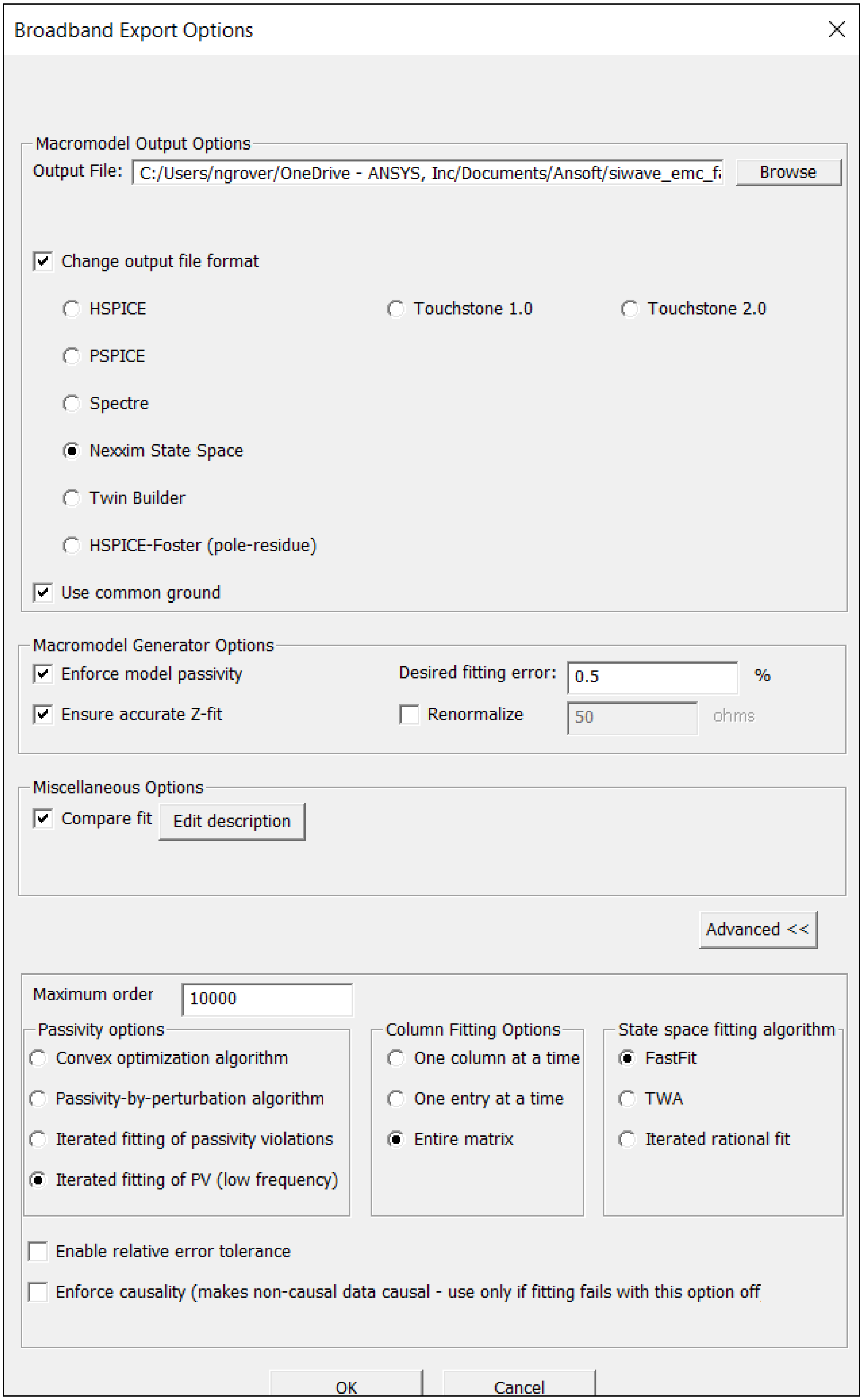


This method yields a colored plot (following). The data is causal, non – causal, or inconclusive.

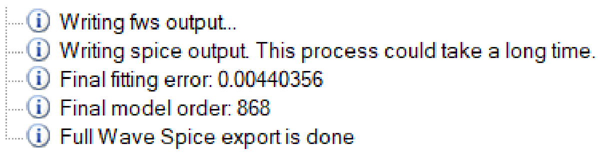


Note that the colored plot is also generated with respect to the tolerance set by the user (in this case, the default fitting error is 2.5%). There is no direct correlation between the tolerance of 2.5% here and 0.5% fitting error set in the Broadband Export Options window. Use this example for guidance, to see where the data is not meeting causality once the fitting error check returns a bad result. This example should not be used as a final basis to check for causality. Once the macromodel is created and returns a satisfactory causality check result, it returns to the Broadband Export Options window.

This time, check the **Enforce model passivity** box and select **Iterated fitting of PV (low frequency)** from the Passivity options group box (i.e., IFPVLF), as shown in the following screenshot.


Broadband Export Options
Macromodel Output Options
Output File: C:/Users/ngrover/OneDrive - ANSYS, Inc/Documents/Ansoft/siwave_emc_fi
Browse
Change output file format
HSPICE
Touchstone 1.0
Touchstone 2.0
PSPICE
Spectre
Nexxim State Space
Twin Builder
HSPICE-Foster (pole-residue)
Use common ground
Macromodel Generator Options
Enforce model passivity
Desired fitting error: 0.5 %
Ensure accurate Z-fit
Renormalize
50 ohms
Miscellaneous Options
Compare fit
Edit description
Advanced <<
Maximum order 10000
Passivity options
Convex optimization algorithm
Passivity-by-perturbation algorithm
Iterated fitting of passivity violations
Iterated fitting of PV (low frequency)
Column Fitting Options
One column at a time
One entry at a time
Entire matrix
State space fitting algorithm
FastFit
TWA
Iterated rational fit
Enable relative error tolerance
Enforce causality (makes non-causal data causal - use only if fitting fails with this option off
OK
Cancel

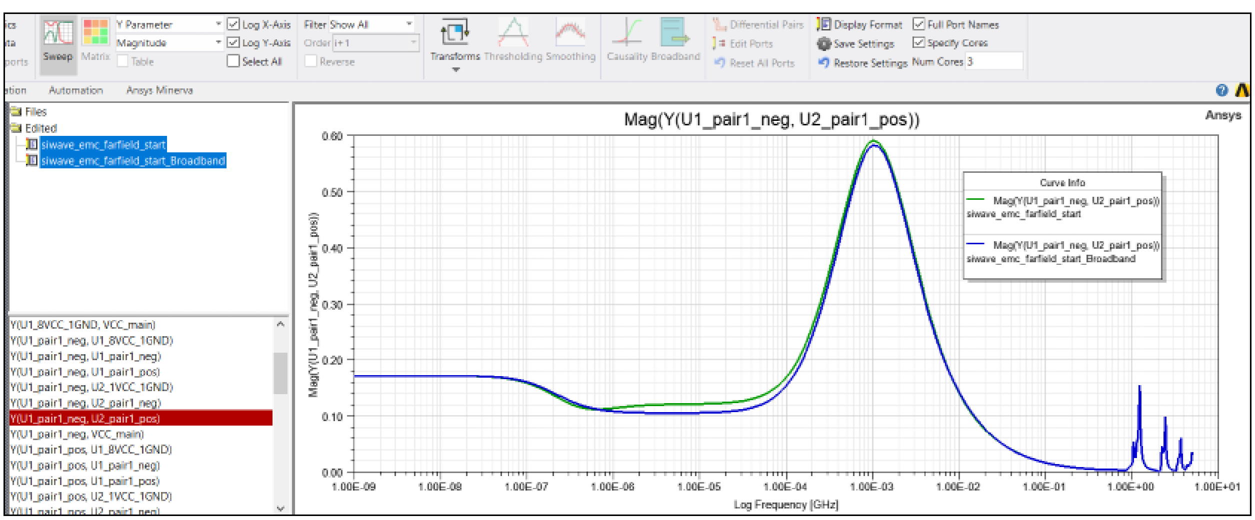
Once the new macromodel is generated, the final fitting error will appear again in the Message Manager window. Fitting error (with IFPVLF considered) will be displayed as in the following screenshot.



If the error is below the fitting tolerance of 0.5%, the generated macromodel is causal and passive. Subsequently, the macromodel can be utilized for the next step of transient simulation. A final fitting error below the fitting tolerance is a necessary condition for a good fitting experience but may not be sufficient when generating a broadband macromodel for a transient simulation. It must be ensured that the transient simulation functional output makes sense.

Consider that though the original S-parameter data may be causal and passive, there is no guarantee that the resultant macromodel is going to remain passive. For example, assuming the user has the following data: 1.0, 1.0, 0.9999, 0.9999 and so on, at several frequency samples, and a rational fit is applied to this data, given that the data is causal and passive, and the target fitting error is 0.5% or 0.005. The resulting fit would be: 1.0049, 1.0049, 1.00001, 1.000001 and so on. Though the fitting error is better than 0.5%, the maximum singular value is 1.0049, which exceeds one. Hence, the result is non-passive. Though the data is causal and passive, the fit may not be.

What is important is what the macromodel fitting tool does with respect to the original S - parameter data. This can be done in NDE, using a “Y” parameter plot in magnitude (with log – log axis), as shown in the following screenshot. The green line is the original data, and the blue line is the fitted data:



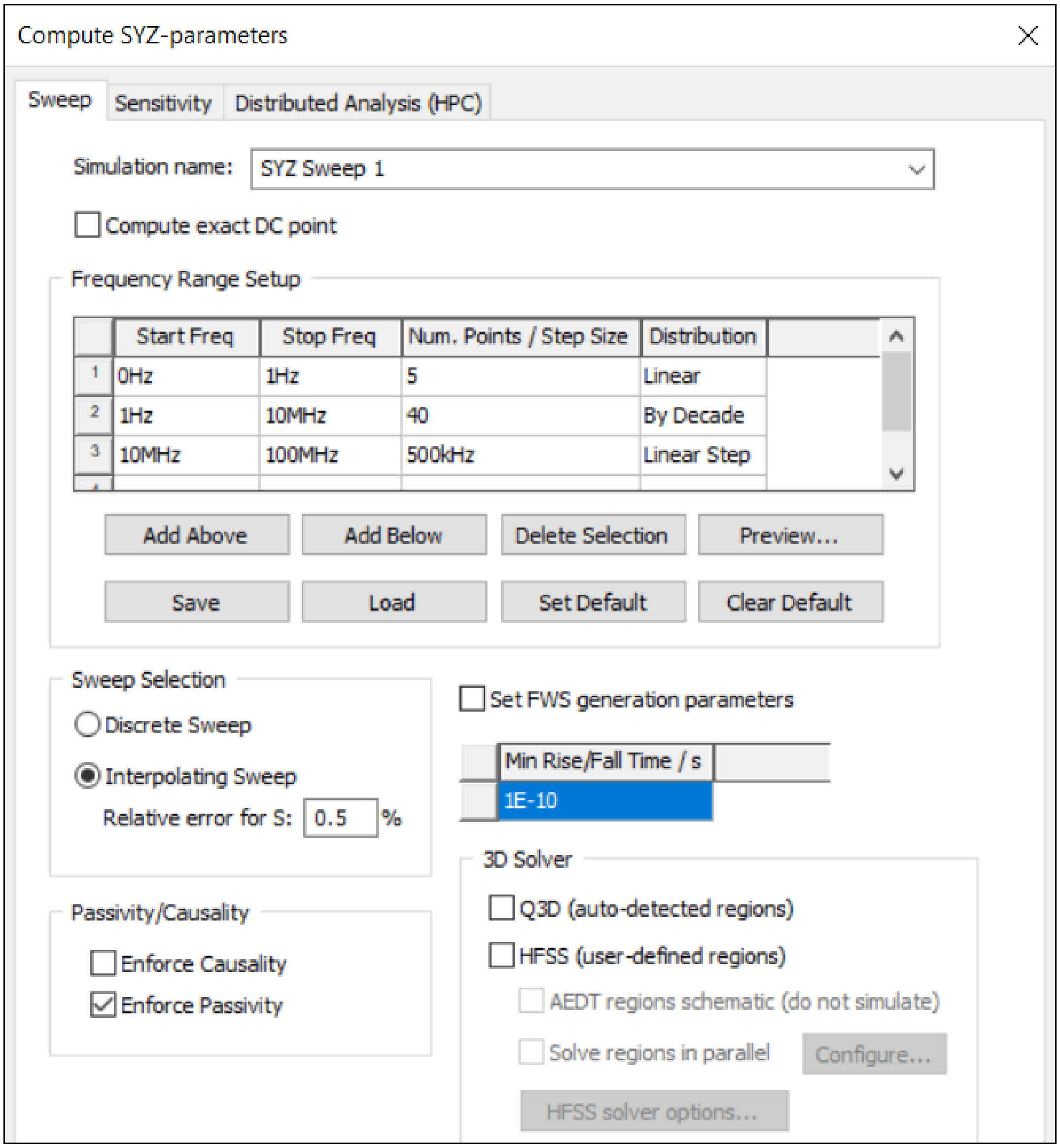
Why IFPVLF?

For many applications (e.g., EMI, EMC, etc.), the desire is to fit to S-parameter data while preserving the fit corresponding to “Z” or “Y” at DC and low frequency to minimize the inaccuracies in steady state results from transient simulations. IFPVLF increases the likelihood of optimal DC and low frequency fit. However, a weighting function (of frequency) to control the fit to low frequency is utilized, and the final fitting error to mid and higher frequency ranges may be less accurate.

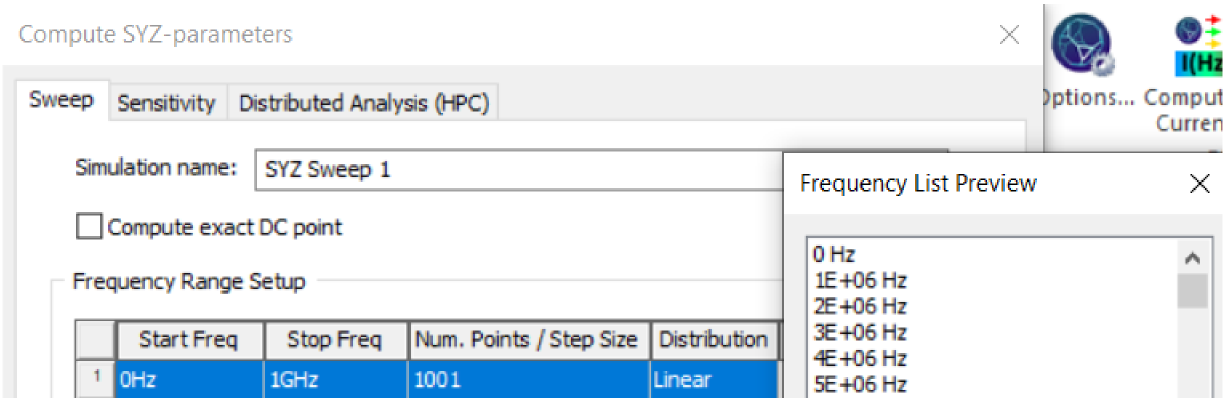
Recommendations to Improve Causality, Passivity & Fitting Error

If the final fitting error is unsatisfactory for any tolerance the user sets. (e.g., anywhere between 0.1% and 3%), it is prudent to return to the field solver to incorporate some additional settings that may improve the fitting experience. The following are just some of the additional settings that should be considered:

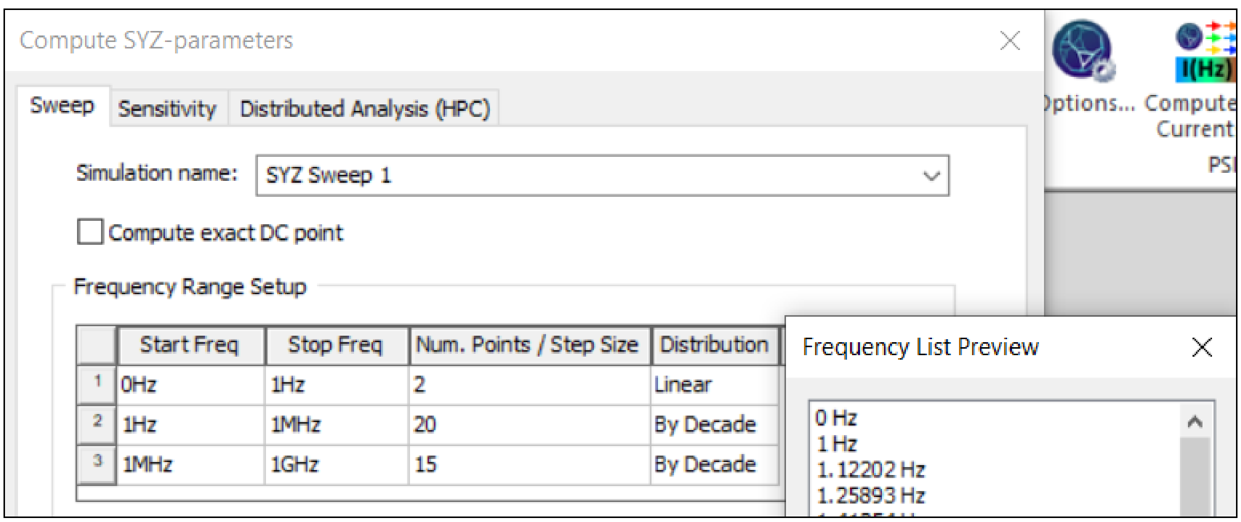
- Use of causal material properties in a PCB stackup, in which the extracted data is also expected to be causal.
- Use the **Enforce Passivity** checkbox. For example, in SIwave, check the **Enforce Passivity** box and the solver will try and make data at the remaining unsolved points (in case of interpolating sweep) passive if they were not already. Then the user only needs to enforce passivity for the interpolation sweep.



- Lower the interpolation error percentage tolerance. In SIwave analysis settings, the default interpolation error is 0.5%. Lower the interpolation error percentage to improve the S-parameter data. Lowering this tolerance will force the solver to solve a greater number of (basis) points. Reducing the relative interpolation error percentage tolerance is more effective if the number of steps is increased. However, increasing the number of steps will also raise the run time of the field solver.
- Refine and distribute the frequency range to properly cover the low frequency range instead of defining one single range up to the highest frequency. For example, one could define frequency range as shown in the following screenshot.



Using this sweep, consider that passed the DC point of 0Hz, the next point is in a higher range (in this example, 1 MHz). There is low-frequency data missing, which could negatively impact the fitting later. The following screenshot demonstrates a sweep will result in a more even distribution.



- Increase the highest frequency of solving. Increasing the max frequency is potentially helpful depending upon the application.

  For example, running a transient simulation with a pulse and small rise time typically requires first ensuring the frequency data covers a bandwidth equal to a factor (e.g., two times one over the rise time).

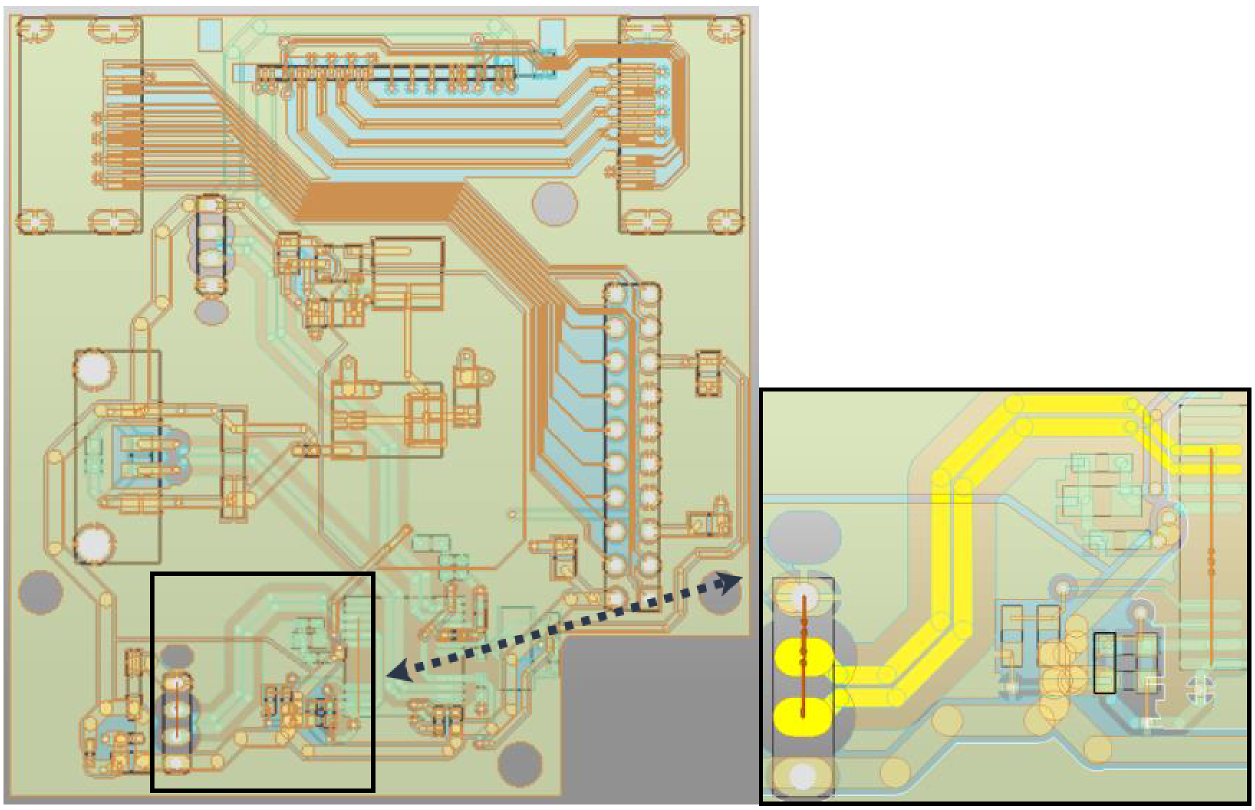
  The robustness of the test in place (i.e., NDE) on causality of the scattering parameter data can potentially be improved by covering more of the data, since much of the uncertainty happens out of the band. The notion of causality and passivity should be understood on the whole real line.

For example, consider increasing the highest frequency of a sweep to 3 GHz, if it is currently 1 GHz.

What is important is what the software has done with respect to the original S-parameter data. This result can also be achieved in NDE first, for exploration, by making sure the fit is adequate before running the project in a Circuit design for transient analysis.

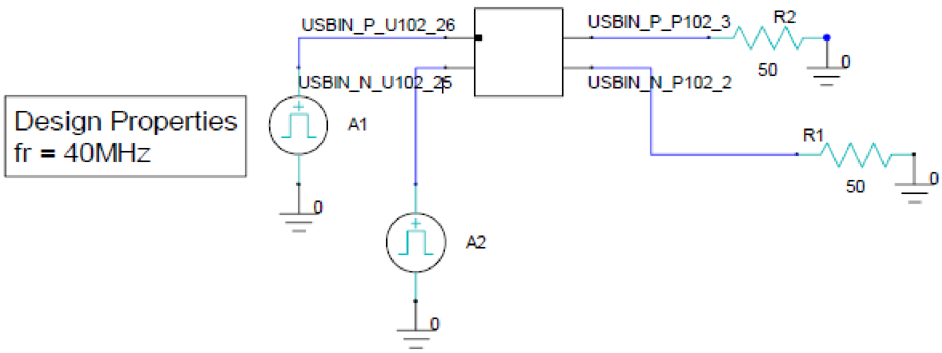
PCB Example

The following screenshots are of a simple PCB with four ports (on the two traces considered) for which the S-parameter model is extracted using SIwave.



When the user creates a Circuit simulation for transient analysis, they feed the two traces with a pulse source each of switching frequency 40 MHz and terminate the other end of the traces with 50 Ohms.

Apart from the touchstone model and the macromodel quality, different settings of extraction and broadband model generation affect the transient analysis output.



After field solver (in this case SIwave) extraction, use Network Data Explorer (NDE) to check the model quality in terms of causality, passivity, and fitting error.

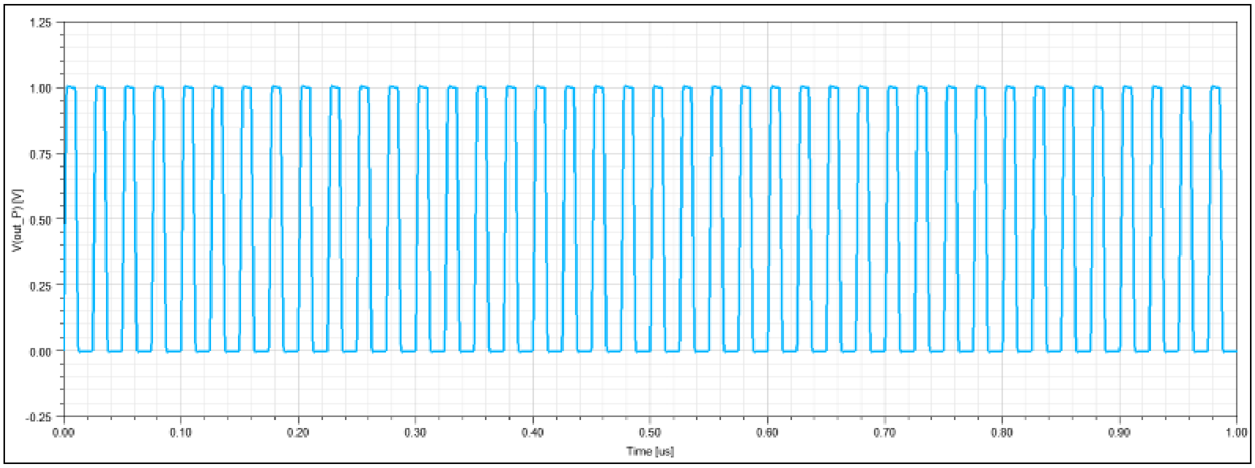
Tabulated below are some examples with different settings in SIwave (when exporting S-parameter data) and NDE (when exporting a broadband macromodel):

	SIWave				Network Data Explorer (NDE)		
S. No.	Interpolation Error Tolerance (%)	Causality/Passivity Enabled in Siwave	Highest Frequency in Siwave Sweep	Sweep Distribution	IFPVLF Yes or No	Fitting Error Tolerance (%)	Final Fitting Error (%)
1.	1	No	100MHz	0Hz-100MHz 51 Points (Linear)	No	0.5	1.46
2.	1	No	100MHz		Yes	0.5	1.38
3.	0.001	Yes	100MHz		Yes	0.5	0.46
4.	0.001	Yes	2GHz	0Hz – 1Hz: 5 points (Linear) 1Hz – 100MHz: 30 points (Decade) 100MHz – 2GHz: 1.5MHz (Linear step)	Yes	0.5	0.17
5.	0.001	Yes	5GHz	0Hz – 1Hz: 5 points (Linear) 1Hz – 100MHz: 30 points (Decade) 100MHz – 5GHz: 1.5MHz (Linear step)	Yes	0.5	0.19
6.	0.001	Yes	10GHz	0Hz – 1Hz: 5 points (Linear) 1Hz – 100MHz: 30 points (Decade) 100MHz – 10GHz: 1.5MHz (Linear step)	Yes	0.5	0.17

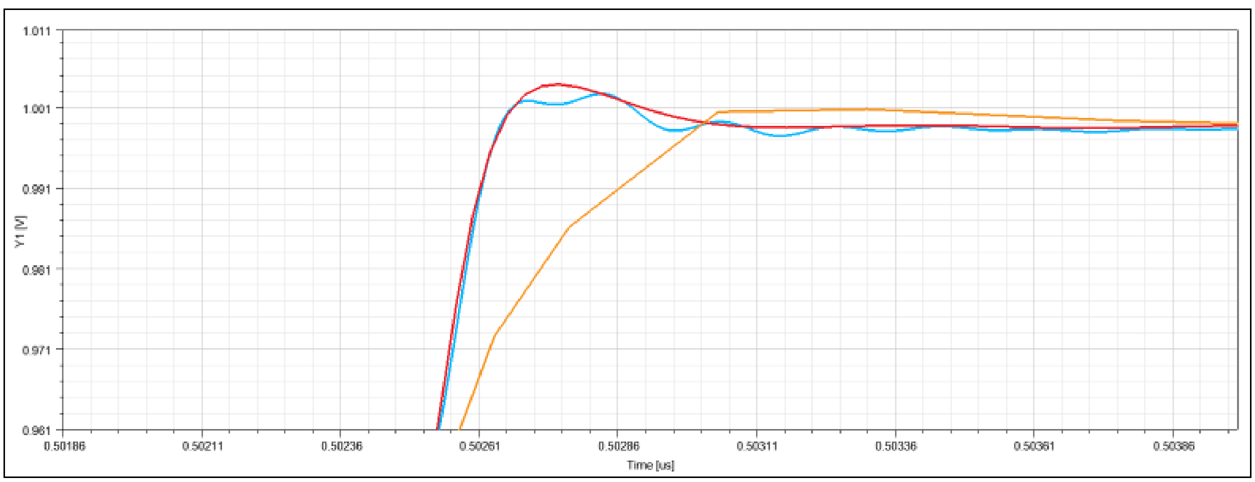
Consider the final fitting error of the models in S. No. 1 without passivity (IFPVLF) imposed, which is above the threshold of 0.5%. The model is not causal with respect to the threshold set.

Consider models S. No. 2 through S. No. 6. Models S. No. 2 through S. No. 6. have final fitting errors with passivity (IFPVLF) imposed, but only models S. No. 3 through 6 are causal and passive. Subsequently, model S. No. 2 should not be used for transient simulation.

To confirm the transient simulation behavior from these models, monitor the voltage across the 50 Ohms load on the USBIN_P trace. A typical output waveform is shown in the following screenshot, but the section that follows focuses on one of the rising edges of the waveform.

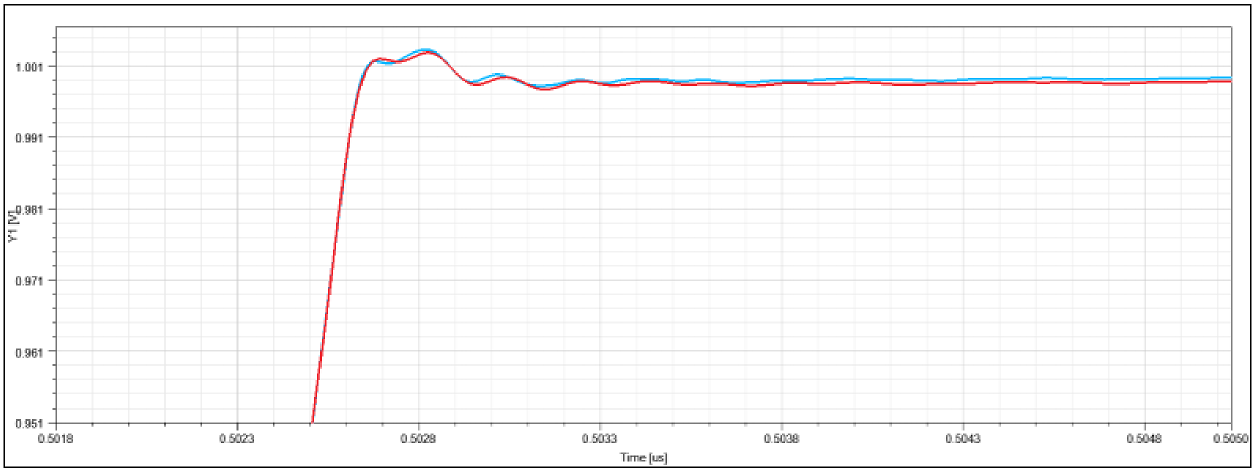


Between models S. Nos. 3, 4, and 5, the maximum frequency extracted in SIwave has been modified to 100 MHz (i.e., the orange line shown in the following screenshot), 2GHz (i.e., the red line), and 5GHz (i.e., the blue line) respectively. The change happening between the transient results of these three models is also shown. The blue line (i.e., 5GHz model) captures the high frequency content (i.e., the ripples) in the rising and falling edges of the voltage waveform more accurately. Thus, increasing the maximum frequency in SIwave extraction corrects the functional voltage output waveform, even though all three models were causal and passive (having final fitting errors below 0.5% threshold).



The only difference between models S. No. 5 and 6 is in the maximum frequency extracted in SIwave, 5GHz (i.e., the blue line in the following example) and 10GHz (i.e., the red line), respectively. Since the transient results of these two models are a trivial difference, it is unnecessary to increase the maximum frequency of extraction beyond 5 GHz. The functional output voltage has started to converge to the correct waveform even with the 5 GHz model.

With a 40 MHz switching source, as shown in the previous screenshot, the maximum frequency of 100 MHz may be insufficient to capture all the switching dynamics. Depending on the application and switching frequencies, the impact of choosing the maximum frequency in the sweep is shown in the following screenshot.



Conclusion

Depending on the model and the desired frequency range, one or a combination of the above recommendations may be required to improve the fitting experience and/or the transient functional outputs. A fundamental understanding of these concepts is important for appropriate analysis.

FAQs

Q1. Is it guaranteed that the discrete sweep in our field solvers will always generate causal and passive S-parameter data?

Answer: This is more expected than guaranteed. For example, if a frequency dependent dielectric (loss) material is not provided by the user, then the calculated discrete sweep S - parameter data could end up being non - causal. However, using the Debye or Djordjevic-Sarkar models will ensure that does not happen, resulting in causal S-parameter data. There have not been any cases reported of non - causality of S - parameter data with discrete sweep using our field solvers. However, as the causality property will depend upon the frequency range and number of points in the sweep giving rise to bandlimited discretized S - parameter data, the above answer is subjective.

Q2. How is the causality check used if the check shows the data is not causal within the tolerance specified? Does this indicate that the S-parameter data is unsuitable?

Answer: Causality and passivity should be considered in the whole real line, from -infinity to +infinity in continuous domain. For example, if the S-parameter data received is discrete bandlimited and the user needs to know whether the sampled bandlimited S-parameter data is causal, then causality is here understood with respect to a prescribed threshold. This threshold should have a correlation to the state space fitting error which uses a causal rational function to fit to the same S-parameter data. Therefore, the best measure of causality relative to a target fitting error (default is 0.5%) is whether the solver produces a better fit than the target fitting error. Meanwhile, the causality checker considers the contributions of discretization error (i.e., the error between samples) as well as the unavailable S -parameter data (i.e., the data outside of the band), by using a tight (upper) bound accounting for the missing data.

Q3. Does a low fitting error below the set threshold guarantee a better macromodel for transient simulation?

Answer: Even models with 100 MHz (i.e., S. No. 3 in the Section VII table) or 2 GHz (i.e., S. No. 4 in the table) - maximum extracted frequency were causal and passive and had a fitting error below the set threshold of 0.5% were not guaranteed the most accurate transient output. Low fitting error is only an indication of how well the solver is working to produce a macromodel that fits the original S-parameter data. Whether the original S-parameter data is of optimum quality (in terms of frequency points, maximum frequency, etc.) needs to be considered.

References:

[1] A. Hochman, "FastAAA: A fast rational-function fitter," 2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), 2017, pp. 1-3, doi: 10.1109/EPEPS.2017.8329756.

[2] S. Lalgudi, S. Asgari and M. Tsuk, "Improved procedure to test causality of tabulated S-parameters," 2012 IEEE 21st Conference on Electrical Performance of Electronic Packaging and Systems, 2012, pp. 191-194, doi: 10.1109/EPEPS.2012.6457874.

[3] S. Asgari, S. N. Lalgudi and M. Tsuk, "Analytical integration-based causality checking of tabulated S-parameters," 19th Topical Meeting on Electrical Performance of Electronic Packaging and Systems, 2010, pp. 189-192, doi: 10.1109/EPEPS.2010.5642578.

[4] S. Lefteriu and S. Grivet-Talocia, "Loewner-based macromodeling with exact interpolation constraints," 2016 IEEE 25th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), 2016, pp. 43-46, doi: 10.1109/EPEPS.2016.7835414.

[5] B. Gustavsen and C. Heitz, "Fast realization of the modal vector fitting method for rational modeling with accurate representation of small eigenvalues," 2009 IEEE Power & Energy Society General Meeting, 2009, pp. 1-1, doi: 10.1109/PES.2009.5275736.

Authors:

1. Nikhil Grover, Manager Application Engineering, Ansys (India)

2. Saeed Asgari, Senior Principal R&D Engineer, Ansys (USA)

Acknowledgments:

1. Flavio Calvano, Manager Application Engineering, Ansys (Italy)

2. Ryan Schimizzi, Lead R&D Engineer, Ansys (USA)

3. Adam Charpentier, Senior R&D Documentation Specialist, Ansys (USA)

4. Wade Smith, Manager Application Engineering, Ansys (USA)

5. Fred German, Director R&D, Ansys (USA)

6. Rajavardhan Talashila, Application Engineer II, Ansys (India)

Convolution Method

By default, Nexxim transient analysis of S-parameter elements uses a state-space formulation.

Setting option **s_element.convolution** to 1, 2, or 3 sets Nexxim to use convolution rather than state-space matrices to model the behavior of S-parameter elements during transient analysis.

The convolution method converts the frequency-domain parameters into time-domain impulse responses. The impulse responses are directly applied in transient simulation via the convolution integral. See Convolution References for further information.

Upon completion, Nexxim reports the error in the least-squares convolution fit. Errors greater than 0.1 suggest non-causality in the S-parameter data. Convolution may also yield inaccurate results without any warning. Several modifications to the data may improve the quality of the convolution response:

- Provide linearly spaced frequency points in the S-parameter file.
- Include DC in the S-parameter file.
- Provide as large a frequency range as possible.
- Avoid sharper-than-necessary input waveforms in the netlist.

The **convolution** options control the generation of the impulse response. **Convolution=1** is recommended. **Convolution=2** and **3** are for advanced users.

- **convolution=1**: The impulse response is a piecewise linear waveform, with breakpoints chosen to exactly match the given frequency-domain data. This setting gives the most accurate response, but may have passivity violations just above the frequency band of interest.

When **convolution=1** has been enabled, setting the S-element parameter **DELAYHANDLE=1** enables convolution to incorporate propagation delay for that element.

> **Note:**
>
> The HSPICE S-Model parameter **DELAYHANDLE=1** has the same effect as option **convolution =1**.

- **convolution=2**: The impulse response is a train of impulses in the time domain, given by the inverse Fast Fourier Transform. This setting yields an impulse response that is accurate in-band and usually passive. However, the transient waveforms may have discontinuity effects.
- **convolution=3**: The impulse response is the linear interpolation of the inverse FFT results. This setting yields an impulse response with a low probability of passivity violations, but there is significant filtering toward the top of the frequency range of the input data.

The **convolution=1**, **2**, **3** options apply the convolution method globally to all S-parameter models in the design. The **convolution** option can be overridden in a particular model by setting model parameter **CONVOLUTION**. The setting of a **CONVOLUTION** model parameter overrides the global option **convolution** setting for S-element instances that reference the given model.

Convolution References

[1] S. Lalgudi, M. Tsuk and S. Asgari. “Accurate Delay-Augmented Convolution-based Transient Simulation using Delayed Least Squares Convolution,” EPEP 2010.

[2] M.J. Tsuk, and S.N. Lalgudi, “Least-squares convolution: A method to improve the fidelity of convolution in transient circuit simulation,” *IEEE 18th Topical Meeting Electr. Performance Electron. Packag.*, Portland, OR, Oct 23-25, 2009.

[3] P. Triviero, and S. Grivet-Talioca, “Causality-constrained interpolation of tabulated frequency responses,”*IEEE 15th Topical Meeting Electr. Performance Electron. Packag.*, Scottsdale, AZ, Oct 23-25, 2006, pp. 181-184.

Reference Nodes on S-Parameter Elements

When an S-parameter element is inserted into a schematic, the import window allows you to specify a reference node. You may choose:

- **Implied reference to ground**: the global ground is the reference port for each N port connection. Its voltage is always 0V.

- **Common reference port**: an additional reference pin is added that is a common reference for the definition of port voltages. This reference can have any voltage and corresponds to the net for that extra pin.
- **Add individual hidden reference pin per port**: an additional reference pin is added for each port. These reference pins are hidden, set to 0V by default, and can each have their own reference voltage. You can connect each reference pin to its own net. Select this option if you do not want to see the reference pins.
- **Add individual reference pin per port**: an additional reference pin is added for each port. Every reference pin can have its own reference voltage and be connected to its own net.

Nexxim does not make any adjustments to the S-parameter matrix when **Implied reference to ground** is selected or when **Common reference port** is chosen , then connected or shorted directly to ground in the schematic.

When a **Common reference port** is unconnected or is connected to ground through a resistor, Nexxim uses matrix operations to adjust the S-parameters. When a **Common reference port** is left unconnected, Nexxim connects the port to ground through a resistor with value 10^{12} ohms. If there are individual reference pins, Nexxim adjusts the S-parameter matrix to be twice the size to accommodate the connections to these reference pins.

The following formulas are for the frequency domain at a single frequency.

It is convenient to convert from S-parameters to the equivalent Z parameters. The Z-matrix takes the vector of input currents into the vector of terminal voltages: V = Z x I.

Suppose that the S-parameters all have a common reference impedance Zref and that the resistor inserted between the reference node of the S-parameter block and ground is of value R>0.

Let matrix **A** be the 2x2 matrix with the value R/Z_{ref} in each of the four locations. The effective Z-matrix of the structure with the resistor V_{eff} = Z + A.

Put in the form of matrix manipulations for the S-parameters:

$$S_{\text{eff}} = \frac{\frac{I+S}{I-S} + A - I}{\frac{I+S}{I-S} + A + I}$$

where ***I*** is the 2x2 identity matrix.

Noise in S-Parameter Elements

By default, DC and frequency-domain analyses of S-parameter models look in the Touchstone file for noise data. If the file does not supply external noise data, an internal noise model generates the noise data. The S-Element option and instance parameter **NOISEMODEL** selects the noise data source:

- Use external data if present, or the internal noise model when no external data is present.
- Use only the internal noise model.
- Turn off all noise calculation.

Nexxim Internal Noise Model

In Nexxim, a noisy electronic circuit is modeled as a noiseless circuit plus external noise sources for which certain noise characteristics are known. The noise sources are represented as current sources, which are defined in terms of a so-called **S** $_{xx}$ matrix. The **S** $_{xx}$ matrix for a two-port device states the noise behavior at a frequency point as:

$$S_{xx} = \begin{bmatrix} \overline{i^2_{n_1}} & \overline{i_{n_2} i^*_{n_1}} \\ \overline{i_{n_2} i^*_{n_1}} & \overline{i^2_{n_2}} \end{bmatrix}$$

where $i_{ni}i_{nj}$ is a noise correlation factor between currents i and j.

The TouchStone format allows for noise specification of circuit devices of up to two ports. The noise characteristics are :

- Minimum noise figure (F_{min}) in dB.
- Source reflection coefficient to realize F_{min} (Γ_{opt}), specified in the Touchstone file as two values, magnitude and phase.
- Normalized effective noise resistance (r_n).

dru

All of these parameters are specified against frequency. The parameter Γ_{opt} is a complex VALUE, . The noise characteristics are specified as a noise block appended after the S-parameter block in a TouchStone file, formatted according to the TouchStone standard. Nexxim performs a transformation on the noise parameters found in the TouchStone file to the noise matrix S_{xx}, as explained in Gonzales [1984] (Reference [1] in the S-Parameter General References).

In the Nexxim internal noise model, circuit noise is inferred on the S-parameters.

Noise Example

Here is a shortened example of a two-port Touchstone file with noise data:

```
!2-port S-parameter file with noise data
# GHZ S RI R 50.0
```

```
!freq ReS11 ImS11 ReS21 ImS21 ReS12 ImS12 ReS22 ImS22
1.000 0.393 -0.121 -0.001 -0.002 -0.001 -0.002 0.393 -0.121
!Noise Parameters
4 0.7 0.64 69 0.38
8 2.7 0.46 -33 0.40
```

Interpolation and Extrapolation of Frequency-Dependent Data

Data may not be available at each frequency point of the simulation. Consequently, the data may require interpolation or extrapolation to produce information at the necessary frequencies.

When the S-parameter data have a low top frequency compared to the frequency content of the input signal, Nexxim extrapolates the data. The Nexxim standard transient solution for S-parameters incorporates a state-space formulation, which is guaranteed causal. This state-space formulation gives a natural extrapolation function, which is also causal. However, The Nexxim extrapolation function can differ on the functions provided by other simulators. Convolution-based simulators generally provide simple constant, linear, or windowing extrapolation functions. Such simple extrapolation functions are not causal, and therefore can give erroneous transient results.

- The designer can avoid the issue of extrapolation by extending the frequency range of the input S-parameter data well above the traditional signal integrity bandwidth of 0.35/risetime.
- The HIGHPASS and LOWPASS parameters on the S-element model control the extrapolation method. See *S-Model* in the Nexxim Components help.
- For interpolation, the S-element model in Nexxim has parameters for selection between LINEAR and STEP interpolation. STEP interpolation takes the interpolated value on the closest table entry that is lower in frequency that the current entry. LINEAR is the default. See *S-Model* in the Nexxim Components help.

Traveling Wave and Power Wave Formulas

There are two common definitions of the scattering matrix, *S*. One definition is based on traveling waves while the other is based on power waves. Traveling waves correspond to the inward and outward propagating solutions to the telegrapher equations (See References [1]). Power waves relate the power transfer in the inward and outward directions (See References [2,3]). The two definitions are equivalent if the characteristic impedances are real. However if the characteristic impedance is complex, important differences arise.

Both *S* parameter formulations define the relation between the outgoing waves **b** and the incoming waves **a**, or more specifically:

$$b = Sa$$

The difference is whether to choose traveling waves or power waves for **a** and **b**. For this derivation, consider voltage traveling waves. There are also current traveling waves and many different normalized voltage waves; however, the most common is the voltage wave:

$$V(z) = V^{+}e^{-j\beta z} + V^{-}e^{j\beta z}$$

Use the subscripts *v* and *p* to denote voltage and power waves respectively. If voltage waves are chosen, which is typical for high frequency (HF) applications, $\mathbf{a}_v$ and $\mathbf{b}_v$ are:

$$a_v = V^{+} = \frac{V + Z_c I}{2}$$

$$b_v = V^{-} = \frac{V - Z_c I}{2}$$

where Z_c is a diagonal matrix of comprising the characteristic impedance of each port. The inverse transformation is:

$$\mathbf{V} = \mathbf{a}_v + \mathbf{b}_v$$
$$\mathbf{I} = Z_c^{-1}\left(\mathbf{a}_v - \mathbf{b}_v\right)$$

And if power waves are chosen, which is typical for circuit applications, $\mathbf{a}_p$ and $\mathbf{b}_p$ are:

$$\mathbf{a}_p = \frac{\mathbf{V} + Z_c\mathbf{I}}{2\sqrt{R_c}}$$
$$\mathbf{b}_p = \frac{\mathbf{V} - Z_c^{H}\mathbf{I}}{2\sqrt{R_c}}$$

where $R_c$is the real part of characteristic impedance, R_c= Re (Z_C) and where the superscript H denotes the Hermitian, or conjugate transpose. The inverse transformation of the power waves is:

$$\mathbf{V} = \frac{Z_c^H \mathbf{a}_p + Z_c \mathbf{b}_p}{\sqrt{R_c}}$$

$$\mathbf{I} = \frac{\mathbf{a}_p - \mathbf{b}_p}{\sqrt{R_c}}$$

The voltage based definitions yield the following familiar relationships between the network parameters:

$$S_v = (\overline{Z}_v - I)^{-1}(\overline{Z}_v + I)$$

$$\overline{Z} = (I - S_v)^{-1}(S_v + I)$$

$$\overline{Y} = (I + S_v)^{-1}(S_v - I)$$

where $\overline{Y}$ and $\overline{Z}$ are the normalized Y and I matrices and where I is the identity matrix. The power based definition yields more complicated expressions for network relationships:

$$S_p = R_c(Z - Z_c^H)(Z + Z_c)^{-1}R_c^{-1}$$

$$S_p = R_c(I - Z_c^H Y)(I + Z_c Y)^{-1}R_c^{-1}$$

$$Z = R_c^{-1}(I - S_p)^{-1}(S_p Z_c + Z_c^H)R_c$$

If the characteristic impedance is completely real (Z_c= $Z^H{}_c$ = R_c), there is no difference between the two definitions. Only a significant imaginary part to the characteristic impedance can cause a noticable difference. If there is an imaginary part to the characteristic impedance, the voltage based definition can yield S parameters greater than unity. This occurs if the reactance of the load has the opposite sign of the line (e.g., the load is inductive and the line is capacitive). It is important to note that this is not wrong, just counter-intuitive. Connect the device to other components within a circuit simulator to get the correct answer.

There are two significant drawbacks to the power based S parameters [" Traveling Wave and Power Wave References" on the next page 4,5]. A waveguide in cutoff has a characteristic impedance that is completely imaginary (R_c= 0); hence the S parameters are undefined. The concepts of normalized impedance and normalized admittance are not applicable to the power based S parameters; in the numerator, normalize with respect to $Z^H{}_c$, but in the denominator, normalize to Z_c. This means that the Smith chart cannot be used to convert power based S parameters to the network parameters *Y* and *Z*. The Smith chart only works with voltage wave or traveling wave S parameters.

Traveling Wave and Power Wave References

[1]R. E. Collin, Foundations for Microwave Engineering, New York: McGraw-Hill, 1992.

[2]K. Kurokawa,"Power waves and the scattering matrix," IEEE Trans. Microwave Theory and Tech, pp. 194-202, March 1965.

[3]D. Frickey, "Conversions Between *S*, *Y*, *Z*, *h*, *ABCD*, and *T* Parameters which are valid for complex source and load impedances'', IEEE Trans. Microwave Theory and Tech, pp. 205-211, February 1994.

[4]D. F. Williams and R. Marks, 'Comments on ''Conversions Between *S*, *Y*, *Z*, *h*, *ABCD*, and *T* Parameters which are valid for complex source and load impedances''', IEEE Trans. Microwave Theory and Tech, pp. 914-915, April 1995.

[5]R. Marks and D. F. Williams, ''A general waveguide circuit theory,'' J. Res. Natl. Inst. Stand. Technol., vol. 97, pp.533-561, Sept.-Oct. 1992.

Running State-Space Fitting on the Command Line

You can operate the Ansys state-space fitter on the command line. Use the executable `genequiv.exe` which is available in the **Electronics Desktop** installation directory, most commonly `C:\Program Files\AnsysEM\vxxx\Win64` (where *xxx* is the product version, such as v242 for 2024 R2).

Command Line Syntax:

```
genequiv [options] "<input_file>" -o "<output_file>" -s
<subcircuit_name>
```

Input file:

Input_file is the file name of input Touchstone S-parameter file. Put quotation marks around the path to accommodate any spaces in the path or file name.

Output file:

Output_file is the name of the model output file that is written to the same directory as the input file. A model_name must always be specified but is overridden if the **-o** option is used. Put quotation marks around the path to accommodate any spaces in the path or file name.

Sub-circuit name:

Subcircuit_name is required when generating SPICE-type equivalent circuits. It is not required for passivity and causality checking operations.

General Options:

All options can be specified with either a single leading dash (-help) or two dashes (--help).

- **-built**

  Display the version, build date, and time. A synonym for **-v** and **-version**.

- **-h**

  Lists the flags that genequiv accepts and their descriptions. A synonym for **-help**.

- **-help**

  Lists the flags that genequiv accepts and their descriptions. A synonym for **-h**.

- **-v**

  Display the version, build date, and time. A synonym for **-built** and **-version**.

- **-version**

  Display the version, build date, and time. A synonym for **-built** and **-v**.

State-Space Fitting Options:

- **-colfit**

  Fits the entire S-matrix using a common set of poles. **-matfit** by default.

- **-entryfit**

  Fits a unique set of poles for each entry of the S matrix.

- **-matfit**

  Fits the entire S-matrix using a common set of poles. See **-colfit**.

- **-maxp** <integer>

  Sets the maximum number of poles to use per matrix entry. 10000 by default.

- **-method** <integer>

  Sets the method to be used for the state-space fitting. Acceptable values are:

  0 for old iterated rational fitting, depricated.

  1 for TWA

  2 for FastFit. This is the default value.

- **-mp** <integer>

  Specifies the number of threads to use for multiprocessing. The default is half of the total number of cores available in your computer.

- **-relerr**

  Enforces per-entry relative error tolerance when performing a state-space fitting. False by default.

- **-tol** <double>

  Specifies the error tolerance for the state-space fitter. Set a double value and is 0.005 by default.

- **-zfit** <integer>

  Improves fitting accuracy for Z-parameters. Acceptable values are:

  0 for deactivate.

  1 for activate. This is the default value.

Passivity Enforcement and Checking Options:

- **-checkpassivity**

  Performs a passivity check on the input data and can be combined with **-checkcausality**. False by default.

- **-passive**

  Enables passivity enforcement via convex optimization. False by default.

- **-passive_ifpv**

  Enables passivity enforcement via Iterated Fitting of Passivity Violations (IFPV). False by default.

- **-passive_ifpvlf**

  Enables IFPV with enforcement of low-frequency fit. False by default.

- **-passive_pert**

  Enables passivity enforcement via Passivity-by-Perturbation (PBP). False by default.

- **-passive_rlfit**

  Enforce passivity on RL (GC) fitting (Pspice export only). False by default.

Causality Enforcement and Checking Options:

- **-autorenormcc**

  Deprecated: activates automatic renormalization of S-parameters for causality check. False by default.

- **-causal**

  Enforce causality on the input data before performing state-space fitting. False by default.

- **-causality_plots**

  Forces the causality checker to write necessary data (into Touchstone files) to get causality plots in Network Data Explorer. False by default.

- **-causality_tol** <double>

  Specifies the error tolerance for the causality checker. This is 0.01 by default.

- **-cccontinuation** <integer>

  Controls whether the causality checker continues automatically when the default interpolation and integration types lead to an inconclusive causality check. Accepted values are:

  0 to deactivate continuation.

  1 to activate. This is the default value.

- **-ccintegration** <integer>

  Controls the integration method applied in the evaluation of the in-band data and the calculation of the discretization error bound. Accepted values are:

  1 for numerical integration. This is the default value.

  2 for analytic integration.

- **-ccinterp** <integer>

  Controls the interpolation method applied in the evaluation of the in-band data and the calculation of the discretization error bound. Accepted values are:

  1 for cubic spline interpolation. This is the default value.

  2 for rational interpolation.

- **-checkcausality**

  Performs a causality check on the input data and can be combined with **-checkpassivity**. False by default.

- **-causality_thresh**

  Error threshold for causality correction on RL model.

Output and Formatting Options:

- **-foster**

  Generates an HSPICE model in Foster (pole-residue) format. It must be paired with the **-hspice** option and is false by default.

- **-gndopt** <integer>

  Sets the ground reference type for ports. Accepted values are:

  0 for common (suppressed) ground. This is the default behavior. (Previously **-cg**)

  2 for one negative reference terminal per port. This was the old default behavior.

- **-hspice**

  Generates HSPICE-compatible circuit model (*.sp file). False by default.

- **-hspice_rfm**

  Experimental: Generates a state-space model in HSPICE rational function matrix format (*.rfm file). False by default.

- **-o** <string>

  Sets the output file name.

- **-printlog**

  Activates more verbose message printing on the state-space fitter. False by default.

- **-prof** <string>

  Specifies an output file for runtime statistics profile.

- **-pspice**

  Generates a PSPICE-compatible circuit model (*.lib file). False by default.

- **-s** <string>

  Sets the subcircuit name in the output file.

- **-simplorer_s** and **-simplorer_hfss**

  Generates a Simplorer S-element circuit model (*.sml file). False by default.

- **-spectre**

  Generates a Spectre-compatible circuit model (*.cir file). False by default.

- **-sss**

  Generates a state-space model in Nexxim/Designer format file (*.sss). False by default.

- **-tstone** <string>

  Generates Touchstone V1 or V2 output file. You must specify a formatting control input file name.

- **-vrfy**

  Produces a Touchstone file and passivity report from fitted state-space model. False by default.

- **-vrfy_sss**

  Produces an additional Nexxim/Designer format file (*.sss) for comparing the fitted results with the original data. This SSS file is generated alongside the primary output file and is read by Network Data Explorer when it generates comparison plots.

Input Options:

- **-designname** <string>

  Sets the design name in the comment block if called from a desktop environment.

- **-i** <string>

  Sets the input file name.

- **-pnames** <string>

  Specifies a text input file containing the port names with one name per line.

- **-projname** <string>

  Sets the project name in the comment block if called from a desktop environment.

- **-projpath** "<string>"

  Sets the path to the project file that created the model. Place the path in quotes.

- **-reducename** <string>

  Sets the reduce matrix name for Q3D/2D Extractor projects.

- **-setupname** <string>

  Sets the solution setup name for desktop projects.

- **-sweepname** <string>

  Sets the frequency sweep name for desktop projects.

- **-zrefs** <string>

  Specifies a text input file containing the port reference impedances with one per line.

Circuit Design File Formats

This topic contains descriptions of the file formats used by Ansys tools.

Related Topics

Circuit Netlist Format

Touchstone Data Format

Compact FLP Format

CITIfile Format

Ansys Neutral File (ANF) Format

Circuit Netlist Format

The circuit simulator reads the circuit description and the simulation controls from a netlist, which must be in ASCII format. The netlist can be created with a text editor and saved in a text file, which can be imported into the schematic for simulation. A Circuit Netlist Design can also be entered directly using the built-in Netlist Editor. A netlist is automatically generated from a Circuit design entered with the **Schematic Editor**.

Related Topics

Netlist File Structure

Names, Numbers, Constructs, and Expressions

Subcircuits

Options and Controls

External Files

Netlist File Structure

A Circuit netlist specifies both the topology of a single circuit and the Circuit analysis to be run on the circuit, The netlist is parsed line by line. This topic describes the structural features of a Circuit netlist.

Input Lines

The netlist file consists of one or more input lines. Each line starts with a non-blank character and ends with a standard (non-continued) carriage return/line feed (CR/LF). An input line can occupy as many physical lines as needed.

> **Note:**
>
> Except for external filenames, entries in the netlist are not case-sensitive.

Statements

The meaningful lines in a netlist file are called statements. Each kind of statement is interpreted according to its predefined syntax or format. (Statements are sometimes also called cards, a terminology recalling the time when a netlist consisted of a physical 'deck' of punched Hollerith cards.)

All statements except device instance statements begin with a period (**.**) followed by a keyword that identifies the type of statement.

TITLE and END Statements

The first input line in a netlist file is treated as the title of the netlist. The title appears at the top of all output files generated by the netlist.

The netlist can use a TITLE statement to identify the title:

.TITLE*title_text*

The title line can be a comment:

*****title_text*

The title line can be a string enclosed in quotes:

'*title_text*'

The title line can be a normal input line without any special identification:

`title_text`

A netlist file ends when the physical end of file is reached. The netlist can use a END statement to delimit the end of the netlist.

.END

Any characters or lines in a source file after the END statement are ignored. However, an END statement in an INCLUDE or LIB file (See External Files) does not end the netlist. Only the END in the main source file terminates the parsing of the netlist.

Note:

- The first line in a netlist file is always used as the title and any other meaning is ignored.
- A netlist file for the Circuit solver can contain only one circuit description. Multiple circuits in one netlist file are not supported.

MEASURE Statements

The MEASURE statement generates a value which measures a quantity in the simulation data. The MEASURE statement is supported in externally generated netlists. This topic describes the categories of available MEASURE statements, along with a description of the syntax that is supported.

Delay Measurements (OPDelay)

The syntax for delay measurements (time between events) is:

```
.MEASURE [DC|AC|TRAN] result trigger_spec target_spec
```

Where *trigger_spec* can be either:

```
+ TRIGoutput_varVAL=trigger_value|'expression'
 + [TD=delay|'expression']
+ [CROSS=crosses|LAST] | [RISE=rises|LAST] | [FALL=falls|LAST]
```

or

```
+ TRIGAT=begin_time
```

And *target_spec* can be either:

```
+ TARGoutput_varVAL=target_value|'expression'
 + [TD=delay|'expression'|TRIG]
+ [CROSS=crosses|LAST] | [RISE=rises|LAST] | [FALL=falls|LAST]
```

or

```
+ TARG AT=end_time
```

Functions Across Two Events

The syntax for measurements across two events is:

```
.MEASURE [DC|AC|TRAN] result
+ FUNCTIONoutput_varFROM=begin_timeTO=end_time
```

Where *FUNCTION* is **MIN**, **MAX**, **PP**, **AVG**, **RMS**, or **INTEG**

Derivative Function

The syntax for the derivative of a quantity at an event is:

```
.MEASURE [DC|AC|TRAN] result
+ DERIVATIVEoutput_var|'expression' AT=at_value
```

```
.MEASURE [DC|AC|TRAN] result
+ DERIVATIVEoutput_var|'expression' AT=at_value
+ WHENoutput_var=value|output_var|PAR('expression')
 + [TD=delay|'expression']
+ [CROSS=crosses|LAST] | [RISE=rises|LAST] | [FALL=falls|LAST]
```

Where *at_value* is time for TRAN, frequency for AC, and a parameter value for DC analysis.

Locating Points in the Data

The syntax for locating single points in the data is:

```
.MEASURE [DC|AC|TRAN] result
+ WHENoutput_var=value|output_var|PAR('expression')
 + [TD=delay|'expression']
+ [CROSS=crosses|LAST] | [RISE=rises|LAST] | [FALL=falls|LAST]
```

```
.MEASURE [DC|AC|TRAN] result
+ FINDoutput_var|PAR('expr') AT=at_value
+ [TD=delay|'expression']
+ [CROSS=crosses|LAST] | [RISE=rises|LAST] | [FALL=falls|LAST]
```

Where *at_value* is time for TRAN, frequency for AC, and a parameter value for DC analysis.

```
.MEASURE [DC|AC|TRAN] result
+ FINDoutput_var|PAR('expr')
+ WHENoutput_var=value|output_var|PAR('expression')
 + [TD=delay|'expression']
+ [CROSS=crosses|LAST] | [RISE=rises|LAST] | [FALL=falls|LAST]
```

Delay Measurements (OPDelay)

Setting MEASURE Output Variables to Expressions

The syntax for setting a MEASURE output variable using an expression is:

```
.MEASUREoutput_var PARAM= 'expression'
```

The *expression* can reference other MEASURE *output_vars*. The calculated *output_var* may be used in subsequent MEASURE statements.

CONNECT Statement

A CONNECT statement connects two nodes in the netlist. A CONNECT statement that appears inside a subcircuit definition is local to that subcircuit. The CONNECT statement syntax is:

```
.CONNECTvisible_node invisible_node
```

After the CONNECT statement, only the first node, termed *visible_node* in the syntax above, is evaluated in the simulation. The second node, termed *invisible_node* in the syntax, is no longer recognized as a node name in netlist statements. For example:

```
.CONNECT n22 50
```

After this statement, references to node n22 also apply to node 50, while references to node 50 are no longer valid.

Multiple CONNECT statements can connect several nodes to one visible node. For example:

```
.CONNECT n22 50
.CONNECT 50 100
```

After these statements, references to node N22 also apply to nodes 50 and 100, while references to either node 50 or node 100 are no longer valid.

PRINT Statements

A netlist can use one or more PRINT statements to specify the outputs from particular types of analysis. The netlist syntax is:

```
.PRINT [analysis] output1 [output2] ...
```

> **Warning:**
>
> By default, the Circuit solver stores all intermediate values for all nodes. If you do not use a PRINT statement, all node voltages and branch currents are retained so they can be output to the results file. With large circuits or analyses, the amount of data to be stored can cause the simulation to slow down, or even to exceed memory limits (usually a fatal error). PRINT statements specify which nodes and values can be retained, allowing the Circuit solver to discard all other values when they are no longer needed. Use of PRINT statements can significantly reduce memory usage. Although the range of output values may be limited, in those cases where memory is an issue a PRINT statement may be required to allow the computation to complete successfully.

> **Note:**
>
> PROBE statements in imported netlists are internally converted to PRINT statements.

ALTER Statements

The ALTER statement directs the Circuit solver to re-simulate a netlist using changed values for parameters. The netlist may contain any number of ALTER statements.

> **Note:** Library and include files cannot contain ALTER statements.

The syntax is:

```
$ Rest of netlist
.ALTER [title1]
$ ALTER block statements
[.ALTER title2
$ ALTER block statements]
...
.END
```

The ALTER statements should come after all other statements in the netlist except for the END statement. An ALTER block can contain the following kinds of statements:

- .DEL LIB
- .INCLUDE
- .IC
- .LIB
- .MODEL
- .NODESET
- .OPTIONS
- .PARAM
- .PRINT
- .SUBCKT
- .ENDS
- .TEMP
- Analysis statements (See Analysis Controls for a list)
- Device instances, including subcircuits.

In the ALTER block, assign a new value to any device instance name, model name, netlist parameter name, or option name on the main netlist. You can also define new device instances, models, subcircuits, netlist parameters, and options. In the simplest case, the netlist contains just one ALTER statement. The Circuit solver first simulates the netlist on the beginning up to the first ALTER statement. The Circuit solver then reanalyzes the netlist, using any changes made in the ALTER statement, up to the END statement. The simulation includes the analyses specified in the original netlist, plus any analyses added in the ALTER block. All the analyses use the values of parameters, etc. set in the ALTER block.

When more than one ALTER block is present, the Circuit solver first simulates the netlist on the beginning up to the first ALTER statement, as before. The Circuit solver then reanalyzes the netlist, using any changes made in the first ALTER block, up to the second ALTER statement. The simulation includes the analyses specified in the original netlist, plus any analyses added in the first ALTER block. All the analyses use the values of parameters, etc. set in the first ALTER block.

The Circuit solver next re-analyzes the netlist, using any changes made in the second ALTER block, up to the next ALTER statement (or up to the END statement, if there are just two ALTER blocks). The simulation includes the analyses specified in the original netlist, plus any analyses added in the first ALTER block, plus any analyses added in the second ALTER block. All the analyses use the values of parameters, etc. set in the second ALTER block.

The Circuit solver continues re-simulating the circuit until the last ALTER statement has been processed. Following is an example:

```
* Test of ALTER blocks

.param myval=1000

V1 1 0 sin(0 1 myval)
R1 1 2 1000
C1 1 2 0.001
R2 2 0 1000

.TRAN 1e-6 2e-3
.PRINT TRAN V(2)

.ALTER first_one
.param myval=2000
.HB TONES=2000 MAXk=3
.PRINT HB V(2)

.ALTER second_one
.param myval=3000
.LNA POI 3 1000 3000
.PRINT LNA V(2)

.END
```

For this example, the Circuit solver runs a transient analysis with netlist parameter **myval** equal to 1000. After reading ALTER block **first_one**, the Circuit solver runs a transient analysis and a harmonic balance analysis, both with **myval** equal to 2000. Finally, after reading ALTER block **second_one**, the Circuit solver runs a transient analysis, a harmonic balance analysis, and a linear network analysis, all with **myval** equal to 3000.

The .DEL LIB and LIB statements allow an ALTER block to control which library files are active for each variation on the netlist. Using these statements, avoid rerunning any previous analyses and only run the analyses specified within specific ALTER blocks. Put all the affected simulation commands into a library file referenced with a LIB statement, either in the outer netlist or in an ALTER block. Then an ALTER block uses DEL LIB to delete the unwanted statements.

Suppose the library file **analyses.lib** contains the following blocks:

```
*analyses.lib
.LIB T1
.TRAN 1e-6 2e-3
.PRINT TRAN V(2)
.ENDL
.LIB H1
.HB TONES=2000 MAXk=3
.PRINT HB V(2)
.ENDL
.LIB L1
.LNA POI 3 1000 3000
.PRINT LNA V(2)
.ENDL
```

The netlist on the previous example could be modified as follows:

```
* Test of ALTER blocks with Library file

.param myval=1000

V1 1 0 sin(0 1 myval)
R1 1 2 1000
C1 1 2 0.001
R2 2 0 1000

.LIB 'analyses.lib' T1

.ALTER first_one
.param myval=2000
.DEL LIB 'analyses.lib' T1
.LIB 'analyses.lib' H1

.ALTER second_one
.param myval=3000
.DEL LIB 'analyses.lib' H1
.LIB 'analyses.lib' L1

.END
```

With this netlist, the Circuit solver first runs the transient analysis specified in the library block T1, using netlist parameter **myval** equal to 1000. Next, after reading ALTER block **first_one**, the Circuit solver deletes the transient analysis library block T1 and executes the harmonic balance analysis specified in library block H1, using netlist parameter **myval** equal to 2000. Last, after reading ALTER block second_one, the Circuit solver deletes the harmonic balance analysis library block H1 and executes the linear network analysis specified in library block L1, using netlist parameter **myval** equal to 3000.

Continuation Lines

To improve readability, an input line can contain intermediate carriage return/line feeds. The intermediate line breaks must be continued or 'escaped' using one of the following character combinations.

- A backslash (\) before a CR/LF suppresses the interpretation of the CR/LF as the end of the input line. Any whitespace (blanks or tabs) before the slash is retained, and any whitespace at the beginning of the continuation line is also retained. A valid comment can appear on the line after the backslash.

```
   M25 nd ng ns nb model_1 L=1.1e-3 W=2.2e-3 \
 M=2 DTEMP=25 OFF
```

- A double backslash (\\) before a CR/LF also suppresses the interpretation of the CR/LF as the end of the input line. Any whitespace before the double slash is ignored. Any whitespace on the same line after the double slash is ignored. Any whitespace at the beginning of the continuation line is ignored. A valid comment can appear on the line after the double backslash. This form is useful when a path must be broken over a line without any intervening spaces.

```
   .INCLUDE './long_directory_name/long_subdirectory_name\\
/file_name'
```

- A plus sign (+) as the first character in a line (after a CR/LF at the end of the previous line) indicates that the new line is a continuation of the previous line. Blank lines and valid comment lines between the previous line and the continuation line are ignored.

```
   M25 nd ng ns nb model_1 L=1.1e-3 W=2.2e-3
+ M=2 DTEMP=25 OFF
```

A line continued with a plus sign is interpreted as part of the statement, even if the previous line ended with a comment. For example:

```
R1 1 2 ; comment

+ 1000
```

Is the same as:

```
R1 1 2 1000
```

Comments

A comment is text that is not part of any statement.

A semicolon (;) anywhere outside of quotes makes the rest of the line a comment.

```
; This is a comment
M1 1 2 3 4 model_1; This is a comment
```

A line that begins with an asterisk (*), a dollar sign ($), or a double slash (//) is treated as a comment and is not interpreted.

```
* This is a comment
$ This is a comment
// This is a comment
```

Characters in a line become comments when they follow an asterisk (*), a dollar sign ($), or a double slash (//) that is both outside of quotes and directly preceded by one or more blanks or tabs (“whitespace”).

```
M2 5 6 7 8 model_1 $ This is a comment
M3 9 10 11 12 model_1 * This is a comment
M4 13 14 15 16 model_1 // This is a comment
```

> **Note:**
>
> The asterisk also serves as the sign for multiplication in expressions. When an expression includes the asterisk character, the expression *must* be enclosed in single or double quotes, otherwise any part of the expression that follows the asterisk is treated as a comment.

Characters in a comment are not interpreted in any way. A single or double backslash (\ or \\) in a comment is not interpreted as a continuation character. Also, a plus sign (+) at the beginning of a continuation line does not continue a comment on the previous line, but instead continues the interpreted statement on the previous line. To continue a comment, the netlist must use one of the comment characters on the continuation line. Here are some examples.

The next two examples of device instance statements show invalid uses of continuation characters for a comment. The second line in each example causes a syntax error.

```
V1 1 2 0.5V ; This is an \\
invalid continued comment
```

```
V2 3 4 1.5V ; This is another
+ invalid continued comment
```

The simplest way to continue a comment is just to begin the next line with a comment character:

```
V1 1 2 0.5V ; This is one way
* to continue a comment
```

You can also continue the command itself, then add the rest of the comment:

```
V2 3 4 // This is another
+ 1.5V // way to continue a comment
```

Names, Numbers, Constructs, and Expressions

A name is a user-defined token for a node, device instance, model, parameter, or other object in the netlist. The rules for names are presented here for comparison.

Node Names

A node name can contain any number of alphanumeric characters, and the following special characters:

```
! # $ % ^ * - + _ [ ] < > /
```

However, a node name containing a special character that also represents an arithmetic operator (**-** **+** ***** /) or a relational operator (< >) cannot be used in expressions, with or without quotes.

Node names can begin with one of the special characters

```
! # % _
```

Net names entered in the **Schematic Editor** may not contain spaces ( ), ampersands (&), or asterisks(*).

> **Note:**
>
> All node names in the Circuit solver are treated as strings, even when they appear as integers such as “12”. No numeric conversion is performed. For example, some simulators treat node “001” as node “1”. The Circuit solver treats these as two different node names.

Device Instance Names

The name for a circuit device instance begins with a predefined character representing the device type.

Element-Identity Characters

Character	Element Type
A	Distributed element or Ansys-developed element
B	Nonlinear dependent voltage source
C	Capacitor
D	Diode
E	Voltage-controlled voltage source (VCVS)
F	Current-controlled current source (CCCS)
G	Voltage-controlled current source (VCCS)
H	Current-controlled voltage source (CCVS)
I	Independent current source
J	JFET or MESFET
K	Coupled inductor
L	Inductor
M	MOSFET
N	Maxwell SPICELink N element
P	Piecewise linear dependent current source
Q	BJT
R	Resistor
S	S-parameter element (N-port) or Voltage-controlled switch
T	Lossless transmission line
U	Lossy transmission line
V	Independent voltage source
W	Lossy transmission line
X	Subcircuit

For example, a resistor instance name must begin with the letter R, a capacitor name must begin with the letter C, a MOSFET name must begin with the letter M. Characters after the first can be alphanumerics or the special characters:

```
! # $ % * - + _ [ ] < > / :
```

Note that the period character (".") is NOT valid in device instance names.

Model Names

The name for a device model begins with any alphabetic character. Characters after the first can be alphanumerics or the special characters:

```
! # $ % * - + _ [ ] < > / :
```

Netlist Parameter Names

The name for a netlist parameter defined in a .PARAM statement begins with any alphabetic character. Characters after the first can be alphanumerics or the special characters:

```
! # $ % _ [ ] : .
```

> **Note:**
>
> For compatibility with existing netlist formats, the Circuit solver accepts parameters whose names include symbols for arithmetic and relational operations, such as (+, -, *, /, <, >, =). However, parameter names that include such symbols cannot be used in quoted expressions.

Device Instance Parameter and Model Parameter Names

Device instance parameters and model parameters have names defined in the Circuit models. They must be spelled as documented, but are case-insensitive.

Delimiters

Delimiters separate names and other tokens in an input line. Delimiters are the blank, tab, comma, equal sign (=), and parentheses ( ). A name may not contain any of these delimiters.

The period (.) character also indicates subcircuit hierarchy (See Subcircuits), and thus becomes another kind of delimiter. The period may not be used in device instance names.

Numbers

Real numbers in the netlist file can be represented as decimal, exponential, or scaled numbers, or as decibels (exponentially scaled numbers).

A decimal number consists of the numerals 0 through 9, with or without a decimal point, and with or without a leading unary plus or minus: 123, 12.30, +5.0, -20.

An exponential number consists of a decimal number mantissa, the letter **e** (for “exponent”) and a decimal number exponent. The decimal equivalent is the mantissa multiplied by 10 to the power of the exponent.

Exponential Number Examples

Exponential Number	Interpretation	Decimal Equivalent
1e5 or 1e+5	1×10^5	100000
1.0e-5	1.0×10^{-5} ($1/10^5$)	.00001
-5e4	-5×10^4	-50000
-2.05e-4	-2.05×10^{-4} ($-2.05/10^4$)	-.000205

A scaled number consists of a decimal number and an alphabetic token representing a scale factor: 20K, 1.5P, -7.25MEG. The following table lists the scale factor tokens and their exponential equivalents.

Numeric Scale Factors

Factor	Interpretation	Exponential Multiplier
ATTO	atto	1e-18
F	femto	1e-15
P	pico	1e-12
N	nano	1e-9
U	micro	1e-6
M	milli	1e-3
K	kilo	1e+3
MEG	mega	1e+6
X	mega	1e+6
G	giga	1e+9
T	tera	1e+12

Note: By default, the single-letter token “A” as a numeric suffix represents amperes (AMP). If the netlist contains the statement **.OPTIONS UNIT_ATTO**, the single-letter suffix “A” represents 1e-18 (ATTO). In either case, the token AMP (with or without any trailing characters) is interpreted as amperes, and the token ATTO (with or without any trailing characters) is interpreted as 1e-18.

A decibel number uses one of the units DB or DBM. A number (*nn* in the following examples) with one of these units scales exponentially:

*nn*DB is scaled as 10 ^ (*nn* * 0.1).

*nn*DBM is scaled as 10 ^ (*nn* * 0.1 – 3).

In both of these examples, the ^ sign denotes exponentiation and the * sign denotes multiplication.

Note: Any extra letters after the DB or DBM prevent a match for DB or DBM (unlike other units, where additional letters are ignored). Here are some examples:

5K = 5000

5KOHM = 5000

5KAPPLES = 5000

5DB = 3.1622777

5DBW = 5

5DBM = 0.0031622777

5DBMW = 5

Netlist Parameters

The netlist can define parameters or variables to represent temporary values. Netlist parameter names follow the rules for names given earlier. Netlist parameters can be global or local, depending on where in the netlist they are defined. Global netlist parameters are defined with a .PARAM statement in the main body of the netlist, as discussed in this topic. Local netlist parameters are defined within a subcircuit definition, see Subcircuits.

The .PARAM statement has the following syntax:

```
.PARAMparam_name =expression [param_name =expression ...]
```

Each .PARAM statement can define one or more netlist parameters. A netlist parameter can be referenced as a value or in an expression anywhere in the netlist, including subcircuit definitions. Global netlist parameter definitions can appear anywhere in the netlist. The same parameter-name can appear in two or more definitions. However, the last value assigned to any parameter is used everywhere in the netlist.

A reference to a global parameter anywhere in the netlist returns the value of the corresponding expression. A global parameter reference can precede or follow the statement that assigns the parameter a value (variable substitution occurs after the entire netlist has been parsed). A reference to a global netlist parameter that has not been given a value somewhere in the netlist results in an error.

Here is an example of a netlist parameter used to sweep the value of a resistor device:

```
.PARAM A=1
R10 10 0 R=A*1000
.TRAN 1e-3 1e-2 SWEEP A POI 3 1 3 5
```

The SWEEP specification generates three simulations. In the first simulation, A is equal to 1 so R10 has resistance 1000 ohms. In the second simulation, A is equal to 3 so R10 has resistance 3000 ohms. In the third simulation, A is equal to 5 so R10 has resistance 5000 ohms.

A parameter name cannot be a keyword such as TEMP.

Distribution Functions

A distribution function (**UNIF**, **AUNIF**, **GAUSS**, **AGAUSS**, **LIMIT**) specifies a statistical distribution of values around a nominal value. See Netlist Parameters for details on defining netlist parameters.

Instances of Circuit Devices

A circuit is described by specifying one or more circuit devices. Each instance of a device is specified with an instance statement.

A device instance statement has the general syntax:

```
instance_name nodes [model_name] [instance_parameters]
```

Models of Devices

A simulator model contains the physical and mathematical description for one type of circuit device. .MODEL statements in the netlist reference the simulator models. The general syntax for a .MODEL statement is:

```
.MODELmodelname type [model_parameters]
```

Each .MODEL statement provides a unique model name to be referenced by the instance statements for devices that use the given model. The *type* entry tells the kind of element, for example, D for diode model or NMOS for N-MOSFET model. The .MODEL statement can also specify values for physical and electrical parameters that then apply to all instances that reference that model name. Many device models have several different implementations identified via LEVEL and VERSION model parameters.

```
.MODEL MOS903 NMOS LEVEL=50 LER=1.92e-007 WER=1.00028e-005
```

Expressions

An expression is a combination of operands, operators, and built-in functions that evaluates to a numeric or Boolean value. An expression should be delimited with single or double quotes to distinguish it from other entry types.

> **Note:**
>
> All expressions evaluate to double precision numeric constants.

A quoted expression can contain blanks separating the operands and operators. Without the quotes, blanks are token delimiters. For example, 'A + B + 1' is interpreted as an expression, while
A + B + 1 represents five separate tokens, whose interpretation depends on the context (most likely generating an error).

> **Note:**
>
> Identifiers (names) containing the characters for arithmetic or relational operators cannot be used in expressions, with or without quotes.

Operands

An operand can be one of the following:

A decimal, exponential, scaled, or exponential scaled (decibel) number.

A voltage or current source (e.g., V2, I34).

A node voltage or branch current (V(23), V(10,12), I(R1))

A user-defined parameter or variable (myparam) [See Netlist Parameters for details on user-defined parameters.]

A valid sub-expression enclosed in parentheses.

> **Note:** Complex numbers are not valid as operands or as arguments to built-in functions in expressions in Circuit netlists. Expressions with this limitation include component parameter expressions, expressions in **.param** statements, and sweep variable expressions.

An expression can consist of just a single operand. However, because a name can include the special characters +, -, and *, even a single-token expression should be enclosed in single or double quotes. For example, node-1 is interpreted as a node name, while ‘node-1’ is evaluated as an expression.

Operators

The Circuit solver supports arithmetic and relational operators.

Arithmetic operators include unary and binary operators.

The supported unary operators are listed in The following table:

Unary Operators

Operator	Description
-	Unary minus or negation.
!	Logical “NOT.” !0 = 1, !1 = 0, !a = 0 for a<>0

The unary operators can precede any operand, for example, –1, !(A<B).

A binary operator appears between its two operands. The supported binary arithmetic operators are the character combinations listed in the following table:

Binary Arithmetic Operators

Operator	Operation
+	Add
–	Subtract
*	Multiply
/	Divide
**	Power
^	Power (synonym for **)

The supported relational operators are the character combinations listed in the following table:

Relational Operators

Operator	Operation
>	Is greater than
<	Is less than
= or ==	Is equal to
<> or !=	Is not equal to

Operator	Operation
>=	Is greater than or equal to
<=	Is less than or equal to

The relational operators are binary, that is, they require two operands. The relational operators return a value of 1 if the relation between the two operands is true, or a value of 0 if the relation is false.

Built-In Functions

A built-in function has the format:

`function_name(arguments)`

Each function has a specified number of arguments and format for entering its arguments. The built-in functions return arithmetic, trigonometric, or Boolean values. The following tables list the built-in functions valid in Circuit netlist expressions. In the tables, arguments *x* and *y* can themselves be any expressions.

Built-In Arithmetic Functions

Function	Return Value
ABS(*x*)	Absolute value of *x*.
CEIL(*x*)	Returns the smallest integer greater than or equal to *x* (rounds *x* toward positive infinity)
DB(*x*)	(sign of *x*)*20*$\log_{10}$(abs(*x*)) [*x* expressed in deciBels, with the sign of x]
DB_ NONINVERTING (*x*)	20*$\log_{10}$(abs(*x*)) [*x* expressed in deciBels]
EXP(*x*)	e^x
FLOOR(*x*)	Returns the largest integer less than or equal to *x* (rounds *x* toward negative infinity)
FMOD(*x*,*y*)	*x* -*i* **y* for some integer *i*, such that if *y* is non-zero the result has the same sign as *x* and magnitude less than *y*.
INT(*x*)	Integer portion of *x*
LOG(*x*)	$\log_e(x)$, -300.0*$\log_e(10)$ if $x \le 0$
LOG10(*x*)	$\log_{10}(x)$, -300 if $x \le 0$
MAX(*x*,*y*)	x if x > y, y if y ≥ x
MIN(*x*,*y*)	x if x < y, y if y ≤ x
NINT(*x*)	Rounds *x* up or down to nearest integer. A fractional part of .5 is rounded UP to the nearest integer. NINT(-1.5) is -1 and NINT(1.5) is 2.

Function	Return Value
POW(*x*,*y*)	x^y
PWR(*x*,*y*)	SGN(*x*)$\|x\|^y$
SIGN(*x*,*y*)	abs(*x*) if y ≥ 0 -abs(*x*) if y < 0
SGN(*x*)	-1 if *x* < 0, 0 if *x* = 0, 1 if *x* > 0
SQRT(*x*)	Positive square root of *x*

Built-In Trigonometric Functions

Function	Return Value
ACOS(*x*)	Inverse cosine of *x* in radians
ASIN(*x*)	Inverse sine of *x* in radians
ATAN(*x*)	Inverse tangent of *x* in radians
ATAN2 (*x*,*y*)	For any real arguments (*x*, *y*) not both zero, the angle in radians between the positive X-axis and the point (*x*, *y*). The angle is positive for *y* > 0 and negative for *y* < 0.
COS(*x*)	Cosine of *x* in radians
COSH(*x*)	Hyperbolic cosine of *x* in radians
HYPOT (*x*,*y*)	SQRT(*x***2 + *y***2)
SIN(*x*)	Sine of *x* in radians
SINH(*x*)	Hyperbolic sine of *x* in radians
TAN(*x*)	Tangent of *x* in radians
TANH(*x*)	Hyperbolic tangent of *x* in radians

Built-In Boolean Functions

Function	Return Value
IS_EQUAL(*x*,*y*)	1 if *x* is equal to *y*, 0 otherwise
IS_NOT_EQUAL(*x*,*y*)	1 if *x* is not equal to *y*, 0 otherwise
IS_GREATER(*x*,*y*)	1 if *x* is greater than *y*, 0 otherwise
IS_GREATER_EQUAL (*x*,*y*)	1 if *x* is greater than or equal to *y*, 0 otherwise
IS_LESS(*x*,*y*)	1 if *x* is less than *y*, 0 otherwise
IS_LESS_EQUAL(*x*,*y*)	1 if *x* is less than or equal to *y*, 0 otherwise

Conditional Expressions

A conditional expression is an expression that returns a Boolean value (1 when the condition is true, 0 when the condition is false). The relational operators in Table 6 return Boolean values (using the format X *relop* Y). The built-in Boolean functions in Table 9 also return Boolean values.

Conditional Assignments

A conditional assignment uses one of the following formats:

```
variable = '(cond_expr)?true_value:false_value'
variable = 'if(cond_expr,true_value,false_value)'
```

The quotes are optional, but are recommended for clarity. The conditional expression *cond_expr* is any expression that evaluates to a Boolean value. When the conditional expression is true, the variable is assigned the *true_value*. When the condition is false, the variable is assigned the *false_value*.

Examples:

```
A = '(X>Y) ? 0 : 1'
A = 'if(X>Y,0,1)'
```

In both examples, when the conditional expression X>Y is true, variable A is set to 0, and when the conditional expression is false, A is set to 1.

Operator Precedence in Circuit Expressions

Expressions are evaluated using the following order of operator precedence.

1. Parentheses may be used to group operations. Operations in the innermost pair of parentheses are evaluated first. The result of the operations within a set of parentheses becomes an operand for the next level of evaluation.
2. The arguments to functions can be expressions. An expression that is used as the argument to a function is evaluated before the function is called.
3. Functions have higher precedence than operators outside of functions. When functions are nested, the innermost function is evaluated first.
4. Unary minus (-) and unary NOT (!) operators are applied to their operands before binary operators are applied. If parentheses are not used, a unary operator applies only to the very next operand.
   - For example, if A=2 and B=4, '-A+B' is evaluated as '(-A)+(B)' with value 2, while '-(A+B)' is evaluated as '(-A)+(-B)' with value -6.
   - Similarly, if A=2 and B=4, the expression '!A<B' is evaluated as '(!2)< 4', with value 1 or true, while '!(A<B)' is evaluated as '!(1)', with value 0 or false.

5. Power (**) is the highest binary operator. When two exponentiation operations are present without parentheses to control the precedence, evaluation is left to right.
   - For example, 'A**B**C' is evaluated as '(A**B)**C)'.
6. Multiply (*) and divide (/) have equal precedence. When two multiplies or divides are present without parentheses to control the precedence, evaluation is left to right.
   - For example, '-A/B/C' is evaluated as '((-A)/B)/C'.
7. Add (+) and subtract (-) have equal precedence. When two additions or subtractions are present without parentheses to control the precedence, evaluation is left to right.
   - For example, 'A+B-C' is evaluated as '(A+B)-C'.
8. Relational operators (>, <, =, ==, <>, >=, <=) have equal precedence. When two relational operators are present without parentheses to control the precedence, evaluation is left to right.
   - For example, 'A>B=C' is evaluated as '(A>B)=C'.

Evaluation of Expressions

With two exceptions, an expression is evaluated once at the beginning of simulation and the resulting value is used thereafter. The two exceptions are expressions in sweep specifications and expressions in the instance statements for behavioral devices.

An expression in a sweep specification (See Variable Sweeps) is evaluated once on every sweep step. Note that an expression in one level of a multi-level sweep cannot depend either on a sweep value that is set in another level of the same sweep, or on a value that is set in any level of another sweep statement.

An expression that determines the value of a behavioral device such as an expression-based resistor or behavioral voltage source is evaluated continuously, that is, on every Newton-Raphson iteration and every step of a discrete-time simulation.

A **.OPTION** statement cannot contain an expression.

Divisors

Division operations in most contexts are subject to the **minimum_divisor** option. This option is provided to improve compatibility with HSPICE®.

This option can be set in the netlist:

```
.option minimum_divisor=val
```

Alternatively, it can be set in the Circuit configuration file, **cirsim.cfg**:

```
minimum_divisor=val
```

or on the command line:

```
-minimum_divisor=val
```

(See Command Line Controls for details on the command line and configuration file.)

If **minimum_divisor** is not set, it defaults to 0. The value to match HSPICE^{tm} is 1e-28.

During expansion of subcircuits any divisor anywhere in the netlist whose absolute value is too small is replaced by the (signed) minimum_divisor. A divisors in the range (-minimum_divisor < divisor < > 0) is replaced by (-minimum_divisor) and a divisor in the range (0 <= divisor < minimum_divisor) is replaced by (minimum_divisor).

Replacement can happen when the entire expression is sweep-dependent or voltage-dependent. But it does not happen if just the divisor is sweep-dependent or voltage-dependent. Expression-based components such as the expression-based resistor are not affected by minimum_divisor.

A parallel option, **behavioral_minimum_divisor**, prevents division by zero during the evaluation of Nexsys behavioral or functional components.

To prevent division by zero, the option must be set when set to a nonzero value in the netlist:

.option behavioral_minimum_divisor=*val*

Subcircuits

A subcircuit is a named set of elements and other circuitry that can be used multiple times in a netlist. The subcircuit definition can include local parameters. Different instances of the subcircuit can have different element values.

Subcircuit Definition

A subcircuit is defined in a block. The syntax of a subcircuit block is:

```
.SUBCKTsub_name [boundary_nodes] [local_params]
subcircuit_statements
.ENDS
```

> **Note:**
>
> .MACRO is a synonym for .SUBCKT and .EOM is a synonym for .ENDS.

Subcircuit Statements

The body of the subcircuit definition can contain device instance statements, model definitions, and the definitions and instances of nested subcircuits, all of which are local to the subcircuit. Node names are generally local to the subcircuit, but can be made global, as discussed in later topics.

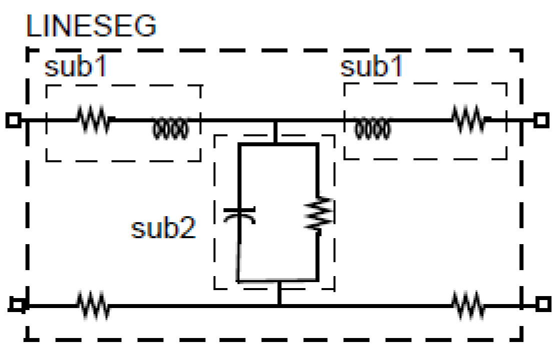
Note:

- Global control statements such as .TEMP and analysis statements such as .DC, .TRAN, and .PRINT that occur within .SUBCKT definitions are treated as if they are in the top-level circuit. .IC and .NODESET statements inside subcircuit definitions apply only to nodes in the subcircuit (See Scope of Initialization Controls).
- Subcircuits automatically inherit higher-level library configurations.

Example with Subcircuits

As an example for use in later discussions, Figure 1 diagrams the circuit topology of a segment of lossy transmission line. Netlist examples to follow shows how this circuit can be implemented with subcircuits:

Lossy Transmission Line Segment with Subcircuits



Here are example netlist definitions for the two internal subcircuits. Later netlist examples shows how to combine the subcircuits into a multi-segment transmission line and illustrate how to modify these definitions to incorporate additional features of subcircuits.

```
.SUBCKT sub1 sub1in sub1out
R1 sub1in sub1local ser_res
L1 sub1local sub1out ser_ind
.ENDS
.SUBCKT sub2 sub2in sub2out
C3 sub2in sub2out par_cap
R3 sub2in sub2out par_res
.ENDS
```

> **Note:**
>
> This example is just for demonstration purposes, not to be used in an actual circuit. The U transmission line model in the Circuit device library allows you to specify both the overall line length and the number of lumped line segments to be used to simulate the transmission line, so you do not have to construct the line segment by segment.

Boundary Nodes

The nodes in the list of *boundary_nodes* on the subcircuit definition line represent points of connection to the external circuit. In subcircuit sub1 above, the nodes sub1in and sub1out are the boundary nodes.

> **Note:**
>
> A subcircuit can be defined without any boundary nodes, in which case it can connect only to nodes defined as **GLOBAL** nodes. See Global Nodes in Subcircuits for further information.

The boundary nodes can have any names except 0, GROUND, GND, and GND!, which stand for the global reference node. The boundary node names do not represent actual nodes in the circuit, they are just dummy names used in the subcircuit.

When a node name inside the subcircuit definition is in the list of *boundary_nodes*, the element node is connected to the corresponding external circuit node when the subcircuit is instantiated.

In this subcircuit example, sub1, nodes sub1in and sub1out are the boundary nodes. Naming nodes sub1in in the R1 instance statement and sub1out in the L1 instance statement create connections to the external circuit nodes:

```
.SUBCKT sub1 sub1in sub1out
R1 sub1in sub1local ser_res
L1 sub1local sub1out ser_ind
.ENDS
```

Local Nodes

Node names in the subcircuit definition that are NOT in the list of boundary nodes are local nodes whose names are valid only inside the subcircuit. In subcircuit sub1 above, the node sub1local is a local node. A local node requires a hierarchical node name to be referenced outside of the subcircuit definition, as described in Hierarchical Names.

CONNECT statements in subcircuit definitions are local to the subcircuit. See CONNECT Statement for further information.

Global Nodes in Subcircuits

The names for the global reference node (0, GROUND, GND, or GND!) are global. These names used anywhere See the single global reference node. A device instance node with one of the global reference node names is connected directly to the global reference node from any level of subcircuit nesting.

A subcircuit local node also becomes global when the node name is specified in a .GLOBAL statement either inside or outside a subcircuit definition, in the main module or in a module imported with an .INCLUDE or .LIB statement.

.GLOBAL*nodename1* [*nodename2* ...]

All nodes with a .GLOBAL node name are connected to that signal at every level of the netlist.

```
.SUBCKT sub4 N1 N2 $ Series resistors
R1 INPUT N1 R=0.05
R2 N2 0 R=0.1
.ENDS
.GLOBAL INPUT $ R1 in subcircuit sub4 connects to this node.
```

> **Note:**
>
> 1. The .GLOBAL statement should appear in the top-level netlist. If a .GLOBAL statement occurs in a .SUBCKT definition, it is treated as if it appeared in the top-level circuit.
> 2. Use the .GLOBAL statement sparingly. Special attention should be paid to .GLOBAL definitions appearing inside a subcircuit library. Common names defined as .GLOBAL inside a circuit library may unintentionally affect the overall circuit that references the library.
> 3. In the **Schematic Editor**, adding a Global port to a schematic node inserts a .GLOBAL statement into the netlist that represents the circuit.

Local Device Instances

Device instance names that appear within a subcircuit definition are local to the subcircuit. Local device names can duplicate names in the outer circuit and in other subcircuits. A local device instance requires a hierarchical device name to be referenced outside of the subcircuit definition, as described in Hierarchical Names.

Subcircuit Instance Statement

To create an instance of a subcircuit, the netlist adds an X-element statement:

X*xxxcircuit_nodessub_name* [*local_params*] [**M=** val]

The subcircuit instance name must begin with the letter **X** (or **x**), followed by any number of valid device name characters. See Device Instance Names for the list of valid characters.

The circuit nodes in the X-element statement are the nodes in the actual circuit that connect to the *boundary_nodes* in the order given by the subcircuit definition.

The *sub_name* is the name of the subcircuit definition to instantiate.

The optional **M** parameter on a subcircuit instance statement specifies a value that multiplies the value of the **M** model parameter in every Circuit device instance referenced in the subcircuit. For example, suppose the netlist contains the following statements:

```
.SUB MTEST 1 2 3 4
M1 x1 x2 x3 x4 mos1
M2 x5 x6 x7 x8 mos1 M=3
...
.ENDS
...
X1 101 102 103 104 MTEST M=2
```

When subcircuit X1 is instantiated, its parameter setting **M=2** causes the instance of MOSFET M1 to have an effective instance parameter **M=2**, since by default the MOSFET instance parameter **M** has value 1. MOSFET M2 has an effective instance parameter **M=6** (3 times 2).

Here is the basic definition for the subcircuit LINESEG that combines the two subcircuits to make one segment of a lossy transmission line:

```
.SUBCKT LINESEG insig inref outsig outref
Xleft insig local1 sub1
Xright outsig local1 sub1
Xmid local1 local2 sub2
Rleft inref local2 ser_res
Rright local2 outref ser_res
.ENDS
```

Here is a circuit fragment that builds a multi-segment transmission line by connecting together two instances of subcircuit LINESEG.

```
* Lossy transmission line fragment
Xtrans1 send1 GND 10 20 LINESEG
Xtrans2 10 20 rcv1 GND LINESEG
```

Subcircuit Parameters

The subcircuit definition can reference global netlist parameters defined with .PARAM statements external to the subcircuit block. The subcircuit definition can also define one or more local subcircuit parameters for use in the subcircuit statements. The names and values of subcircuit parameters are valid only inside the subcircuit. A subcircuit parameter can be created, or the value of an existing subcircuit parameter can be changed, by the X-element subcircuit instance statement.

The subcircuit statements can contain references to global and local parameters. A reference to a global parameter returns the value set in the external netlist. A reference to a local parameter returns the value set by the subcircuit definition or by the subcircuit instance. If both the subcircuit definition and the subcircuit instance assign values to the same local parameter, the value assigned in the subcircuit instance takes precedence. A subcircuit parameter value assigned in a subcircuit instance statement affects all references to that parameter in all subcircuits that are instantiated as a result of the called subcircuit.

> **Note:**
>
> A subcircuit may define a parameter named **M** using the **.param** statement, but this subcircuit parameter does not affect the value of the **M** parameter on subcircuit instance statements or on Circuit device instances in the subcircuit.

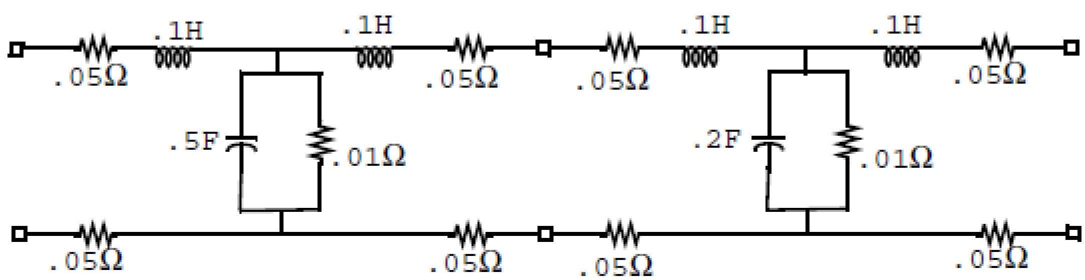
The example subcircuits given earlier reference the following parameters: ser_res and par_res for the serial and parallel resistors, ser_ind for the serial inductors, and par_cap for the parallel capacitor. The earlier examples do not show where the parameters get their values. Here is a more complete netlist fragment which gives definitions for all the parameters.

```
* Lossy transmission line fragment with parameters
.PARAM ser_res=.05
Xtrans1 send1 GND 10 20 LINESEG
Xtrans2 11 21 rcv1 GND LINESEG par_cap=.2
.SUBCKT LINESEG insig inref outsig outref ser_ind=.1
Xleft insig local1 sub1
Xright outsig local1 sub1
Xmid local1 local2 sub3
Rleft inref local2 ser_res
Rright local2 outref ser_res
.ENDS $ End of LINESEG
.SUBCKT sub1 sub1in sub1out
R1 sub1in sub1local ser_res
```

```
L1 sub1local sub1out ser_ind
.ENDS $ End of sub1
.SUBCKT sub2 sub2in sub2out par_cap=.5 par_res=.01
C3 sub2in sub2out par_cap
R3 sub2in sub2out par_res
.ENDS $ End of sub2
.END $ End of Netlist
```

Here is the resulting circuit fragment, with device values:

Lossy Transmission Line Segment with Device Values



- The parallel resistors in subcircuits sub1 and LINESEG all reference the global parameter ser_res, which is set in the .PARAM statement to 0.05 Ohm.
- The serial inductors in subcircuit sub1 reference local parameter ser_ind, which is given the value 0.1 Henry in the definition of subcircuit LINESEG, which instantiates sub1.
- The parallel capacitor in subcircuit sub2 references local parameter par_cap. Parameter par_cap is given a default value 0.5 Farad in the definition of subcircuit sub2, and this value is applied in the instance of sub2 that results on the subcircuit instance Xtrans1. However, the subcircuit instance Xtrans2 sets the value of par_cap to 0.2 Farad. The subcircuit instance Xtrans2 instantiates subcircuit LINESEG, which in turn instantiates subcircuit sub2, so the subcircuit instance value of par_cap, 0.2 Farad, overrides the default.
- The parallel resistor in subcircuit sub3 references local parameter par_res. Parameter par_res is given the value 0.01 Ohm in the definition of subcircuit sub2, and this value is applied in both instances of sub2 that are created.

Overriding the Precedence of Subcircuit Parameters

A subcircuit parameter can be given a value at different levels of nesting. There are two orders of precedence, global and local. Here is an example; consider the definitions of parameter foo.

```
.PARAM foo=1
```

```
X1 10 20 SUBA foo=2
.SUBCKT SUBA n1 n2 foo=3
X2 n1 n2 SUBB foo=4
.ENDS $ End of SUBA
.SUBCKT SUBB n3 n4 foo=5
R1 n3 n4 R=foo
.ENDS $ End of SUBB
.END
```

Global Precedence

With global precedence, the outermost assignment of subcircuit parameter foo has highest precedence. In the example above, the precedence of these assignment statements is as follows:

- The .PARAM statement has the highest precedence.
- The assignment in statement X1 has next highest precedence.
- The assignment in .SUBCKT SUBA has next highest precedence.
- The assignment in statement X2 has next highest precedence.
- The assignment in .SUBCKT SUBB has lowest precedence.

Thus, resistor R1 in subcircuit SUBB is instantiated with resistance R = 1 Ohm.

Local Precedence

Setting the global option PARHIER=LOCAL invokes local precedence, the reverse of global precedence. With PARHIER=LOCAL in effect, the precedence in the example above is:

- The assignment in .SUBCKT SUBB has highest precedence.
- The assignment in statement X2 has next highest precedence.
- The assignment in .SUBCKT SUBA has next highest precedence.
- The assignment in statement X1 has next highest precedence.
- The .PARAM statement has the lowest precedence.

With PARHIER=LOCAL in effect, resistor R1 in subcircuit SUBB has resistance 5 Ohm.

> **Note:**
>
> Setting PARHIER to any value starting with the letter L enables local precedence. Omitting PARHIER or setting it to any value not starting with the letter L leaves the precedence global.

Overriding the Default

Circuit netlists generated from schematics insert PARHIER=LOCAL by default. To override this default, uncheck the **Use local parameter scoping** group box on the Nexxim Circuit Design Global Options or Nexxim Circuit Design-Level Options window. See Circuit Design Settings Reference for further information.

Hierarchical Names

subcircuit_element[.*subcircuit_element* ...].*name*

The subcircuit element is the X-element that instantiates the subcircuit (See Subcircuit Instance Statement). For example, to reference resistor Rleft in the instance of subcircuit LINESEG that was instantiated by subcircuit instance Xtrans1, the netlist has:

```
Xtrans1.Rleft
```

The hierarchical device name can include any number of lower level subcircuit calls, separating them with periods. For example, to reference node n25 in the instance of subcircuit sub3 that was instantiated indirectly by subcircuit instance Xtrans2, the netlist has:

```
Xtrans2.Xmid.n25
```

Subcircuit element Xtrans2 calls subcircuit LINESEG, which in turn uses subcircuit element Xmid to reference subcircuit sub3 containing the appropriate node.

Nested Subcircuits

A subcircuit definition can contain the definitions of other “nested” subcircuits. A .SUBCKT definition that is nested within another .SUBCKT definition is local to the enclosing subcircuit. Instances of the nested subcircuit can appear only in the scope of the enclosing subcircuit definition. Here is a skeletal example:

```
.SUBCKT Outer 1 2
* Definition of subcircuit Outer
.SUBCKT Inner 1 2
* Definition of nested subcircuit Inner
.ENDS Inner
X1 3 4 Inner $ Valid subcircuit instance
.ENDS Outer
* Start of top-level circuit
X2 10 20 Outer $ Valid subcircuit instance
X3 30 40 Inner $ Invalid subcircuit instance
.END top-level circuit
```

In this example, the definition of subcircuit Inner is nested in the definition of subcircuit Outer. Instance X1 is valid since subcircuit Inner is known inside the definition of subcircuit Outer. However, instance X3 is not valid, since subcircuit Inner is not recognized outside of subcircuit Outer.

Note that instances of non-nested subcircuits are valid both inside and outside of subcircuit definitions, as shown in the transmission line example given earlier.

Options and Controls

The following options and controls are available for working with Circuit netlist formatting.

Analysis Controls

To perform simulations, the netlist uses one or more of the analysis control statements:

Analysis Controls

Statement	Analysis
.DC	DC operating point analysis
.HB	Harmonic balance
.LNA	Linear network analysis
.OSC	Oscillator analysis
.TRAN	Transient analysis
.TV_ NOISE	Time-varying noise analysis

A **SWEEP** specification can be added to any of these analysis statements to perform a series of analyses using a parameter sweep.

Separate help topics explain the complete syntax for each of the analysis statements.

> **Note:**
>
> .AC analysis statements are processed correctly in netlists imported from other simulator environments. Linear Network Analysis (.LNA) matches their functionality.

Netlist Options

The netlist can include one or more .OPTIONS statements to specify controls that apply to the entire netlist.

Simulator options are used by one or more simulation tools. See the chapters on the individual tools for further information.

Model options are used to create instances of particular kinds of device models. See the Circuit Component Models help topics for further information.

A **.OPTIONS** statement cannot contain an expression.

Initialization Controls

The netlist also contains initialization controls: NODESET statements for DC operating point and IC statements for transient analyses.

NODESETs

The netlist can include one or more .NODESET statements. Voltages assigned with .NODESET are applied at the start of the DC operating point analysis, to provide the initial values for the first Newton-Raphson iteration. Any node voltages not specified with NODESET are initialized to zero.

The syntax is:

.NODESET V(*node1***)=***value1* [**V(***node2***)=***value2* **...**]

For example, to initialize node 12 to 4.5 volts and node in2 to 2.23 volts:

```
.NODESET V(12)=4.5 V(in2)=2.23
```

Initial Conditions Statements

The netlist can include one or more .IC statements to force nodes to particular voltages. Voltages set with .IC are applied throughout the DC operating point analysis. Any node voltages not specified with .IC are computed.

The syntax is:

.IC V(*node1***)=***value1* [**V(***node2***)=***value2* **...**]

For example, to force node 12 to 4.5 volts and node in2 to 2.23 volts:

```
.IC V(12)=4.5 V(in2)=2.23
```

Scope of Initialization Controls

.NODESET and .IC statements at the outer level of the netlist can reference nodes that are defined in the top-level circuit directly, as in the examples above.

.NODESET and .IC statements at the outer level of the netlist can also use hierarchical names to reference nodes within subcircuit instances. For example:

```
.SUBCKT test in out
R1 in middle 100
R2 middle out 200
.ENDS test
Xtest 1 2 test
.NODESET V(Xtest.middle)=5
```

.NODESET and .IC statements inside a subcircuit definition apply only to those nodes that appear in the subcircuit definition. For example:

```
.SUBCKT test in out
R1 in middle 100
R2 middle out 200
.NODESET V(middle)=6
.ENDS test
```

In this subcircuit definition, the voltage on node middle is initialized at 6 volts in all instances of the subcircuit. Since the .NODESET is internal to the subcircuit definition, the node it references is treated as a subcircuit internal node.

External Files

The Circuit solver allows input lines to be copied into the netlist from three types of external files: include files, library files, and vector files. External files can also be referenced in netlist statements defining the N-element, a frequency-dependent S-parameter element compatible with Ansys Maxwell Spice, and the SP frequency-dependent model for S-elements and W-element transmission lines.

File References

A reference to an external file in a Circuit netlist can use one of the following formats:

```
filename
"filename"
'filename'
"path/filename"
'path/filename'
"$environment_variable/path/filename"
'$environment_variable/path/filename'
```

A *filename* that contains only characters valid for node names can be entered without quotation marks. If the *filename* contains any special characters not valid for node names, it must be enclosed in single or double quotation marks.

The *path* specifies the directory where the *filename* is to be found. When *path* is given, the combined *path* and *filename* must be enclosed in single or double quotes. Pathnames must See subdirectories using a syntax that is compatible with the platform operating system. Some operating systems use a forward slash (/) as a path separator, while others use a backslash (\). The examples above use a forward slash, but a backslash is the valid separator used in Windows™ systems. The system ensures the consistent and compatible use of slash designators in netlist file paths for each of the following platforms: Windows and Linux™.

> **Note:**
>
> The forward slash (/) is a valid separator for file path resolution in Circuit netlists under the Windows operating system.

A *filename* with a *path* that contains a fully rooted path (identifying the drive and all intermediate directories) is called an *absolute* file reference. An absolute file reference is searched for along the specified path. Absolute file references that include the drive are valid only in Windows systems.

The value of an environment variable can be used to supply the leading part of the *path*. For example, if an environment variable MyLibrary has been defined as ‘C:\Project5\Models’, the file reference ‘$MyLibrary\Model5’ is expanded to ‘C:\Project5\Models\Model5’.

A *filename* without a *path*, or with a *path* that is not fully rooted, is called a *relative* file reference. The search for a relative file reference depends on the location of the netlist file that is the source for the file reference. Linux systems should use relative paths.

- If the source netlist file is a top-level netlist that is referenced by a Circuit .ADSN or a netlist .NDSN file, the relative-referenced *filename* is searched for first in the directory where the .ADSN or .NDSN file resides, then in the directory where the application was started.
- If the source netlist file is not a top-level netlist, or the Circuit solver is being run as a standalone process, the relative-referenced *filename* is searched for starting in the directory where the source netlist file resides, then in the directory where the Circuit solver was started.

Include Files

The **.INCLUDE** statement identifies a file that is to be inserted in its entirety into the netlist at the point of the **.INCLUDE** statement:

```
.INCLUDE file_reference
```

The file-reference has the form discussed in the previous topic. For example:

```
.INCLUDE 'diode_model_statements'
.INCLUDE 'models\diode_models'
.INCLUDE 'C:\simulator\model_statements\diode_model_statements'
```

The complete contents of the include file are added to the calling netlist. The included file can contain further **.INCLUDE** files to be processed.

> **Note:**
>
> None of the included files can See any other file in the sequence of included files. A loop of included files causes an unrecoverable error condition. However, the Circuit solver cannot detect this error in advance, so no warning message is given.
>
> Include files cannot contain ALTER statements.

Library Files

Library files contain one or more blocks of netlist statements. Each block in the library file has the format:

.LIB*block_name*

lib_block_statements

.ENDL

The library file block can contain any valid netlist statements. LIB blocks are processed after all the other statements in the netlist have been processed, and the order of execution of LIB block statements cannot be predetermined.

To read and process the statements from a library file block, the netlist uses a library call the form:

.LIB*file_referenceblock_name*

The rules for *file_reference* are the ones given in the module on File References.

A **.LIB** block in a given library file can contain calls to blocks in the same library file or in other library files, but a **.LIB** block may not call a block with its own block name in its own file (that is, a **.LIB** block cannot call itself).

> **Note:**
>
> Library files cannot contain ALTER statements.

To delete a library file on the netlist, use the DEL LIB statement:

`.DEL LIB`*`file_referenceblock_name`*

Vector Files

Digital vector files contain vector pattern definitions to be used as input stimuli to the circuit.

To read and process the statements from a vector file, the netlist uses a vector file statement of the form:

`.VEC`*`file_reference`*

The rules for *file_reference* are the ones given in the module on File References.

> **Note:**
>
> The Circuit solver supports vector files containing only input vector types. Output and bidirectional vector types are not processed.

Touchstone Data Format

Touchstone data files contain S-, Y-, Z-, H-, or G-parameter data for a single device. The data file typically contains information about the device type and measurement conditions followed by sequential data lines. Touchstone® is a registered trademark of Agilent Technologies, Inc.

The Circuit solver supports both Version 1.1 and Version 2.0 of the Touchstone specification.

- *Touchstone® File Format Specification*, Rev 1.1, Copyright© 2002 by the EIA/IBIS Open Forum.
- *Touchstone® File Format Specification*, Version 2.0, Copyright© 2009 by TechAmerica.

> **Note:**
>
> The Circuit solver does not support the Mixed-Mode N-Port Parameters in the Touchstone 2.0 Specification.

See Circuit S-Parameter Technical Notes for details on how the Circuit solver handles Touchstone files.

Compact FLP Data File

Compact FLP files contain network parameter data for one or more devices. Each device's data is entered on sequential lines to form a dataset. Each such dataset begins with a mandatory header line that is used to identify the set followed by a mandatory description line. The header format is the same for all *N*-port data sets. Two-port network data sets may contain a subsequent secondary header for associated noise data. When multiple data sets are stored in a data bank file, they must be uniquely named.

FLP data bank files may contain multiple data sets, but they must all be of the same port size—that is, all two-port, all three-port, et cetera —but not mixed.

> **Note:**
>
> Ansys tools can read and import, but not write and export, data in the FLP format. The FLP format is documented to support backward compatibility only.

Related Topics

FLP Data Header Form

One-Port Data

Two-Port Data

Three-Port Data

Four-Port Data

NPORT Data (Five or More Ports)

FLP Noise Data

Conventions for Compact FLP File Formats

FLP Data Header Form

This header form is used in the following forms:

DEVname Min_Freq Max_Freq N_Freq R0 Data_Code
Mandatory_description_line

where:

Table 37:

DEVname	An alphanumeric device name beginning with a letter character
Min_Freq	Lowest frequency for which measured data is provided
Max_Freq	Highest frequency for which measured data is provided
N_Freq	Number of frequencies for which measured data is provided
R0	Reference resistance of *S*-parameter data
Data_Code	A numeric code identifying the type of signal parameters data: • 1 or S identifies S-parameter data in magnitude phase

	format • 2 or Y identifies Y-parameter data in real imaginary format • 3 or Z identifies Z-parameter data in real imaginary format • 31 or DB identifies S-parameter data where magnitudes are in [dB] • –1 or RI identifies S-parameter data in real imaginary format
Mandatory_description_ line	Data set description text line must not begin with ! or *
Comment lines	Must begin with *

One-Port Data

Form

```
header
f1 a1 b1
f2 a1 b1:
```

where **header** is the FLP data bank header.

The entries **f1, f2, . .** represent frequencies, and the data pairs **a1, b1** represent real values identifying the data for each dataset. The actual numbers that replace **a1** and **b1** for each frequency are either the magnitude and angle (in degrees) or real and imaginary parts of the complex reflection coefficient, admittance or impedance.

Example

```
DIODE2 6GHZ 8GHZ 3 50 1
PIN diode in 'ON' state with 20mA in pill package
* From the HP device library
 6GHZ 0.073 6.3
 7GHZ 0.09 9.9
 8GHZ 0.103 10.4
```

In this example, S-parameter data is given for three frequency points. The name of the dataset is **DIODE2**.

Two-Port Data

Form

```
header
[f1] a11 b11 a21 b21 a12 b12 a22 b22
[f2] a11 b11 a21 b21 a12 b12 a22 b22
 :
```

where **header** is the FLP data bank header.

The entries **f1, f2, ...** represent frequencies. The indexing system in the form above identifies the 11, 21, 12 and 22 matrix elements of the network parameters at each sweep point. Depending on the format used, the matrix coefficients **a** and **b** are magnitude and phase (in degrees) or real and imaginary components of the data type that is specified by the code.

Three-Port Data

Form

```
header
* Data for first frequency
[f1] a11 b11 a12 b12 a13 b13
 a21 b21 a22 b22 a23 b23
 a31 b31 a32 b32 a33 b33
* Data for second frequency
[f2] a11 b11 a12 b12 a13 b13
 a21 b21 a22 b22 a23 b23
 a31 b31 a32 b32 a33 b33
.
.
.
```

where **header** is the FLP data bank header.

The entries **f1, f2, ...** represent frequencies. The indexing system in the form above identifies the matrix elements of the network parameters at each sweep point. Depending on the format that is used, the matrix coefficients **a** and **b** are magnitude and phase (in degrees) or real and imaginary components of the data type that is specified by the code.

Example

```
MY_DIPLEXER 2GHZ 6GHZ 3 50 1
2-GHz low-pass and 6-GHz high-pass diplexer, distributed design
 2.000GHZ 0.183 7.6 0.993 -130.7 0.016 151.6
 0.993 -130.7 0.183 -73.7 0.0 22.1
 0.016 151.6 0.0 22.1 0.993 125.5
 4.000GHZ 0.345 -39.9 0.8 49.8 0.504 76.7
 0.8 49.8 0.526 -51.5 0.3 136.4
 0.504 76.7 0.3 136.4 0.81 -4.3
 6.000GHZ 0.036 91.8 0.0 -145.9 0.999 -122.5
 0.0 -145.9 0.999 -115.9 0.0 89.6
 0.999 -122.5 0.0 89.6 0.04 -155.5
```

Four-Port Data

Form

```
header
* Data for first frequency
[f1] a11 b11 a12 b12 a13 b13 a14 b14
 a21 b21 a22 b22 a23 b23 a24 b24
 a31 b31 a32 b32 a33 b33 a34 b34
 a41 b41 a42 b42 a43 b43 a44 b44
* Data for second frequency
[f2] a11 b11 a12 b12 a13 b13 a14 b14
 a21 b21 a22 b22 a23 b23 a24 b24
 a31 b31 a32 b32 a33 b33 a34 b34
 a41 b41 a42 b42 a43 b43 a44 b44
.
```

where **header** is the FLP data bank header.

The entries **f1, f2, ...** represent frequencies. The indexing system in the form above identifies the matrix elements of the network parameters at each sweep point. Depending on the format that is used, the matrix coefficients **a** and **b** are magnitude and phase (in degrees) or real and imaginary components of the data type that is specified by the code.

Example

```
DIST_NWK_A45 4GHZ 4.5GHZ 3 50 1
Bias and signal distribution network P/N A-45
 4.000GHZ 0.196 -59.8 1.0 -166.2 0.032 135.0 0.001 -98.6
 1.0 -166.2 0.197 -92.9 0.0 -22.4 0.001 103.9
 0.032 135.0 0.0 -22.4 1.00 107.6 0.000 45.2
 0.001 -98.6 0.001 103.9 0.000 45.2 1.000 0.1
 4.250GHZ 0.410 153.3 0.6 -59.9 0.641 -19.7 0.000 -73.7
 0.6 -59.9 0.426 3.2 0.6 -95.7 0.000 -149.8
 0.641 -19.7 0.6 -95.7 0.44 85.0 0.000 -109.6
 0.000 -73.7 0.000-149.8 0.000 -109.6 1.000 0.1
 4.500GHZ 0.045 92.5 0.0 -148.2 0.999 -136.9 0.000 -87.3
 0.0 -148.2 1.000-121.6 0.0 72.4 0.000 121.9
 0.999-136.9 0.0 72.4 0.05 174.2 0.000 133.2
 0.000 -87.3 0.000 121.9 0.000 133.2 1.000 0.0
```

NPORT Data (Five or More Ports)

Form

```
header
* Data for first frequency
```

```
[f1] a11 b11 a12 b12 a13 b13 ... a1N b1N
 a21 b21 a22 b22 a23 b23 ... a2N b2N
 a31 b31 a32 b32 a33 b33 ... a3N b3N
 :
 aN1 bN1 aN2 bN2 aN3 bN3 ... a1N b1N
* Data for second frequency
[f2] a11 b11 a12 b12 a13 b13 ... a1N b1N
 a21 b21 a22 b22 a23 b23 ... a2N b2N
 a31 b31 a32 b32 a33 b33 ... a3N b3N
 :
 aN1 bN1 aN2 bN2 aN3 bN3 ... a1N b1N
 :
```

where **header** is the FLP data bank header.

The entries **f1, f2, ...** represent frequencies. The indexing system in the form above identifies the matrix elements of the network parameters at each sweep point. Depending on the format that is used, the matrix coefficients **a** and **b** are magnitude and phase (in degrees) or real and imaginary components of the data type that is specified by the code.

Example

```
* Six-port data
NWA6P 1GHz 2GHz 2 50 1
1GHz 0.11 111 0.12 112 0.13 113 0.14 114 0.15 115
+ 0.16 116
 0.21 121 0.22 122 0.23 123 0.24 124 0.25 125
+ 0.26 126
 0.31 131 0.32 132 0.33 133 0.34 134 0.35 135
+ 0.36 136
 0.41 141 0.42 142 0.43 143 0.44 144 0.45 145
+ 0.46 146
 0.51 151 0.52 152 0.53 153 0.54 154 0.55 155
+ 0.56 156
 0.61 161 0.62 162 0.63 163 0.64 164 0.65 165
+ 0.66 166
2GHz 0.11 111 0.12 112 0.13 113 0.14 114 0.15 115
+ 0.16 116
 0.21 121 0.22 122 0.23 123 0.24 124 0.25 125
+ 0.26 126
 0.31 131 0.32 132 0.33 133 0.34 134 0.35 135
+ 0.36 136
 0.41 141 0.42 142 0.43 143 0.44 144 0.45 145
+ 0.46 146
 0.51 151 0.52 152 0.53 153 0.54 154 0.55 155
+ 0.56 156
```

```
 0.61 161 0.62 162 0.63 163 0.64 164 0.65 165
+ 0.66 166
```

FLP Noise Data

Noise data may also be supplied using a header line that follows the network data:

Form

```
NOI { RN | NC } [x]
[f1] fmin g1 g2 r1
[f2] fmin g1 g2 r1
 :
```

where

RN signifies that the normalized equivalent noise resistance is given as r1
NC signifies that the noise figure in dB is given as r1
x is an optional reference normalization impedance whose default value is 50 ohms.

Each data line that follows begins with the frequency and is a separate dataset, where

fmin = minimum noise figure in dB
g1 and **g2** = magnitude and phase in degrees of the optimum noise input reflection coefficient
r1 = normalized equivalent noise resistance when the RN option is used or the noise figure in dB when the NC option is used.

The noise data does not require the same number of data sets as those given for the preceding network parameters data, however, the dataset entries follow the same rules as the other data types.

Example

The dataset in this example is identified by the set name **HPM02**. The header indicates the frequency range of the S-parameter data and seven data points and reference termination of 50 ohms. The second header line explains that the data is for a transistor with parasitic lead inductances attached to the three terminals. The seven S-parameter data lines follow the header. Three noise data lines are appended to the end of the S-parameter data following an appropriate noise data header.

```
HPM02 6GHZ 12GHZ 7 50 1
HPMP with 0.2nH inductance connected to gate, drain and 0.1nH to
source
* From the HP device library
6GHZ 0.732 -96.3 1.973 99.3 0.051 69.8 0.642 -26.0
7GHZ 0.699 -110.9 1.835 88.2 0.058 70.8 0.629 -30.3
8GHZ 0.673 -124.4 1.708 78.4 0.066 73.2 0.618 -34.8
9GHZ 0.644 -136.9 1.570 69.8 0.073 77.3 0.613 -39.5
10GHZ 0.622 -147.5 1.449 61.7 0.081 82.7 0.610 -44.9
```

```
11GHZ 0.616 -154.8 1.378 53.5 0.093 87.7 0.605 -51.1
12GHZ 0.623 -160.1 1.332 44.6 0.108 90.7 0.602 -58.1
NOI RN
1GHZ 2.4 .632 34.2 0.25
2GHZ 1.8 .758 68.4 0.36
4GHZ 2.6 .470 92.8 0.37
```

Conventions for Compact FLP File Formats

In the Compact FLP file format topic, optional entries such as [f1], [f2] may be used to represent frequencies that must be included if interpolation is required or necessary. They need not be included if a set of data for each simulation frequency is given.

When two or more data sets are supplied in a single data file, they must all follow the same form. If a frequency is included, it is the first entry in each dataset and all the data sets that belong to the same data group must also have a frequency entry. The frequency entry may include a unit code (e.g., 100MHz) or be a number with an optional exponent (e.g., 1E8) in which case the unit code (Hz) is assumed. When frequencies are supplied as part of the data sets, the order of data sets is unimportant within each group; the data sets are automatically sorted internally.

CITIfile Format

CITIfile (“Common Instrumentation Transfer and Interchange”) is an ASCII-based format originally developed by Hewlett-Packard (now Agilent Technologies). CITIfile format files containing frequency-dependent data can be both imported into and exported. A CITIfile file contains a single “package” consisting of a header followed by one or more arrays of data.

> **Note:**
>
> The CITIfile format is case-insensitive. In these help topics, CITIfile keywords are shown in upper case, but that is not required.
>
> A CITIfile file can contain only a single package (header plus data). Packages after the first one in a file is ignored.

Related Topics

CITIfile File Header

CITIfile Data Arrays

CITIfile Example

CITIfile File Header

A CITIfile package must begin with a header. The header consists of the following entries, each occupying one or more lines.

CITIfile Version

The first line in the file identifies the version of the CITIfile format supported by Ansys. This line must be the following:

`CITIFILE A.01.00`

> **Note:**
>
> CITIfile revision A.01.01 defines the statement type “CONSTANT” with or without an accompanying “COMMENT.” The import facility silently ignores all such lines.

Package Name

The second line gives a name for the package. This line is optional. The format is:

`NAME`_`name`_

For example:

`NAME BJTInverter`

> **Note:**
>
> The CITIfile reference defines several keywords for the NAME statement (such as DATA for error-corrected data). The netlist parser does not interpret the NAME entry.

Frequency Information

The next line in the file identifies the data as dependent on frequency magnitude, and specifies the number of frequencies. The format for this line is:

`VAR FREQ MAG`_`number_of_freqs`_

For example, if the data is for ten frequencies, the VAR line is:

`VAR FREQ MAG 10`

Data Types

The next line or lines in the file define the types of the data. Each data type is defined in a line with the format:

`DATA`_`type format`_

The following DATA types are supported by Ansys:

- **S** or **S[***port***,***port***]** (Scattering parameter data. **S** with no port identification is interpreted as **S[1,1]**. Explicit port identification such as **S[1,2]** is required for any other multi-port combination.)
- **Y** or **Y[***port***,***port***]** (Admittance parameter data. **Y** with no port identification is interpreted as **Y[1,1]**. Explicit port identification such as **Y[1,2]** is required for any other multi-port combination.)
- **Z** or **Z[***port***,***port***]** (Impedance parameter data. **Z** with no port identification is interpreted as **Z[1,1]**. Explicit port identification such as **Z[1,2]** is required for any other multi-port combination.)
- **Zo** or **Zo[***port***]** (Terminal characteristic impedance. **Zo** with no port identification is interpreted as **Zo[1]**. Explicit port identification such as **Zo[2]** is required for any port.)
- **Zpi** or **Zpi[***port***]** (Terminal characteristic impedance, power/current. **Zpi** with no port identification is interpreted as **Zpi[1]**. Explicit port identification such as **Zpi[2]** is required for any other port.)
- **Gamma** or **Gamma[***port***]** (Propagation constant parameters. **Gamma** with no port identification is interpreted as **Gamma[1]**. Explicit port identification such as **Gamma[2]** is required for any other port.)

> **Note:**
>
> Zpi is identical to Zo in Circuit designs, and upon import, Zpi becomes Zo. The CITIfile should not specify both Zpi and Zo. If both Zo and Zpi are specified with different values, the values of Zpi take precedence.

The only DATA format supported is **RI** (real/imaginary).

There can be multiple DATA lines in the package. For example:

```
DATA S[1,1] RI
DATA Gamma[1] RI
DATA Zo[1] RI
```

Frequency List

The file header should contain a list of the frequencies at which the data is generated. The number of frequencies in the list should correspond to the number given in the VAR line described earlier. Two formats are supported for frequency listings: VAR_LISTs and SEG_ LISTs.

The VAR_LIST explicitly lists all the frequencies in Hz. The format is:

VAR_LIST_BEGIN

frequency_1

```
...
frequency_n
VAR_LIST_END
```

Here is an example with ten frequencies:

```
VAR_LIST_BEGIN
1000000000
2000000000
3000000000
4000000000
5000000000
6000000000
7000000000
8000000000
9000000000
10000000000
VAR_LIST_END
```

Instead of a VAR_LIST, the CITIfile can use a SEG_LIST. The SEG_LIST specifies a start frequency, stop frequency, and number of frequency points. The format is:

```
SEG_LIST_BEGIN
SEGstart_freq stop_freq number_of_freqs
SEG_LIST_END
```

Ansys tools convert the SEG_LIST into individual frequencies using the following formula:

$F_n = start\_freq + \{(n\text{-}1) \times [(stop\_freq\ \text{-}start\_freq) / (number\_of\_freqs\ \text{-}1)]\}$

The index *n* goes from 1 to *number_of_frequencies*. For example, to produce the same set of frequencies as the VAR_LIST given above, the SEG_LIST is:

```
SEG_LIST_BEGIN
SEG 1000000000 10000000000 10
SEG_LIST_END
```

The VAR_LIST or SEG_LIST is the last item in the header.

> **Note:**
>
> A CITIfile header may contain only a single VAR_LIST or SEG_LIST. Ansys does not support the use of multiple VAR_LISTS.

CITIfile Data Arrays

The CITIfile package should contain one data array for each DATA line in the header. The first array is the data whose type and format was specified in the first DATA line, the second array is for the second DATA line, et cetera. Each array contains one line of data for each frequency in the VAR_LIST block in the header, described earlier.

Here is a data array for the DATA S[1,1] RI line given earlier:

```
BEGIN
-8.515589E-001, -2.006537E-001
-9.498286E-001, -1.361623E-001
-9.761472E-001, -9.717214E-002
-9.862472E-001, -7.472031E-002
-9.910952E-001, -6.048325E-002
-9.937765E-001, -5.072857E-002
-9.954099E-001, -4.365178E-002
-9.964769E-001, -3.829255E-002
-9.972115E-001, -3.409734E-002
-9.977385E-001, -3.072603E-002
END
```

Each Real/Imaginary pair of data points corresponds to one of the ten frequencies in the VAR_LIST given earlier.

> **Note:**
>
> Multiple, indexed data arrays are not supported.

CITIfile Example

Here is a brief CITIfile example:

```
CITIFILE A.01.00
NAME BJTInverter
VAR FREQ MAG 10
DATA S[1,1] RI
VAR_LIST_BEGIN
1000000000
2000000000
3000000000
4000000000
5000000000
6000000000
7000000000
8000000000
```

```
9000000000
10000000000
VAR_LIST_END
BEGIN
-8.515589E-001, -2.006537E-001
-9.498286E-001, -1.361623E-001
-9.761472E-001, -9.717214E-002
-9.862472E-001, -7.472031E-002
-9.910952E-001, -6.048325E-002
-9.937765E-001, -5.072857E-002
-9.954099E-001, -4.365178E-002
-9.964769E-001, -3.829255E-002
-9.972115E-001, -3.409734E-002
-9.977385E-001, -3.072603E-002
END
```

Ansys Neutral File Format

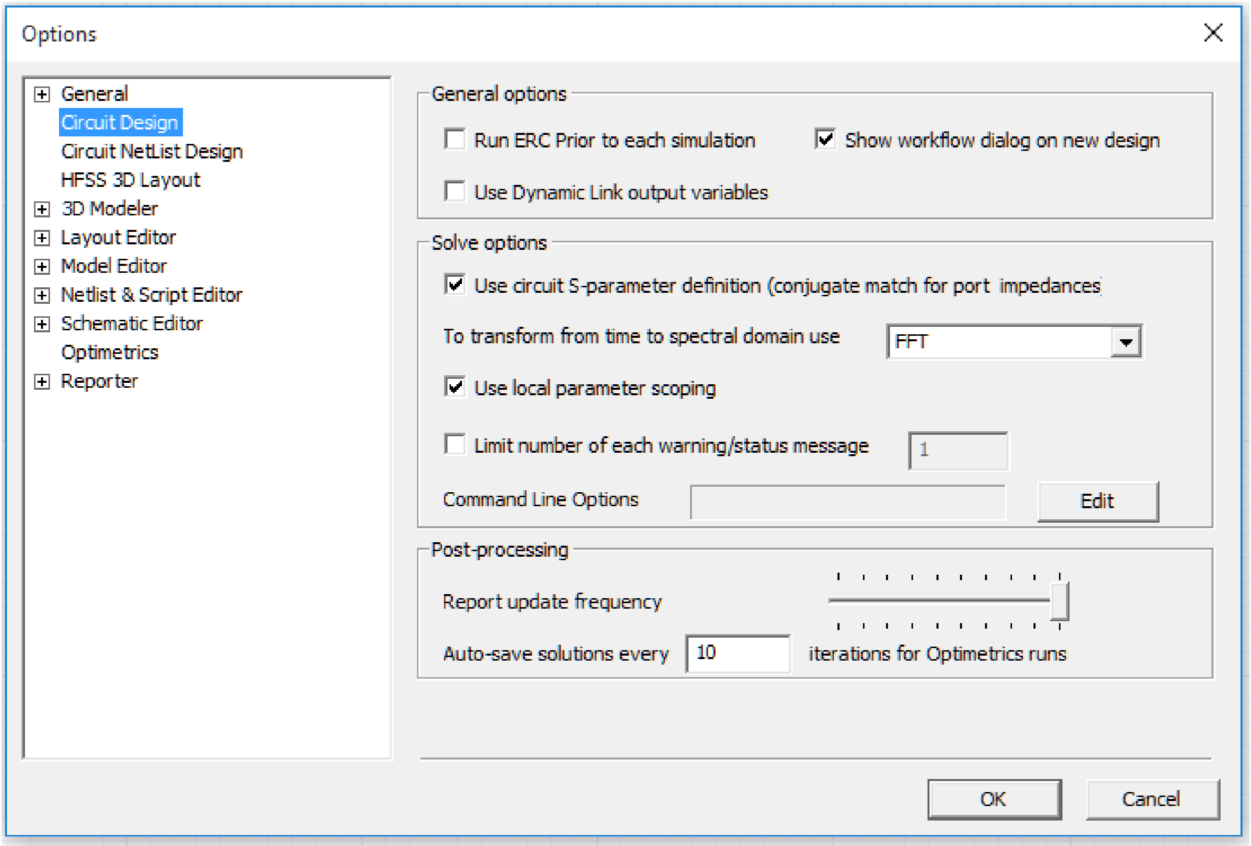
ANF is a public, neutral file format that allows third party tools to exchange design data with Ansys products. Ansys Neutral File (ANF) formatted files are generated either by third party tools, by the “ALinks” program that translates third-party designs, or by Ansys tools when exporting schematic or layout data. An ANF file can contain schematic data, 2D geometry (layout), 3D geometry, and component data. None of the data is required. A given ANF file may contain just schematic data with no layout data, or may contain just layout data with no schematic data.

Circuit Design Settings

Nexxim options specify settings or values that control Nexxim analyses. Nexxim options can apply globally to all designs, to analyses in a specific design, to specific types of analysis, or to the treatment of S-elements and W-elements.

Nexxim Circuit Design — Global Options

To set option defaults that affect all designs, click **Tools>Options>General** to open the **Options** window, then select **Circuit Design** on the left group box.



General Options

Run ERC Prior to each simulation — Runs electrical rules checking for nets or nodes that are electrically connected but not physically connected. The ERC results display in the message window. The simulation proceeds regardless of the results of ERC.

Show Workflow window on new design — Controls the display of the Workflow window when a new circuit design is inserted. When a new design is inserted, a window opens that allows you to associate the new design with a workflow template. For more information, see Workflow Projects.

Use Dynamic Link output variables — Enables the Nexxim simulation to use output variables from dynamically linked HFSS designs. Using the HFSS output variables gives Nexxim access to nearly all types of HFSS data instead of just S-parameters. However, accessing the HFSS output variables can slow down the analysis. See Access to HFSS Output Variables for further information.

Solve Options

Use circuit S-parameter definition — Controls the definition for port impedances. For Nexxim, this option should typically be selected (check box checked). The definition of a “matched port” depends on the application. In applications that deal mostly with circuit quantities (including the Nexxim simulator), a “matched port” has a characteristic impedance that maximizes the power transfer. This is also called a “conjugate match.” In applications that deal mostly with electromagnetic quantities (including HFSS and Planar EM), a “matched port” has a characteristic impedance that maximizes the transfer of the voltage wave. The two definitions differ only by the conjugate of the characteristic impedance of the port. For real port impedances, there is no difference.

To transform from time to spectral domain — Allows you to select the method used to transform on the time domain to the spectral domain for all projects.

- Fourier Integration is very accurate but may be time-consuming for large data sets. The Fourier Integration method computes the projection of the signal on each of the requested harmonics via integration of the inner-product integral.
- FFT (Fast Fourier Transform, the default) achieves faster conversion with a slight reduction in accuracy. The FFT method samples the signal at 2*(maximum harmonic number) points, and outputs the FFT of the sampled signal. The FFT method may not be robust enough to avoid aliasing while converting some time-domain signals to the frequency domain.

Use local parameter scoping – Controls the precedence of parameter handling in subcircuits. By default, option PARHIER=local is inserted in the Nexxim netlist. Uncheck the box to restore global parameter scoping. See Overriding the Precedence of Subcircuit Parameters for further information.

Limit number of each warning message – Lets you specify a maximum number for multiple instances of the same warning message (all error messages are always displayed). Click the check box and enter a whole number, 0 or greater. If the Nexxim solver generates multiple warning messages for the same issue, the number of messages displayed is limited to the value specified. The default is one (1) warning. The logfile includes all status messages, info messages, warning messages, and error messages regardless of this option.

Command Line Options — Additional options that the user can pass to the HPSICE solver during an HSPICE analysis.

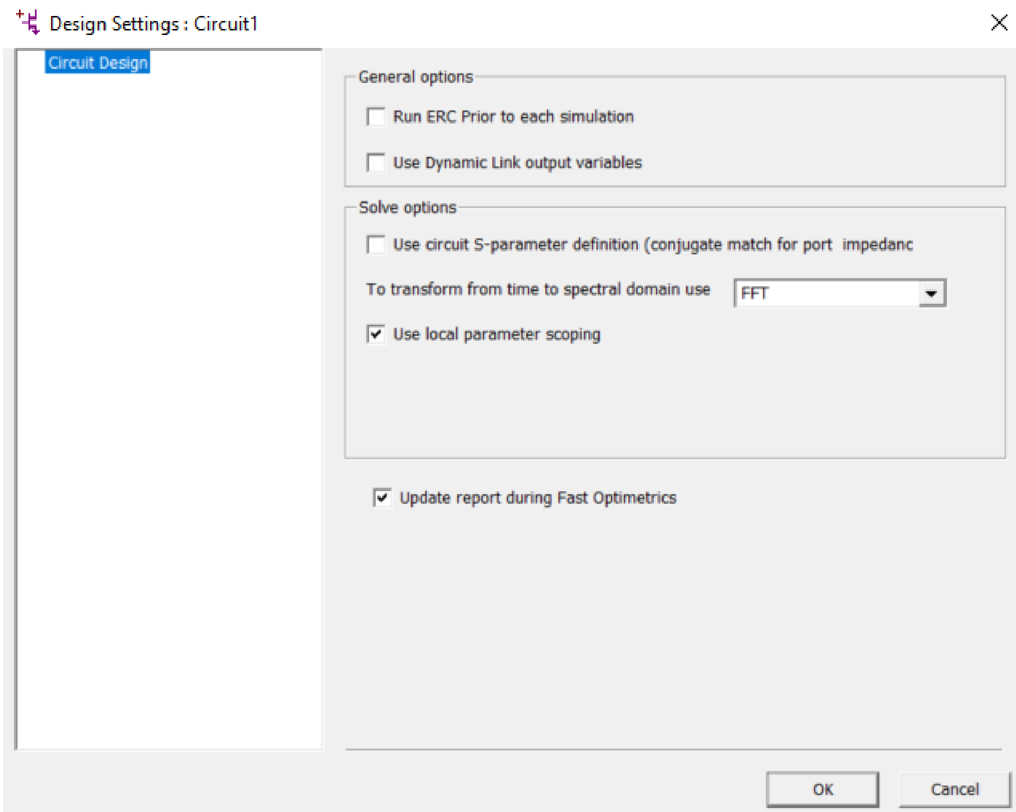
Post-processing

Report update frequency — When dynamic probes (DYNIPROBE or DYNVPROBE) are used in circuit designs, the user can set up plots in schematic. These plots are updated as the simulation progresses. "Report update frequency" controls the frequency at which these reports are updated.

Auto-save solutions every — Used to set **Iterations for Optimetrics runs**, which is the number of iterations after which to do an auto save during an optimetrics analysis.

Nexxim Circuit Design — Local Options

To set option defaults that affect a particular circuit netlist design, on the **Project Manager** window, expand the **Project Tree**. Then right-click the active design and select **Design Settings ...** to open the **Design Settings** window.



General Options

Run ERC Prior to each simulation — Runs electrical rules checking for nets or nodes that are electrically connected but not physically connected. The ERC results display in the message window. The simulation proceeds regardless of the results of ERC.

Use Dynamic Link output variables — Enables the Nexxim simulation to use output variables from dynamically linked HFSS designs. Using the HFSS output variables gives Nexxim access to nearly all types of HFSS data instead of just S-parameters. However, accessing the HFSS output variables can slow down the analysis. See Access to HFSS Output Variables for further information.

Solve Options

Use circuit S-parameter definition — Controls the definition for port impedances. For Nexxim, this option should typically be selected (check box checked). The definition of a “matched port” depends on the application. In applications that deal mostly with circuit quantities (including the Nexxim simulator), a “matched port” has a characteristic impedance that maximizes the power transfer. This is also called a “conjugate match.” In applications that deal mostly with electromagnetic quantities (including HFSS and Planar EM), a “matched port” has a characteristic impedance that maximizes the transfer of the voltage wave. The two definitions differ only by the conjugate of the characteristic impedance of the port. For real port impedances, there is no difference.

To transform from time to spectral domain use — Allows you to select the method used to transform on the time domain to the spectral domain for all projects.

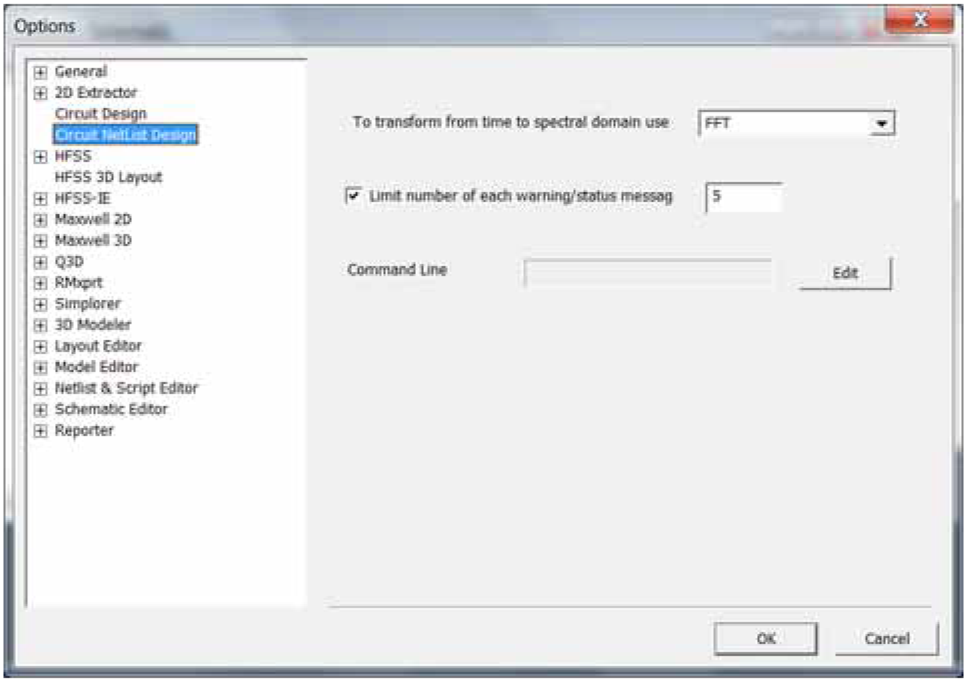
- Fourier Integration is very accurate but may be time-consuming for large data sets. The Fourier Integration method computes the projection of the signal on each of the requested harmonics via integration of the inner-product integral.
- FFT (Fast Fourier Transform, the default) achieves faster conversion with a slight reduction in accuracy. The FFT method samples the signal at 2*(maximum harmonic number) points, and outputs the FFT of the sampled signal. The FFT method may not be robust enough to avoid aliasing while converting some time-domain signals to the frequency domain.

Use local parameter scoping – Controls the precedence of parameter handling in subcircuits. By default, option PARHIER=local is inserted in the Nexxim netlist. Uncheck the box to restore global parameter scoping. See Overriding the Precedence of Subcircuit Parameters for further information.

Update report during Fast Optimetrics – If a report has all variables set to "nominal", the report is updated during fast-optimetrics if this option is checked.

Nexxim Circuit Netlist Design — Global Options

Selecting **Netlist format** on the **Circuit** menu allows you to set the format for netlists opened with **File>Open**. The choices are **HSPICE** and **Spectre**. The initial setting is **HSPICE**. The format is reset automatically to correspond to the format of the file being opened. To set option defaults that affect all circuit netlist designs, click **Tools>Options >General** to open the **Options** window, then select **Netlist Design** from the left group box.



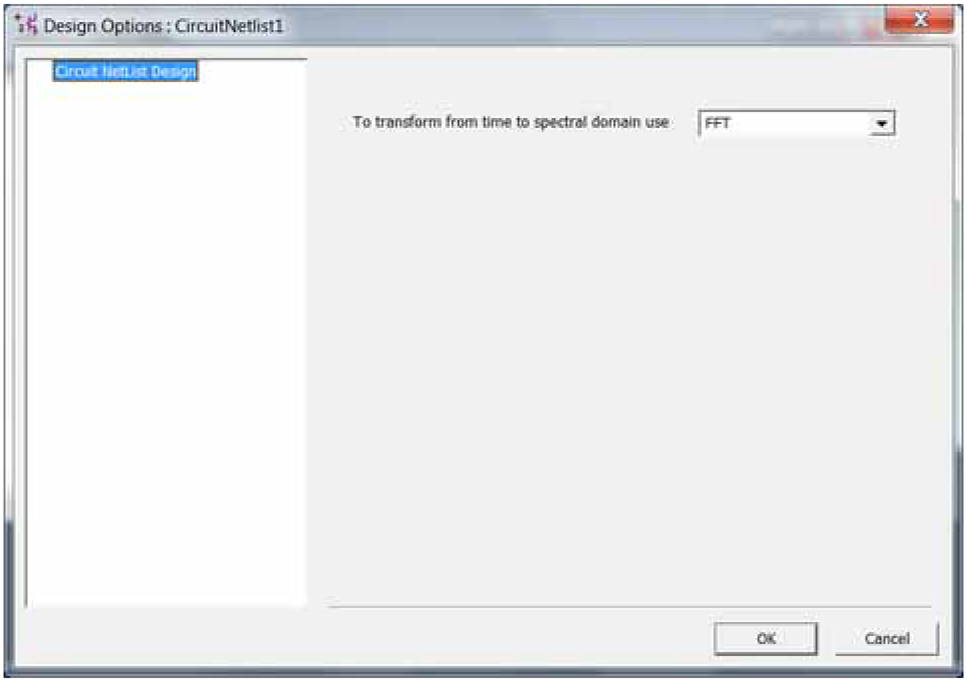
To transform on the time domain to the spectral domain for the selected netlist design. The default method, Fourier Integration, is very accurate but may be time-consuming for large data sets. The alternative method uses the Fast Fourier Transform to achieve faster conversion with a slight reduction in accuracy.

Limit number of each warning message – Lets you specify a maximum number for multiple instances of the same warning message (all error messages are always displayed). Click the check box and enter a whole number, 0 or greater. If the Nexxim solver generates multiple warning messages for the same issue, the number of messages displayed is limited to the value specified. The default is one (1) warning. The logfile includes all status messages, info messages, warning messages, and error messages regardless of this option.

Command Line — Additional options that the user can pass to the HPSICE solver during an HSPICE analysis.

Nexxim Circuit Netlist Design — Local Options

To set option defaults that affect a particular circuit netlist design, select the design in the **Project Manager** window, then select **Design Options** to open the **Nexxim Options** window.



To transform on the time domain to the spectral domain allows selection of the method used to transform on the time domain to the spectral domain for the selected netlist design.

- The default method, Fourier Integration, is very accurate but may be time-consuming for large data sets.
- The alternative method, Fast Fourier Transform, uses the FFT to achieve faster conversion with a slight reduction in accuracy.

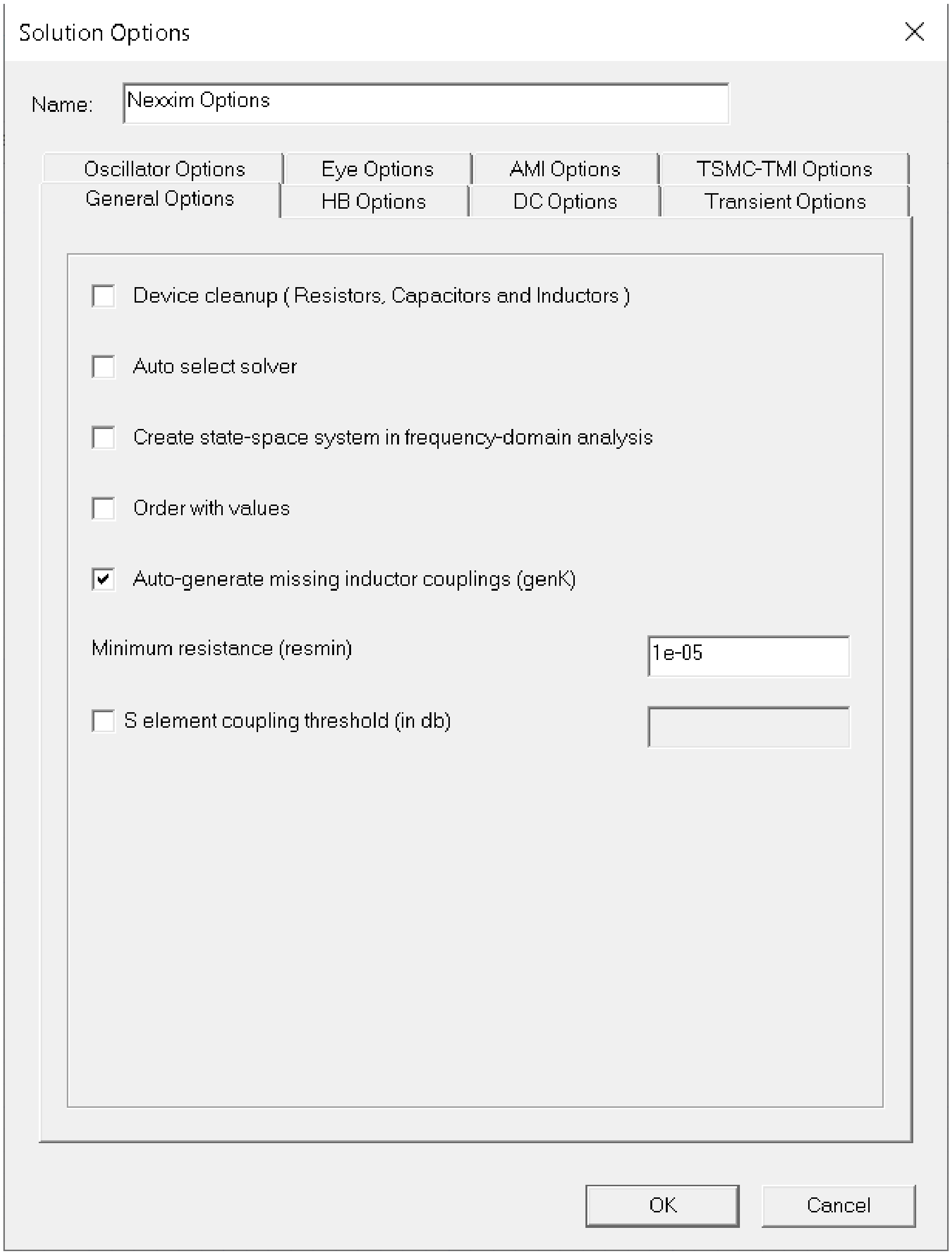
Analysis-Specific Options

In a schematic, define sets of analysis-specific options with the **Add Nexxim Solution Options ...** selection. Then select a defined option set when creating or editing a solution setup. Using this mechanism, use the same option set in multiple analysis setups, or use different option sets in the same solution setup by editing the setup.

There are multiple ways to create an option set:

- From the **Circuit** menu, select **Add Nexxim Solution Options ...**.
- From the **Project Manager** window, expand the **Project Tree** and [*active design folder*]. Then right-click **Analysis** and select **Add Nexxim Solution Options ...** to open the **Solution Setup** window.
- Click **Select** in the **Solution Options** group box in a **Solution Setup** window.

Any of these actions opens the **Solution Options** window.



- Click the **Name** field in the **Solution Options** window to assign a unique name to the set of options you are defining, or accept the default name.
- The **Solution Options** window has a tab for each of the Circuit analysis types: General Options, HB Options, DC Options, Transient Options, Oscillator Options, Eye Options, AMI Options, and TSMC-TMI Options.

General Options Tab

- The **General Options** tab sets options that apply to all Nexxim analyses. See Global Analysis Options and Global Device Options for more information on these options.
- **Device Cleanup** enables Nexxim to delete unused passive elements (sets **device_cleanup=1**)
- **Auto select solver** enables Nexxim to automatically select the sparse linear solver (sets **auto_select_solver=yes**). The default for **auto_select_solver** is No. When **auto_select_solver** is set to Yes, Nexxim selects the sparse linear solver that gives the best performance for all simulations specified in the netlist. Setting **auto_select_solver** can improve performance. However, for large circuits with a short simulation time (e.g., with a short-time transient analysis), **auto_select_solve** r may incur a time penalty due to the initialization.
- **Create state-space system in frequency-domain analysis** controls the treatment of S-parameter elements (sets **time_domain_s_model=1**).
- **Order with values** controls the matrix ordering algorithm (sets **order_with_values**). By default, **order_with_values=0**, selecting the default sparse matrix solver. Setting **order_with_values=1** selects an earlier solver.
- **Minimum Resistance** sets a minimum resistance for floating nodes (sets **RESMIN**).
- **S element coupling threshold (in db)** uses specified threshold to break s element models into smaller ones if applicable.

HB Options Tab

- The **HB Options** tab sets options that apply only to Nexxim harmonic balance analyses. See Harmonic Balance Options for details on the HB analysis options.

DC Options Tab

- The **DC Options** tab sets options that apply only to Nexxim DC operating point analyses. See DC Analysis Options for details on the DC analysis options.

  Transient Options Tab

- The **Transient Options** tab sets options that apply only to Nexxim transient analyses. See Transient Analysis Options for details on the transient analysis options.

Oscillator Options Tab

- The **Oscillator Options** tab sets options that apply only to Nexxim oscillator analyses. See Oscillator Analysis Options for details.

Eye Options Tab

- The **Eye Options** tab sets options that apply only to Nexxim VerifEye and Quick Eye analyses. See VerifEye and Quick Eye Analysis Options for details on the Eye analysis options.

AMI Options Tab

- The **AMI Options** tab sets options that apply only to Nexxim AMI analyses. See AMI Analysis Options for details on the AMI analysis options.

TSMC-TMI Options Tab

- The **TSMC-TMI Options** tab contains settings that apply only to Nexxim simulation using TMI models. Your installation must include DLLs from TSMC to use these options. Click the **Enable** check box to enable the Circuit solver to access the TSMC-TMI DLLs and activate the fields in the **TSMC-TMI Options** tab. See TSMC-TMI Options for details on the TSMC-TMI analysis options.

When you have completed setting options, click **OK** to exit the window. The named set of options is added to the **Analysis** icon in the **Project Manager** window.

Editing Solution Options

To edit a set of Solution options that has previously been defined:

- Click the icon for the option set under the Analysis group in the **Project Manager** window to open the **Solution Options** window. Edit the options, then click **OK** to exit.
- Alternatively, click **Select** in the **Solution Options** group box in a **Solution Setup** window to open the **Select Solution Options** window . Click the option set you wish to edit, and click **Edit** to open the **Solution Options** window. Edit the options, then click **OK** to exit the **Solution Options** window.

> **Note:** To apply a set of options in a Nexxim analysis, you must add the options in the **Solution Setup** window for a particular analysis. Click **Select** in the **Solution Options** group box of the **Solution Setup** window to open the **Select Solution Options** window. Click the name of the option set you want to use, then click **OK** on the **Select Solution Options** window. The name of the selected option appears in the **Name** field in the **Solution Options** group box. Complete the **Solution Setup** window as described in the topics on the individual analyses.

Setting Options in a Netlist

In a netlist, both device and analysis-specific options are set with the **.OPTIONS** statement. The syntax is:

```
.OPTIONS [qualifier.]option[=val][ [analysis.]option[=val] ...]
```

The **.OPTIONS** statement can contain multiple option settings, separated by spaces.

The *qualifier* makes its option specific to one analysis type:

dc for DC analysis

tran for transient analysis

hb for harmonic balance analysis

hb.env for harmonic balance envelope analysis

lna for linear network analysis

osc for oscillator analysis

osc.phase_noise for oscillator phase noise analysis

tv_noise for time-varying noise analysis

eye for VerifEye and Quick Eye analyses

ami for Algorithmic Modeling Interface (AMI) analysis

s_element for S-element calculations

w_element for W-element calculations

Nexxim also allows statement parameters that set options for just one particular analysis. For example, here is a partial syntax for the TRAN statement:

```
.TRANstepstoptime
 +ERRPRESET=relaxed|moderate|strict
 +RELREF=pointlocal|allocal|sigglobal|allglobal
```

ERRPRESET and **RELREF** are statement parameters that set the corresponding options.

An analysis-specific option setting takes precedence over a non-specific setting of the same option, for the given analysis. For example, during a harmonic balance analysis, the option setting:

```
hb.abstol=4e-12
```

takes precedence over a setting of **abstol=2e-12** set elsewhere in the netlist. When you set an option with the **AddSolution Options** selection on the **Circuit** drop-down, or from in the individual **Solution Setup** windows, Ansys Electronics Desktop automatically puts the correct prefix on the option in the netlist.

See the help topics on the individual analyses for examples and discussions of the analysis-specific options.

Some options are flags that do not have a value. They are enabled when they are present in the **.OPTIONS** statement and unavailable otherwise. An example is WL, which reverses the order of the width and length model parameters in a MOSFET instance or element statement. Other options take a value, in which case the statement adds an equals sign after the name of the option.

When the same option has been set both inside and outside of the netlist, the following precedence is applied:

Command line options > netlist options > default (configuration file) options.

Within a netlist, options on the **.OPTIONS** statement can be overridden by model and instance parameters. The order of precedence is: instance parameter > model parameter > option.

If the netlist contains multiple **.OPTIONS** statements that set the same option, the last setting overrides any earlier settings.

Circuit Temperature Options

Properties of circuit elements may be a function of the element's temperature. The TC1 and TC2 parameters of a resistor, for example, relate the resistance to the resistor's temperature.

A Nexxim circuit includes two representations of temperature: the *actual* temperature and the *nominal* temperature.

The nominal temperature is the temperature at which the elements' baseline parameters are valid. The actual temperature is the temperature at which the circuit is being simulated. Temperature dependency equations deal with the difference between the element's actual and nominal temperature. Therefore, each element's actual and nominal temperature must be known.

By default, the global nominal temperature is 25 degrees Celsius. The global nominal temperature can be set in the netlist with the statement:

```
.OPTIONS TNOM=val
```

The value on the last **.OPTIONS TNOM** statement in the netlist becomes the global nominal temperature for all devices in the netlist that do not have another nominal temperature specification. The use of **TNOM** as a global option is available only via the netlist. There is currently no way to set **TNOM** globally from in the **Schematic Editor**.

Many device models include a **TNOM** or **TREF** model parameter. A **TNOM=***val* or **TREF=***val* entry in a **.MODEL** statement sets the nominal temperature for all device instances that reference that model. The **TNOM** or **TREF** parameter can be set in the netlist, or on the Properties window of individual models in the **Schematic Editor**. A **TNOM** or **TREF** parameter overrides the value of the **TNOM** option, when both are present.

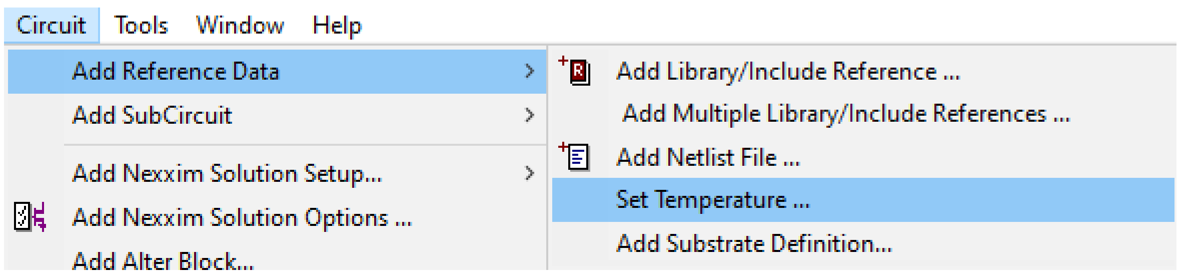
Note that it is also legal to give the value of the **TNOM** or **TREF** model parameter in the netlist statement defining a device instance. This overrides the value of the **TNOM** or **TREF** parameter in the model.

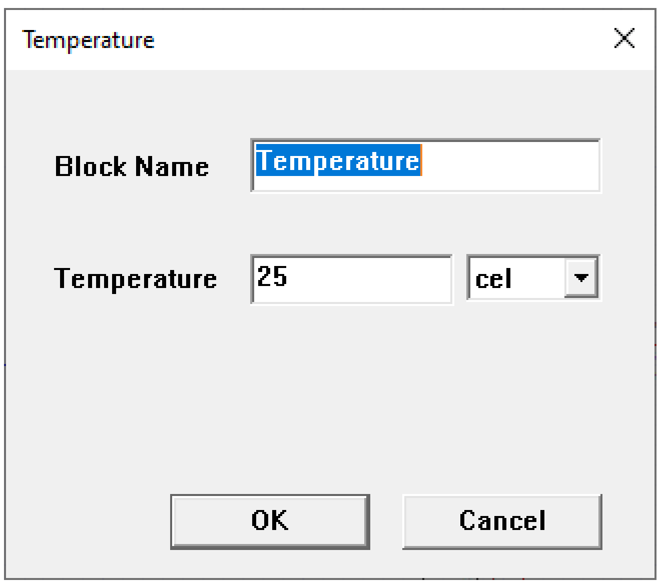
By default, the global circuit temperature is equal to the global nominal temperature. The global circuit temperature can be set in a netlist with the statement:

```
.TEMPtemperature [temperature ...]
```

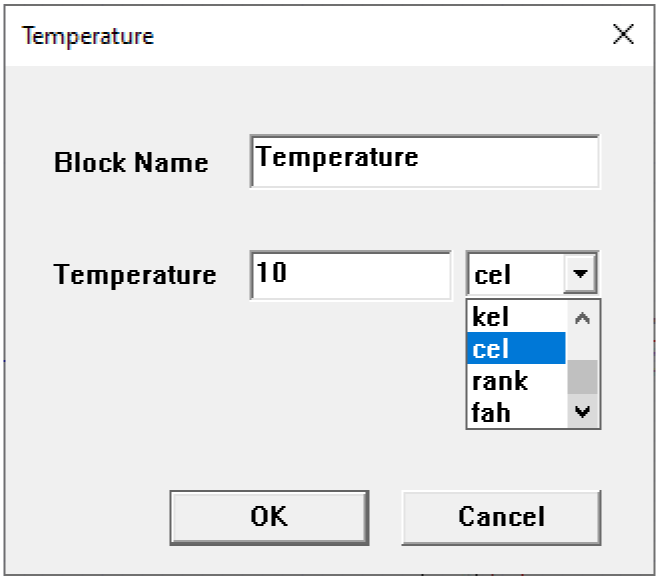
When more than one temperature is specified, Nexxim analyses perform a sweep of the multiple temperature points (unless the individual analysis statement already includes a temperature sweep). See Variable Sweep for further information. If the netlist contains multiple **.TEMP** statements, the last temperature overrides any earlier settings.

1. Set the global temperature by completing the following steps.
2. From the **Circuit** menu, select **Add Reference Data** >**Set Temperature ...** to open the **Temperature** window.





3. Choose the appropriate scale (i.e., **mkel**, **ckel**, **dkel**, **kel**, **cel**, **rank** or **fah**) and type the appropriate temperature into the field.



4. Click **OK**.

> **Note:** A device instance statement can include the entry **DTEMP=***val*. The value *val* is added to the global circuit temperature to determine the actual temperature of the device instance.

Setting Options or Temperature in External Files

.OPTIONS and **.TEMP** statements can be supplied from external **.INCLUDE** and **.LIB** files (See External Files in the Netlist Format help topic).

A file imported with **.INCLUDE** is copied into the netlist at the point of the **.INCLUDE** statement. Thus, a setting made with **.OPTIONS** or **.TEMP** statements in a **.INCLUDE** file overrides settings made in earlier statements, and is overridden by later statements.

A portion of a file imported with **.LIB** is processed after all other statements in the netlist have been processed, and the order of processing of statements in .LIB files versus the main file cannot be determined in advance. Thus, if a **.LIB** block contains a **.OPTIONS** statement for a particular option, it should be the only setting for that option in the entire netlist. If a **.LIB** block contains a **.TEMP** statement, it should be the only **.TEMP** statement in the entire netlist.

If a **.OPTIONS** statement is deprecated, it generates a warning and be ignored.

Using GPUs with Nexxim Analyses

Nexxim can take advantage of a Graphics Processing Unit (GPU) to accelerate simulation, eye rendering, or both. Nvidia® GPUs with compute capability 2.0 are supported.

To enable Nexxim to use a GPU installed in your computer, set the global analysis option **use_gpu=1**.

- From the Ansys Electronics Desktop, select **Tools>Options>HPC and Analysis Options**.
- From the **HPC and Analysis Options** window, select the **Options** tab.
- Click **Enable GPU for Eye Analysis**. Select **True** on the menu. Click **OK** to close the options window.

When enabled, Nexxim uses the GPU to accelerate the rendering of eye diagrams for Transient, QuickEye, and AMI analyses. The GPU also speeds up QuickEye and AMI simulations.

Setting **use_gpu=1** also enables Ansys Electronics Desktop to speed up the rendering of eye diagram reports in general.

> **Warning:**
>
> GPU acceleration cannot be accessed by remote users. Setting **use_gpu=1** takes effect only with direct logins.

Nexxim Circuit Netlist Design — Options Reference

This topic contains tables for all the customer-accessible Nexxim Circuit netlist design-level options.

Global Analysis Options Reference

Global analysis options apply to multiple Nexxim analyses.

Global Analysis Options

Option	Default Value	Description
abstol	1e-9	Absolute error tolerance for currents (Amps)
alpha	0.0	Minimum conductance of the resistor placed in parallel with each branch (mho)
auto_enforce_passivity	1	1=Enable auto_enforce_passivity for all W-elements and S-elements.

Option	Default Value	Description
		Passivity enforcement should be enabled or deactivated globally for both S-elements and W-elements. Use of the local auto_enforce_passivity options to selectively deactivate passivity enforcement for 0=Turns off passivity enforcement for all w-elements and s-elements in the circuit. When the Global option is deactivated (set to 0), the **auto_enforce_passivity** option set in a W-element or S-element is ignored. **auto_enforce_passivity** is valid only in a transient simulation and is ignored in a frequency-domain analysis (HB or LNA).
auto_select_ solver	No	When Yes, Nexxim selects the sparse linear solver that gives the best performance for all simulations in the netlist
autoconverge	1	1 = Enable automatic cyclic progression through convergence methods upon convergence failure. 0 = Turns off automatic cycling of methods.
behavioral_ minimum_ divisor	0	Minimum divisor to use for behavioral components in Nexsys simulations. Set to nonzero value to prevent division by zero.
beta	0.0	Minimum conductance of the resistor placed between each node and ground (mho)
dcic	1	Flag to ignore initial conditions
delmax, tmax, maxstep	min (*stoptime*/100, 15**tstep*)	Maximum step size
foster_ bundling	1 if the number of Foster elements in the design is 25 or more and AMI, EYE or Tran	1=Detect port-to-port transfer functions (of type VCCS or VCVS) of an N-port macromodel in pole-residue format (also known as the Foster format) that share the same poles, and eliminate redundant

Option	Default Value	Description
	analysis is requested — otherwise 0.	calculations. This is the default setting. Macromodels are usually constructed with port-to-port transfer functions that share the same set of poles. In these cases, the **foster_bundling** option significantly reduces the run time and memory consumption of the simulation, especially for macromodels with large numbers of ports and/or large numbers of poles. This option does not affect the accuracy of the simulation. 0=Treat each port-to-port transfer function independently. When this option is set to 1 (bundling enabled), error messages on the simulator due to some errors in the macromodel (e.g., a complex pole-pair with only real values for the residues) does not pinpoint the Foster element that is causing the problem; instead, they is more generic. If this kind of limitation is a concern, then set this option to 0. This option is ignored for LNA, OSC, HB and TV analyses.
gmax	1e2	Maximum conductance of the resistor placed in parallel with each branch (mho)
gmindc	0.0	Minimum conductance of the resistor placed in parallel with each branch (mho)
gshunt	0.0	Minimum conductance of the resistor placed between each node and ground (mho)
iabstol	1e-9	Absolute error tolerance for currents (Amps)
limiting	1	1=allow limiting on Newton steps
limiting_step	0.3 for nonlinear circuits. Very large (unavailable) for linear	Maximum absolute voltage update allowed per N-R iteration (Volts)

Option	Default Value	Description
	circuits.	
ltratio	7.0	Ratio of transient analysis tolerance to Newton-Raphson tolerance, for timestep calculation
macmod	0	1=Use subckt definition for *modelname* in MOSFET/BJT/Diode instance when no .MODEL reference exists. 2=Use .MODEL definition for *subckt_ name* in X-element instance when no subckt exists. 3=Both 1 and 2. When **TMIFLAG** is set, **macmod=2** is automatically set.
max_ continuation_ iterations	500	Maximum number of continuation steps allowed per strategy
max_newton_ iterations	10 (TRAN) 200 (DC) 20 (HB)	Maximum number of Newton-Raphson (N-R) iterations per continuation step
maxiters	10 (TRAN)	Maximum number of Newton-Raphson iterations for each timestep
max_ messages	1	Maximum number of warning messages of the same type to be displayed in the **Message Manager** window. All error messages are always displayed. If **max_ messages** is not present, no message filtering is done. The logfile includes all messages, warnings, and errors regardless of this option.
method	trap	Integration method (parentheses show synonyms for methods): **trap** (**trapezoidal**) **gear** (**bdf2**) **ndf2**
minimum_ divisor	0	Minimum divisor. Use 1e-28 for HSPICE compatibility.
num_threads	1	Maximum number of independent processor threads to use during analyses. To prevent a scenario where parallel simulation is slower than serial

Option	Default Value	Description
		simulation, Nexxim may use fewer threads than requested by the user.
order_with_ values	0	0=default sparse matrix solver 1=earlier solver, uses values in Jacobian in computing the fill reduction ordering
parhier	None	PARHIER=L reverses the precedence of parameter assignment in subcircuits
password	None	To access encrypted password-protected external files.
reltol	1e-3	Relative error tolerance for currents and voltages
result_as_ initial_guess	1	1=Use of the result from each step in a sweep as the initial guess for the next sweep step.
rmax	5	Factor for calculating maximum first stage timestep
seed	None	Used by random-value generator to initialize the sequence of pseudorandom values. Using the same seed generates the same sequence of values.
skip_dc	no	yes = Field solvers need not solve for DC. Use this option only for circuits with linear devices and no non-zero-valued sources except AC sources. no = Solve for DC.
time_domain_ s_model	0	1=Create state-space system in a frequency-domain analysis like LNA where one is not usually generated. The state-space system may not be passive unless passivity enforcement is enabled. (**s_element.enforce_passivity** =non-zero). **time_domain_s_model** is processed only in a frequency-domain analysis (HB or LNA) that is run WITHOUT passivity enforcement activated. If passivity is enforced in a frequency-domain analysis, then time-domain modeling is automatically enabled, eliminating the need for the **time_domain_s_model**

Option	Default Value	Description
		option. If this option is specified in a netlist along with passivity enforcement and LNA, then this option is silently ignored.
time_domain_w_model	0	1=create state-space system, even in a frequency-domain analysis like LNA where one is not usually generated. The state-space system may not be passive unless passivity enforcement is enabled. (**w_element.enforce_passivity** =non-zero). **time_domain_w_model** is processed only in a frequency-domain analysis (HB or LNA) that is run WITHOUT passivity enforcement activated. If passivity is enforced in a frequency-domain analysis, then time-domain modeling is automatically enabled, eliminating the need for the **time_domain_w_model** option. If this option is specified in a netlist along with passivity enforcement and LNA, then this option is silently ignored.
trtol	7.0	Ratio of transient analysis tolerance to Newton-Raphson tolerance, for timestep calculation
unit_atto	None	When set, A as a numeric suffix means e-18 (atto) rather than Ampere.
use_gpu	0	1=Use installed GPU to accelerate simulation and eye rendering for AMI, QE, and TRAN
vabstol	50e-6	Absolute error tolerance for voltages (Volts)
vntol	50e-6	Absolute error tolerance for voltages (Volts)

Global Device Options Reference

Global device options affect all devices of a given type unless overridden by model and instance parameters.

Global Device Options

Option	Description	Used By	Default
DEVICE_CLEANUP	1 = delete zero-valued resistors, capacitors, and inductors 0 = do not delete (default)	Resistor, Capacitor, Inductor	0
DAMPINDUCTORS	Damping of inductors	Power electronics simulations	1e12
DIODECJO	Specify minimum value of parameter CJO for simulation of PSPICE diode models.	PSPICE diodes	5e-17
GENK	Enable/disable computing of missing mutual inductances (for HSPICE compatibility). • **1** =compute • **0** =do not compute. When GENK=0, all missing mutual inductance coupling coefficients are set to 1e-10 (no coupling).	Mutual inductor	1
GFARAD	Conductivity leakage of capacitors.	Power electronics simulations	1e-12
GMAX	Maximum conductance used in NODESET calculations	Resistor	1.0
MOSFETCJO	Supply a minimum capacitance for MOSFET source/drain junctions.	Power electronics simulations.	5e-12
PARHIER	PARHIER=L reverses the precedence of parameter assignment in subcircuits	Subcircuits	Not set (default precedence)
RESMIN	Minimum resistance (positive or negative)	Resistor	If GMAX is defined, calculated from GMAX Else: 1.0e-5
SCALE	Scaling multiplier for instance parameters. The parameters affected vary across model implementations.	Device Instances	1.0
SCALM	Multiplier for model parameter values. The parameters affected vary across model implementations.	MOSFETs	1.0

Option	Description	Used By	Default
SKIP_PWL_DIODE	Determines whether to keep the original model. Possible values are: • **0** - Replace idealized semiconductor diode models (n < 0.5) with an equivalent piece-wise linear model to improve convergence in Power Electronics simulations. Active only for PSPICE model import. • **1** - Override replacement (keep original models).	Diodes	0
THEV_INDUC	Turns a default 1 mOhm series resistance on or off for inductors.	Power electronics simulations	1(on)
WL	Reverses the default order of width and length parameters on MOSFET instances	MOSFETs	Not set (default parameter order)

DC Analysis Options Reference

These options apply to Nexxim DC operating point analyses. See DC Analysis Options for an explanation of these options in the **Schematic Editor** and netlist editor.

Nexxim DC Analysis Options

Option	Default Value	Description
abstol	1e-9	Absolute current tolerance (Amps). The DC simulator starts with this options at an initial value.
alpha	1e-13	Minimum conductance of the resistor placed in parallel with each branch (mho, equivalent to GMINDC)
autoconverge	1	1 = Enable automatic cyclic progression through convergence methods starting with **conv** setting, upon convergence failure. 0 = Turns off automatic cycling of methods.
beta	1e-12	Minimum conductance of the resistor placed between each node and ground (mho, equivalent to GSHUNT)
conv	1	1 = Beta and device continuation 2 = Device continuation

Option	Default Value	Description
		3 = Pseudotransient 4 = Modified beta/device continuation 5 = Alpha, beta, and device continuation
create_ic_file	0	1=Create initial conditions (.ic) file from DC operating point.
dcic	1	Flag to ignore initial conditions
gmax	1e2	Maximum conductance of the resistor placed in parallel with each branch (mho). Used for HSPICE compatibility in NODESETs and initial conditions
limiting	1	1=allow limiting on Newton steps
limiting_ method	Standard Limiting	Standard Limiting = Damped Newton-Raphson. Modified Limiting = Newton-Raphson with enhanced limiting algorithm. Faster convergence rate for moderately nonlinear circuits.
limiting_step	0.3 for nonlinear circuits. Very large (unavailable) for linear circuits	Maximum absolute voltage update allowed per N-R iteration (Volts). The DC simulator starts with this options at an initial value.
max_ continuation_ iterations	500	Maximum number of continuation steps allowed per strategy
max_newton_ iterations	200	Maximum number of Newton-Raphson (N-R) iterations per continuation step. The DC simulator starts with this options at an initial value.
max_newton_ iterations_limit	None	Maximum post-lambda pseudo-transient continuation
max_pt_ continuation_ iterations (Used only when **conv=3** has been specified.)	5000	Maximum iterations of pseudotransient continuation
min_lambda_ step	1e-4	Minimum lambda step allowed in continuation

Option	Default Value	Description
output_ic_file_name	None	Name of the initial condition file
reltol	1e-3	Relative current and voltage tolerance. The DC simulator starts with this options at an initial value.
result_as_initial_guess	1	Turned off, when set to 0, otherwise the use of the result from each step in a sweep as the initial guess for the next sweep step.
vntol	50e-6	Absolute voltage tolerance (Volts). The DC simulator starts with this options at an initial value.

Transient Analysis Options Reference

These options apply to Nexxim transient analysis.

Nexxim Transient Analysis Options

Option	Default Value	Description
abstol	1e-9	Absolute error tolerance for currents (Amps)
accurate_points.num	0	Force transient analysis to place time points at N equidistant points (simulate N events)
accurate_points.start	0.0	First accurate point
accurate_points.stop	0.0	Last accurate point
alpha	0.0	Minimum conductance of the resistor placed in parallel with each branch (mho)
beta	0.0	Minimum conductance of the resistor placed between each node and the ground (mho)
CMIN	0.0	Minimum capacitance placed between each node and the ground (F). The CMIN option should be used when transient analyses fail to converge despite the step size becoming very small. The minimum capacitance specified should be in the range of 1e-18 to 1e-14 F for the most accurate results. If CMIN is set to a value larger than 1e-12 F, the CMIN value is automatically reset to 1e-12 F.
convergence_enhancement	0	To enable, set to 1. Increases the chance of convergence for some designs.

Option	Default Value	Description
		The following solver settings are modified if enabled: **step** = **step**/100 **maxiters** = 100 **update_jacobian_period** = 1 **method** = ndf2 **maxstep** = 25*step **alpha** = **beta** = 1e-5 **relref** = **alllocal** if it was **pointlocal** **relref** = **allglobal** if it was **alllocal**
delmax, tmax, maxstep	min (*stoptime*/100, 15**tstep*)	Maximum step size
errpreset	None	Overrides parameter defaults for speed/accuracy trade-off. See Table 4 in this topic.
freq_ restriction	1	Ensure at least 8 time points in one period.
limiting	1	1=allow limiting on Newton steps
limiting_step	0.3 for nonlinear circuits. Very large (unavailable) for linear circuits.	Maximum absolute voltage update allowed per N-R iteration (Volts)
max_damp	0	Nonlinearity-based time stepping decreasing factor controller. Must be no greater than 1.
max_ frequency	50e9	Maximum frequency resolved by transient; used for S-parameters
maxiters	10	Maximum number of Newton-Raphson iterations for each timestep
max_mem_ size	200e6	Maximum size in bytes for internal storage of unknowns. With 64-bit solvers, max_mem_size may be set to 500e6 or higher to handle circuits with large numbers of nodes and branches, or circuits with W-elements that have long delays.
max_stretch	0	Nonlinearity-based time stepping increasing factor controller. Must be no greater than 2.

Option	Default Value	Description
method	trap	Integration method (parentheses show synonyms for methods): **trap** (**trapezoidal**) **gear** (**bdf2**) **ndf2**
NOISEFMAX	0	The highest frequency to analyze for pseudorandom noise. Must be non-zero to enable transient noise analysis. Maximum transient timestep is set to 1/NOISEFMAX.
NOISEFMIN		Lowest frequency to analyze for noise. Under NOISEFMIN, noise power density is constant. 1/NOISEFMIN must be less than the transient analysis stoptime. **Not used currently.**
NOISESCALE	1	Scale factor
NOISESEED	0	Random number seed. Using the same seed generates the same set of random values
NOISESTART	0	Noise start time
NOISETMIN	0	Noise minimum transient step. **Not used currently.**
num_ frequencies	500	Number of frequencies resolved by transient plus DC; used for S-parameters
outputstart	0.0	Time to start saving results to a file
relref	alllocal	The **RELREF** argument controls how the norms are computed for the delta check, function check, and LTE checks. *Point-local* means the norm is local to the signal and to the timestep. *Local* means the norm is local to the signal but considers the maximums over all time-steps. *Global* means the norm is global for the entire circuit over all time-steps. The settings for **relref** are: **pointlocal** = Point-local for delta, function, and LTE **alllocal** = Local for delta, function, and LTE **sigglobal** = Local for function, global for delta & LTE **allglobal** = Global for delta, function, and LTE
reltol	1e-3	Relative error tolerance for currents and voltages
read_state	None	Read initial state on the specified file
result_as_ initial_guess	1	1=Use of the result from each step in a sweep as the initial guess for the next sweep step.
rmax	5	Factor for calculating maximum first stage timestep
sig_storage_	0	1=Skip time domain probe data storage

Option	Default Value	Description
off		
skipResult	0	By default, transient results are stored into the SDF file. For the Eye Scope, setting skipResult to 1 allows the user to see the eye without storing all the data. WARNING: Nexxim does not check that there is an Eye Scope present, risking loss of data.
step	None	Suggested transient step size
stop	None	Stop time
stop_pwl_ times	1	0=Transient does not stop at the data points in any PWL source 1=Transient stops and starts backward Euler at every point specified in all PWL sources 2=Transient stops at every data point in all PWL sources, but maintains the current integration order 3=Transient stops only at data points that deviate on the extrapolation using the previous two points by more than the reltol and abstol specified for transient
tpbs	0	1=Enable transient parallel bus speedup
tpbs.maxiters	Depends on num_threads	Number of parallel iterations
tpbs.overlap	Depends on num_threads	Length of parallel overlap in seconds
tpbs.reltol	0.005	Relative error tolerance for voltages and currents within parallel iterations
tran.calc_stat_ eye	0	Enables statistical eye plots for Transient setups when there is an eyeprobe or eyescope present in the design
trtol	7.0	Ratio of transient analysis tolerance to Newton-Raphson tolerance, for timestep calculation
update_ jacobian_ period	3	Number of Newton steps per Jacobian update
write_final_ state		Write the final values of node voltages to specified file
write_initial_ state		Write the initial (DC) values of node voltages to specified file
vntol	50e-6	Absolute error tolerance for voltages (Volts)

Table 5: Nexxim Transient Analysis Errpreset Option

Option	errpreset			
	None	**relaxed**	**moderate**	**strict**
reltol	x1	x10	x1	/10
trtol	7	7	7	10
relref	alllocal	sigglobal	sigglobal	alllocal
method	trap	trap	trap	ndf2
maxstep	stoptime/100	stoptime/12	stoptime/50	stoptime/100
update_jacobian_ period	3	3	3	1

Harmonic Balance Options Reference

These options apply to Nexxim harmonic balance analyses.

Nexxim Harmonic Balance Analysis Options

Option	Default Value	Description
abstol	1e-12	Absolute error tolerance for currents for the nonlinear equation solver (Amp)
alpha	1e-12	Conductance in parallel with each branch (mho, equivalent to G_{MIN})
auto_refine_solution	no	For power sweeps, yes = increase number of harmonics as power increases
beta	0.0	Conductance between each node and ground (mho, equivalent to G_{SHUNT})
cmin	0.0	Capacitance between each node and ground (Farad)
cluster_tol	1e-2	Multitone frequencies within this tolerance are clustered to save memory (Hz)
continuation	**source_ stepping_ last_tone**	Continuation strategy **source_stepping_last_tone** (solve low-voltage case for last tone in **TONES** group box, then increase voltages until true circuit is solved.) **pseudo_transient** (use additive transient term)
continuation.pseudo_ transient.max_h Used only when	5e3	Initial value of internal pseudotransient variable

Option	Default Value	Description
continuation=pseudo_ transient is in effect		
continuation.pseudo_ transient.pt_reltol_scale Used only when **continuation=pseudo_ transient** is in effect	0.8	Relative tolerance of pseudotransient reltol to standard HB
continuation.pseudo_ transient.step_reltol_ scale Used only when **continuation=pseudo_ transient** is in effect	0.8	Relative tolerance of pseudotransient continuation to standard HB
initial_guess	**dc** for single-tone, **fewer_ tones** for multi-tone	Source for initial guess at voltages and currents **dc** (Run DC) **fewer_tones** (iterative guesses) **fewer_tones_transient** (iterative guesses starting with result of transient analysis) **file** (read from HB file) **transient** (Run transient analysis) **zero** (Zero)
initial_guess.fewer_ tones_transient.num_ cycles (Used only when **fewer_ tones_transient** is selected for the **initial_ guess** option.)	20	Number of cycles of 1st tone period to run initial transient analysis
initial_guess.fewer_ tones_transient.tmax (Used only when **fewer_ tones_transient** is selected for the **initial_ guess** option.)	None	Suggested maximum step size for transient analysis
initial_guess.fewer_ tones_transient.tstop (Used only when **fewer_ tones_transient** is	None	Stop time for transient analysis

Option	Default Value	Description
selected for the **initial_ guess** option.)		
initial_ guess.transient.num_ cycles (Used only when **transient** is selected for the **initial_ guess** option.)	20	Number of cycles of 1st tone period to run initial transient analysis
initial_ guess.transient.tmax (Used only when **transient** is selected for the **initial_ guess** option.)	None	Suggested maximum step size for transient analysis
initial_ guess.transient.tstop (Used only when **transient** is selected for the **initial_ guess** option.)	None	Stop time for transient analysis
initial_guess_file (Used only when **file** is selected for the **initial_ guess** option.)	None	Pathname and filename of SDF-format file with HB results to use as initial guess
limiting_step	0.3	Maximum absolute voltage update allowed per iteration for the nonlinear equation solver
linearsolver_reltol	1e-2	Reltol for linear solver
linearsolver_restarts	1	Maximum linear solver restarts
limiting_step	0.3 for nonlinear circuits. Very large (unavailable) for linear circuits.	Maximum absolute voltage update allowed per N-R iteration (Volts)
maxk	None	Maximum number of harmonics at each frequency
max_linearsolver_ iterations	100	Maximum number of iterations for linear equation solver
max_newton_iterations	20	Maximum number of iterations for the nonlinear equation solver

Option	Default Value	Description
memory_over_speed	0	0 = Run to maximize speed, memory use may be excessive 1 = Run to conserve memory
method	hb	**hb** = standard HB calculation. **shooting** = HB method for strongly nonlinear circuits
power_of_two_ expansion	1 for single-tone, 0 for multi-tone	Expand MAXK to
preconditioner	0	Select preconditioner (1=time_ frequency, 0=bdf)
read_state	None	Read initial state on the specified file
reltol	1e-3	Relative error tolerance for currents and voltages (dimensionless)
result_as_initial_guess	1	Turned off, when set to 0, otherwise the use of the result from each step in a sweep as the initial guess for the next sweep step.
strongly_nonlinear	0	0=linear/mild, 1=strongly nonlinear/shooting
td_preconditioner	1	1=use td preconditioner. **Mainly for internal use.**
tones	None	List of frequencies to analyze
trim_tol	0	Harmonics voltage tolerance. Harmonics less than trim_tol after the initial guess are pruned. 0=no pruning. **Mainly for internal use.**
trtol	7	Ratio of transient analysis tolerance to Newton-Raphson tolerance, for timestep calculation
tstab	None	Initial stabilization time. Can be used to change the starting phase of the HB algorithm
vntol	1e-7	Absolute error tolerance for voltages for the nonlinear equation solver (Volt)
write_final_state		Write the final values of node voltages to specified file.
write_initial_state		Write the initial (DC) values of node voltages to specified file.
zero_freq_at_end	1	Return to DC at end of simulation. Only for single-tone HB

Linear Network Analysis Options Reference

These options apply to Nexxim linear network analyses.

Nexxim Linear Network Analysis Options

Option	Default Value	Description
flag	LNA	LNA = simple AC analysis NOISE_DC = noise analysis LNAN=AC and noise LNAG=AC and group delay LNANG=AC, noise, and group delay
group_ delay_ pert	None	Perturbation (percentage of the test frequency) to be applied in group delay calculation
input_ referred_ noise	None	Node voltage, branch, or current to be used as the source for an input-referred noise calculation
result_ as_ initial_ guess	1	Use LNA solution as initial guess in sweep
noise_ output	None	Output nodes where noise should be computed in noise analysis. No noise factor or conversion gain is computed for these outputs
skip_dc	no	yes = Field solvers need not solve for DC. Use this option only for circuits with linear devices and no non-zero-valued sources except AC sources. no = Solve for DC.
small_ signal_ beta	1e-12	Minimum conductance from node to ground in small-signal analysis
ZERO_ PORT_ VALUES	TRUE	Enable use of port impedance resistors **Note:** **May be obsolete.**

Oscillator and Phase Noise Analysis Options Reference

These options apply to Nexxim oscillator and oscillator phase noise analyses.

Nexxim Oscillator and Phase Noise Analysis Options

Option	Default Value	Description
freq_abstol	10	Relative tolerance for oscillation frequency
harmonic_number	None	Oscillator harmonic to compute phase noise for (Phase noise analysis only)
initial_guess	**transient**	Select method for initial guess at probe frequency **dc** = use **FREQ** parameter on the oscillator probe. **freq_sweep** = run frequency sweep **transient** = run transient analysis
initial_guess.freq_sweep.method	linear	Linear or nonlinear solution type for frequency sweep initial guess
initial_guess.freq_sweep.num_steps	50	Number of steps for frequency sweep
initial_guess.freq_sweep.start	1e9 Hz	Start frequency for frequency sweep
initial_guess.freq_sweep.stop	5e9 Hz	End frequency for frequency sweep
initial_guess.transient.num_cycles	Calculated	Number of periods for transient initial guess (like stop time)
initial_guess.transient.output_start	0	Initial time during transient analysis that is ignored by initial guess
initial_guess.transient.tmax	Calculated	Maximum step size for transient analysis
initial_guess.transient.tstep	Calculated	Maximum step size for transient analysis
initial_guess.transient.tstop	Calculated	Stop time for transient analysis initial guess
iprobe_abstol	1e-11	abstol for probe current
maxk	None	Maximum number of harmonics at each frequency (Phase noise analysis only)
noise_output	None	Node or branch to compute the phase noise on (Phase noise analysis only)

Option	Default Value	Description
nonlinear_method	**newton**	Newton-Raphson iteration strategy **newton** = standard Newton-Raphson **damped_newton** = use ∆V damping **descent** = follow slope of derivative
nonlinear_method.descent.num_iterations	100	Maximum number of steps in descent initial guess. **Mainly for internal use.**
nonlinear_method.descent.param.f_tolerance	1e-4	Function reltol in descent **Mainly for internal use.**
nonlinear_method.descent.param.max_step	0.01	Maximum normalized step size in descent **Mainly for internal use.**
nonlinear_method.descent.param.step	1e-4	Normalized step size in descent **Mainly for internal use.**
nonlinear_method.descent.param.tolerance	0.01	reltol in descent **Mainly for internal use.**
nonlinear_method.descent.param.	1e-4	Function reltol in descent **Mainly for internal use.**
nonlinear_method.newton.barrier	2	Power of the barrier that prevents probe from zero voltage. **Mainly for internal use.**
phase_noise.method	**stochastic**	Select stochastic or time_varying method for phase noise computation
reltol	1e-4	Relative voltage tolerance for NR iterations
result_as_initial_guess	**false**	When set to **true**, result from previous sweep step is used as the initial guess for the current sweep
tones	None	Approximate oscillator frequency
vprobe_abstol	1e-5	abstol for voltage probe

Envelope Analysis Options Reference

These options apply to Nexxim envelope analyses.

Nexxim Envelope Analysis Options

Option	Default Value	Description
clockname	None	Name of the device providing the clock fundamental. Nexxim determines the clock period by searching the sources with this name. (Mutually exclusive with period and fund)
envmethod	trap	Integration method for transient analysis. trap=trapezoidal rule, bdf2=second-order backward difference formula (Gear)
fund	None	Clock or carrier frequency (mutually exclusive with period and clockname)
maxk	None	Maximum number of harmonics at each frequency (Phase noise analysis only)
modulationbw	None	Estimate of the bandwidth of the nonperiodic RF modulation input signal
period	None	Period of clock fundamental (mutually exclusive with fund and clockname)
step	None	Used in maximum timestep calculation for transient
stop	None	Stop time for transient. Required to distinguish envelope analysis from ordinary HB
tones	None	Estimated frequencies of the periodic inputs for HB
hb.auto_ refine_ solution	no	For power sweeps, yes=increase number of harmonics as power increases
hb.env.fixed_ step	0	If >0 envelope analysis uses this as a fixed step. **Mainly for internal use.**
hb.method	hb	**hb** =standard HB calculation. **shooting** =HB method for strongly nonlinear circuits
hb.trim_tol	0	Harmonics voltage tolerance. Harmonics less than trim_tol after the initial guess are pruned. 0=no pruning. **Mainly for internal use.**

TV Noise and PXF Analysis Options Reference

These options apply to Nexxim time-varying (TV) noise and periodic transfer function (PXF) analyses.

Nexxim TV Noise and PXF Analysis Options

Option	Default Value	Description
abstol	1e-12	Absolute error tolerance for currents (Amp)
freqaxis	absin	Output results versus the input frequency, the output frequency, or the absolute value of the input frequency. Only for PXF.
fund	None	Steady state analysis fundamental frequency (or estimate for autonomous circuits).
iprobe	None	Input RF port, current source, or voltage source for small-signal input. For a port, Nexxim computes the noise figure and conversion gain in Volts (in the sense of available power)
linearsolver_ restarts	1	Maximum linear solver restarts
max_ linearsolver_ iterations	20	Maximum number of iterations for linear equation solver
maxsideband	None	Maximum sideband for PXF
noise_output		Output nodes where noise should be computed in noise analysis. No noise factor or conversion gain is computed for these outputs
oprobe	None	Output port, current source, voltage source, resistor, or current probe for small-signal output.
probe	None	PXF computes every transfer function to the specified probe component
pxf	false	Enable PXF analysis
refsideband	None	Coefficients for calculation of the input frequency
relharmnum	None	Specify a single relative harmonic number
relharmvec	None	Specify a list of relative harmonic numbers
saveallsidebands	no	yes=save all sideband results
sidebands	None	List of relevant sidebands
ss_analysis	hb	Select hb or osc to compute the steady-state values
ss_maxk	None	Maximum number of harmonics at each frequency in the steady-state calculations
ss_tones	None	List of frequencies to use in the steady-state HB calculations
reltol	1e-3	Relative error tolerance for currents and voltages (dimensionless)

Option	Default Value	Description
tstab	None	Initial stabilization time. Can be used to change the starting phase of the HB algorithm
hb.auto_refine_ solution	no	For power sweeps, yes=increase number of harmonics as power increases
hb.method	hb	**hb** =standard HB calculation. **shooting** =HB method for strongly nonlinear circuits
hb.trim_tol	0	Harmonics voltage tolerance. Harmonics less than trim_tol after the initial guess are pruned. 0=no pruning. **Mainly for internal use.**

VerifEye and Quick Eye Analysis Options Reference

These options apply to both VerifEye and Quick Eye analyses. All statistical eye analysis options use the prefix **eye**.

Nexxim VerifEye and Quick Eye Analysis Options

Option	Default Value	Description
eye.ampl_bins	500	Number of histogram bins to use for generating the 3D and contour data
eye.auto_ extend_step_ response	0	By default, the step response is extended only to the end of simulation time. To extend the step response beyond the simulation time, set this option to 1.
eye.eye_start_ time	0	For statistical plots of eye diagrams, discard all data from time 0 to **eye_start_time** in seconds.
eye.maxstep	0.25xUI	Maximum second-stage timestep size for the transient step response calculation. See Note.
eye.normalize_ ffe_weights	1	1=Normalize user-supplied tap weights (Sum of absolute values of all weights equals one) 0=no normalization
eye.num_ initial_ui	100	Number of unit intervals (UI) over which the initial transient step response should run after any channel or source delay. For example, with num_initial_ui set to 100 and a UI of 1ns, the initial transient analysis should run for at least 100ns + source delay + channel delay. There are additional constraints on the values this parameter can take. Please see **eye.resp_tmax** for additional information.
eye.q_scale_	1	Q-scale extrapolation:

Option	Default Value	Description
extrap		0=never extrapolate 1=extrapolate only when transmit jitter is present 2=always extrapolate. This setting can minimize noise at the bottom of a VerifEye bathtub curve when no transmit jitter is present.
eye.qe_only_cmf	0	By default, QuickEye stores the eye diagram and the transient-like result. Setting qe_only_cmf=1 turns off the storage of the transient-like result.
eye.resp_tmax	1e-6	Maximum stop time for the transient simulation needed to calculate the step response. If the step response calculation does not settle to a stable state value before the time in **resp_tmax**, simulation stops at this time. To have transient simulation run until the step response settles, increase the value of **resp_tmax**. NOTE: QuickEye and VerifEye use the minimum of (**num_initial_ui** x UI) and **resp_tmax** to set the transient stop time for the step response calculation.
eye.rmax	5	Factor for calculating maximum first-stage timestep for the transient step response calculation. See Note.
eye.sym_step_reponse	0	By default, QuickEye and VerifEye use different rise and fall times. To use the rise time as the fall time, set this option to 1.
eye.ui_bins	500	Number of histogram bins to use for generating the 3D and contour data
eye.verifeye_correlated_crosstalk	1	zero=enables VE uncorrelated crosstalk calculation

Warning:

QuickEye and VerifEye use transient analysis to calculate the step response. The transient analysis uses the two-stage timestep control (See"Timestep Control " on page 25-266). Additional accuracy can be achieved by reducing the maximum step size in both stages. For best performance, start by reducing the first-stage time maximum timestep **eye.rmax** from 5 gradually to 1. Then, if additional accuracy is appropriate, reduce the second-stage maximum timestep **eye.maxstep**.

AMI Analysis Options Reference

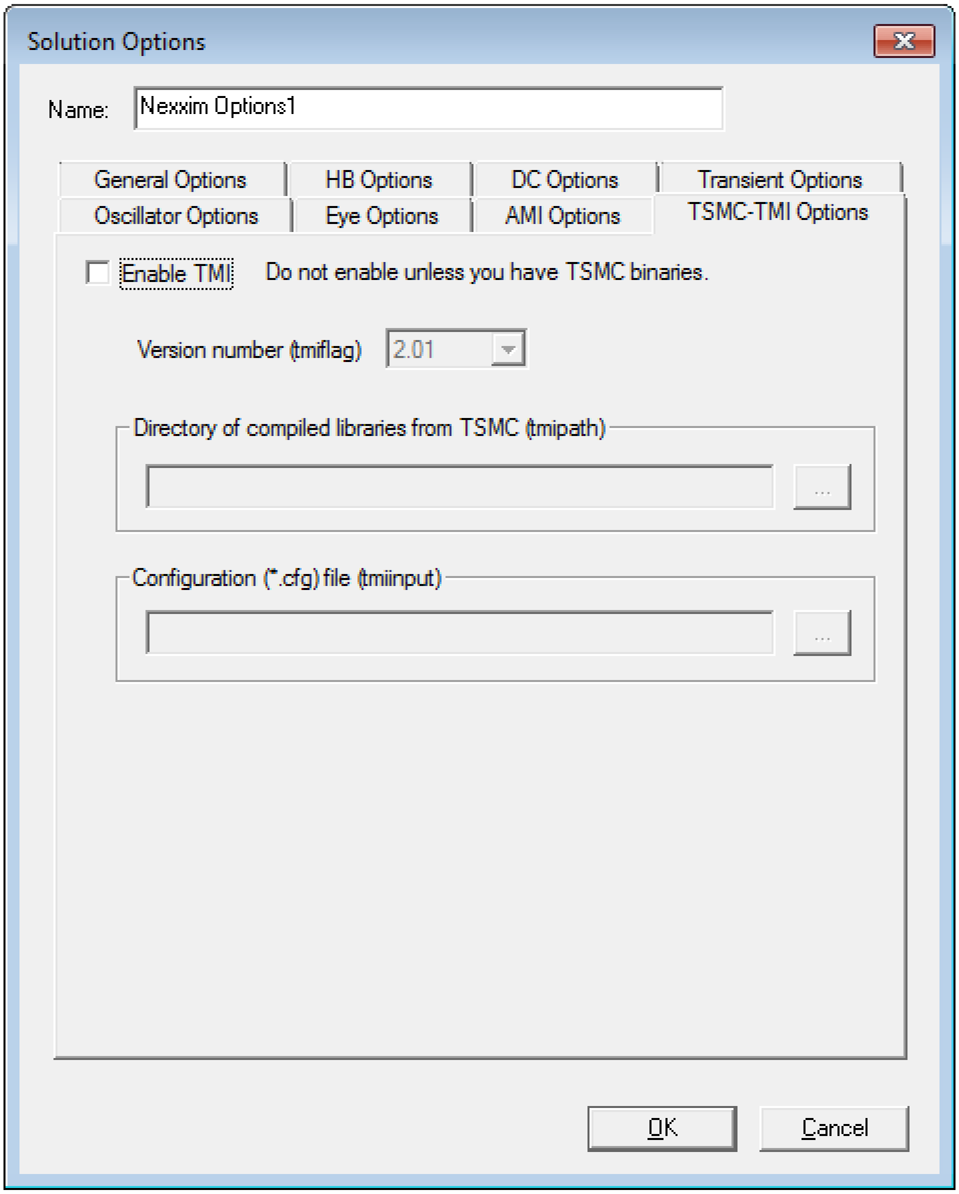
These options apply to AMI analyses. All these options use the prefix **ami**.

AMI Analysis Options

Option	Default Value	Description
ami.ampl_ bins	500	Number of histogram bins to use for generating the 3D and contour data
ami.delay	1	Delay before computing impulse response
ami.eye_ start_time	0	Start time for recording eye data
ami.resp_ tmax	1e-6	Maximum time for the transient impulse response calculation
ami.skip_ tran_result	0	0=Save transient data for AMI 1=Skip transient result generation for AMI (useful for long runs of data that need not be saved)
ami.ui_ bins	500	Number of histogram bins to use for generating the 3D and contour data
ami.use_ clock_ times	1	1=Turn on sampling of the eye diagram based on the clock time values produced by the AMI receiver model. 0=Do not use clock times on the AMI receiver. The eye diagram is sampled at even UI intervals. The results of clock time sampling can be observed only in the Statistical Eye Plot for the "EyeAfterProbe" quantity.

TSMC-TMI Options Reference

These options apply to the Circuit simulation of MOSFETs using your TSMC Modeling Interface (TMI) model files and DLLs from TSMC. From the **Solution Options** window, click the **TSMC-TMI Options** tab:



Click the **Enable TMI** check box to enable Circuit simulation of MOSFET components with TSMC-TMI models. The fields in the tab are activated.

Table 15: TMI Options

Option	Default Value	Description
tmiflag	2.01	TMI specification version (1.00, 2.00, or 2.01)
tmipath	None	Path and directory containing the TMI compiled library

Option	Default Value	Description
tmiinput	None	Path and name of TMI configuration file.

The current release of the Circuit solver supports additional TSMC-TMI options not listed on the tab or table only with their default values: **tmiage** =0 (off), **tmisave** =0 (off), and **modmonte** =0 (off).

Layout Editor Layer Colors

The Electronics Desktop supports layer mapping files (.tech) with the following format:

- A forward slash, “/”, is the comment character.
- Units may be specified with a “UNITS <string>” entry that precedes the layer information t entries, where “<string>” is any of the allowed desktop length units; the default is “nm”.
- Each layer information entry is specified by a line that contains the following:

  <import layer> — Name of the DXF layer

  <destination layer> — The layout layer in the **Layout Editor** (i.e., the name to map the DXF layer to)

  <layer color> — A color string choice from those listed

  <layer elevation> — Double

  <layer height> — Double

  The follow list shows the various color string choices with corresponding RGB values:

  snow RGB = { 255, 250, 250},

  ghost white RGB = { 248, 248, 255},

  GhostWhite RGB = { 248, 248, 255},

  white smoke RGB = { 245, 245, 245},

  WhiteSmoke RGB = { 245, 245, 245},

  gainsboro RGB = { 220, 220, 220},

  floral white RGB = { 255, 250, 240},

  FloralWhite RGB = { 255, 250, 240},

  old lace RGB = { 253, 245, 230},

  OldLace RGB = { 253, 245, 230},

linen RGB = { 250, 240, 230},

antique white RGB = { 250, 235, 215},

AntiqueWhite RGB = { 250, 235, 215},

papaya whip RGB = { 255, 239, 213},

PapayaWhip RGB = { 255, 239, 213},

blanched almond RGB = { 255, 235, 205},

BlanchedAlmond RGB = { 255, 235, 205},

bisque RGB = { 255, 228, 196},

peach puff RGB = { 255, 218, 185},

PeachPuff RGB = { 255, 218, 185},

navajo white RGB = { 255, 222, 173},

NavajoWhite RGB = { 255, 222, 173},

moccasin RGB = { 255, 228, 181},

cornsilk RGB = { 255, 248, 220},

ivory RGB = { 255, 255, 240},

lemon chiffon RGB = { 255, 250, 205},

LemonChiffon RGB = { 255, 250, 205},

seashell RGB = { 255, 245, 238},

honeydew RGB = { 240, 255, 240},

mint cream RGB = { 245, 255, 250},

MintCream RGB = { 245, 255, 250},

azure RGB = { 240, 255, 255},

alice blue RGB = { 240, 248, 255},

AliceBlue RGB = { 240, 248, 255},

lavender RGB = { 230, 230, 250},

lavender blush RGB = { 255, 240, 245},

LavenderBlush RGB = { 255, 240, 245},

misty rose RGB = { 255, 228, 225},

MistyRose RGB = { 255, 228, 225},

white RGB = { 255, 255, 255},

black RGB = { 0, 0, 0},

dark slate gray RGB = { 47, 79, 79},

DarkSlateGray RGB = { 47, 79, 79},

dark slate grey RGB = { 47, 79, 79},

DarkSlateGrey RGB = { 47, 79, 79},

dim gray RGB = { 105, 105, 105},

DimGray RGB = { 105, 105, 105},

dim grey RGB = { 105, 105, 105},

DimGrey RGB = { 105, 105, 105},

slate gray RGB = { 112, 128, 144},

SlateGray RGB = { 112, 128, 144},

slate grey RGB = { 112, 128, 144},

SlateGrey RGB = { 112, 128, 144},

light slate gray RGB = { 119, 136, 153},

LightSlateGray RGB = { 119, 136, 153},

light slate grey RGB = { 119, 136, 153},

LightSlateGrey RGB = { 119, 136, 153},

gray RGB = { 190, 190, 190},

grey RGB = { 190, 190, 190},

light grey RGB = { 211, 211, 211},

LightGrey RGB = { 211, 211, 211},

light gray RGB = { 211, 211, 211},

LightGray RGB = { 211, 211, 211},

midnight blue RGB = { 25, 25, 112},

MidnightBlue RGB = { 25, 25, 112},

navy RGB = { 0, 0, 128},

navy blue RGB = { 0, 0, 128},

NavyBlue RGB = { 0, 0, 128},

cornflower blue RGB = { 100, 149, 237},

CornflowerBlue RGB = { 100, 149, 237},

dark slate blue RGB = { 72, 61, 139},

DarkSlateBlue RGB = { 72, 61, 139},

slate blue RGB = { 106, 90, 205},

SlateBlue RGB = { 106, 90, 205},

medium slate blue RGB = { 123, 104, 238},

MediumSlateBlue RGB = { 123, 104, 238},

light slate blue RGB = { 132, 112, 255},

LightSlateBlue RGB = { 132, 112, 255},

medium blue RGB = { 0, 0, 205},

MediumBlue RGB = { 0, 0, 205},

royal blue RGB = { 65, 105, 225},

RoyalBlue RGB = { 65, 105, 225},

blue RGB = { 0, 0, 255},

dodger blue RGB = { 30, 144, 255},

DodgerBlue RGB = { 30, 144, 255},

deep sky blue RGB = { 0, 191, 255},

DeepSkyBlue RGB = { 0, 191, 255},

sky blue RGB = { 135, 206, 235},

SkyBlue RGB = { 135, 206, 235},

light sky blue RGB = { 135, 206, 250},

LightSkyBlue RGB = { 135, 206, 250},

steel blue RGB = { 70, 130, 180},

SteelBlue RGB = { 70, 130, 180},

light steel blue RGB = { 176, 196, 222}, LightSteelBlue RGB = { 176, 196, 222},

light blue RGB = { 173, 216, 230},

LightBlue RGB = { 173, 216, 230},

powder blue RGB = { 176, 224, 230},

PowderBlue RGB = { 176, 224, 230},

pale turquoise RGB = { 175, 238, 238},

PaleTurquoise RGB = { 175, 238, 238},

dark turquoise RGB = { 0, 206, 209},

DarkTurquoise RGB = { 0, 206, 209},

medium turquoise RGB = { 72, 209, 204},

MediumTurquoise RGB = { 72, 209, 204},

turquoise RGB = { 64, 224, 208},

cyan RGB = { 0, 255, 255},

light cyan RGB = { 224, 255, 255},

LightCyan RGB = { 224, 255, 255},

cadet blue RGB = { 95, 158, 160},

CadetBlue RGB = { 95, 158, 160},

medium aquamarine RGB = { 102, 205, 170},

MediumAquamarine RGB = { 102, 205, 170},

aquamarine RGB = { 127, 255, 212},

dark green RGB = { 0, 100, 0},

DarkGreen RGB = { 0, 100, 0},

dark olive green RGB = { 85, 107, 47},

DarkOliveGreen RGB = { 85, 107, 47},

dark sea green RGB = { 143, 188, 143},

DarkSeaGreen RGB = { 143, 188, 143},

sea green RGB = { 46, 139, 87},

SeaGreen RGB = { 46, 139, 87},

medium sea green RGB = { 60, 179, 113},

MediumSeaGreen RGB = { 60, 179, 113},

light sea green RGB = { 32, 178, 170},

LightSeaGreen RGB = { 32, 178, 170},

pale green RGB = { 152, 251, 152},

PaleGreen RGB = { 152, 251, 152},

spring green RGB = { 0, 255, 127},

SpringGreen RGB = { 0, 255, 127},

lawn green RGB = { 124, 252, 0},

LawnGreen RGB = { 124, 252, 0},

green RGB = { 0, 255, 0},

chartreuse RGB = { 127, 255, 0},

medium spring green RGB = { 0, 250, 154},

MediumSpringGreen RGB = { 0, 250, 154},

green yellow RGB = { 173, 255, 47},

GreenYellow RGB = { 173, 255, 47},

lime green RGB = { 50, 205, 50},

LimeGreen RGB = { 50, 205, 50},

yellow green RGB = { 154, 205, 50},

YellowGreen RGB = { 154, 205, 50},

forest green RGB = { 34, 139, 34},

ForestGreen RGB = { 34, 139, 34},

olive drab RGB = { 107, 142, 35},

OliveDrab RGB = { 107, 142, 35},

dark khaki RGB = { 189, 183, 107},

DarkKhaki RGB = { 189, 183, 107},

khaki RGB = { 240, 230, 140},

pale goldenrod RGB = { 238, 232, 170},

PaleGoldenrod RGB = { 238, 232, 170},

light goldenrod yellow RGB = { 250, 250, 210},

LightGoldenrodYellow RGB = { 250, 250, 210},

light yellow RGB = { 255, 255, 224},

LightYellow RGB = { 255, 255, 224},

yellow RGB = { 255, 255, 0},

gold RGB = { 255, 215, 0},

light goldenrod RGB = { 238, 221, 130},

LightGoldenrod RGB = { 238, 221, 130},

goldenrod RGB = { 218, 165, 32},

dark goldenrod RGB = { 184, 134, 11},

DarkGoldenrod RGB = { 184, 134, 11},

rosy brown RGB = { 188, 143, 143},

RosyBrown RGB = { 188, 143, 143},

indian red RGB = { 205, 92, 92},

IndianRed RGB = { 205, 92, 92},

saddle brown RGB = { 139, 69, 19},

SaddleBrown RGB = { 139, 69, 19},

sienna RGB = { 160, 82, 45},

peru RGB = { 205, 133, 63},

burlywood RGB = { 222, 184, 135},

beige RGB = { 245, 245, 220},

wheat RGB = { 245, 222, 179},

sandy brown RGB = { 244, 164, 96},

SandyBrown RGB = { 244, 164, 96},

tan RGB = { 210, 180, 140},

chocolate RGB = { 210, 105, 30},

firebrick RGB = { 178, 34, 34},

brown RGB = { 165, 42, 42},

dark salmon RGB = { 233, 150, 122},

DarkSalmon RGB = { 233, 150, 122},

salmon RGB = { 250, 128, 114},

light salmon RGB = { 255, 160, 122},

LightSalmon RGB = { 255, 160, 122},

orange RGB = { 255, 165, 0},

dark orange RGB = { 255, 140, 0},

DarkOrange RGB = { 255, 140, 0},

coral RGB = { 255, 127, 80},

light coral RGB = { 240, 128, 128},

LightCoral RGB = { 240, 128, 128},

tomato RGB = { 255, 99, 71},

orange red RGB = { 255, 69, 0},

OrangeRed RGB = { 255, 69, 0},

red RGB = { 255, 0, 0},

hot pink RGB = { 255, 105, 180},

HotPink RGB = { 255, 105, 180},

deep pink RGB = { 255, 20, 147},

DeepPink RGB = { 255, 20, 147},

pink RGB = { 255, 192, 203},

light pink RGB = { 255, 182, 193},

LightPink RGB = { 255, 182, 193},

pale violet red RGB = { 219, 112, 147},

PaleVioletRed RGB = { 219, 112, 147},

maroon RGB = { 176, 48, 96},

medium violet red RGB = { 199, 21, 133},

MediumVioletRed RGB = { 199, 21, 133},

violet red RGB = { 208, 32, 144},

VioletRed RGB = { 208, 32, 144},

magenta RGB = { 255, 0, 255},

violet RGB = { 238, 130, 238},

plum RGB = { 221, 160, 221},

orchid RGB = { 218, 112, 214},

medium orchid RGB = { 186, 85, 211},

MediumOrchid RGB = { 186, 85, 211},

dark orchid RGB = { 153, 50, 204},

DarkOrchid RGB = { 153, 50, 204},

dark violet RGB = { 148, 0, 211},

DarkViolet RGB = { 148, 0, 211},

blue violet RGB = { 138, 43, 226},

BlueViolet RGB = { 138, 43, 226},

purple RGB = { 160, 32, 240},

medium purple RGB = { 147, 112, 219},

MediumPurple RGB = { 147, 112, 219},

thistle RGB = { 216, 191, 216},

snow1 RGB = { 255, 250, 250},

snow2 RGB = { 238, 233, 233},

snow3 RGB = { 205, 201, 201},

snow4 RGB = { 139, 137, 137},

seashell1 RGB = { 255, 245, 238},

seashell2 RGB = { 238, 229, 222},

seashell3 RGB = { 205, 197, 191},

seashell4 RGB = { 139, 134, 130},

AntiqueWhite1 RGB = { 255, 239, 219},

AntiqueWhite2 RGB = { 238, 223, 204},

AntiqueWhite3 RGB = { 205, 192, 176},

AntiqueWhite4 RGB = { 139, 131, 120},

bisque1 RGB = { 255, 228, 196},

bisque2 RGB = { 238, 213, 183},

bisque3 RGB = { 205, 183, 158},

bisque4 RGB = { 139, 125, 107},

PeachPuff1 RGB = { 255, 218, 185},

PeachPuff2 RGB = { 238, 203, 173},

PeachPuff3 RGB = { 205, 175, 149},

PeachPuff4 RGB = { 139, 119, 101},

NavajoWhite1 RGB = { 255, 222, 173},

NavajoWhite2 RGB = { 238, 207, 161},

NavajoWhite3 RGB = { 205, 179, 139},

NavajoWhite4 RGB = { 139, 121, 94},

LemonChiffon1 RGB = { 255, 250, 205},

LemonChiffon2 RGB = { 238, 233, 191},

LemonChiffon3 RGB = { 205, 201, 165},

LemonChiffon4 RGB = { 139, 137, 112},

cornsilk1 RGB = { 255, 248, 220},

cornsilk2 RGB = { 238, 232, 205},

cornsilk3 RGB = { 205, 200, 177},

cornsilk4 RGB = { 139, 136, 120},

ivory1 RGB = { 255, 255, 240},

ivory2 RGB = { 238, 238, 224},

ivory3 RGB = { 205, 205, 193},

ivory4 RGB = { 139, 139, 131},

honeydew1 RGB = { 240, 255, 240},

honeydew2 RGB = { 224, 238, 224},

honeydew3 RGB = { 193, 205, 193},

honeydew4 RGB = { 131, 139, 131},

LavenderBlush1 RGB = { 255, 240, 245},

LavenderBlush2 RGB = { 238, 224, 229},

LavenderBlush3 RGB = { 205, 193, 197},

LavenderBlush4 RGB = { 139, 131, 134},

MistyRose1 RGB = { 255, 228, 225},

MistyRose2 RGB = { 238, 213, 210},

MistyRose3 RGB = { 205, 183, 181},

MistyRose4 RGB = { 139, 125, 123},

azure1 RGB = { 240, 255, 255},

azure2 RGB = { 224, 238, 238},

azure3 RGB = { 193, 205, 205},

azure4 RGB = { 131, 139, 139},

SlateBlue1 RGB = { 131, 111, 255},

SlateBlue2 RGB = { 122, 103, 238},

SlateBlue3 RGB = { 105, 89, 205},

SlateBlue4 RGB = { 71, 60, 139},

RoyalBlue1 RGB = { 72, 118, 255},

RoyalBlue2 RGB = { 67, 110, 238},

RoyalBlue3 RGB = { 58, 95, 205},

RoyalBlue4 RGB = { 39, 64, 139},

blue1 RGB = { 0, 0, 255},

blue2 RGB = { 0, 0, 238},

blue3 RGB = { 0, 0, 205},

blue4 RGB = { 0, 0, 139},

DodgerBlue1 RGB = { 30, 144, 255},

DodgerBlue2 RGB = { 28, 134, 238},

DodgerBlue3 RGB = { 24, 116, 205},

DodgerBlue4 RGB = { 16, 78, 139},

SteelBlue1 RGB = { 99, 184, 255},

SteelBlue2 RGB = { 92, 172, 238},

SteelBlue3 RGB = { 79, 148, 205},

SteelBlue4 RGB = { 54, 100, 139},

DeepSkyBlue1 RGB = { 0, 191, 255},

DeepSkyBlue2 RGB = { 0, 178, 238},

DeepSkyBlue3 RGB = { 0, 154, 205},

DeepSkyBlue4 RGB = { 0, 104, 139},

SkyBlue1 RGB = { 135, 206, 255},

SkyBlue2 RGB = { 126, 192, 238},

SkyBlue3 RGB = { 108, 166, 205},

SkyBlue4 RGB = { 74, 112, 139},

LightSkyBlue1 RGB = { 176, 226, 255},

LightSkyBlue2 RGB = { 164, 211, 238},

LightSkyBlue3 RGB = { 141, 182, 205},

LightSkyBlue4 RGB = { 96, 123, 139},

SlateGray1 RGB = { 198, 226, 255},

SlateGray2 RGB = { 185, 211, 238},

SlateGray3 RGB = { 159, 182, 205},

SlateGray4 RGB = { 108, 123, 139},

LightSteelBlue1 RGB = { 202, 225, 255},

LightSteelBlue2 RGB = { 188, 210, 238},

LightSteelBlue3 RGB = { 162, 181, 205},

LightSteelBlue4 RGB = { 110, 123, 139},

LightBlue1 RGB = { 191, 239, 255},

LightBlue2 RGB = { 178, 223, 238},

LightBlue3 RGB = { 154, 192, 205},

LightBlue4 RGB = { 104, 131, 139},

LightCyan1 RGB = { 224, 255, 255},

LightCyan2 RGB = { 209, 238, 238},

LightCyan3 RGB = { 180, 205, 205},

LightCyan4 RGB = { 122, 139, 139},

PaleTurquoise1 RGB = { 187, 255, 255},

PaleTurquoise2 RGB = { 174, 238, 238},

PaleTurquoise3 RGB = { 150, 205, 205},

PaleTurquoise4 RGB = { 102, 139, 139},

CadetBlue1 RGB = { 152, 245, 255},

CadetBlue2 RGB = { 142, 229, 238},

CadetBlue3 RGB = { 122, 197, 205},

CadetBlue4 RGB = { 83, 134, 139},

turquoise1 RGB = { 0, 245, 255},

turquoise2 RGB = { 0, 229, 238},

turquoise3 RGB = { 0, 197, 205},

turquoise4 RGB = { 0, 134, 139},

cyan1 RGB = { 0, 255, 255},

cyan2 RGB = { 0, 238, 238},

cyan3 RGB = { 0, 205, 205},

cyan4 RGB = { 0, 139, 139},

DarkSlateGray1 RGB = { 151, 255, 255},

DarkSlateGray2 RGB = { 141, 238, 238},

DarkSlateGray3 RGB = { 121, 205, 205},

DarkSlateGray4 RGB = { 82, 139, 139},

aquamarine1 RGB = { 127, 255, 212},

aquamarine2 RGB = { 118, 238, 198},

aquamarine3 RGB = { 102, 205, 170},

aquamarine4 RGB = { 69, 139, 116},

DarkSeaGreen1 RGB = { 193, 255, 193},

DarkSeaGreen2 RGB = { 180, 238, 180},

DarkSeaGreen3 RGB = { 155, 205, 155},

DarkSeaGreen4 RGB = { 105, 139, 105},

SeaGreen1 RGB = { 84, 255, 159},

SeaGreen2 RGB = { 78, 238, 148},

SeaGreen3 RGB = { 67, 205, 128},

SeaGreen4 RGB = { 46, 139, 87},

PaleGreen1 RGB = { 154, 255, 154},

PaleGreen2 RGB = { 144, 238, 144},

PaleGreen3 RGB = { 124, 205, 124},

PaleGreen4 RGB = { 84, 139, 84},

SpringGreen1 RGB = { 0, 255, 127},

SpringGreen2 RGB = { 0, 238, 118},

SpringGreen3 RGB = { 0, 205, 102},

SpringGreen4 RGB = { 0, 139, 69},

green1 RGB = { 0, 255, 0},

green2 RGB = { 0, 238, 0},

green3 RGB = { 0, 205, 0},

green4 RGB = { 0, 139, 0},

chartreuse1 RGB = { 127, 255, 0},

chartreuse2 RGB = { 118, 238, 0},

chartreuse3 RGB = { 102, 205, 0},

chartreuse4 RGB = { 69, 139, 0},

OliveDrab1 RGB = { 192, 255, 62},

OliveDrab2 RGB = { 179, 238, 58},

OliveDrab3 RGB = { 154, 205, 50},

OliveDrab4 RGB = { 105, 139, 34},

DarkOliveGreen1 RGB = { 202, 255, 112},

DarkOliveGreen2 RGB = { 188, 238, 104},

DarkOliveGreen3 RGB = { 162, 205, 90},

DarkOliveGreen4 RGB = { 110, 139, 61},

khaki1 RGB = { 255, 246, 143},

khaki2 RGB = { 238, 230, 133},

khaki3 RGB = { 205, 198, 115},

khaki4 RGB = { 139, 134, 78},

LightGoldenrod1 RGB = { 255, 236, 139},

LightGoldenrod2 RGB = { 238, 220, 130},

LightGoldenrod3 RGB = { 205, 190, 112},

LightGoldenrod4 RGB = { 139, 129, 76},

LightYellow1 RGB = { 255, 255, 224},

LightYellow2 RGB = { 238, 238, 209},

LightYellow3 RGB = { 205, 205, 180},

LightYellow4 RGB = { 139, 139, 122},

yellow1 RGB = { 255, 255, 0},

yellow2 RGB = { 238, 238, 0},

yellow3 RGB = { 205, 205, 0},

yellow4 RGB = { 139, 139, 0},

gold1 RGB = { 255, 215, 0},

gold2 RGB = { 238, 201, 0},

gold3 RGB = { 205, 173, 0},

gold4 RGB = { 139, 117, 0},

goldenrod1 RGB = { 255, 193, 37},

goldenrod2 RGB = { 238, 180, 34},

goldenrod3 RGB = { 205, 155, 29},

goldenrod4 RGB = { 139, 105, 20},

DarkGoldenrod1 RGB = { 255, 185, 15},

DarkGoldenrod2 RGB = { 238, 173, 14},

DarkGoldenrod3 RGB = { 205, 149, 12},

DarkGoldenrod4 RGB = { 139, 101, 8},

RosyBrown1 RGB = { 255, 193, 193},

RosyBrown2 RGB = { 238, 180, 180},

RosyBrown3 RGB = { 205, 155, 155},

RosyBrown4 RGB = { 139, 105, 105},

IndianRed1 RGB = { 255, 106, 106},

IndianRed2 RGB = { 238, 99, 99},

IndianRed3 RGB = { 205, 85, 85},

IndianRed4 RGB = { 139, 58, 58},

sienna1 RGB = { 255, 130, 71},

sienna2 RGB = { 238, 121, 66},

sienna3 RGB = { 205, 104, 57},

sienna4 RGB = { 139, 71, 38},

burlywood1 RGB = { 255, 211, 155},

burlywood2 RGB = { 238, 197, 145},

burlywood3 RGB = { 205, 170, 125},

burlywood4 RGB = { 139, 115, 85},

wheat1 RGB = { 255, 231, 186},

wheat2 RGB = { 238, 216, 174},

wheat3 RGB = { 205, 186, 150},

wheat4 RGB = { 139, 126, 102},

tan1 RGB = { 255, 165, 79},

tan2 RGB = { 238, 154, 73},

tan3 RGB = { 205, 133, 63},

tan4 RGB = { 139, 90, 43},

chocolate1 RGB = { 255, 127, 36},

chocolate2 RGB = { 238, 118, 33},

chocolate3 RGB = { 205, 102, 29},

chocolate4 RGB = { 139, 69, 19},

firebrick1 RGB = { 255, 48, 48},

firebrick2 RGB = { 238, 44, 44},

firebrick3 RGB = { 205, 38, 38},

firebrick4 RGB = { 139, 26, 26},

brown1 RGB = { 255, 64, 64},

brown2 RGB = { 238, 59, 59},

brown3 RGB = { 205, 51, 51},

brown4 RGB = { 139, 35, 35},

salmon1 RGB = { 255, 140, 105},

salmon2 RGB = { 238, 130, 98},

salmon3 RGB = { 205, 112, 84},

salmon4 RGB = { 139, 76, 57},

LightSalmon1 RGB = { 255, 160, 122},

LightSalmon2 RGB = { 238, 149, 114},

LightSalmon3 RGB = { 205, 129, 98},

LightSalmon4 RGB = { 139, 87, 66},

orange1 RGB = { 255, 165, 0},

orange2 RGB = { 238, 154, 0},

orange3 RGB = { 205, 133, 0},

orange4 RGB = { 139, 90, 0},

DarkOrange1 RGB = { 255, 127, 0},

DarkOrange2 RGB = { 238, 118, 0},

DarkOrange3 RGB = { 205, 102, 0},

DarkOrange4 RGB = { 139, 69, 0},

coral1 RGB = { 255, 114, 86},

coral2 RGB = { 238, 106, 80},

coral3 RGB = { 205, 91, 69},

coral4 RGB = { 139, 62, 47},

tomato1 RGB = { 255, 99, 71},

tomato2 RGB = { 238, 92, 66},

tomato3 RGB = { 205, 79, 57},

tomato4 RGB = { 139, 54, 38},

OrangeRed1 RGB = { 255, 69, 0},

OrangeRed2 RGB = { 238, 64, 0},

OrangeRed3 RGB = { 205, 55, 0},

OrangeRed4 RGB = { 139, 37, 0},

red1 RGB = { 255, 0, 0},

red2 RGB = { 238, 0, 0},

red3 RGB = { 205, 0, 0},

red4 RGB = { 139, 0, 0},

DeepPink1 RGB = { 255, 20, 147},

DeepPink2 RGB = { 238, 18, 137},

DeepPink3 RGB = { 205, 16, 118},

DeepPink4 RGB = { 139, 10, 80},

HotPink1 RGB = { 255, 110, 180},

HotPink2 RGB = { 238, 106, 167},

HotPink3 RGB = { 205, 96, 144},

HotPink4 RGB = { 139, 58, 98},

pink1 RGB = { 255, 181, 197},

pink2 RGB = { 238, 169, 184},

pink3 RGB = { 205, 145, 158},

pink4 RGB = { 139, 99, 108},

LightPink1 RGB = { 255, 174, 185},

LightPink2 RGB = { 238, 162, 173},

LightPink3 RGB = { 205, 140, 149},

LightPink4 RGB = { 139, 95, 101},

PaleVioletRed1 RGB = { 255, 130, 171},

PaleVioletRed2 RGB = { 238, 121, 159},

PaleVioletRed3 RGB = { 205, 104, 137},

PaleVioletRed4 RGB = { 139, 71, 93},

maroon1 RGB = { 255, 52, 179},

maroon2 RGB = { 238, 48, 167},

maroon3 RGB = { 205, 41, 144},

maroon4 RGB = { 139, 28, 98},

VioletRed1 RGB = { 255, 62, 150},

VioletRed2 RGB = { 238, 58, 140},

VioletRed3 RGB = { 205, 50, 120},

VioletRed4 RGB = { 139, 34, 82},

magenta1 RGB = { 255, 0, 255},

magenta2 RGB = { 238, 0, 238},

magenta3 RGB = { 205, 0, 205},

magenta4 RGB = { 139, 0, 139},

orchid1 RGB = { 255, 131, 250},

orchid2 RGB = { 238, 122, 233},

orchid3 RGB = { 205, 105, 201},

orchid4 RGB = { 139, 71, 137},

plum1 RGB = { 255, 187, 255},

plum2 RGB = { 238, 174, 238},

plum3 RGB = { 205, 150, 205},

plum4 RGB = { 139, 102, 139},

MediumOrchid1 RGB = { 224, 102, 255},

MediumOrchid2 RGB = { 209, 95, 238},

MediumOrchid3 RGB = { 180, 82, 205},

MediumOrchid4 RGB = { 122, 55, 139},

DarkOrchid1 RGB = { 191, 62, 255},

DarkOrchid2 RGB = { 178, 58, 238},

DarkOrchid3 RGB = { 154, 50, 205},

DarkOrchid4 RGB = { 104, 34, 139},

purple1 RGB = { 155, 48, 255},

purple2 RGB = { 145, 44, 238},

purple3 RGB = { 125, 38, 205},

purple4 RGB = { 85, 26, 139},

MediumPurple1 RGB = { 171, 130, 255},

MediumPurple2 RGB = { 159, 121, 238},

MediumPurple3 RGB = { 137, 104, 205},

MediumPurple4 RGB = { 93, 71, 139},

thistle1 RGB = { 255, 225, 255},

thistle2 RGB = { 238, 210, 238},

thistle3 RGB = { 205, 181, 205},

thistle4 RGB = { 139, 123, 139},

gray0 RGB = { 0, 0, 0},

grey0 RGB = { 0, 0, 0},

gray1 RGB = { 3, 3, 3},

grey1 RGB = { 3, 3, 3},

gray2 RGB = { 5, 5, 5},

grey2 RGB = { 5, 5, 5},

gray3 RGB = { 8, 8, 8},

grey3 RGB = { 8, 8, 8},

gray4 RGB = { 10, 10, 10},

grey4 RGB = { 10, 10, 10},

gray5 RGB = { 13, 13, 13},

grey5 RGB = { 13, 13, 13},

gray6 RGB = { 15, 15, 15},

grey6 RGB = { 15, 15, 15},

gray7 RGB = { 18, 18, 18},

grey7 RGB = { 18, 18, 18},

gray8 RGB = { 20, 20, 20},

grey8 RGB = { 20, 20, 20},

gray9 RGB = { 23, 23, 23},

grey9 RGB = { 23, 23, 23},

gray10 RGB = { 26, 26, 26},

grey10 RGB = { 26, 26, 26},

gray11 RGB = { 28, 28, 28},

grey11 RGB = { 28, 28, 28},

gray12 RGB = { 31, 31, 31},

grey12 RGB = { 31, 31, 31},

gray13 RGB = { 33, 33, 33},

grey13 RGB = { 33, 33, 33},

gray14 RGB = { 36, 36, 36},

grey14 RGB = { 36, 36, 36},

gray15 RGB = { 38, 38, 38},

grey15 RGB = { 38, 38, 38},

gray16 RGB = { 41, 41, 41},

grey16 RGB = { 41, 41, 41},

gray17 RGB = { 43, 43, 43},

grey17 RGB = { 43, 43, 43},

gray18 RGB = { 46, 46, 46},

grey18 RGB = { 46, 46, 46},

gray19 RGB = { 48, 48, 48},

grey19 RGB = { 48, 48, 48},

gray20 RGB = { 51, 51, 51},

grey20 RGB = { 51, 51, 51},

gray21 RGB = { 54, 54, 54},

grey21 RGB = { 54, 54, 54},

gray22 RGB = { 56, 56, 56},

grey22 RGB = { 56, 56, 56},

gray23 RGB = { 59, 59, 59},

grey23 RGB = { 59, 59, 59},

gray24 RGB = { 61, 61, 61},

grey24 RGB = { 61, 61, 61},

gray25 RGB = { 64, 64, 64},

grey25 RGB = { 64, 64, 64},

gray26 RGB = { 66, 66, 66},

grey26 RGB = { 66, 66, 66},

gray27 RGB = { 69, 69, 69},

grey27 RGB = { 69, 69, 69},

gray28 RGB = { 71, 71, 71},

grey28 RGB = { 71, 71, 71},

gray29 RGB = { 74, 74, 74},

grey29 RGB = { 74, 74, 74},

gray30 RGB = { 77, 77, 77},

grey30 RGB = { 77, 77, 77},

gray31 RGB = { 79, 79, 79},

grey31 RGB = { 79, 79, 79},

gray32 RGB = { 82, 82, 82},

grey32 RGB = { 82, 82, 82},

gray33 RGB = { 84, 84, 84},

grey33 RGB = { 84, 84, 84},

gray34 RGB = { 87, 87, 87},

grey34 RGB = { 87, 87, 87},

gray35 RGB = { 89, 89, 89},

grey35 RGB = { 89, 89, 89},

gray36 RGB = { 92, 92, 92},

grey36 RGB = { 92, 92, 92},

gray37 RGB = { 94, 94, 94},

grey37 RGB = { 94, 94, 94},

gray38 RGB = { 97, 97, 97},

grey38 RGB = { 97, 97, 97},

gray39 RGB = { 99, 99, 99},

grey39 RGB = { 99, 99, 99},

gray40 RGB = { 102, 102, 102},

grey40 RGB = { 102, 102, 102},

gray41 RGB = { 105, 105, 105},

grey41 RGB = { 105, 105, 105},

gray42 RGB = { 107, 107, 107},

grey42 RGB = { 107, 107, 107},

gray43 RGB = { 110, 110, 110},

grey43 RGB = { 110, 110, 110},

gray44 RGB = { 112, 112, 112},

grey44 RGB = { 112, 112, 112},

gray45 RGB = { 115, 115, 115},

grey45 RGB = { 115, 115, 115},

gray46 RGB = { 117, 117, 117},

grey46 RGB = { 117, 117, 117},

gray47 RGB = { 120, 120, 120},

grey47 RGB = { 120, 120, 120},

gray48 RGB = { 122, 122, 122},

grey48 RGB = { 122, 122, 122},

gray49 RGB = { 125, 125, 125},

grey49 RGB = { 125, 125, 125},

gray50 RGB = { 127, 127, 127},

grey50 RGB = { 127, 127, 127},

gray51 RGB = { 130, 130, 130},

grey51 RGB = { 130, 130, 130},

gray52 RGB = { 133, 133, 133},

grey52 RGB = { 133, 133, 133},

gray53 RGB = { 135, 135, 135},

grey53 RGB = { 135, 135, 135},

gray54 RGB = { 138, 138, 138},

grey54 RGB = { 138, 138, 138},

gray55 RGB = { 140, 140, 140},

grey55 RGB = { 140, 140, 140},

gray56 RGB = { 143, 143, 143},

grey56 RGB = { 143, 143, 143},

gray57 RGB = { 145, 145, 145},

grey57 RGB = { 145, 145, 145},

gray58 RGB = { 148, 148, 148},

grey58 RGB = { 148, 148, 148},

gray59 RGB = { 150, 150, 150},

grey59 RGB = { 150, 150, 150},

gray60 RGB = { 153, 153, 153},

grey60 RGB = { 153, 153, 153},

gray61 RGB = { 156, 156, 156},

grey61 RGB = { 156, 156, 156},

gray62 RGB = { 158, 158, 158},

grey62 RGB = { 158, 158, 158},

gray63 RGB = { 161, 161, 161},

grey63 RGB = { 161, 161, 161},

gray64 RGB = { 163, 163, 163},

grey64 RGB = { 163, 163, 163},

gray65 RGB = { 166, 166, 166},

grey65 RGB = { 166, 166, 166},

gray66 RGB = { 168, 168, 168},

grey66 RGB = { 168, 168, 168},

gray67 RGB = { 171, 171, 171},

grey67 RGB = { 171, 171, 171},

gray68 RGB = { 173, 173, 173},

grey68 RGB = { 173, 173, 173},

gray69 RGB = { 176, 176, 176},

grey69 RGB = { 176, 176, 176},

gray70 RGB = { 179, 179, 179},

grey70 RGB = { 179, 179, 179},

gray71 RGB = { 181, 181, 181},

grey71 RGB = { 181, 181, 181},

gray72 RGB = { 184, 184, 184},

grey72 RGB = { 184, 184, 184},

gray73 RGB = { 186, 186, 186},

grey73 RGB = { 186, 186, 186},

gray74 RGB = { 189, 189, 189},

grey74 RGB = { 189, 189, 189},

gray75 RGB = { 191, 191, 191},

grey75 RGB = { 191, 191, 191},

gray76 RGB = { 194, 194, 194},

grey76 RGB = { 194, 194, 194},

gray77 RGB = { 196, 196, 196},

grey77 RGB = { 196, 196, 196},

gray78 RGB = { 199, 199, 199},

grey78 RGB = { 199, 199, 199},

gray79 RGB = { 201, 201, 201},

grey79 RGB = { 201, 201, 201},

gray80 RGB = { 204, 204, 204},

grey80 RGB = { 204, 204, 204},

gray81 RGB = { 207, 207, 207},

grey81 RGB = { 207, 207, 207},

gray82 RGB = { 209, 209, 209},

grey82 RGB = { 209, 209, 209},

gray83 RGB = { 212, 212, 212},

grey83 RGB = { 212, 212, 212},

gray84 RGB = { 214, 214, 214},

grey84 RGB = { 214, 214, 214},

gray85 RGB = { 217, 217, 217},

grey85 RGB = { 217, 217, 217},

gray86 RGB = { 219, 219, 219},

grey86 RGB = { 219, 219, 219},

gray87 RGB = { 222, 222, 222},

grey87 RGB = { 222, 222, 222},

gray88 RGB = { 224, 224, 224},

grey88 RGB = { 224, 224, 224},

gray89 RGB = { 227, 227, 227},

grey89 RGB = { 227, 227, 227},

gray90 RGB = { 229, 229, 229},

grey90 RGB = { 229, 229, 229},

gray91 RGB = { 232, 232, 232},

grey91 RGB = { 232, 232, 232},

gray92 RGB = { 235, 235, 235},

grey92 RGB = { 235, 235, 235},

gray93 RGB = { 237, 237, 237},

grey93 RGB = { 237, 237, 237},

gray94 RGB = { 240, 240, 240},

grey94 RGB = { 240, 240, 240},

gray95 RGB = { 242, 242, 242},

grey95 RGB = { 242, 242, 242},

gray96 RGB = { 245, 245, 245},

grey96 RGB = { 245, 245, 245},

gray97 RGB = { 247, 247, 247},

grey97 RGB = { 247, 247, 247},

gray98 RGB = { 250, 250, 250},

grey98 RGB = { 250, 250, 250},

gray99 RGB = { 252, 252, 252},

grey99 RGB = { 252, 252, 252},

gray100 RGB = { 255, 255, 255},

grey100 RGB = { 255, 255, 255},

dark grey RGB = { 169, 169, 169},

DarkGrey RGB = { 169, 169, 169},

dark gray RGB = { 169, 169, 169},

DarkGray RGB = { 169, 169, 169},

dark blue RGB = { 0, 0, 139},

DarkBlue RGB = { 0, 0, 139},

dark cyan RGB = { 0, 139, 139},

DarkCyan RGB = { 0, 139, 139},

dark magenta RGB = { 139, 0, 139},

DarkMagenta RGB = { 139, 0, 139},

dark red RGB = { 139, 0, 0},

DarkRed RGB = { 139, 0, 0},

light green RGB = { 144, 238, 144},

LightGreen RGB = { 144, 238, 144}

Circuit and Layout Unit Types

The Electronics Desktop supports a subset of the unit types shown in this list. To confirm that a particular unit is supported, consult the default-value drop-down-menus for the various unit specifications listed in the Default Units Tab of the **General Options** window.

Unit Type	String Representation	Description
Angle	deg	
Angle	rad	
AngularSpeed	deg_per_sec	
AngularSpeed	deg_per_min	
AngularSpeed	deg_per_hr	
AngularSpeed	rad_per_sec	
AngularSpeed	rad_per_min	
AngularSpeed	rad_per_hr	
AngularSpeed	rpm	
Capacitance	farad	
Capacitance	fF	

Capacitance	mF	
Capacitance	nF	
Capacitance	pF	
Capacitance	uF	
Conductance	fSie	
Conductance	mSie	
Conductance	nSie	
Conductance	pSie	
Conductance	sie	
Conductance	uSie	
Current	A	
Current	fA	
Current	kA	
Current	mA	
Current	nA	
Current	pA	
Current	uA	
ElectricFieldStrength	V_per_meter	
Flux	Wb	
Force	fNewton	
Force	pNewton	
Force	nNewton	
Force	uNewton	
Force	mNewton	
Force	newton	
Force	kNewton	
Force	megNewton	
Force	gNewton	
Force	PoundsForce	
Frequency	GHz	
Frequency	Hz	
Frequency	kHz	
Frequency	MHz	
Frequency	rps	
Frequency	thz	
Inductance	fH	femto henri

Inductance	H
Inductance	mH
Inductance	nH
Inductance	pH
Inductance	uH
Length	cm
Length	ft
Length	in
Length	km
Length	meter
Length	mm
Length	mil
Length	nm
Length	uin
Length	um
MagFieldStrength	Oe
MagFieldStrength	kOe
MagFieldStrength	A_per_meter
MagFieldStrength	kA_per_meter
MagInduction	uGauss
MagInduction	mGauss
MagInduction	gauss
MagInduction	kGauss
MagInduction	uTesla
MagInduction	mTesla
MagInduction	tesla
MagInduction	kTesla
NoiseSpectrum	dBc/Hz
Power	dBm
Power	dBW
Power	fW
Power	pW
Power	nW
Power	uW
Power	mW
Power	W

Power	kW
Power	HP
Power	Btu_per_hr
Pressure	n_per_meter_sq
Pressure	kn_per_meter_sq
Pressure	megn_per_meter_sq
Pressure	gn_per_meter_sq
Pressure	psi
Pressure	kpsi
Pressure	megpsi
Pressure	gpsi
Resistance	GOhm
Resistance	kOhm
Resistance	megohm
Resistance	mOhm
Resistance	ohm
Speed	mm_per_sec
Speed	cm_per_sec
Speed	m_per_sec
Speed	m_per_hr
Speed	inches_per_sec
Speed	feet_per_sec
Speed	feet_per_min
Speed	miles_per_hour
Speed	miles_per_min
Speed	miles_per_sec
Speed	km_per_hour
Speed	km_per_min
Speed	km_per_sec
Temperature	cel
Temperature	fah
Temperature	kel
Time	fs
Time	ms
Time	ns

Time	ps	
Time	s	
Time	us	
Torque	fNewtonMeter	
Torque	pNewtonMeter	
Torque	nNewtonMeter	
Torque	uNewtonMeter	
Torque	mNewtonMeter	
Torque	NewtonMeter	
Torque	kNewtonMeter	
Torque	megNewtonMeter	
Torque	gNewtonMeter	
Torque	FootPounds	
Voltage	dBV	
Voltage	fV	
Voltage	kV	
Voltage	megV	
Voltage	mV	
Voltage	nV	
Voltage	pV	
Voltage	uv	
Voltage	V	
Volume	l	
Volume	ml	
Weight	ug	
Weight	mg	
Weight	gram	
Weight	kg	
Weight	oz	
Weight	lb	
Frequency	Hz	Hertz
Frequency	kHz	Kilo-Hertz
Frequency	milliHz	Milli-Hertz
Frequency	MHz	Mega-Hertz
Frequency	per_sec	Per second
Resistance	Ohm	Ohm

Resistance	megohm	Mega-Ohm
Resistance	mOhm	Milli-Ohm
Resistance	kOhm	Kilo-Ohm
Resistance	uOhm	Micro-Ohm
Resistance	Gohm	Giga-Ohm
Reactance	Ohm	Ohm
Reactance	megohm	Mega-Ohm
Reactance	mOhm	Milli-Ohm
Reactance	kOhm	Kilo-Ohm
Reactance	uOhm	Micro-Ohm
Reactance	Gohm	Giga-Ohm
Impedance	Ohm	Ohm
Impedance	megohm	Mega-Ohm
Impedance	mOhm	Milli-Ohm
Impedance	kOhm	Kilo-Ohm
Impedance	uOhm	Micro-Ohm
Impedance	Gohm	Giga-Ohm
Conductance	Sie	Siemens
Conductance	megSie	Mega-Siemens
Conductance	uSie	Micro-Siemens
Conductance	mSie	Milli-Siemens
Conductance	kSie	Kilo-Siemens
Conductance	mho	Mhos
Conductance	perohm	Reciprocal Ohm
Conductance	ApV	Ampere per Volt
Admittance	Sie	Siemens
Admittance	megSie	Mega-Siemens
Admittance	uSie	Micro-Siemens
Admittance	mSie	Milli-Siemens
Admittance	kSie	Kilo-Siemens
Admittance	mho	Mhos
Susceptance	Sie	Siemens
Susceptance	megSie	Mega-Siemens
Susceptance	uSie	Micro-Siemens
Susceptance	mSie	Milli-Siemens
Susceptance	kSie	Kilo-Siemens

Susceptance	mho	Mhos
Inductance	h	Henry
Inductance	uh	Micro-Henry
Inductance	nh	Nano-Henry
Inductance	mh	Milli-Henry
Capacitance	farad	Farad
Capacitance	pf	Pico-Farad
Capacitance	uf	Micro-Farad
Capacitance	nf	Nano-Farad
Capacitance	mf	Milli-Farad
Length	meter	Meter
Length	mm	Milli-Meter
Length	km	Kilo-Meter
Length	cm	Centi-Meter
Length	um	Micro-Meter
Length	nm	Nano-Meter
Length	micron	Microns
Length	dm	Deci-Meter
Length	pm	Pico-Meter
Length	fm	Femto-Meter
Length	mil	Milli-inch
Length	in	Inch
Length	ft	Foot (am.)
Length	yd	Yard (am.)
Length	mile	Mile (am.)
Length	mileTerr	Terrestrial mile
Length	mileNaut	Nautical mile
Length	lightyear	Light year
Time	s	Second
Time	hour	Hour
Time	ps	Pico-Second
Time	us	Micro-Second
Time	ns	Nano-Second
Time	ms	Milli-Second
Time	min	Minute
Time	fs	Fempto-second

Time	day	Day
Angle	rad	Radian
Angle	deg	Degree
Angle	degsec	Second
Angle	degmin	Minute
Power	W	Watt
Power	megW	Mega-Watt
Power	uW	Micro-Watt
Power	mW	Milli-Watt
Power	VA	Volt_Amps
Power	kW	Kilo-Watt
Power	gW	Giga-Watt
Power	J_per_s	Joule per second
Voltage	v	Volt
Voltage	fv	Femto-Volt
Voltage	pv	Pico-Volt
Voltage	uv	Micro-Volt
Voltage	megv	Mega-Volt
Voltage	nv	Nano-Volt
Voltage	mv	Milli-Volt
Voltage	kv	Kilo-Volt
Voltage	gv	Giga-Volt
Current	a	Ampere
Current	ua	Micro-Ampere
Current	na	Nano-Ampere
Current	ma	Milli-Ampere
Current	ka	Kilo-Ampere
Current	C_per_s	Coulomb per Second
Temperature	kel	Kelvin
Temperature	mkel	Milli-Kelvin
Temperature	cel	Celsius
Temperature	dkel	Deci-Kelvin
Temperature	fah	Farenheit
Temperature	ckel	Centi-Kelvin
Weight	kg	Kilogram
Weight	gram	Gram

Weight	ug	Micro-Gram
Weight	mg	Milli-Gram
Weight	ton	Tons
Volume	m3	Cubic-Meter
Volume	galUS	Gallon (am.)
Volume	cup	Cup
Volume	galUK	Gallon (br.)
Volume	l	Liter
Volume	mm3	Cubic-Centi-Meter
Volume	ml	Cubic-Centi-Meter
MagInduction	tesla	Tesla
MagInduction	gauss	Gauss
MagFieldStrength	A_per_meter	Ampere per meter
MagFieldStrength	oersted	Oersted
Force	newton	Newton
Force	megnewton	Mega-Newton
Force	unewton	Micro-Newton
Force	mnewton	Milli-Newton
Force	dyne	Dyne
Force	knewton	Kilo-Newton
Force	kp	Kilo-pond
Force	poundsForce	Pound-force
Torque	NewtonMeter	Newton meter
Torque	kNewtonMeter	Kilo-Newton meter
Torque	mNewtonMeter	Milli-Newton meter
Torque	uNewtonMeter	Micro-Newton meter
Torque	cNewtonMeter	Centi-Newton meter
Speed	m_per_sec	Meter per second
Speed	km_per_hour	Kilometer per hour
Speed	miles_per_hour	Miles per hour
AngularSpeed	rad_per_sec	Radian per second
AngularSpeed	deg_per_sec	Degree per second
AngularSpeed	rev_per_min	Revolutions per minute
AngularSpeed	rev_per_sec	Revolutions per second
AngularSpeed	per_second	Revolutions per second
Flux	weber	Weber

Flux	vh	Volt-hour
Flux	mx	Maxwell
Flux	vs	Volt-second
ElectricFieldStrength	v_per_meter	Volt per meter
ElectricFieldStrength	v_per_cm	Volt per centi-meter
Pressure	pascal	Pascal
Pressure	psi	Pounds per square inch
Pressure	hPascal	Hekto-Pascal
Pressure	gPascal	Giga-Pascal
Pressure	techAtm	Technical atmosphere
Pressure	kPascal	Kilo-Pascal
Pressure	mbar	Milli-bar
Pressure	stAtm	Standard atmosphere
Pressure	mPascal	Milli-Pascal
Pressure	n_per_meter_sq	Newton per square-meter
Pressure	megPascal	Mega-Pascal
Pressure	uPascal	Micro-Pascal
Pressure	torr	Torr
Pressure	bar	Bar
Pressure	cPascal	Centi-Pascal
Pressure	mmHg	Milli-meter mercury
Pressure	dPascal	Deci-Pascal
Pressure	mmh2o	Milli-meter water column
Acceleration	m_per_s2	Meter per square-second
Acceleration	cm_per_s2	Centi-Meter per square-second
Acceleration	in_per_s2	Inch per second-squared
Acceleration	mm_per_s2	Milli-Meter per square-second
AIGB	F_per_ghalfs_ per_m	(Farad per gram) root second per meter
AIGB	Fs2_per_ghalf_ per_m	(Farad second squared per gram) root per meter
AmountOfSubstance	mol	Mol
AmountOfSubstance	umol	Micro-Mol
AmountOfSubstance	nmol	Nano-Mol
AmountOfSubstance	mmol	Milli-Mol
AmountOfSubstance	kmol	Kilo-Mol

AngleCoefficient	per_rad	Per radian
AngleCoefficient	V_per_V_per_deg	Volt per Volt per degree
AngleCoefficient	per_deg	Per degree
AngleCoefficient	V_per_V_per_rad	Volt per Volt per radian
AngularAcceleration	rad_per_s2	Radian per square-second
AngularAcceleration	deg_per_s2	Degrees per square-second
AngularAcceleration	per_s2	Revolutions per square-second
AngularDamping	Nms_per_rad	Newton-meter-second per radian
AngularDamping	dNms_per_rad	Deci-Newton-meter-second per radian
AngularDamping	kNms_per_rad	Kilo-Newton-meter-second per radian
AngularJerk	rad_per_sec2	Radian per second-cubed
AngularJerk	deg_per_sec2	Degrees per second-cubed
AngularMomentum	kgm2_per_sec	Kilo-gram square-meter per second
AngularMomentum	nms	Newton-meter-second
AngularStiffness	nm_per_rad	Newton-meter per radian
AngularStiffness	nm_per_deg	Newton-meter per degree
AngularWindage	nms2_per_rad2	BasicUnit
AngularWindage	kgm2_per_rad2	Kilogram meter per rad squared
Area	m2	Square-Meter
Area	km2	Square-Kilo-Meter
Area	ft2	Square foot
Area	in2	Square Inch
Area	mm2	Square-Milli-Meter
Area	cm2	Square-Centi-Meter
Area	um2	Micro-meter squared
AreaCoefficient	per_m2	Per square-meter
AreaCoefficient	per_cm2	Per square-centi-meter
ArealFlowRate	m2_per_s	Square-meter per second
ArealFlowRate	m2_per_min	Square-meter per minute
ArealFlowRate	m2_per_hour	Square-meter per hour
AreaPerPower	m2_per_W	Square-meter per Watt
AreaPerPower	m2_per_Js	Square-meter per Joule-second
AreaPerVoltage	m2_per_V	Square-meter per volt

AreaPerVoltage	Am2_per_W	Ampere-square-meter per Volt
AreaPerVoltage	Am2_per_kW	Ampere-square-meter per Kilo-Watt
AreaPerVoltage	m2_per_kV	Square-meter per Kilo-volt
AreaPerVoltageTemperature	m2_per_Vkel	Square-meter per Volt and Kelvin
AreaPerVoltageTemperature	Am2_per_Wkel	Ampere-square-meter per Watt-Kelvin
BIGB	F_per_ghalfs_ per_Vm	(Farad per gram) root second per Volt-meter
BIGB	Fs2_per_ghalf_ per_Vm	(Farad second squared per gram) root per Volt-meter
CapacitancePerArea	F_per_m2	Farad per square-meter
CapacitancePerArea	nF_per_m2	Nano-Farad per square-meter
CapacitancePerArea	uF_per_m2	Micro-Farad per square-meter
CapacitancePerArea	F_per_cm2	Farad per square-cent-meter
CapacitancePerArea	mF_per_m2	Milli-Farad per square-meter
CapacitancePerAreaPerVoltage	F_per_Vm2	Farad per Volt and square-meter
CapacitancePerAreaPerVoltage	mF_per_Vm2	Milli-Farad per Volt and square-meter
CapacitancePerAreaPerVoltage	pF_per_Vm2	Pico-Farad per Volt and square-meter
CapacitancePerAreaPerVoltage	nF_per_Vm2	Nano-Farad per Volt and square-meter
CapacitancePerAreaPerVoltage	uF_per_Vm2	Micro-Farad per Volt and square-meter
CapacitancePerLength	F_per_m	Farad per meter
CapacitancePerLength	pF_per_m	Pico-Farad per meter
CapacitancePerLength	nF_per_m	Nano-Farad per meter
CapacitancePerLength	mF_per_m	Milli-Farad per meter
CapacitancePerLength	uF_per_m	Micro-Farad per meter
CapacitanceTemperatureCoeff	F_per_Kel	Farad per Kelvin
CapacitanceTemperatureCoeff	pF_per_Cel	Pico-Farad per degree Celsius
CapacitanceTemperatureCoeff	pF_per_Kel	Pico-Farad per Kelvin
CapacitanceTemperatureCoeff	mF_per_Cel	Milli-Farad per degree Celsius
CapacitanceTemperatureCoeff	pF_per_Fah	Pico=Farad per degree Farenheit
CapacitanceTemperatureCoeff	nF_per_Kel	Nano-Farad per Kelvin
CapacitanceTemperatureCoeff	mF_per_Kel	Milli-Farad per Kelvin

CapacitanceTemperatureCoeff	F_per_Cel	Farad per degree Celsius
CapacitanceTemperatureCoeff	mF_per_Fah	Milli-Farad per degree Farenheit
CapacitanceTemperatureCoeff	nF_per_Cel	Nano-Farad per degree Celsius
CapacitanceTemperatureCoeff	F_per_Fah	Farad per degree Farenheit
CapacitanceTemperatureCoeff	uF_per_Kel	Micro-Farad per Kelvin
CapacitanceTemperatureCoeff	nF_per_Fah	Nano-Farad per degree Farenheit
CapacitanceTemperatureCoeff	uF_per_Cel	Micro-Farad per degree Celsius
CapacitanceTemperatureCoeff	uF_per_Fah	Micro-Farad per degree Farenheit
Charge	Charge	Coulomb
Charge	Ahour	Ampere hour
Charge	uC	Micro-Coulomb
Charge	mC	Milli-Coulomb
Charge	nC	Nano-Coulomb
Charge	kC	Kilo-Coulomb
Charge	As	Ampere second
Compliance	m_per_N	Meter per Newton
Compliance	in_per_lbf	Inch per pound-force
Compliance	cm_per_N	Centi-meter per Newton
ConductancePerLength	S_per_m	Siemens per meter
ConductancePerLength	mS_per_m	Milli-Siemens per meter
ConductancePerLength	kS_per_m	Kilo-Siemens per meter
ConductancePerLength	uS_per_m	Micro-Siemens per meter
ConductancePerLength	cS_per_m	Centi-Siemens per meter
CurrentChangeRate	A_per_s	Ampere per second
CurrentChangeRate	A_per_hour	Ampere per hour
CurrentChangeRate	mA_per_s	Milli-Ampere per second
CurrentChangeRate	kA_per_s	Kilo-Ampere per second
CurrentChangeRate	uA_per_s	Micro-Ampere per second
CurrentChangeRate	A_per_min	Ampere per minute
CurrentDensity	A_per_m2	Ampere per square-meter
CurrentDensity	uA_per_m2	Micro-Ampere per square-meter
CurrentDensity	mA_per_cm2	Milli-Ampere per square-centi-meter
CurrentDensity	mA_per_m2	Milli-Ampere per square-meter
CurrentDensity	A_per_cm2	Ampere per square-centi-meter
CurrentDensity	uA_per_cm2	Micro-Ampere per square-centi-meter

CurrentGain	A_per_A	Ampere per Ampere
CurrentGain	A_per_mA	Ampere per Milli-Ampere
CurrentGain	mA_per_A	Milli-Ampere per Ampere
CurrentLengthPerVoltage	Am_per_V	Ampere-meter per Volt
CurrentLengthPerVoltage	mAm_per_V	Milli-Ampere-meter per Volt
CurrentLengthPerVoltage	uAm_per_V	Micro-Ampere-meter per Volt
CurrentPerCharge	A_per_C	Ampere per Coulomb
CurrentPerCharge	A_per_Ahour	Ampere per Ampere-hour
CurrentPerCharge	A_per_As	Ampere per Ampere-second
CurrentPerIrradiance	A_per_W_per_m2	Ampere per Watt per meter-squared
CurrentPerIrradiance	kA_per_W_per_m2	Kilo-Ampere per Watt per meter-squared
CurrentPerIrradiance	uA_per_W_per_m2	Micro-Ampere per Watt per meter-squared
CurrentPerIrradiance	mA_per_W_per_m2	Milli-Ampere per Watt per meter-squared
CurrentPerLength	A_per_m	Ampere per meter
CurrentPerLength	mA_per_m	Milli-Ampere per meter
CurrentPerLength	kA_per_m	Kilo-Ampere per meter
CurrentPerLength	uA_per_m	Micro-Ampere per meter
CurrentPerTemperature2_half	A_per_Kel2half	Ampere per Kelvin to the power of 2.5
CurrentPerTemperatureCubed	A_per_Kel3	Ampere per Kelvin-cubed
CurrentPerTemperatureDiffCubed	A_per_diff_Kel3	Ampere per Kelvin-cubed
CurrentPerTemperatureDiffCubed	A_per_Cel3	Ampere per degree Celsius cubed
CurrentSquaredTime	A2s	Ampere-squared second
CurrentSquaredTime	mA2s	Milli-Ampere-squared second
CurrentTemperatureCoeff	A_per_Kel	Ampere per Kelvin
CurrentTemperatureCoeff	A_per_Cel	Ampere per degree Celsius
CurrentTemperatureCoeff	mA_per_Kel	Milli-Ampere per Kelvin
CurrentTemperatureCoeff	mA_per_Fah	Milli-Ampere per degree Farenheit
CurrentTemperatureCoeff	kA_per_Kel	Kilo-Ampere per Kelvin
CurrentTemperatureCoeff	A_per_Fah	Ampere per degree Farenheit
CurrentTemperatureCoeff	uA_per_Kel	Micro-Ampere per Kelvin

CurrentTemperatureCoeff	uA_per_Cel	Micro-Ampere per degree Celsius
CurrentTemperatureCoeff	mA_per_Cel	Milli-Ampere per degree Celsius
Damping	Ns_per_m	Newton second per meter
Damping	kNs_per_m	Kilo-Newton second per meter
Damping	mNs_per_m	Milli-Newton second per meter
Damping	cNs_per_m	Centi-Newton second per meter
Damping	dNs_per_m	Deci-Newton second per meter
Density	kg_per_m3	Kilogram per Cubic-Meter
Density	g_per_cm3	Gram per Cubic-Centimeter
Density	kg_per_l	Kilo-Gram per Liter
Density	kg_per_dm3	Kilo-gram per cubic-deci-meter
Density	g_per_l	Gram per Liter
Displacement	meter	Meter
Displacement	mm	Milli-Meter
Displacement	km	Kilo-Meter
Displacement	cm	Centi-Meter
Displacement	um	Micro-Meter
Displacement	nm	Nano-Meter
Displacement	micron	Microns
Displacement	dm	Deci-Meter
Displacement	pm	Pico-Meter
Displacement	fm	Femto-Meter
Displacement	mil	Milli-inch
Displacement	in	Inch
Displacement	ft	Foot (am.)
Displacement	yd	Yard (am.)
Displacement	mile	Mile (am.)
Displacement	mileTerr	Terrestrial mile
Displacement	mileNaut	Nautical mile
Displacement	lightyear	Light year
ElectricFlux	C	Coulomb
ElectricFlux	uC	Micro-Coulomb
ElectricFlux	nC	Nano-Coulomb
ElectricFlux	mC	Milli-Coulomb
ElectricFlux	As	Ampere-second
ElectricFluxDensity	C_per_m2	Coulomb per square-meter

ElectricFluxDensity	nC_per_m2	Nano-Coulomb per square-meter
ElectricFluxDensity	uC_per_m2	Micro-Coulomb per square-meter
ElectricFluxDensity	mC_per_m2	Milli-Coulomb per square-meter
Energy	J	Joule
Energy	Whour	Watt-hour
Energy	kJ	Kilo-Joule
Energy	eV	Electron-Volt
Energy	GJ	Giga-Joule
Energy	kWhour	kilo-Watt-hours
Energy	erg	Erg
Energy	Ws	Watt-second
Energy	megJ	Mega-Joule
Energy	uJ	Micro-Joule
Energy	mJ	Milli-Joule
FluidicConductance	m3_per_Pas	Cubic-meter per Pascal-second
FluidicConductance	cm3_per_Pas	Cubic-centi-meter per Pascal-second
FluidicCapacitance	m3_per_Pa	Cubic-meter per Pascal
FluidicCapacitance	cm3_per_Pa	Centi-meter-cubed per Pascal
FluidicResistance	Pas_per_m3	Pascal-second per cubic-meter
FluidicResistance	Ns_per_m5	Newton-second per meter to the power of 5
HeatFlow	W	Watt
HeatFlow	J_per_s	Joule per second
Illuminance	lx	Lux
Illuminance	klx	Kilo-Lux
Illuminance	meglx	Mega-Lux
Illuminance	lm_per_in2	Lumen per square-inch
Illuminance	W_per_m2	Watt per square-meter
Illuminance	W_per_cm2	Watt per square-centi-meter
Illuminance	lm_per_cm2	Lumen per square-centi-meter
Illuminance	lm_per_m2	Lumen per square-meter
InductancePerLength	H_per_m	Henry per meter
InductancePerLength	pH_per_m	Pico-Henry per meter
InductancePerLength	nH_per_m	Nano-Henry per meter
InductancePerLength	mH_per_m	Milli-Henry per meter

InductancePerLength	uH_per_m	Micro-Henry per meter
Inertance	Pas2_per_m3	Pascal square-second per cubic-meter
Inertance	Ns2_per_m5	Newton square second per meter to power 5
Inertance	kg_per_m4	Kilo-gram per meter to the power of 4
Irradiance	irrad_W_per_m2	Watt per meter-squared
Irradiance	megW_per_m2	Mega-Watt per meter-squared
Irradiance	mW_per_m2	Milli-Watt per meter-squared
Irradiance	irrad_W_per_cm2	Watt per cent-meter-squared
Irradiance	kW_per_m2	Kilo-Watt per meter-squared
Irradiance	uW_per_m2	Micro-Watt per meter-squared
Irradiance	W_per_in2	Watt per inch-squared
Jerk	m_per_s3	Meter per second cubed
Jerk	cm_per_s3	Centi-Meter per second cubed
Jerk	nm_per_s3	Nano-Meter per second cubed
Jerk	in_per_s3	Inch per second-cubed
Jerk	um_per_s3	Micro-Meter per second cubed
Jerk	mm_per_s3	Milli-Meter per second cubed
Length2PerVoltage2	m2_per_V2	Meter-squared per Volt-squared
LengthCoefficient	per_m	Per meter
LengthCoefficient	V_per_V_per_m	Volt per Volt per meter
LengthCoefficient	per_km	Per kilo-meter
LengthCoefficient	per_um	Per micro-meter
LengthCoefficient	per_in	Per inch
LengthCoefficient	per_cm	Per centi-meter
LengthCoefficient	V_per_V_per_in	Volt per Volt per inch
LengthCoefficient	per_mm	Per milli-meter
LengthPerVoltage	m_per_V	Meter per Volt
LengthPerVoltage	mm_per_V	Milli-Meter per Volt
LengthPerVoltage	km_per_V	Kilo-Meter per Volt
LengthPerVoltage	um_per_V	Micro-Meter per Volt
LengthPerVoltage	dm_per_V	Deci-Meter per Volt
LengthPerVoltage	cm_per_V	Centi-Meter per Volt
LengthPerVoltageRoot	m_per_Vhalf	Meter per Volt-root

LengthPerVoltageRoot	dm_per_Vhalf	Deci-Meter per Volt-root
LengthPerVoltageRoot	mm_per_Vhalf	Milli-Meter per Volt-root
LengthPerVoltageRoot	um_per_Vhalf	Micro-Meter per Volt-root
LengthPerVoltageRoot	km_per_Vhalf	Kilo-Meter per Volt-root
LengthPerVoltageRoot	cm_per_Vhalf	Centi-Meter per Volt-root
LuminousFlux	lm	Lumen
LuminousFlux	gm2_per_s3	Kilo-gram meter-squared per second-cubed
LuminousFlux	klm	Kilo-Lumen
LuminousFlux	mlm	Milli-Lumen
LuminousFlux	meglm	Mega-Lumen
LuminousIntensity	Cd	Candela
LuminousIntensity	GCd	Giga-Candela
LuminousIntensity	kCd	Kilo-Candela
LuminousIntensity	mCd	Milli-Candela
LuminousIntensity	megCd	Mega-Candela
MagneticReluctance	A_per_Wb	Ampere per Weber
MagneticReluctance	A_per_Vs	Ampere per Volt-second
MassFlowRate	kg_per_s	Kilogram per second
MassFlowRate	g_per_s	Gram per second
MMF	a	Ampere
MMF	ua	Micro-Ampere
MMF	na	Nano-Ampere
MMF	ma	Milli-Ampere
MMF	ka	Kilo-Ampere
MolarDensity	mol_per_m3	Mol per cubic-meter
MolarDensity	mol_per_dm3	Mol per cubic-deci-meter
MolarDensity	mol_per_cm3	Mol per cubic-centi-meter
MolarDensity	mol_per_l	Mol per liter
MolarEnergy	J_per_mol	Joule per mole
MolarEnergy	megJ_per_mol	Mega-Joule per mole
MolarEnergy	mJ_per_mol	Milli-Joule per mole
MolarEnergy	uJ_per_mol	Micro-Joule per mole
MolarEnergy	kJ_per_mol	Kilo-Joule per mole
MolarEnergy	gJ_per_mol	Giga-Joule per mole
MolarVelocity	mol_per_s	Mole per second

MolarVelocity	cmol_per_s	Centi-Mole per second
MolarVelocity	mmol_per_s	Milli-Mole per second
MolarVelocity	kmol_per_s	Kilo-Mole per second
MolarVelocity	umol_per_s	Micro-Mole per second
MolarViscosity	Pas_per_mol	Pascal second per Mol
MomentInertia	kgm2	Kilo-gram square-meter
MomentInertia	lbin2	Pound inch-squared
MomentInertia	lbft2	Pound foot-squared
Momentum	kgm_per_s	Kilo-gram meter per second
Momentum	gm_per_s	Gram meter per second
Percentage	percent	Per cent
PercentagePerTime	percent_per_s	Percent per second
PercentagePerTime	per_day	Per day
PercentagePerTime	percent_per_min	Percent per minute
PercentagePerTime	per_hour	Per hour
PercentagePerTime	percent_per_day	Percent per day
PercentagePerTime	percent_per_hour	Percent per hour
PercentagePerTime	per_min	Per minute
PercentagePerTime	per_s	Per second
Permeance	Vs_per_A	Volt-second per Ampere
Permeance	Wb_per_A	Weber per Ampere
PressureChangeRate	Pa_per_s	Pascal per second
PressureChangeRate	torr_per_hour	Torr per hour
PressureChangeRate	mmH2O_per_s	Milli-meter water column per second
PressureChangeRate	psi_per_s	PSI per second
PressureChangeRate	techatm_per_hour	Technical atmosphere per hour
PressureChangeRate	bar_per_hour	Bar per hour
PressureChangeRate	statm_per_s	Standard atmosphere per second
PressureChangeRate	mbar_per_s	Milli-Bar per second
PressureChangeRate	mmH2O_per_hour	Milli-meter water column per hour
PressureChangeRate	Pa_per_hour	Pascal per hour
PressureChangeRate	psi_per_hour	PSI per hour
PressureChangeRate	mmH2O_per_min	Milli-meter water column per minute

PressureChangeRate	techatm_per_min	Technical atmosphere per minute
PressureChangeRate	statm_per_hour	Standard atmosphere per hour
PressureChangeRate	torr_per_s	Torr per second
PressureChangeRate	mbar_per_hour	Milli-bar per hour
PressureChangeRate	techatm_per_s	Technical atmosphere per second
PressureChangeRate	bar_per_s	Bar per second
PressureChangeRate	statm_per_min	Standard atmosphere per minute
PressureChangeRate	torr_per_min	Torr per Minute
PressureChangeRate	Pa_per_min	Pascal per minute
PressureChangeRate	bar_per_min	Bar per minute
PressureChangeRate	mbar_per_min	Milli-bar per minute
PressureChangeRate	psi_per_min	PSI per minute
PressureCoefficient	per_Pa	Per Pascal
PressureCoefficient	per_statm	Per standard atmosphere
PressureCoefficient	per_psi	Per PSI
PressureCoefficient	per_mmH2O	Per Milli-meter water column
PressureCoefficient	per_mbar	Per Milli-bar
PressureCoefficient	V_per_Vper_Pa	Volt per Volt per Pascal
PressureCoefficient	per_mmHg	Per Milli-meter mercury
PressureCoefficient	per_techatm	Per technical atmosphere
PressureCoefficient	per_bar	Per bar
Ratio	db	Decibel
Ratio	bel	Bel
ReciprocalPower	per_W	One over Watt
ReciprocalPower	per_megW	One over Mega-Watt
ReciprocalPower	per_Js	One over Joule-second
ReciprocalPower	per_mW	One over Milli-Watt
ReciprocalPower	per_kW	One over Kilo-Watt
ReciprocalPower	per_gW	One over Giga-Watt
ReciprocalResistanceCharge	per_OhmC	One over Ohm-coulomb
ReciprocalResistanceCharge	per_OhmAs	One over Ohm-Ampere-second
ReciprocalResistanceCharge	per_OhmAhour	One over Ohm-Ampere-hour
ReciprocalResistanceTime	per_Ohms	One over Ohm-second
ReciprocalResistanceTime	per_Ohmmin	One over Ohm-minute
ReciprocalResistanceTime	per_Ohmhour	One over Ohm-hour
ResistancePerCharge	Ohm_per_C	Ohm per Coulomb

ResistancePerCharge	Ohm_per_Ahour	Ohm per Ampere-hour
ResistancePerCharge	megOhm_per_C	Mega-Ohm per Coulomb
ResistancePerCharge	nOhm_per_C	Nano-Ohm per Coulomb
ResistancePerCharge	mOhm_per_C	Milli-Ohm per Coulomb
ResistancePerCharge	uOhm_per_C	Micro-Ohm per Coulomb
ResistancePerCharge	kOhm_per_C	Kilo-Ohm per Coulomb
ResistancePerCharge	Ohm_per_As	Ohm per Ampere-second
ResistancePerCharge	gOhm_per_C	Giga-Ohm per Coulomb
ResistancePerLength	Ohm_per_m	Ohm per meter
ResistancePerLength	megOhm_per_m	Mega-Ohm per meter
ResistancePerLength	mOhm_per_m	Milli-Ohm per meter
ResistancePerLength	uOhm_per_m	Micro-Ohm per meter
ResistancePerLength	kOhm_per_m	Kilo-Ohm per meter
ResistancePerLength	Ohm_per_um	Ohm per micro-meter
ResistanceTemperatureCoeff	Ohm_per_Kel	Ohm per Kelvin
ResistanceTemperatureCoeff	mOhm_per_Kel	Milli-Ohm per Kelvin
ResistanceTemperatureCoeff	kOhm_per_Kel	Kilo-Ohm per Kelvin
ResistanceTemperatureCoeff	Ohm_per_Cel	Ohm per degree Celsius
Resistivity	Ohmm	Ohm Meter
Resistivity	Ohmcm	Ohm Centi-Meter
Resistivity	Ohmum	Ohm micro-meter
Resistivity	Ohmmm2_per_mm	Ohm Square-Milli-Meter per Meter
SpecificHeatCapacity	J_per_Kelkg	Joule per Kelvin and Kilo-gram
SpecificHeatCapacity	mJ_per_Kelkg	Milli-Joule per Kelvin and Kilo-gram
SpecificHeatCapacity	kJ_per_Kelkg	Kilo-Joule per Kelvin and Kilo-gram
Stiffness	N_per_m	Newton per Meter
Stiffness	lbf_per_in	Pound-force per inch
Stiffness	N_per_cm	Newton per centi-meter
Stiffness	kN_per_m	Kilo-Newton per Meter
SurfaceChargeDensity	surf_charge_C_per_m2	Coulomb per meter-squared
SurfaceChargeDensity	surf_charge_nC_per_m2	Nano-Coulomb per meter-squared
SurfaceChargeDensity	kC_per_m2	Micro-Coulomb per meter-

		squared
SurfaceChargeDensity	As_per_m2	Ampere-second per meter-squared
SurfaceChargeDensity	surf_charge_mC_per_m2	Milli-Coulomb per meter-squared
SurfaceMobility	m2_per_Vs	Square-meter per Volt-second
SurfaceMobility	cm2_per_Vs	Square-centi-meter per Volt-second
SurfaceMobilityPerVoltage	m2_per_V2s	Square-meter per Volt-second per Volt
SurfaceMobilityPerVoltage	cm2_per_V2s	Square-centi-meter per Volt-second per Volt
TemperatureAreaPerPower	kelm2_per_w	Kelvin square-meter per Watt
TemperatureAreaPerPower	celm2_per_w	Degree Celsius square-meter per Watt
TemperatureCoefficient	per_Kel	Per Kelvin
TemperatureCoefficient	percent_per_Cel	Percent per degree Celsius
TemperatureCoefficient	per_Cel	Per degree Celsius
TemperatureCoefficient	percent_per_Fah	Percent per degree Farenheit
TemperatureCoefficient	per_Fah	Per degree Farenheit
TemperatureCoefficient	percent_per_Kel	Percentage per Kelvin
TemperatureCoefficient2	per_Kel2	Per Kelvin-squared
TemperatureCoefficient2	per_Fah2	Per degree Fahrenheit squared
TemperatureCoefficient2	per_Cel2	Per degree Celsius squared
TemperatureDifference	keldiff	Kelvin
TemperatureDifference	mkeldiff	Milli-Kelvin
TemperatureDifference	celdiff	Celsius
ThermalCapacitance	J_per_Kel	Joule per Kelvin
ThermalCapacitance	Ws_per_Kel	Watt-second per Kelvin
ThermalConductance	W_per_Kel	Watt per Kelvin
ThermalConductance	kW_per_Cel	Kilo-Watt per degree Celsius
ThermalConductance	mW_per_Kel	Milli-Watt per Kelvin
ThermalConductance	kW_per_Kel	Kilo-Watt per Kelvin
ThermalConductance	mW_per_Cel	Milli-Watt per degree Celsius
ThermalConductance	W_per_Cel	Watt per degree Celsius
ThermalConductivity	W_per_Kelm	Watt per Kelvin and Meter
ThermalConductivity	mW_per_Kelm	Milli-Watt per Kelvin and Meter

ThermalConvection	w_per_m2kel	Watt per square-meter and Kelvin
ThermalConvection	w_per_cm2kel	Watt per square centi-meter and Kelvin
ThermalRadiationCoeff	W_per_Kel4	Watt per Kelvin to the power of 4
ThermalRadiationCoeff	kW_per_Kel4	Kilo-Watt per Kelvin to the power of 4
ThermalRadiationCoeff	mW_per_Kel4	Milli-Watt per Kelvin to the power of 4
ThermalRadiationConstant	W_per_m2Kel4	Watt per square meter and Kelvin to the power of 4
ThermalRadiationConstant	W_per_cm2Kel4	W per square centi-meter and Kelvin to the Power of 4
ThermalResistance	Kel_per_W	Kelvin per Watt
ThermalResistance	Kels_per_J	Kelvin-second per Joule
TimePerAngle	s_per_rad	Second per radian
TimePerAngle	s_per_deg	Second per degree
TimePerAngle	ms_per_rad	Milli-Second per radian
TimePerAngle	s_per_rev	Second per revolution
TimeSqPerAngleSq	s2_per_rad2	(seconds per rad) squared
TimeSqPerAngleSq	s2_per_deg2	(seconds per degree) squared
TransconductanceParameter	A_per_V	Ampere per Volt
TransconductanceParameter	mA_per_V	Milli-Ampere per Volt
TransconductanceParameter	kA_per_V	Kilo-Ampere per Volt
TransistorConstant	A_per_V2	Ampere per square Volt
TransistorConstant	mA_per_V2	Milli-Ampere per square Volt
TranslationalAcceleration	trans_accel_m_per_s2	Meter per square second
TranslationalAcceleration	trans_accel_cm_per_s2	Centi-Meter per square second
TranslationalAcceleration	dm_per_s2	Deci-Meter per square second
TranslationalAcceleration	trans_accel_in_per_s2	Inch per second-squared
VelocitySaturation	vel_sat_m_per_V	Meter per Volt
VelocitySaturation	vel_sat_mm_per_V	Milli-Meter per Volt
VelocitySaturation	vel_sat_um_per_V	Micro-Meter per Volt
VelocitySaturation	vel_sat_cm_per_	Centi-Meter per Volt

	V	
VelocitySaturationPerVoltage	m_per_V2	Meter per Volt-squared
VelocitySaturationPerVoltage	cm_per_V2	Centi-Meter per Volt-squared
VelocitySaturationPerVoltage	um_per_V2	Micro-Meter per Volt-squared
VelocitySaturationPerVoltage	mm_per_V2	Milli-Meter per Volt-squared
Viscocity	Pas	Pascal-second
Viscocity	hPas	Hekto-Pascal-second
Viscocity	kPas	Kilo-Pascal-second
Viscocity	mPas	Milli-Pascal-second
Viscocity	poise	Poise
Viscocity	Ns_per_m2	Newton-second per square-meter
Viscocity	cpoise	Centi-poise
Viscocity	cPas	Centi-Pascal-second
Viscocity	dPas	Deci-Pascal-second
Viscocity	uPas	Micro-Pascal-second
ViscousFriction	vis_fric_Ns_per_m	Newton-second per meter
ViscousFriction	vis_fric_cNs_per_m	Centi-Newton-second per meter
ViscousFriction	vis_fric_mNs_per_m	Milli-Newton-second per meter
ViscousFriction	vis_fric_kNs_per_m	Kilo-Newton-second per meter
VoltageAccelerationCoefficient	V_per_m2_per_s2	Voltage per meter per second-squared
VoltageAccelerationCoefficient	mV_per_m2_per_s2	Milli-Voltage per meter per second-squared
VoltageChangeRate	V_per_s	Volt per second
VoltageChangeRate	V_per_min	Volt per minute
VoltageChangeRate	V_per_hour	Volt per hour
VoltageChangeRate	mV_per_s	Milli-Volt per second
VoltageChangeRate	kV_per_s	Kilo-Volt per second
VoltageCoefficient	per_V	Per Volt
VoltageCoefficient	per_mV	Per Milli-Volt
VoltageCoefficient	per_kV	Per Kilo-Volt
VoltageCoefficient2	per_V2	Per Voltage-square
VoltageCubed	V3	Voltage-cubed

VoltageCubed	mV3	Milli-volt-cubed
VoltageGain	V_per_V	Volt per Volt
VoltageGain	V_per_mV	Volt per Milli-Volt
VoltageGain	mV_per_V	Milli-Volt per Volt
VoltageJerkCoefficient	V_per_m_per_s3	Volt per meter per second-cubed
VoltageJerkCoefficient	mV_per_m_per_s3	Milli-Volt per meter per second-cubed
VoltageLength	Vm	Volt meter
VoltageLength	kVm	Kilo-Volt meter
VoltageLength	mVm	Milli-Volt meter
VoltageLength	uVm	Micro-Volt meter
VoltagePerCell	V_per_cell	Volt per cell
VoltagePerCell	megV_per_cell	Mega-Volt per cell
VoltagePerCell	kV_per_cell	Kilo-Volt per cell
VoltagePerCell	uV_per_cell	Micro-Volt per cell
VoltagePerCell	mV_per_cell	Milli-Volt per cell
VoltagePerCell	nV_per_cell	Nano-Volt per cell
VoltagePerCell	pV_per_cell	Pico-Volt per cell
VoltagePerCell	gV_per_cell	Giga-Volt per cell
VoltagePerLengthRoot	V_per_mhalf	Volt per meter-root
VoltagePressureRootCoeff	V_per_Pahalf	Volt per Pascal-root
VoltagePressureRootCoeff	mV_per_Pahalf	Milli-Volt per Pascal-root
VoltageRoot	Vhalf	Volt to the power of 0.5
VoltageRootCoefficient	per_Vhalf	Per Volt-root
VoltageTemperature10Coeff	V_per_Kel10	Volt per Kelvin power 10
VoltageTemperature10Coeff	V_per_Cel10	Volt per degree Celsius power 10
VoltageTemperature10Coeff	uV_per_Kel10	Micro-Volt per Kelvin power 10
VoltageTemperature10Coeff	mV_per_Kel10	Milli-Volt per Kelvin power 10
VoltageTemperature10Coeff	uV_per_Cel10	Micro-Volt per degree Celsius power 10
VoltageTemperature10Coeff	mV_per_Cel10	Milli-Volt per degree Celsius power 10
VoltageTemperature11Coeff	V_per_Kel11	Volt per Kelvin power 11
VoltageTemperature11Coeff	V_per_Cel11	Volt per degree Celsius power 11
VoltageTemperature11Coeff	uV_per_Kel11	Micro-Volt per Kelvin power 11
VoltageTemperature11Coeff	mV_per_Kel11	Milli-Volt per Kelvin power 11
VoltageTemperature11Coeff	uV_per_Cel11	Micro-Volt per degree Celsius

		power 11
VoltageTemperature11Coeff	mV_per_Cel11	Milli-Volt per degree Celsius power 11
VoltageTemperature12Coeff	V_per_Kel12	Volt per Kelvin power 12
VoltageTemperature12Coeff	V_per_Cel12	Volt per degree Celsius power 12
VoltageTemperature12Coeff	uV_per_Kel12	Micro-Volt per Kelvin power 12
VoltageTemperature12Coeff	mV_per_Kel12	Milli-Volt per Kelvin power 12
VoltageTemperature12Coeff	uV_per_Cel12	Micro-Volt per degree Celsius power 12
VoltageTemperature12Coeff	mV_per_Cel12	Milli-Volt per degree Celsius power 12
VoltageTemperature13Coeff	V_per_Kel13	Volt per Kelvin power 13
VoltageTemperature13Coeff	V_per_Cel13	Volt per degree Celsius power 13
VoltageTemperature13Coeff	uV_per_Kel13	Micro-Volt per Kelvin power 13
VoltageTemperature13Coeff	mV_per_Kel13	Milli-Volt per Kelvin power 13
VoltageTemperature13Coeff	uV_per_Cel13	Micro-Volt per degree Celsius power 13
VoltageTemperature13Coeff	mV_per_Cel13	Milli-Volt per degree Celsius power 13
VoltageTemperature14Coeff	V_per_Kel14	Volt per Kelvin power 14
VoltageTemperature14Coeff	V_per_Cel14	Volt per degree Celsius power 14
VoltageTemperature14Coeff	uV_per_Kel14	Micro-Volt per Kelvin power 14
VoltageTemperature14Coeff	mV_per_Kel14	Milli-Volt per Kelvin power 14
VoltageTemperature14Coeff	uV_per_Cel14	Micro-Volt per degree Celsius power 14
VoltageTemperature14Coeff	mV_per_Cel14	Milli-Volt per degree Celsius power 14
VoltageTemperature15Coeff	V_per_Kel15	Volt per Kelvin power 15
VoltageTemperature15Coeff	V_per_Cel15	Volt per degree Celsius power 15
VoltageTemperature15Coeff	uV_per_Kel15	Micro-Volt per Kelvin power 15
VoltageTemperature15Coeff	mV_per_Kel15	Milli-Volt per Kelvin power 15
VoltageTemperature15Coeff	uV_per_Cel15	Micro-Volt per degree Celsius power 15
VoltageTemperature15Coeff	mV_per_Cel15	Milli-Volt per degree Celsius power 15
VoltageTemperature2Coeff	V_per_Kel2	Volt per Kelvin-squared
VoltageTemperature2Coeff	V_per_Cel2	Volt per degree Celsius-squared
VoltageTemperature2Coeff	uV_per_Kel2	Micro-Volt per Kelvin-squared

VoltageTemperature2Coeff	mV_per_Kel2	Milli-Volt per Kelvin-squared
VoltageTemperature2Coeff	uV_per_Cel2	Micro-Volt per degree Celsius-squared
VoltageTemperature2Coeff	mV_per_Cel2	Milli-Volt per deree Celsius-squared
VoltageTemperature3Coeff	V_per_Kel3	Volt per Kelvin-cubed
VoltageTemperature3Coeff	V_per_Cel3	Volt per degree Celsius-cubed
VoltageTemperature3Coeff	uV_per_Kel3	Micro-Volt per Kelvin-cubed
VoltageTemperature3Coeff	mV_per_Kel3	Milli-Volt per Kelvin-cubed
VoltageTemperature3Coeff	uV_per_Cel3	Micro-Volt per degree Celsius-cubed
VoltageTemperature3Coeff	mV_per_Cel3	Milli-Volt per degree Celsius-cubed
VoltageTemperature4Coeff	V_per_Kel4	Volt per Kelvin power 4
VoltageTemperature4Coeff	V_per_Cel4	Volt per degree Celsius power 4
VoltageTemperature4Coeff	uV_per_Kel4	Micro-Volt per Kelvin power 4
VoltageTemperature4Coeff	mV_per_Kel4	Milli-Volt per Kelvin power 4
VoltageTemperature4Coeff	uV_per_Cel4	Micro-Volt per degree Celsius power 4
VoltageTemperature4Coeff	mV_per_Cel4	Milli-Volt per degree Celsius power 4
VoltageTemperature5Coeff	V_per_Kel5	Volt per Kelvin power 5
VoltageTemperature5Coeff	V_per_Cel5	Volt per degree Celsius power 5
VoltageTemperature5Coeff	uV_per_Kel5	Micro-Volt per Kelvin power 5
VoltageTemperature5Coeff	mV_per_Kel5	Milli-Volt per Kelvin power 5
VoltageTemperature5Coeff	uV_per_Cel5	Micro-Volt per degree Celsius power 5
VoltageTemperature5Coeff	mV_per_Cel5	Milli-Volt per degree Celsius power 5
VoltageTemperature6Coeff	V_per_Kel6	Volt per Kelvin power 6
VoltageTemperature6Coeff	V_per_Cel6	Volt per Kelvin power 6
VoltageTemperature6Coeff	uV_per_Kel6	Micro-Volt per Kelvin power 6
VoltageTemperature6Coeff	mV_per_Kel6	Milli-Volt per Kelvin power 6
VoltageTemperature6Coeff	uV_per_Cel6	Micro-Volt per degree Celsius power 6
VoltageTemperature6Coeff	mV_per_Cel6	Milli-Volt per degree Celsius power 6
VoltageTemperature7Coeff	V_per_Kel7	Volt per Kelvin power 7

VoltageTemperature7Coeff	V_per_Cel7	Volt per degree Celsius power 7
VoltageTemperature7Coeff	uV_per_Kel7	Micro-Volt per Kelvin power 7
VoltageTemperature7Coeff	mV_per_Kel7	Milli-Volt per Kelvin power 7
VoltageTemperature7Coeff	uV_per_Cel7	Micro-Volt per degree Celsius power 7
VoltageTemperature7Coeff	mV_per_Cel7	Milli-Volt per degree Celsius power 7
VoltageTemperature8Coeff	V_per_Kel8	Volt per Kelvin power 8
VoltageTemperature8Coeff	V_per_Cel8	Volt per degree Celsius power 8
VoltageTemperature8Coeff	uV_per_Kel8	Micro-Volt per Kelvin power 8
VoltageTemperature8Coeff	mV_per_Kel8	Milli-Volt per Kelvin power 8
VoltageTemperature8Coeff	uV_per_Cel8	Micro-Volt per degree Celsius power 8
VoltageTemperature8Coeff	mV_per_Cel8	Milli-Volt per degree Celsius power 8
VoltageTemperature9Coeff	V_per_Kel9	Volt per Kelvin power 9
VoltageTemperature9Coeff	V_per_Cel9	Volt per degree Celsius power 9
VoltageTemperature9Coeff	uV_per_Kel9	Micro-Volt per Kelvin power 9
VoltageTemperature9Coeff	mV_per_Kel9	Milli-Volt per Kelvin power 9
VoltageTemperature9Coeff	uV_per_Cel9	Micro-Volt per degree Celsius power 9
VoltageTemperature9Coeff	mV_per_Cel9	Milli-Volt per degree Celsius power 9
VoltageTemperatureCoeff	V_per_Kel	Volt per Kelvin
VoltageTemperatureCoeff	V_per_Cel	Volt per degrees Celsius
VoltageTemperatureCoeff	uV_per_Kel	Micro-Volt per Kelvin
VoltageTemperatureCoeff	mV_per_Kel	Milli-Volt per Kelvin
VoltageTemperatureCoeff	uV_per_Cel	Micro-Volt per degree Celsius
VoltageTemperatureCoeff	mV_per_Cel	Milli-Volt per degree Celsius
VolumeCoefficient	per_m3	Per cubic-meter
VolumeCoefficient	per_cm3	Per cubic-centi-meter
VolumeFlowConductance	vol_flow_con_ m3_per_Pas	Cubic-meter per Pascal-second
VolumeFlowConductance	vol_flow_con_ cm3_per_Pas	Centi-meter cubed per Pascal-second
VolumeFlowPerPressureRoot	m3_per_sPahalf	Meter-cubed per second per Pascal-root
VolumeFlowRate	m3_per_s	Cubic-meter per second

VolumeFlowRate	cm3_per_s	Cubic-centi-meter per second
VolumeFlowRate	m3_per_hour	Cubic-meter per hour
VolumeFlowRate	m3_per_min	Cubic-meter per minute
VolumeFlowRateChangeRate	m3_per_s2	Cubic-meter per square-second
VolumeFlowRateChangeRate	cm3_per_s2	Centi-meter cubed per second squared
WireCrossSection	m2wire	Meter squared
WireCrossSection	um2wire	Micro-meter squared
WireCrossSection	ft2wire	Square foot
WireCrossSection	in2wire	Square inch
WireCrossSection	mm2wire	Milli-meter squared
WireCrossSection	cm2wire	Centi-meter squared
WireCrossSection	AWG	American wire gauge
EnergyDensity	J_per_m3	joules per meter cubed
EnergyDensity	kJ_per_m3	kilo joules per meter cubed

Controlling Nexxim Output

For large circuits, significant memory space can be saved by specifying a set of values to output, rather than the default method of storing values for every node.

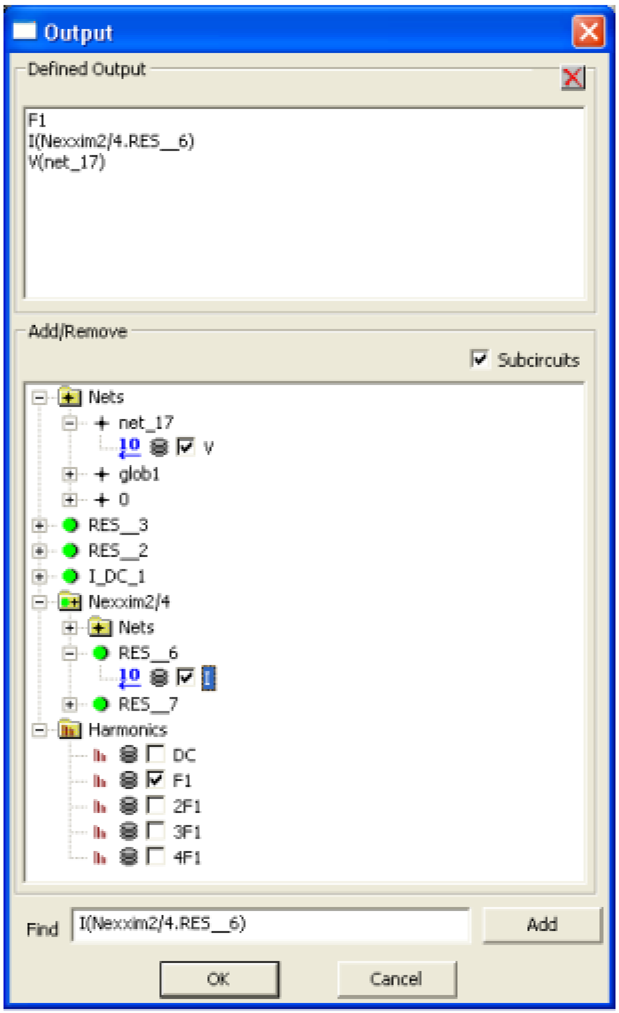
Schematic Design Output Control

In the **Schematic Editor**, Click the **Edit Output Quantities** feature in the **Solution Setup** window to specify particular outputs. Ports and probes in the schematic are automatically added as outputs

> **Warning:**
>
> By default, Nexxim stores all intermediate values for all nodes. If you do not specify any output quantities at setup, all node voltages and branch currents are retained so they can be output to the results file. With large circuits or analyses, the amount of data to be stored can cause the simulation to slow down, or even to exceed memory limits (usually a fatal error for Ansys Electronics Desktop/Nexxim). The Edit Output Quantities window specifies which nodes/values can be retained, allowing Nexxim to discard all other values when they are no longer needed. Use of the window can significantly reduce memory usage. Although the range of output values may be limited, in those cases where memory is an issue, use of the Edit Output Quantities facility may be required to allow the computation to complete successfully.

Edit Output Quantities

1. In the **Circuit** drop-down, select **Add Solution Setup**.
2. In the **Solution Setup** window, select the type of analysis to be run. Click **Next** to open a setup window for the selected analysis.
3. In the window, click **Edit Quantities** in the **Output Quantities** group box. (This facility is not available for Linear Network Analysis).
4. The **Output** window opens:



For all analysis types, the **Output** window identifies the nets available for voltage outputs. After running an analysis, voltage values are available for these nodes. Expand the **Nets** icon to see the individual nodes. Expand each node to select its voltage as an output.

For all analysis types, the **Output** window identifies devices available for current outputs. After running an analysis, current values are available for all nodes of the requested devices. For example, if you request the current through a resistor **R1**, the current outputs includes the values **Iterminals_0(R1)** and **Iterminals_1(R1)**.

For Harmonic Balance, Time-Varying Noise, and Oscillator analyses, the **Output** window identifies harmonics available for output analysis.

If the circuit contains a subcircuit, the **Output** window displays its nets and devices for selection as outputs when you expand the icon for the subcircuit.

5. Click its check box to select an output quantity. The selected quantities are listed in the **Defined Output** field.
6. Click **OK** to close the **Output** window and return to the window for the selected analysis. The output quantities you selected are listed in the **Output Quantities** group box of the analysis window. Complete the solution setup as described in the topics for the individual analyses.

Ports in Schematics

In schematic circuit designs, add hierarchical ports to nodes that represent the circuit outputs. The nodes with attached ports are automatically added to the list of output quantities available for reports of simulation results.

Probes in Schematics

In schematic circuit designs, add voltage and current probes to nodes whose values are of interest. The nodes with attached probes are automatically added to the list of output quantities available for reports of simulation results.

.MEASURE Quantities in Schematics

A Nexxim netlist may set up a MEASURE variable to contain a value calculated during simulation. After simulation has run successfully, all variables defined in .MEASURE statements are available in the **MeasureCategory** listing of the Report window. See .MEASURE Statements in the File Formats topic for further information.

Netlist Design Output Control

Several types of output controls are available in Nexxim netlists.

Netlist PRINT Statements

A netlist can use one or more **.PRINT** statements to specify the outputs from particular types of analysis.

> **Warning:**
>
> By default, Nexxim stores all intermediate values for all nodes. If you do not use a .PRINT statement, all node voltages and branch currents are retained so they can be output to the results file. With large circuits or analyses, the amount of data to be stored can cause the simulation to slow down, or even to exceed memory limits (usually a fatal error for Ansys Electronics Desktop/Nexxim). .PRINT statements specify which nodes/values are to be retained, allowing Nexxim to discard all other values when they are no longer needed. Use of .PRINT statements can significantly reduce memory usage. Although the range of output values may be limited, in those cases where memory is an issue a .PRINT statement may be required to allow the computation to complete successfully.

> **Note:**
>
> .PROBE statements in imported netlists are internally converted to .PRINT statements.

There are three forms of the .PRINT statement, one for simple circuit values, one for expressions involving circuit values, and one for the output of values calculated internally by selected device models. The output specification for all three PRINT statement types can use wild card characters.

PRINT Statements with Simple Values

The netlist syntax for simple outputs is:

```
.PRINT [analysis] output1 [output2] ...
```

The following topics explain the analysis type and the output specifications.

Analysis Type

The *analysis* is one of:

dc for DC analysis

tran for transient analysis

hb for harmonic balance analysis

lna for linear network analysis

ac for AC analysis (when linear network analysis has been performed)

osc for oscillator analysis

tv_noise for periodic time-varying noise analysis

A **.PRINT** statement that does not have an explicit *analysis* is automatically supplied with one using the following rules:

- A **.PRINT** that occurs before ANY analysis is filled in as "**.PRINT dc**."
- Any other **.PRINT** statement is filled in with the name of the analysis specified most recently above it.
- All **.PRINT** statements that apply to a given analysis type are then applied to that analysis.

So, for example, a netlist that specifies a circuit and the following:

```
.print V(1)
.tran 1e-2 1
.print V(2)
.dc sweep R1 100 200 10
.print V(3)
.dc sweep R2 100 200 50
.print V(4)
```

is interpreted as

```
.print dc V(1)
.tran 1e-2 1
.print tran V(2)
.dc sweep R1 100 200 10
.print dc V(3)
.dc sweep R2 100 200 50
.print dc V(4)
```

and performs three analyses:

1. The transient analysis, in which only V(2) is saved.
2. A DC analysis sweeping the value of R1, in which only V(1), V(3), and V(4) are saved.
3. A DC analysis sweeping the value of R2, in which only V(1), V(3), and V(4) are saved.

Output Quantities

The output quantities can be node voltages or currents through individual elements. For a node voltage, the syntax is:

V(*n1***)**

The simulator outputs the voltage difference between the specified node name (*n1*) and ground.

The output can specify a differential voltage between nodes, using the syntax:

V(n1, n2**)**

The output is the differential voltage for the node pair V(n1) **-** V(n2). If n1 and n2 are the same node, the output evaluates to zero volts.

For a current through a branch containing any single device, the syntax is:

I(*device_name***)**

The output includes the current entering the device through each node.

The following **.PRINT** statement outputs the current through resistor R10 in a harmonic balance analysis:

```
.PRINT HB I(R10)
```

The outputs includes the current values **Iterminals_0(R1)** and **Iterminals_1(R1)** (See the following illustration).

The following statement outputs the current through voltage source VSRC in a transient analysis:

```
.PRINT TRAN I(VSRC)
```

The outputs includes the current values **Ipositive(VSRC)** and **Inegative(VSRC)** (See the following illustration).

To obtain voltage and current output variables within one or more subcircuit levels, use the following syntax:

V (*subcircuit1*. [*subcircuit2*.]...[*subcircuitn*.]*element*)

I (*subcircuit1*. [*subcircuit2*.]...[*subcircuitn*.]*element*)

For example, if subcircuit instance XSUB1 contains voltage source V6, the **.PRINT** statement to output the current through V6 in transient analysis is:

```
.PRINT TRAN I(XSUB1.V6)
```

If XSUB1 is an instance of a subcircuit whose definition includes a node N25, the **.PRINT** statement to output the voltage through node N25 in transient analysis is:

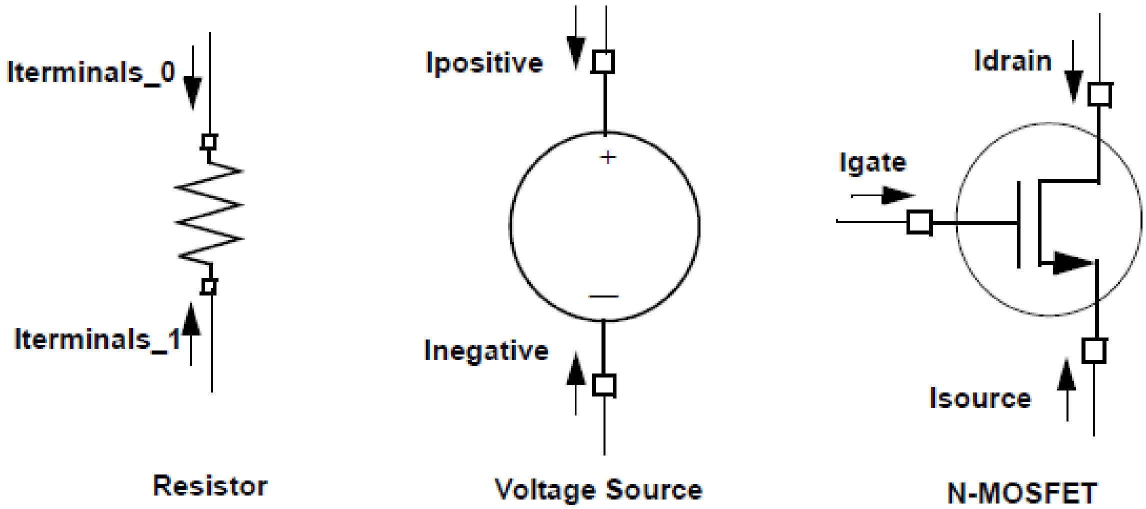
```
.PRINT TRAN V(XSUB1.N25)
```

If subcircuit instance XSUB1 contains a nested subcircuit instance XSUB2 which in turn includes NMOS MOSFET M23, the **.PRINT** statement to output the current through D23 is:

```
.PRINT TRAN I(XSUB1.XSUB2.M23)
```

The outputs includes the current values **Isource(M23)**, **Igate(M23)**, and **Idrain(M23)**.

The following diagram illustrates the directions of the current values for the resistor, voltage source, and MOSFET examples given earlier.



PRINT Statements with Expressions

Output can be specified with an expression, with or without a label. The syntax is a variation on the **.PRINT** statement:

```
.PRINT [analysis] [label=] PAR(expression)
```

The types for *analysis* are the ones given above for simple outputs (See "PRINT Statements with Simple Values " on page 25-242).

The *expression* can reference node voltages, branch currents, and parameter values. Nexxim stores the expression and the values in the results file, along with the *label* specified in the **.PRINT** statement.

Examples:

```
.PRINT PAR(V(1) + V(2))
.PARAM Y = 42
.PRINT X=PAR(Y + V(12)/V(13))
```

Calculated Model Outputs

Certain device models have been set up to provide calculated quantities as outputs. To enable the output quantities, the netlist uses a variation of the .PRINT statement:

```
.PRINTanalysisO(instance_name)
```

The *analysis* is one of the Nexxim analysis types listed above. The *instance_name* is the name of a device instance defined elsewhere in the netlist. The outputs are calculated values particular to the device model referenced by the instance statement.

Here is a netlist fragment:

```
.MODEL bsim_model NMOS LEVEL=53 // Device model
M101 1 2 3 bsim_model // Device instance

.TRAN 1us 100us // Analysis
.PRINT tran O(M101) // Outputs model internal values
// Rest of netlist is not shown in this example
```

The output quantities are listed in the documentation for the individual devices. Since the >PRINT statement references the instance name, the output quantities are documented under the instance statement for the device types. See the documentation for the *MOSFET instance, Level 49 or 53* for an example.

To open the calculated quantities in a Report window:

1. Select **Create Report** on the **Results** icon in the **Project Manager** window. Click **OK** to accept the defaults on the **Create Report** window and open the **Report** window.
2. Select **Device Properties** as the **Category** in the **Report** window. Select one or more of the outputs in the **Quantity** listing, along with the corresponding **Functions**. Click **Add Trace** and **Done** to close the **Report** window and display the outputs.

Wildcard Characters in Output Values

The output specification can contain the wildcard characters asterisk (*****) and question mark (**?**):

***** means “match zero or more characters.” For example “V(net*)” outputs the voltage of any node whose name begins with the characters **net**, such as **net**, **net55**, or **net_whatever**.

? means “match any single character.” For example “V(net?)” outputs the voltage of any node whose name consists of the characters **net** followed by one more character, such as **net1** or **netA**.

> **Note:**
>
> Instances of “V(*,*)” or “V(?,?) that produce “V(x,x),” the differential voltage between two identical nodes, are not normally included in the output.
>
> However, if a wildcard pattern of two nodes matches one or more “V(x,x)” outputs, just one of those matches is included in the result. This guarantees that a statement such as .PRINT PAR(‘V(x) + V(?,?)’) has a value when a valid result exists.

Output of Small-Signal AC Transfer Functions

When linear network analysis is being performed, the netlist can include a **.PRINT AC** statement. Nexxim LNA computes AC transfer functions for any outputs listed in the **.PRINT AC** statement. If the **.PRINT AC** statement is omitted, no AC transfer functions are output. See the Nexxim Linear Network Analysis help topic for further information.

Port Impedance Resistors in LNA

In Nexxim linear network analysis of netlist circuit designs, you add Port Impedance resistors to the netlist to specify outputs. See the Nexxim Linear Network Analysis help topic for details on the use of Port Impedance resistors to control simulation output.

> **Note:**
>
> Port impedance resistors are not required for a linear network analysis that consists only of Noise calculations or only of AC small-signal analyses.

Nexxim Command Line Controls

This topic describes how to control the Nexxim simulator on the operating system command line. Nexxim allows many options to be entered on the command line. All command-line options and arguments are case-insensitive.

> **Note:**
>
> On MS Windows[tm] systems, the command-line version of **nexxim.exe** is supported only for the Windows MS-DOS[tm] command prompt. Other shells such as Cygwin[tm] may not successfully run **nexxim.exe**.

Invoking the Nexxim Simulator

To run the Nexxim simulator, type the following command line at the prompt:

```
nexxim [-ooutput_file] [options] input_file
```

The input netlist file contains the circuit description, simulator controls, and options.

Valid formats and filename extensions for the input netlist file are:

- **.cir** —Nexxim internal netlist format
- **.sp** —HSPICE netlist format
- **.scs** —Spectre® netlist format.

See Netlist Format for a description of the Nexxim internal netlist syntax.

The **–o***output_file* entry outputs the quantities specified by the **.PRINT** statements in the input *netlist_file* to a file called *output_file***.sdf**, located by default in the simulator directory. To output the file under a different directory, include the directory path with the filename. If the **–o** entry is omitted, the simulation results are output to a file called *input_file***.sdf** in the directory containing the netlist file. The output file is written in the Nexxim SDF (Simulation Data File) format. See Viewing Results for information on viewing SDF-format files in **Electronics Desktop**.

Configuration File

Simulation options can be provided in a configuration file (*filename***.cfg**) specified in a **-config_file** option to the command line:

```
nexxim [-config_file=filename.cfg] [-ooutputfile] input_file
```

If no **config_file** option is provided, the simulator looks for a file named **cirsim.cfg** in the directory where **Electronics Desktop** is installed. If the configuration *filename* is not located in the install directory, its path must be provided with the filename. Within the configuration file, each option should be on a separate line, and can start with a prefix to identify the simulation tool or device category to which it applies:

option
Option for all analyses

dc.*option*
Option for dc analysis

tran.*option*
Option for transient analysis

hb.*option*
Option for harmonic balance analysis

lna.*option*
Option for linear network analysis

osc.*option*
Option for oscillator analysis

tvnoise.*option*
Option for time-varying noise analysis

s_element.*option*
Option for analysis of s-parameter element

When the same option has been set more than once, the following precedence is applied:

- Command line options > netlist options > default (configuration file) options.

See Invoking the Nexxim Simulator for details on the other command line entries. See the individual help topics for details on the options recognized by each of the simulation tools.

HPC Options

The Nexxim solver ignores any High Performance Computing (HPC) options in a netlist. Instead, they must be provided on the command line in this form:

```
nexxim input_file [num_threadsworkgroup] [-hpc_type#] [-use_gpu#] [-
enable_tpbs#]
```

HPC Options

Option	Description	Acceptable Values	Default
-enable_tpbs	Enables Transient Parallel Bus Speedup to speed up transient solutions for large non-linear designs	**0** = False **1** = True	1
-hpc_type	Determines which license to use.	**0** = One HPC workgroup license enables one unit of HPC. **1** = One HPC pack license enables eight units of HPC. Additional packs multiply by four, enabling 32, 128, 512, etc., in the context of a single simulation.	1
num_threads	Specifies the number of cores to use for the project. Place no dash before this option.	Integer	4
-use_gpu	Enables GPU to accelerate simulation and eye rendering.	**0** = False **1** = True	1

Log File

By default, error and status messages on the netlist parser and on the simulation tools are displayed on the standard output. Messages can be directed to a logfile specified in a **-logfile** option to the command line:

```
nexxim [-logfilemessagefile] [-ooutputfile] input_file
```

The **–logfile***messagefile* option outputs any messages into a file called *messagefile*.

Message Controls

By default, status, error, and warning messages from the netlist processor, the simulation tools, and the device models are output to standard out or to a specified logfile. You can control the type of error logging with the following options.

All Messages

The **–error.filter** option controls all messages:

`nexxim -error.filter=`*`level`* `[`**`-o`***`outputfile`*`]` *`input_file`*

The *level*is an integer characterizing errors based on their severity:

0 Trace messages

1 Status

2 Info

3 Warnings

4 Errors

The option **–error.filter=***level* allows errors with severity *level* or higher to go through and be posted. For example, **-error.filter=3** allows warnings and errors to be displayed, suppressing all trace, status, and info messages.

> **Note:** Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

Messages from Analysis Tools

The option **-error.filter.analysis** allows messages from all analysis tools with severity *level* or higher to go through:

`nexxim -error.filter.analysis=`*`level`* `[`**`-o`***`outputfile`*`]` *`input_file`***`.cir`**

For example, **–error.filter.analysis=0** provides screen output on the convergence of the DC and transient simulations.

You can specify a filter for messages from individual analysis tools:

`-error.filter.analysis:dc=`*`level`*

Allows DC analysis errors with severity *level* or higher to go through.

-error.filter.analysis:tran=*level*

Allows transient analysis errors with severity *level* or higher to go through.

-error.filter.analysis:hb=*level*

Allows harmonic balance analysis errors with severity *level* or higher to go through.

Messages from Device Models

The option **-error.filter.models=***level* controls error and warning messages on the device models. More specifically:

-error.filter.models=*level*

Allows all model errors with severity *level* or higher to go through.

-error.filter.models:*modelname=level*

Allows errors related to the model called *modelname* and with severity *level* or higher to go through. For example:

-error.filter.models:resistor=0

Device Listing

The **–print_inventory** option generates a listing of device types and instance count:

nexxim -print_inventory=1 [**-o***outputfile*] *input_file*

Here is an example listing:

```
device type_______________number of devices
-------------------------------------------
s_element_________________1
port_impedance____________2
s_element_core_fdomain____1
```

Viewing Results in the GUI

The result of a Nexxim simulation is written out to a file in Simulation Data File (SDF) format, an Ansys-proprietary binary format. To view the results of a simulation that has been run on the command line, start **Electronics Desktop**.

1. From the **File** menu, select **Open** to open an explorer window.
2. In the **Look in** field, browse to the directory containing the SDF file you want to display.

3. In the **Files of type** drop-down, select **Ansoft Nexxim Solution (*.sdf)** or **All Ansoft Nexxim Files (*.adsn, *.cir, *.sdf, *.sp)**.
4. Select the SDF file you want to display, then click **Open**.
5. A Project icon appears in the Projects window of **Electronics Desktop**. Click the + sign for the project, then on the + sign for the circuit. A **Results** icon appears. Right-click on the **Results** icon, and select one of the Report types to open the **Report** window. Select the Y-data on the group box on the right, then click **New Report** to open the graphic display of the trace.
6. Repeat steps 5 through 8 as often as appropriate to display all the simulation data in the results file.
7. From the **File** menu, select **Save as**, then save your settings as a **Electronics Desktop .aedt** file.

Viewing the Nexxim Solution Log File

After a simulation has completed, Nexxim prints messages about the success or failure of the analysis to the Ansys Electronics Desktop Message Manager window.

Nexxim writes details regarding the simulation to a solution log file. The log file information may be useful for diagnosing why a particular analysis failed.

- From a schematic design, right-click the analysis setup under the Analysis icon in the **Project Manager** window and select **Browse Solution Log File**.
- From a netlist design, right-click the circuit icon in the **Project Manager** window and select **Browse Solution Log File**.

The log file opens in a standard Notepad window.

DC Analysis Technical Notes

The following topics provide technical details on Nexxim Direct Current (DC) analysis.

Nexxim DC Analysis Overall Flow

Figure 1 shows the sequence of operations performed in Nexxim DC analysis.

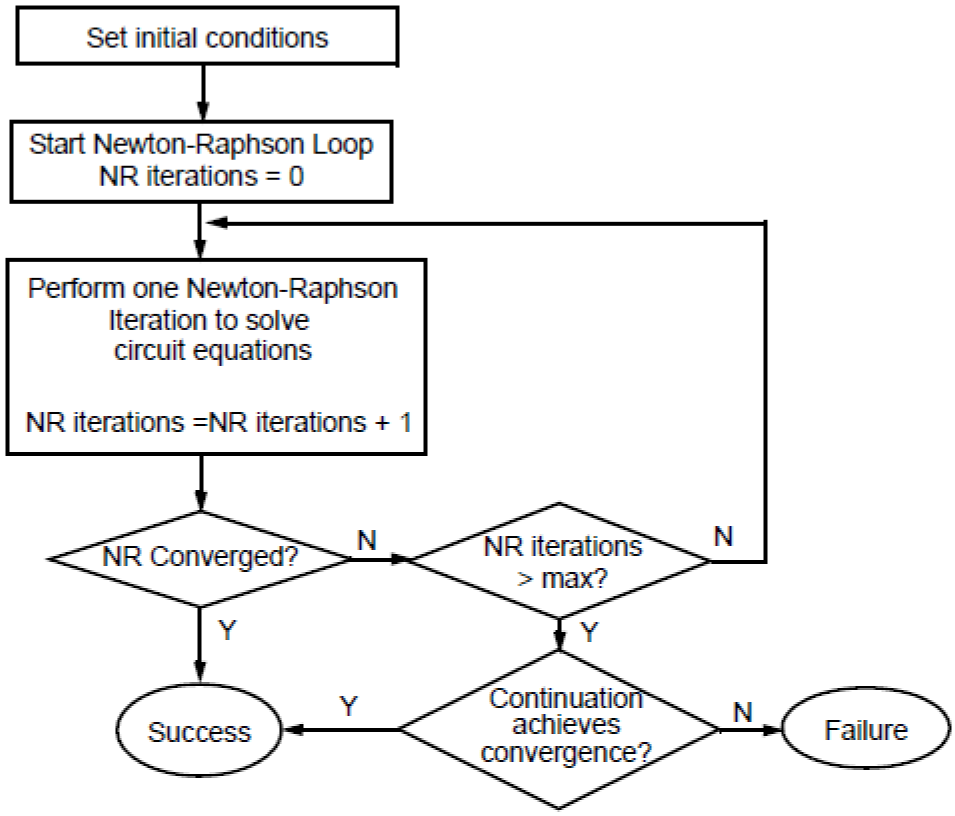


Figure 1. DC Analysis—Overall Flow

Initialization of Nexxim DC Circuit Parameters

DC analysis begins by initializing the circuit to a known state.

- Capacitors are modeled as open circuits.
- Inductors are modeled as short circuits.
- Voltage sources are set to zero and held constant (dv(t)/dt = 0).
- Current sources are set to zero and held constant (di(t)/dt = 0)
- Initial voltages from .NODESET are applied to the specified nodes. See NODESET Statements.
- Non-floating nodes are set to zero volts for the first Newton-Raphson iteration.

- Floating nodes that are not specified with .NODESET are undefined.

> **Note:**
>
> Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

- Conductances with value **alpha** (G_{MIN}) are added between all branches.
- Conductances with value **beta** (G_{SHUNT}) are added between each node and the ground reference node (node 0).
- The nominal circuit temperature is set to **TEMP**.

NODESET Statements

The netlist can include one or more .NODESET statements. Voltages assigned with .NODESET are applied at the start of the DC operating point analysis, to provide the initial values for the first Newton-Raphson iteration. Any node voltages not specified with NODESET are initialized to zero.

The syntax is:

.NODESET V(*node1***)=***value1* [**V(***node2***)=***value2* ...]

For example, to initialize node 12 to 4.5 volts and node in2 to 2.23 volts:

```
.NODESET V(12)=4.5 V(in2)=2.23
```

IC Statements

The netlist can include one or more .IC statements to force nodes to particular voltages. Voltages set with .IC are applied throughout the DC operating point analysis. Any node voltages not specified with .IC are computed.

The syntax is:

.IC V(*node1***)=***value1* [**V(***node2***)=***value2* ...]

For example, to force node 12 to 4.5 volts and node in2 to 2.23 volts:

```
.IC V(12)=4.5 V(in2)=2.23
```

> **Note:**
>
> Nexxim recognizes .IC statements during .OP analysis. Other simulators may not.

Initialization of Nexxim DC Simulation Parameters

The DC simulator starts with a set of simulation parameters at initial values:

- The maximum number of Newton-Raphson iterations is set to **max_newton_iterations**, and the maximum voltage change between NR iterations is set to **limiting_step**.
- Tolerance values are set to the values of **abstol** (absolute current tolerance in amperes), **vntol** (absolute voltage tolerance in volts), and **reltol** (relative current and voltage tolerance, dimensionless), as appropriate.

DC Diagnostics

- If DC analysis fails, Nexxim checks the solution and finds the component that violates tolerance. The device information corresponding to the problematic node or branch is printed in the log file.
- DC analysis starts with DC continuation. If DC continuation fails, diagnostic information of Device continuation is provided.

Solving the Circuit Equations in Nexxim DC Analysis

In general, the equations for a system with a large number of nonlinear elements cannot be solved directly. The DC solver uses the Newton-Raphson (NR) method, a well-known iterative method.

The Newton-Raphson loop begins with vectors of initial values for the node voltages and branch currents. In each NR iteration, the simulator linearizes the circuit around the solution calculated in the previous iteration, and calculates the Jacobian matrix associated with the linearization point. The linear circuit models are represented in matrix/vector form. The simulator then uses a matrix solver to calculate a new vector of unknowns. At the end of each iteration, the Newton-Raphson method updates the solution and checks to see if the new solution satisfies the specified tolerances. If the result is within tolerance, the loop terminates. Otherwise, the loop uses the new result to start the next iteration.

Convergence is a critical issue with the Newton-Raphson method. Convergence to a correct solution is guaranteed by NR only when the initial starting point is sufficiently close to the final solution. Initial voltages set with .NODESET can help, but in many cases the DC solver starts with zero values, which may be too far on the correct values. Thus, convergence is often a serious problem that causes the DC solver to fail, which in turn prevents any further analysis such as transient analysis to be performed. In the Nexxim circuit simulator, the basic Newton-

Raphson method is combined with sophisticated continuation schemes to ensure robust convergence without loss of simulation speed and accuracy.

Linearization of Nonlinear Device Equations

The current through a device is typically expressed as a function of the voltages at its nodes. Linear device currents are linear functions of node voltages. For example, the current through a resistor is described by:

$$f(v) = i = \frac{v_2 - v_1}{r}$$

For nonlinear devices, the relationships between current and voltage are expressed by one or more nonlinear functions of voltage. The Newton-Raphson method linearizes each nonlinear function *f*(*v*) using the following iterative matrix/vector form, where on the *k*th iteration:

$$\Delta v_k = v_{k+1} - v_k = \frac{-f(v_k)}{\partial f(v_k)/\partial v_k}$$

Newton-Raphson Initial Values

The Newton-Raphson method starts with initial guesses for each of the unknown voltages. Some voltages are preset by .NODESETs and by the input voltages and currents. Voltages for connected nodes that are not preset default to zero.

Circuit Matrix Solver

On each NR iteration, the matrix solver computes the linearized solution using the formula:

$$\Delta v_k \times \frac{\partial f(v_k)}{\partial v_k} = -f(v_k)$$

and updates the solution at the end of each NR iteration with:

$$v_{k+1} = v_k + \Delta v_k$$

The functions and derivative formulas are specific to the device models. The simulator uses either a direct solver or a Krylov subspace solver.

Nexxim DC Analysis Convergence Criteria

After each Newton-Raphson iteration, the simulator uses two criteria, the delta check and the function check, to determine whether the solution meets required tolerances. The delta check guarantees that the solution converges in the voltage tolerances set by options **reltol** (relative voltage and current tolerance) and **vntol** (absolute voltage tolerance). The function check guarantees that the solution satisfies the nonlinear equations in the current tolerances set by options **reltol** and **abstol** (absolute current tolerance).

Solving nonlinear circuits may also require limiting the size of voltage changes at each Newton-Raphson iteration.

Delta Check

The delta check is also known as the update criterion. As the solution of the Newton-Raphson solver gets close to the correct values, the norm of the voltage vector Δv_k should approach zero. The norm of Δv_k is the square root of the sum of the squares of the vector elements, represented by $||\Delta v_k||$. The delta check is satisfied when:

$$\left\|\Delta v_k\right\| < \left|(reltol \times \left|\Delta v_k\right| + vntol)\right|$$

The default value for **reltol** is 0.001 (dimensionless). The default for **vntol** is 50μV (50e-6 Volts).

Function Check

$$F(v_k) = B$$

The norm of the residue of this equation is:

$$\|r\| = \|F(v_k) - B\|$$

The norm of the residue should approach zero as the solution converges. The function check ensures that NR has converged to a correct solution, not just to a local minimum. The function check is satisfied when:

$$\|r\| < \left|(reltol \times \left|F(v_k)\right| + abstol)\right|$$

The default value for **reltol** is 0.001 (dimensionless). The default for **abstol** is 1nA (1e-9 Amperes).

Limiting Step and Limiting Method

Nexxim automatically identifies circuits that contain nonlinear elements. For large circuits with many nonlinear devices such as BJTs and MOSFETs, it is necessary to reduce the magnitude of the Δv_k update to aid in obtaining convergence. The DC solver limits the size of voltage updates to a maximum value given by the **limiting_step** option. The default **limiting_step** for circuits with nonlinear devices is 0.3 Volts.

For circuits that contain only linear devices, Nexxim sets the default for **limiting_step** to a large number, effectively disabling it. A large **limiting_step** speeds the convergence of circuits that are mostly linear, while a smaller value may help nonlinear circuits to converge more reliably.

The default method, **Standard Limiting**, reduces the magnitude of the Δv_k update that corresponds to all nonlinear elements and thus slows down convergence. The **Modified Limiting** method aims to avoid unnecessary limiting of moderately nonlinear elements, thus improving the DC convergence rate.

Nexxim DC Continuation Strategies

The Newton-Raphson method guarantees convergence only when its initial guess is close enough to the final solution. Without specific information about the circuit, the DC solver starts from an initial guess of zero. For many nonlinear circuits, zero is too far from the final solution, and Newton-Raphson fails to converge. In these cases, continuation methods can be applied to achieve convergence.

Nexxim uses a fixed sequence of continuation strategies to help achieve convergence. Each strategy uses a set of optimal parameters and a combination of several basic methods—device continuation, alpha (G_{min}) continuation, beta (G_{shunt}) continuation, and pseudotransient continuation. The **conv** option selects the initial combination of methods to use (See DC Analysis Options for details). By default, when one strategy fails to converge, the simulator cycles through the available combinations, and analysis fails only after all continuation strategies fail. You can deactivate the automatic cycling, in which case analysis fails when the first combination of methods fails to converge.

Alpha Continuation

Alpha continuation (also called G_{min} stepping) improves the robustness of the DC simulator. Some circuits may have floating nodes due to open devices. A floating node does not have a unique voltage solution, so the circuit is a singular system that does not have a unique solution. Alpha continuation is one way to reach a solution that is unique and meaningful.

For the alpha continuation method, the simulator inserts a small conductance G_{min} (equivalent to a large resistor) in parallel with every branch. The value of the conductances is equal to the value of the **alpha** option, which defaults to 1e-13 mho. This conductance is small enough that the effect of the conductances on the circuit is typically negligible.

If the Newton-Raphson method fails to converge with the initial small conductance value, alpha stepping attempts to reach convergence by setting the conductances to a relatively large value, so the circuit is affected by the conductances and becomes more “linear.” The solver then attempts to solve this more “linear” circuit. The solution of the modified circuit is not the same as that of the original. However, if the modified circuit converges, the solution should be close to the solution of the original circuit. The solver uses the solution for the modified circuit as the input to the next NR loop, and decreases the value of the alpha conductances to decrease their effect on the circuit.

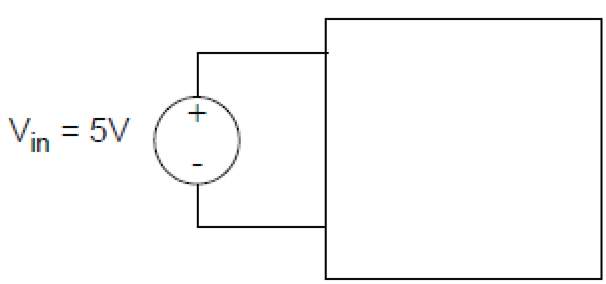
Alpha stepping progresses through smaller and smaller conductance values until convergence is achieved with the minimum value, unless an intermediate value of conductance fails to achieve convergence. The solution is accepted when the conductance equals the **alpha** option, which defaults to 1e-13 mho. If the behavior of the circuit is strongly affected by the G_{min} conductances, reduce **alpha** or even set it to zero.

Device Continuation

Device continuation (also called source stepping) attempts to reach convergence by initializing all voltage and current sources to values close to zero. In this condition, the initial guess of zero for all circuit values is a “correct” solution. Device continuation then attempts to reach convergence on the true solution by solving a sequence of circuits with progressively increasing source values.

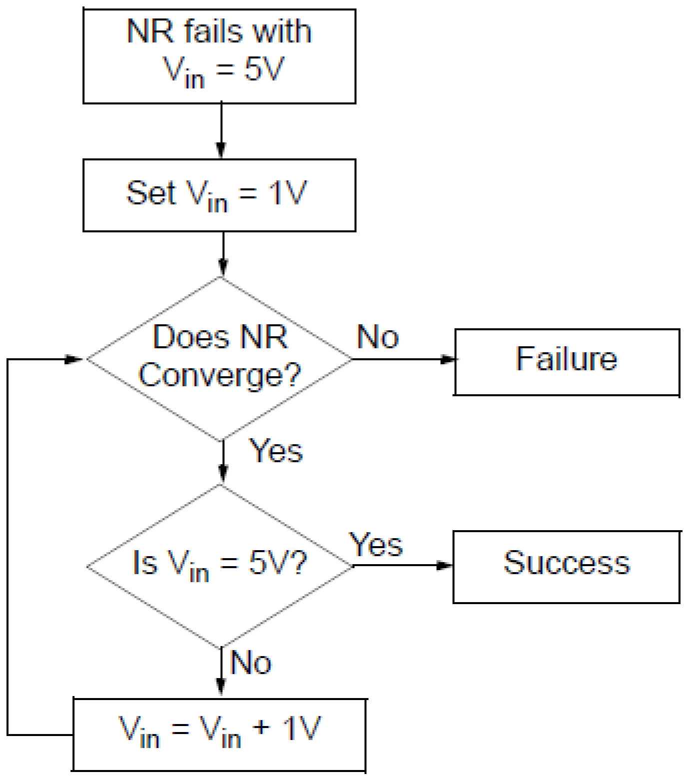
For example, Figure 2 diagrams a circuit block diagram excited by a single 5V source, V_{in}.

Figure 2. Device Continuation Example



If NR fails to converge when V_{in} = 5V, source stepping may begin by trying to solve the circuit with V_{in} set to 1V. If that attempt converges, the solver uses that solution as the initial guess to a new circuit with V_{in} set to 2V, and tries NR again. This process repeats with V_{in} = 3V, V_{in} = 4V, and finally V_{in} = 5V. If each step converges, the fifth step with V_{in} = 5V is the correct solution and source stepping succeeds. If any step fails to converge, the device continuation fails. The device continuation algorithm for this example is illustrated in Figure 3.

Figure 3. Device Continuation Algorithm for Example in Figure 2



Beta Continuation

Beta continuation (also called G_{shunt} stepping) is similar to alpha continuation. With beta continuation, a small conductance (large resistor) G_{shunt} is added between each node and ground. The conductance is equal to the value of the **beta** option, which defaults to 1e-12 mho. If Newton-Raphson fails with the initial small beta conductances, the DC solver attempts to continue by setting the conductances to a relatively large value and executing Newton-Raphson on this more “linear” circuit. If this step succeeds, the solution to the modified solution is used to start the next NR loop with a decreased value for the conductances. Stepping continues until convergence is attained with the conductance equal to the **beta** option, which defaults to 1e-12 mho.

Pseudotransient Continuation

Direct DC analysis sometimes fails to solve a circuit. For such circuits, transient analysis can be employed. The transient problem is solved over a long enough time to reach the DC steady state. This continuation method is called pseudotransient continuation.

Convergence, Speed, and Accuracy

DC operating point simulation provides the large-signal bias point that is important for small-signal analyses such as AC analysis, transfer function analysis, noise analysis, and linear network analysis. These small-signal simulations cannot be performed without a DC solution. Because the duration of the DC analysis is added to the small-signal simulations, it is important to achieve fast DC convergence with high accuracy.

The DC analysis must of course converge in order to obtain a usable result. Once convergence is achieved, the trade-off between solution accuracy and convergence speed can be considered. The options in the following topics affect the convergence, accuracy, or speed.

Options that Affect Nexxim DC Convergence Only

The options in the following categories control the ability of the simulator to achieve convergence.

DC Convergence Algorithms

The option **conv** sets the DC convergence algorithm.

- **conv=1** is the default algorithm, a combination of beta and device continuation which is both aggressive and robust.
- **conv=2** uses only device continuation, and is usually not as robust as **conv=1**.
- **conv=3** uses the more robust pseudotransient algorithm, but is not as fast as **conv=1** when the circuit is linear.
- **conv=4** is a slight modification of the **conv=1** beta and device continuation algorithm.
- **conv=5** combines alpha, beta, and device continuation.

When **conv=3** (pseudotransient continuation) has been specified, the option **max_pt_continuation_iterations** controls the maximum iterations to use (default is 5000 iterations).

The Boolean option **autoconverge** enables and disables automatic cycling through the convergence strategies.

- **autoconverge=1** is the default. Suppose **conv=1** is the current selection. If DC analysis fails with **conv=1**, the simulator automatically tries **conv=2**, **3**, **4**, and **5** until convergence is achieved. If **conv** has been set to a value other than 1, autoconvergence cycles from that method through the remaining methods. For example, if **conv=3** has been selected then fails, autoconvergence cycles through **conv=4**, **5**, **1**, and **2** until convergence is achieved. If no method achieves convergence, analysis halts.

- **autoconverge=0** disables autoconvergence cycling. If the chosen strategy fails, analysis halts.

Initial Guess in Sweeps

By default, DC sweeps use the result on the previous sweep step as the initial guess for the current step. The option **result_as_initial_guess** controls this behavior.

- **result_as_initial_guess=0** turns off the use of previous results.
- **result_as_initial_guess=1** enables the use of previous results. However, the use of previous results is the default, so this option setting is never needed.

Maximum NR Iterations

The option **max_newton_iterations** determines how many iterations is attempted before the Newton-Raphson loop exits without achieving convergence. The default is 200 iterations. Increasing the number of iterations allows the DC simulator to try more iterations per continuation step. As a result, each continuation step takes a longer time. However, this may help large nonlinear circuits to converge with fewer continuation steps.

NR Limiting Step

The option **limiting_step** sets the maximum size for changes in node voltages at each step in the Newton-Raphson loop. The default is 0.3 Volts. Use of a limiting step is referred to as “damped Newton-Raphson.” This option trades off the ability to converge and the speed of the simulation. Making the limiting step smaller improves the chance of convergence, but usually increases the number of NR iterations needed to reach convergence. For circuits with nearly linear functions, increasing the limiting step can speed up the simulation.

Maximum Continuation Iterations

The option **max_continuation_iterations** determines the maximum number of incremental steps to be used for G_{min}, G_{shunt}, and source stepping. This option thus takes effect only when the circuit has failed to converge normally. Increasing the maximum number of continuation iterations might improve the chance that the circuit converges. However, the default number, 500, should be large enough for all cases. If the circuit does not converge with the default continuation parameters, increasing **max_continuation_iterations** is unlikely to succeed.

Options that Affect Convergence, Speed, and Accuracy

The options in the next categories trade off the guarantee of convergence, the speed of the simulation, and the accuracy of the solution.

Relative Voltage and Current Tolerance

The **reltol** option controls the relative tolerance used for both the delta voltage check and the current function check (See "Nexxim DC Analysis Convergence Criteria" on page 25-257). The default **reltol** is 0.001 (0.1%).

Both the delta check and the function check have two parts, a relative part and an absolute part. The relative part is a function of **reltol** and some reference value, typically calculated voltage or current. When the reference values are large, the relative part of the criterion is much larger than the absolute part, and **reltol** in effect controls the convergence. In these cases, making **reltol** larger (“loosening" **reltol**) allows the circuit to converge in fewer iterations (and shorter simulation time), but with reduced accuracy.

Conversely, decreasing (or “tightening”) **reltol** increases the accuracy of the result, but may require more Newton-Raphson iterations to reach convergence. Thus, if **reltol** is tightened, the **max_newton_iterations** option may need to be increased.

Absolute Voltage Tolerance

The **vntol** option controls the absolute part of the delta voltage convergence criterion. The default is 50e-6 Volts or 50µV. When the reference values are close to zero volts, the absolute part of the criterion controls the convergence. **vntol** should be a value that is reasonable given the smallest voltage that is expected. One possibility is to set **vntol** equal to the smallest expected voltage times **reltol**.

Like **reltol**, tightening **vntol** increases accuracy but may require more NR iterations.

Absolute Current Tolerance

The **abstol** option controls the absolute part of the current function convergence criterion. The default is 1e-9 Amps or 1nA. When the current values in the circuit are close to zero amperes, the absolute part of the criterion controls the convergence. **abstol** should be a value that is reasonable given the smallest current that is expected. One possibility is to set **abstol** equal to the smallest expected current times **reltol**.

Like **reltol**, tightening **abstol** increases accuracy but may require more NR iterations.

Alpha Initial Value

The **alpha** option sets the initial value for the G_{min} conductances inserted across all branches in the circuit. The default value is 1e-13 mho (0.1 picoSiemens). The default value is usually too small to affect the operation of most circuits. The user may reduce G_{min} or even set it to zero to avoid affecting the circuit. Changing alpha can affect the convergence.

Beta Initial Value

The **beta** option sets the initial value for the G_{shunt} conductances inserted from all nodes to ground. The default value is 1e-12 mho (1 picoSiemens). The default value is normally small enough, but the user may change it. Changing beta can affect the convergence.

Summary of Effects of DC Options

The following table summarizes the options that affect the convergence, speed, and accuracy of the DC operating point simulation.

Effect of Options on DC Operating Point Simulation

Simulation Result	Effect of Option	
	Improve	**Worsen**
Convergence	Increase **max_newton_iterations**	Decrease **max_newton_iterations**
	Decrease **limiting_step**	Increase **limiting_step**
	Increase **max_continuation_ iterations**	Decrease **max_continuation_ iterations**
	Increase **reltol**, **vntol**, **abstol**	Decrease **reltol**, **vntol**, **abstol**
	Increase **alpha**, **beta**	Decrease **alpha**, **beta**
	Set **conv** =3, 4, or 5	Set **conv** =2
		Set **autoconverge** =0
Speed	Increase **limiting_step**	Decrease **limiting_step**
	Increase **reltol**, **vntol**, **abstol**	Decrease **reltol**, **vntol**, **abstol**
	Set **conv** =2	Set **conv** =3, 4, or 5
Accuracy	Decrease **reltol**, **vntol**, **abstol**	Increase **reltol**, **vntol**, **abstol**
	Decrease **alpha**, **beta**	Increase **alpha**, **beta**

Transient Analysis Technical Notes

The following topics provide technical information on Nexxim transient analysis.

Initial Conditions for Transient Analysis

Transient analysis obtains a set of initial conditions (at time = 0) by performing a DC operating point analysis as described in Nexxim DC Analysis. As in DC analysis, .NODESET and .IC statements may be used to affect the computation of initial conditions.

Any options specified for DC analysis affects the DC calculation done to obtain the initial conditions for transient analysis. See DC Analysis Options.

Some options (in particular, **abstol** and **reltol**) can affect both DC analysis and transient analysis. See "Circuit Transient Analysis Options" on page 11-32 for further information.

Floating Node Warning

Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

Numerical Integration Methods

Transient analysis computes the time-domain response of a circuit by numerically integrating a system of differential/algebraic equations derived on the circuit topology and the circuit device models. Transient analysis uses Newton-Raphson iteration with the following numerical integration methods to solve the circuit equations at each discretized timestep.

Backward Euler

The Nexxim transient analysis engine always uses the backward Euler method when automatically starting or restarting integration.

Trapezoidal Rule

The trapezoidal rule is the Nexxim circuit simulator's default method for transient analysis.

The trapezoidal rule is an accurate integration method that does not produce numerical damping. It can produce "trapezoidal ringing," an oscillation in calculated values that is not the true circuit behavior. If ringing is observed, switch to BDF2 or NDF2.

See Transient Analysis Method Option RF in the Nexxim Design Examples help topic for an example of trapezoidal ringing and the use of the NDF2 integration method.

Second Order Backward Difference Formula

The Second Order Backward Difference Formula or BDF2 is also called the Gear method.

BDF2 is not quite as accurate as the trapezoidal rule, but is an A-stable numerical method. This means that BDF2 does not produce ringing, but instead produces numerical damping (the calculated circuit output eventually decays to zero although the true behavior does not). If BDF2 exhibits severe damping, use TRAP or NDF2.

Numerical Differentiation Formula

The Nexxim simulator offers a proprietary variable-step integration method called the Numerical Differentiation Formula (NDF2). NDF2 offers nearly the accuracy of the trapezoidal method. It creates less numerical damping than BDF2, but like BDF2 it is an A-stable method.

Selecting the Integration Method

The integration method can be specified with the .OPTIONS statement.

.OPTIONS method=*selection*

The default is **TRAP** (the Trapezoidal rule). You can also specify **GEAR** (**BDF2**) or **NDF2**.

Newton-Raphson Iterations

At each timestep, the simulator solves the discrete time approximation equations using the Newton-Raphson (NR) iterative method. Because the solution on the previous timestep is close to the solution for the current step, the number of Newton-Raphson iterations required to reach convergence at each time point is typically much less than the number of iterations needed for the initial DC operating point calculations.

You can set the maximum number of iterations to use with the **maxiters** option on the .OPTIONS statement. The default for transient analysis is 10 iterations.

The option **limiting_step** sets the maximum size for changes in node voltages at each step in the Newton-Raphson loop. Nexxim automatically identifies circuits that contain nonlinear elements. For large circuits with many nonlinear devices such as BJTs and MOSFETs, it is necessary to reduce the magnitude of the Δv_k update to aid in obtaining convergence. The Transient solver limits the size of voltage updates to a maximum value given by the **limiting_step** option. The default **limiting_step** for circuits with nonlinear devices is 0.3 Volts.

For circuits that contain only linear devices, Nexxim sets the default for **limiting_step** to a large number, effectively disabling it. A large **limiting_step** speeds the convergence of circuits that are mostly linear, while a smaller value may help nonlinear circuits to converge more reliably.

The maximum change in voltage per timestep is given by **maxiters** times **limiting_step**. In general, it is safer to change **maxiters** to adjust the size of the allowed voltage change per timestep, rather than changing the **limiting_step**.

The **update_jacobian_period** option can affect the overall speed of the Newton-Raphson process. By default, the Jacobian matrix is updated after every third iteration. Increasing the number of NR iterations per Jacobian update can speed up the simulation, but is risky when the circuit contains nonlinear elements. Reducing the **update_jacobian_period** option to 2 or 1 slows the simulation but may be necessary for highly nonlinear circuits. The Jacobian update period is sometimes called the Šamanskii step size.

Timestep Control

Transient analysis (including the transient analysis used to calculate the step response for QuickEye and VerifEye) uses a variable timestep that depends on several factors:

- Failure of the Newton-Raphson iterations to converge. If the Newton-Raphson iterations on any timestep fail to converge, the simulator decreases the size of the timestep and tries again.
- The Local Truncation Error and the Local Error. If the Local Error (which is calculated based on the Local Truncation Error) on any timestep is too large, the simulator decreases the size of the timestep and tries again.
- The **rmax** and **delmax** options, which control the maximum timestep size. Options **maxstep** (**eye.maxstep** in QuickEye and VerifEye options) and **tmax** are equivalent to **delmax** and can be used interchangeably. A two-stage timestep control algorithm is used to control the step size. The first stage is controlled by **rmax** (**eye.rmax** in QuickEye and VerifEye options). The maximum step size for the first stage is calculated by multiplying the value of the **rmax** option times the *tstep* argument from the .**TRAN** statement. The second stage limit uses the value of **delmax** and incorporates circuit information such as input frequencies. **delmax** should be greater than or equal to **rmax** times the *tstep*. The step size control algorithm behaves more liberally in the first stage, where the actual step size is smaller than **rmax** times the *tstep*, and more conservatively in the second stage, where the step size is larger than **rmax** times the *tstep* yet smaller than **delmax**. The two-stage algorithm allows a good trade-off between simulation time and accuracy.
- The two-stage timestep control algorithm increases/decreases the timestep based on the Local Truncation Error (LTE) and the number of Newton iterations (an indicator of nonlinearity). A small/large LTE leads to increase/decrease the timestep. **max_stretch** and **max_damp** determine the weight of nonlinearity-based factor of timestep control. Nonzero **max_stretch** is expected to reduce simulation time for circuits of weak/moderate nonlinearity. Nonzero **max_damp** increase the change of convergence for circuits of strong nonlinearity. By default, time stepping adaptivity will be triggered, that is a nonzero **max_stretch** will be generated automatically, if time step size is too small.
- The **trtol** option, which controls the relative solution tolerance. The **trtol** option specifies the (dimensionless) ratio between the tolerance of the overall transient analysis solution and the tolerance used in the Newton-Raphson iterations. The default of seven (7) means that tolerance used for the transient analysis solution is seven times looser than the tolerance used for the Newton-Raphson iterations. Increasing **trtol** has the effect of decreasing solution accuracy and increasing the timesteps.

> **Note:**
>
> Nexxim does not use the HSPICE “NPDELAY” parameter when it is present in imported netlists. The Nexxim timestep methodology is more advanced than the HSPICE NPDELAY method.

Controlling Output from Transient Analysis

By default, the output shows all node voltage values at the timesteps produced by the numerical integration. From the Schematic Editor, limit the output to specific variables with the Output Quantities window. From the netlist, specify a list of particular values with a .PRINT statement. For example, if the netlist input contains the following statements:

```
.TRAN 10ns 1us
.PRINT TRAN V(12), V(15)
```

Nexxim outputs just the voltages on nodes 12 and 15 at the internal timesteps. Without the .PRINT statement, the output includes voltages at all nodes in the circuit.

Nexxim performs quadratic interpolation between time steps. Unlike some Spice simulators, the output is not automatically interpolated to (uniform) TSTEP increments.

Stable Undershoot and Numerical Beating

When a periodic signal such as a sinusoidal or piecewise linear waveform is analyzed, the timesteps are selected to give an accurate representation of the periodic signal. Chance differences in the location of timepoints can lead to apparent errors when the wave is plotted.

Undershoot. In some cases, no timestep falls at the exact high or low point of the waveform. When this happens, interpolation may cause the maximum and minimum values as plotted to appear less than the known high and low values. Quadratic interpolation used by Nexxim helps with this “stable undershoot” problem, but cannot fully resolve it.

Beating. Nexxim tries to set constant time points to sample at the same places in each period of the waveform. When the period of the signal is not an integral multiple of the timestep, the apparent frequency implied by the timesteps can be slightly different on the frequency of the signal. This frequency difference generates a “numerical beating” which appears in the plotted waveform. Decreasing the timestep improves the plots, but uses more analysis time.

Transient Continuation

If transient analysis fails, Nexxim checks to see whether a nonpassive S-parameter element could be the cause. If a suspicious element is detected, Nexxim sets option **s_element.enforce_passivity=7** to enable passivity checking and enforcement by the IFPV method, and then retries the transient analysis. To disable automatic transient continuation for S-elements, set option **s_element.auto_enforce_passivity=0**. For W-elements, the option to disable automatic transient continuation is **w_element.auto_enforce_passivity=0**.

> **Note:**
>
> The global option **auto_enforce_passivity=0** overrides any passivity enforcement set for s_element or w_element.

See Circuit S-Parameter Technical Notes for more details.

Transient Diagnostics

- If Transient analysis fails, Nexxim checks the solution and finds the component that violates tolerance. The device information corresponding to the problematic component is printed in the log file.
- The ratio of the max node voltage and the min node voltage is tracked. If the ratio is large, a suggestion on relref is included in the log file.
- The first *N* largest/smallest time steps that lead to convergence are collected. Such information can be printed in the log file if setting the message controller to `error.filter=0`. The value of *N* for largest and smallest time steps can be modified by **`tran.num_large_h_record`** and **`tran.num_small_h_record`**, respectively.

Virtual EMI Receiver Probe

This FFT-based (i.e., Fast Fourier Transform) virtual EMI receiver (i.e., Time Domain Scan) probe reduces EMI/EMC compliance test time, allowing users to verify their designs with shorter bench and chamber time.

> **Note:** The EMI receiver probe enables post-processing of a signal available after completion of the transient analysis of a circuit. Refer to the Circuit Help for instructions.

The virtual EMI receiver functions with the following major steps for EMI/EMC compliance tests:

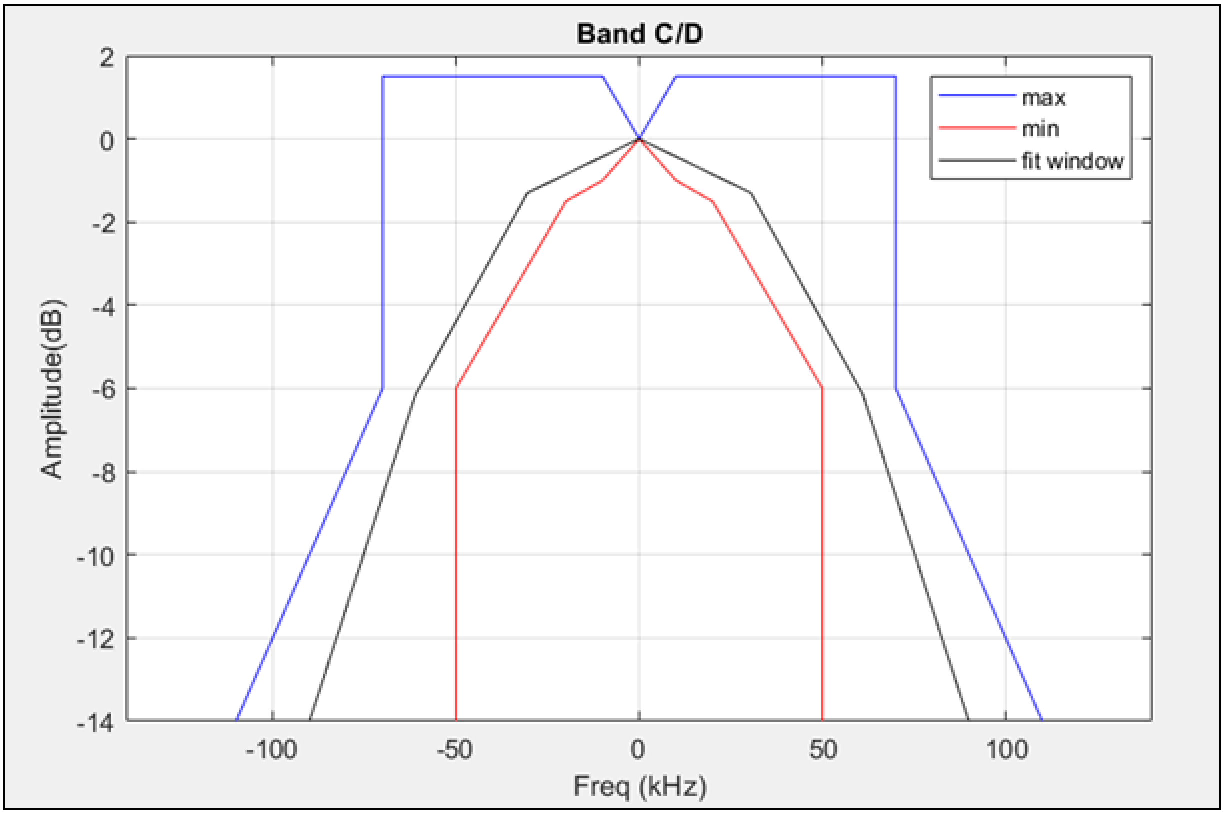
Transient signal mixed with EMI noise of DUT (Device Under Test) first undergoes Short-Time-Fourier-Transform (STFT) analysis with Gaussian Window ( ) to output 3D Spectrogram data which shows frequency component change in the signal with respect to time. Then the Spectrogram data is passed through the desired EMI detectors to obtain signal strength (e.g., dBmV) in the selected frequency range. Users can apply limit lines defined according to various standards (e.g., CISPR25 for automobile tests, IEC 61967 for IC circuits), to analyze if the DUT data passes EMC testing.

The accuracy of the virtual EMI receiver is dependent upon the length of signal (time duration), of Gaussian window, window length, size of FFT, and the overlap rate of windows during calculation of STFT. CISPR 16-1-1 defines the band pass selectivity requirement of the

Gaussian window for each band. Resolution bandwidth (RBW) is the bandpass width at 6dB below peak of a Gaussian window.

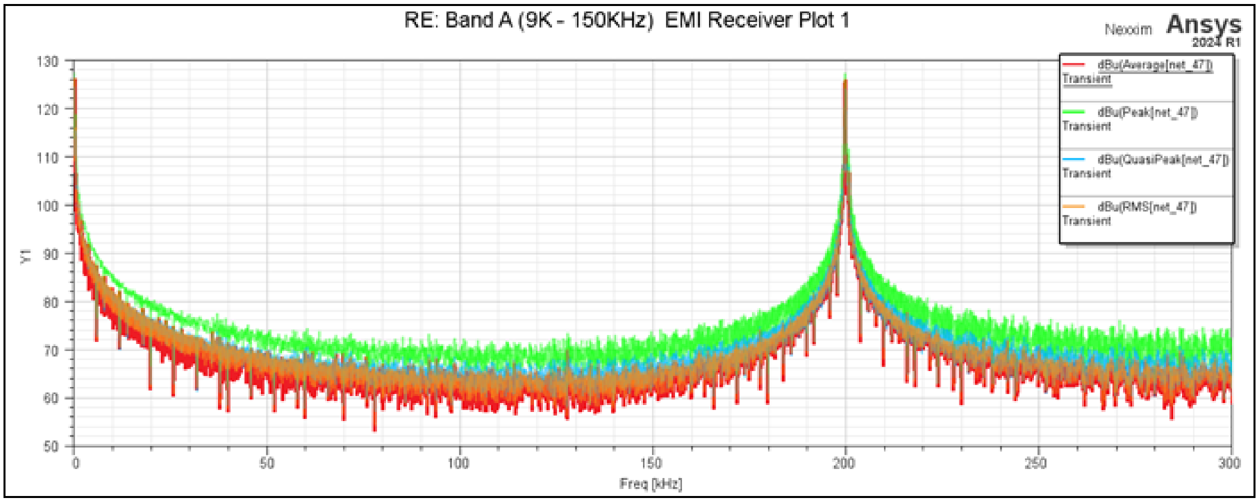
Characteristics	Frequency Band		
	Band A (9 kHz to 150 kHz)	Band B (0.15 MHz to 30 MHz)	Band C and D (30 MHz to 1,000 MHz)
Bandwidth at the -6 dB points (kHz)	**0.20**	**9**	**120**
Detector electrical charge time constant (ms)	**45**	**1**	**1**
Detector electrical discharge time constant (ms)	**500**	**160**	**550**

The following figure demonstrates a Gaussian window that successfully fits within the tolerance defined in CISPR 16-1-1 for Band C and D.

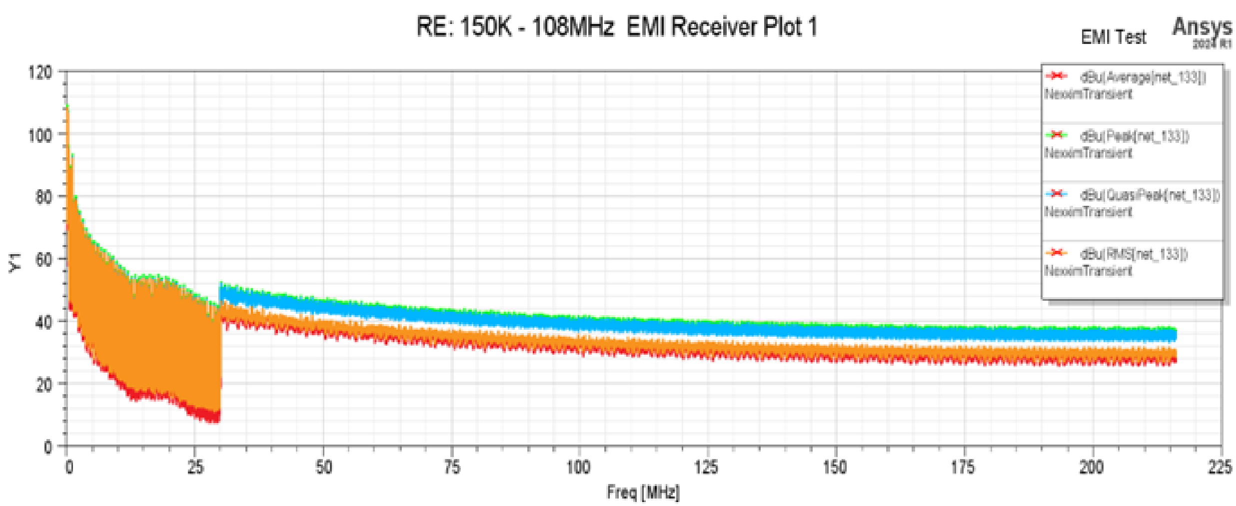


RBW affects EMI receiver's ability to resolve closely spaced signals. Two narrow signals can only be separated, if the resolution bandwidth is smaller than the distance between these two signals. RBW also determines the noise floor of the detected spectrum. Larger RBW results in higher measured noise floor because more frequency components can pass through a wider RBW filter for various detectors.

Exemplary detected EMI signals are shown below with P, QP, RMS and Ave detectors.



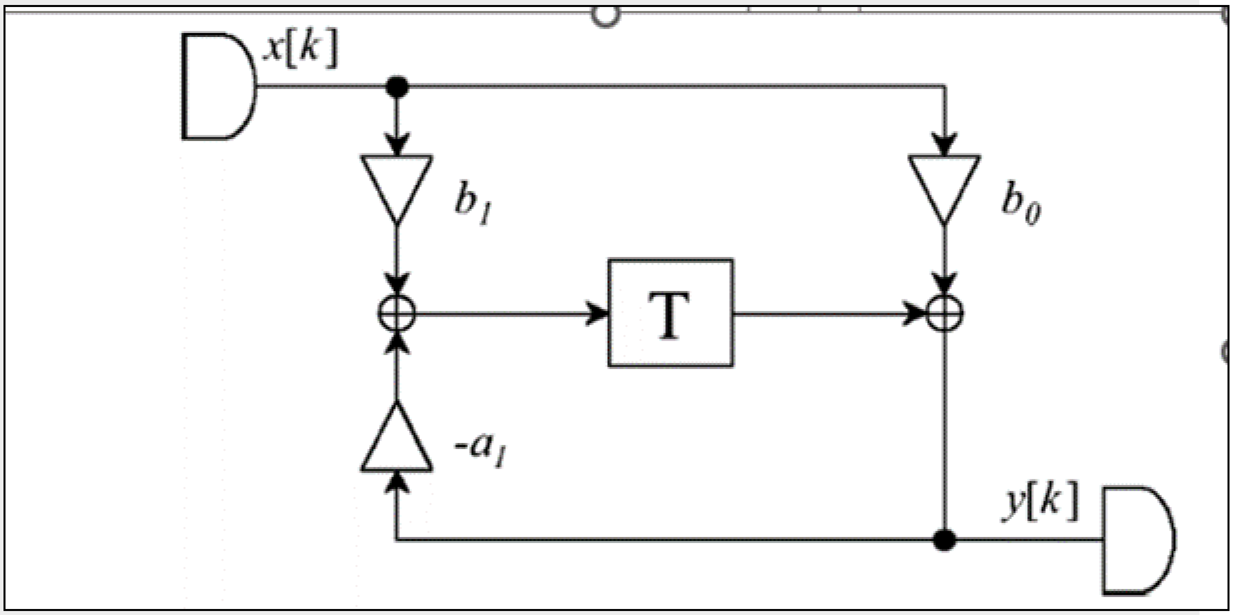
When a Band includes multiple CISPR16 Bands, detected EMI signals show steps in noise level due to RBW value change. Following is an example plot for Band 150K-108MHz(RBW = 9KHz is applied for frequency range from 150K to 30M, and RBW=120K is used for 30M-108MHz):



By default, Gassian parameter sigma, the size of the FFT, and the window size is automatically set to meet the band pass selectivity requirements defined in CISPR 16-1-1. When the signal length is less than the window size appropriate for the chosen frequency resolution (e.g., 2 frequency samples per RBW), the signal is automatically expanded by repeating the signal.

The virtual EMI receiver supports four types of detectors defined in CISPR 16-1-1 (i.e., Average, Peak, Quasi-Peak (QP) and RMS).

- Average computes the average signal over time
- Peak saves the maximum signal value over time at each frequency point
- QP provides weighted signal magnitude at each frequency point; component signals with a high repetition rate are weighted more than rarer signals
- RMS computes root-mean-square signals over time at each frequency point.

> **Note:** There are two time-constants associated with the QP detector for each band. Such a detector can be implemented as an IIR filter.
>
> 
>
>
> To achieve accurate QP results with large time constants, about 1 second of simulated signal is required. To overcome this limitation, as long as the signal length is 2.2 x charge time (rise to 90% of signal strength), the IIR filter structure is used to compute QP. Otherwise, a simplified estimate of QP detection is applied. Due to these assumptions, QP results should be used with consideration unless the signal is long enough to support the required time constants. The result of Peak and RMS detections provides upper and lower bounds for QP detection.

QuickEye and VerifEye Technical Notes

These topics provides background information and technical details about QuickEye and VerifEye. The formulas and relationships in these topics are conceptual, and do not describe the actual algorithms used by the solver.

The Bit Error Rate

The bit error rate (BER) is the figure of merit for signal integrity issues. The transmitter sends a sequence of bits through a channel to the receiver. In a perfect channel, every received bit value (0 or 1) is equal to the transmitted value. In the typical channel, some bit errors occur. The BER is the ratio of bits incorrectly received to the total number of bits transmitted.

Inter-symbol interference (ISI) is the major cause of bit errors. ISI occurs when the response of the channel to a change in bit value (from 0 to 1 or from 1 to 0) is longer than the unit interval (UI) or bit time. When a transition takes more than one UI to settle, the values of previous bits affect the waveform for the current bit.

Here is a simple example. Suppose the channel to analyze is a 100-ohm differential driver feeding into an R-C load with R=100 ohms and C=40 pF. Figure 1 shows the input bit sequence to the channel. 1V represents a logic “1”, while -1V represents a logic “0”. The unit interval (UI) is 1ns.

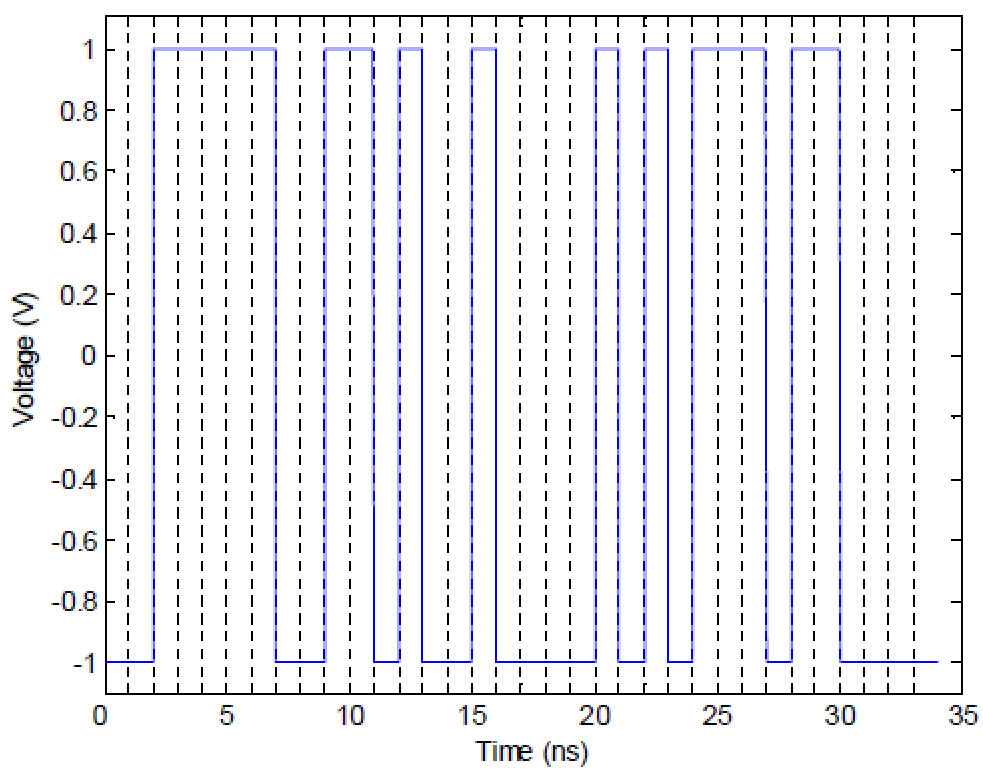


Figure 1: Input to Simple R-C Channel.

Figure 2 graphs the measured output at the receiver end of the channel, showing the degradation of the original stream of bits as it goes through the channel.

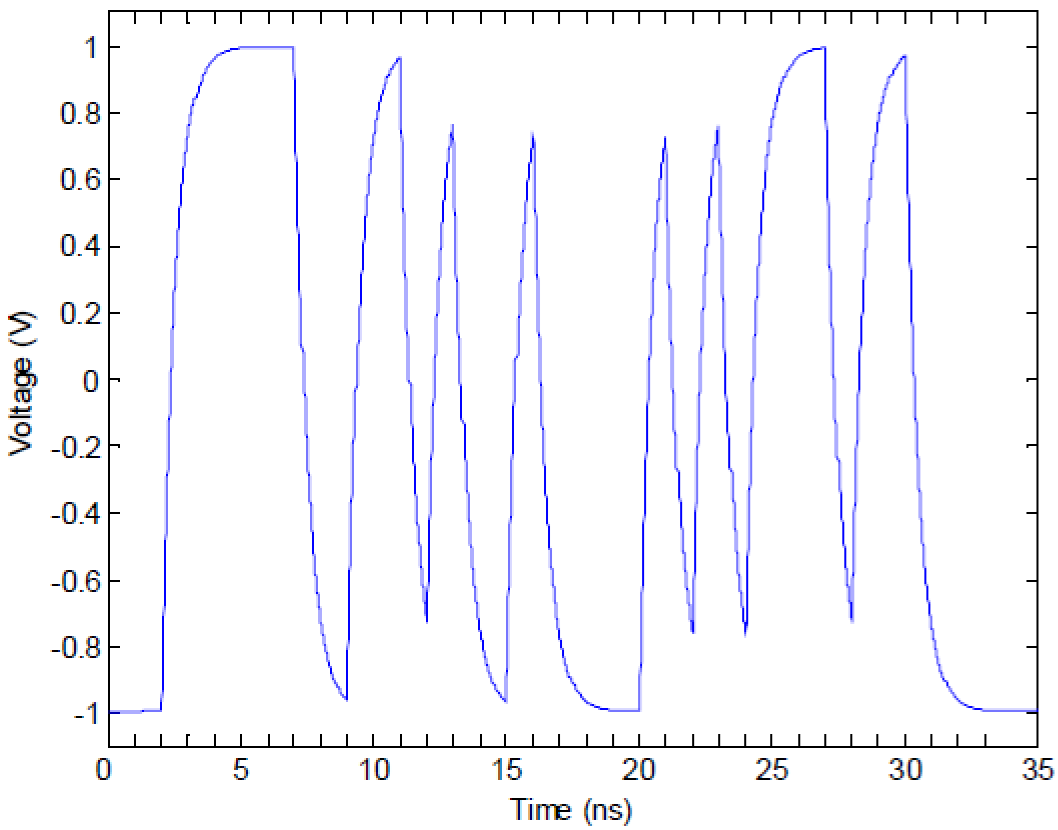


Figure 2: The output of the simple R-C channel.

The waveform in Figure 2 shows the effects of intersymbol interference (ISI). With this channel, the waveform for an isolated 1 surrounded by zeros never gets close to the nominal +1V value, and an isolated 0 surrounded by ones does not fall to -1V. Clearly, this channel has a non-zero bit error rate.

The eye diagram is a convenient tool for estimating the BER given the waveform. Copies of the waveform generated at the receiver end of the channel are overlaid at a spacing of one unit interval (UI). The eye diagram illustrates the allowable window for distinguishing bits from each other at the receiver end. The width and height of the eye opening are the dimensions of the window. The minimum allowable width of the eye depends on the jitter budget. The required height of the window is given by the noise margin of the receivers. Given the required width and height, how often does the waveform violate this window? This is the Bit-Error Rate, or BER.

Figure 3 shows the eye diagram created by shifting and overlaying the waveform of Figure 2. The area of white space in the center is the eye. (More bits is needed to obtain a better “eye.”)

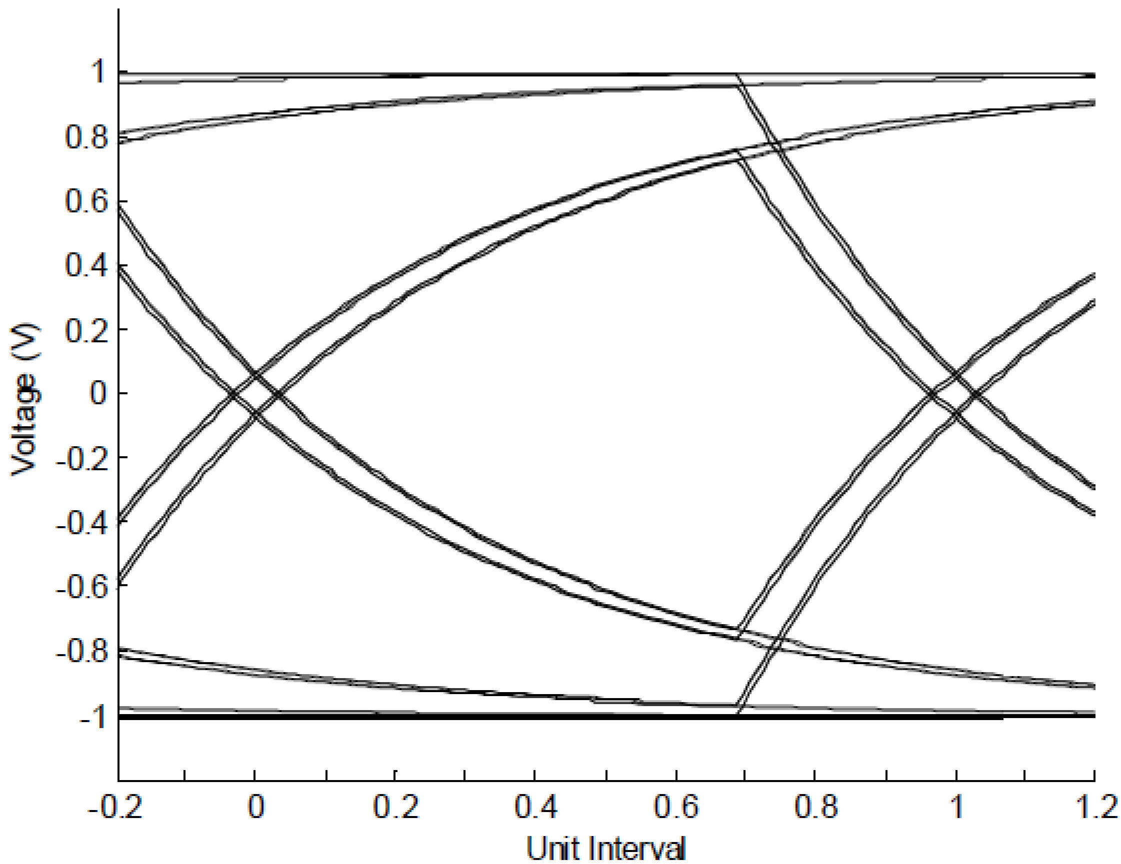


Figure 3: Eye Diagram for Simple R-C Channel

Traditionally, eye diagrams have been measured in the lab or simulated with transient analysis. However, with increasing data rates, particularly in the case of high-speed serial links, the required BER has decreased significantly. The need to predict very low BERs leads to a corresponding increase in the number of bits to be simulated, and thus to a search for alternatives.

The *probability* that a waveform violates the eye window can be measured from test data or calculated from channel parameters. Both transient analysis and QuickEye measure the BER probability by running sufficient data and counting errors. Quick Eye analysis uses simplifying assumptions to speed up the simulation, but requires the same large data sets to measure low BERs. VerifEye analysis uses a fully statistical approach to calculate the BER without requiring any test data.

The LTI Channel Model

A fundamental assumption for both QuickEye and VerifEye is the channel can be treated as a linear time-invariant (LTI) system characterized by a step response. QuickEye and VerifEye are based on the principle of superposition, which makes use of this assumption. Forcing an LTI approximation allows a convolution-based approach to solving the time domain problem. The following topics show how the step response (and the related edge response) can be understood in terms of LTI systems theory.

The LTI Criteria

A system is linear and time-invariant if it meets two criteria:

Linearity: If input $x_1(t)$ produces output $y_1(t)$ and input $x_2(t)$ produces output $y_2(t)$, the linear combination $a_1x_1(t) + a_2x_2(t)$ generates output $a_1y_1(t) + a_2y_2(t)$.

Time invariance: If input $x(t)$ produces output $y(t)$, input $x(t+t_0)$ produces output $y(t+t_0)$, where the offset t_0 can be positive or negative.

The Impulse Response

As a direct consequence of the LTI criteria, an LTI system can be characterized completely by its impulse response. The impulse response *h(t)* is the output of an LTI system when the input is a Dirac delta function δ(t). The Dirac delta function is a distribution that is 0 everywhere except at the origin, where it is of infinite magnitude (Figure 1).

$$\delta(t) = \begin{vmatrix} \infty, & t = 0 \\ 0, & t \neq 0 \end{vmatrix}$$

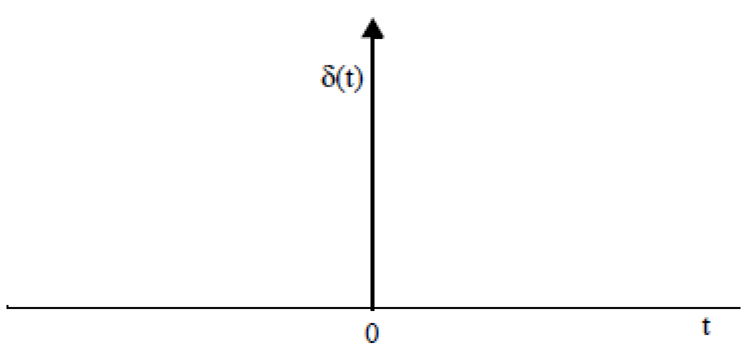


Figure 1. The Dirac Delta Function δ(t)

In time-domain formulas, the Dirac delta function is the unit impulse.

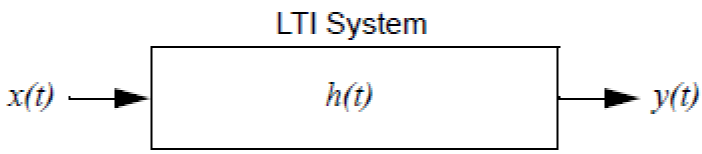


Figure 2. LTI System

For the overall LTI system, the output *y(t)* is the convolution (*) of the input signal *x(t)* and the impulse response *h(t)*:

$$y(t) = x(t)*h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

The Ideal Step Response

The step response *u(t)* is the response of the LTI system when the input is a pure step function. The unit step function (*D(t)* in this topic) is the cumulative distribution function (CDF) of the Dirac delta function:

$$D(t) = \begin{vmatrix} 1, & t >= 0 \\ 0, & t < 0 \end{vmatrix}$$

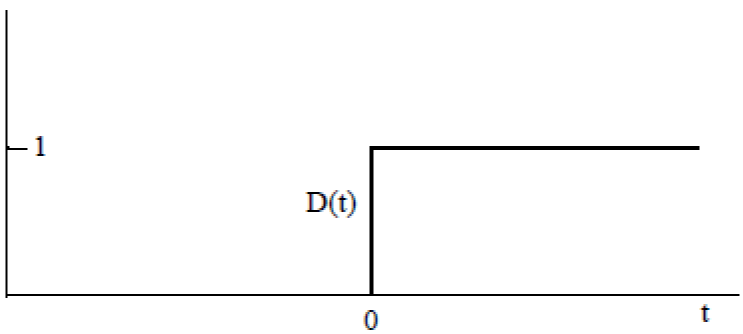


Figure 3. The Step Function D(t)

The Edge Response

Physical transmitters cannot produce a pure step function. The input to the channel on the transmitter is not a true step function but an edge function, rising or falling. The edge functions $E_R(t)$ and $E_F(t)$ incorporate the rise time and fall time specified in the eye source. These transition times cannot be zero.

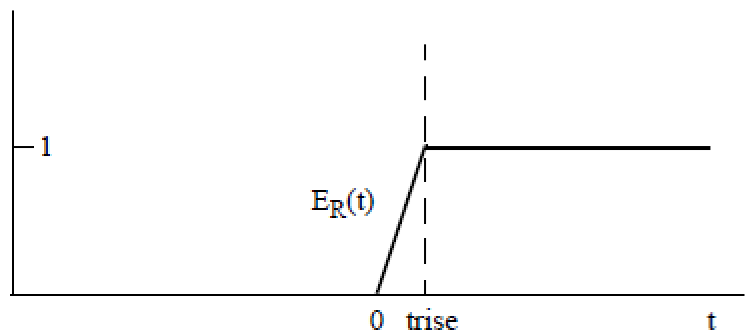


Figure 4. Rising Edge Function E $_R$(t)

By default, the rise and fall times are the same. The eye source can set the rise and fall times to different values.

The edge response *w(t)* is the response of the LTI system when the input is an edge function.

Both QuickEye and VerifEye run transient analysis to calculate the rising and falling edge responses of the channel.

By default, the solver simulates over a time period of 100 UIs to calculate the edge response. The **STEP_RESP_NUM_UI** parameter on the Eye Source sets the number of unit intervals needed to allow the response to reach a steady state (high or low).

Despite the formula differences, the term *step response* is used in the GUI window and in the text of the help topics to represent either the step or edge response.

Figure 5 shows typical rising and falling step responses.

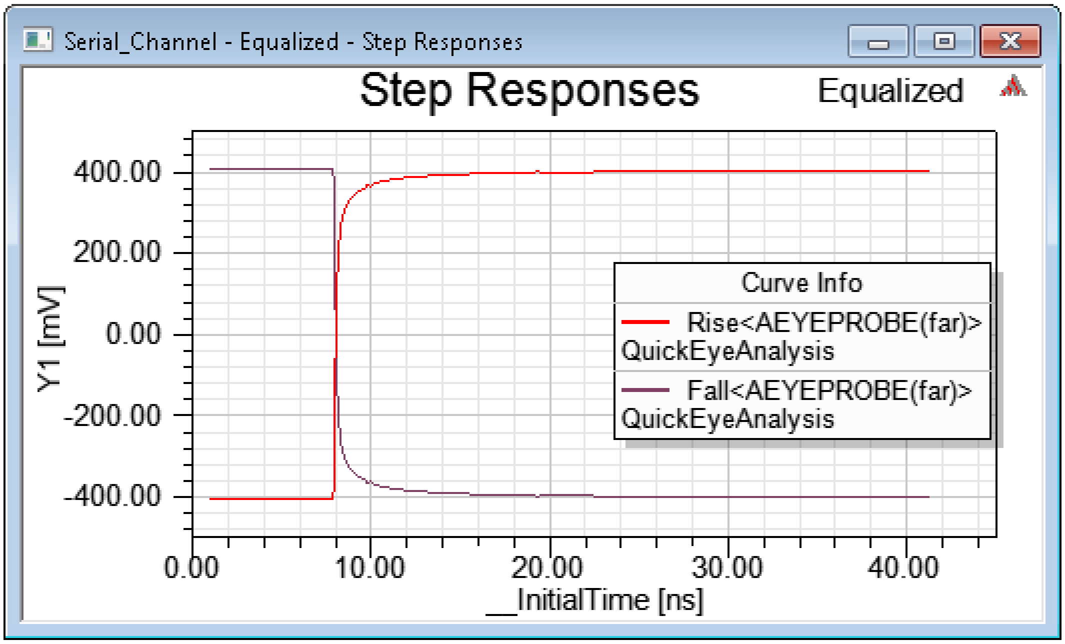


Figure 5. Typical Rising and Falling Step Responses *u(t)* or Edge Responses *w(t)*

QuickEye Technical Notes

Quick Eye analysis employs pattern-dependent convolution. The step responses are simulated with transient analysis for a single rising edge and a single falling edge. The step responses are then combined with the data stream to create an intermediate waveform that approximates the behavior of the fully simulated channel. With QuickEye, inter-symbol interference (ISI) is easy to calculate, but random transmit jitter is more difficult to handle.

QuickEye convolves the step responses for rising and falling edges with the time derivative of the input waveform to calculate the output. Here is a brief derivation.

The output of the LTI channel on the input and the impulse response once more:

$$y(t) = x(t)*h(t)$$

From the assumption of superposition, the impulse response *h(t)* is the time derivative of the step response *u(t).* This allows the following substitution:

$$y(t) = x(t)*\frac{d}{dt}u(t)$$

We can transform this formula into the complex frequency **s** domain by applying the Laplace transform to both sides:

$$Y(s) = X(s)sU(s)$$

Where *Y(s)* and *X(s)* are the Laplace transforms of *y(t)* and *x(t)*, and *sU(s)* is the Laplace transform of *du(t)/dt*. Rearranging the product terms, to get:

$$Y(s) = sX(s)U(s)$$

Take the inverse Laplace transform of both sides to obtain the formula used by QuickEye:

$$y(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t)*u(t)$$

This formulation captures the edge-based nature of the calculation, since the derivative *dx(t)/dt* is non-zero only during the transitions.

QuickEye creates an intermediate waveform by applying the rising and falling step responses to the transitions in the input data. Figure 1 shows the intermediate QuickEye waveform for the RC example. Rising edges are solid, falling edges are dashed. Summing these transitions creates the waveform in Figure 2.

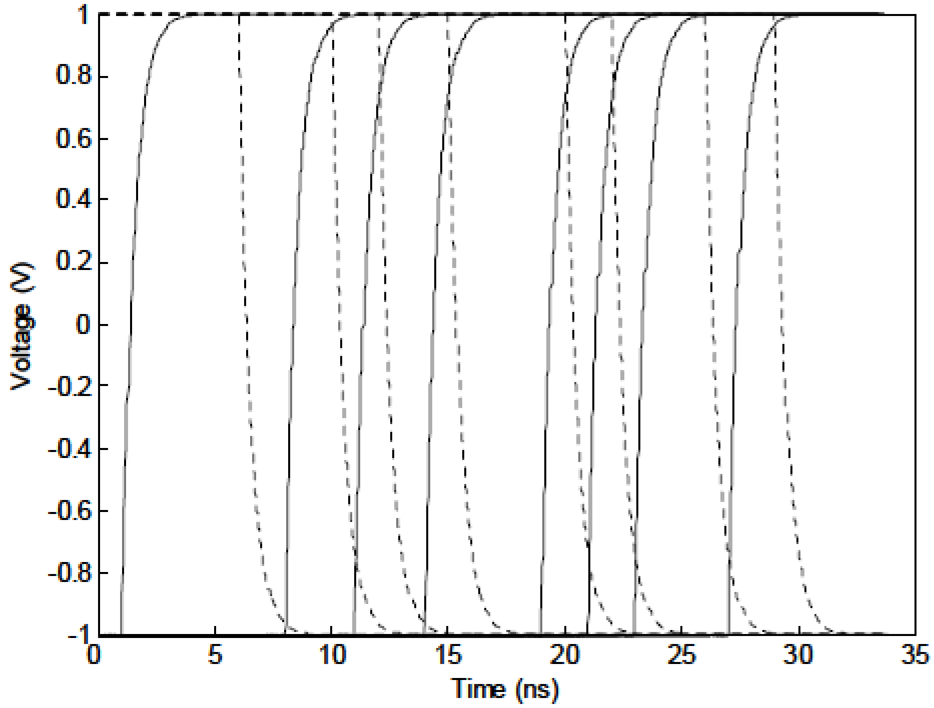


Figure 1. QE Intermediate Waveform for RC Example Data

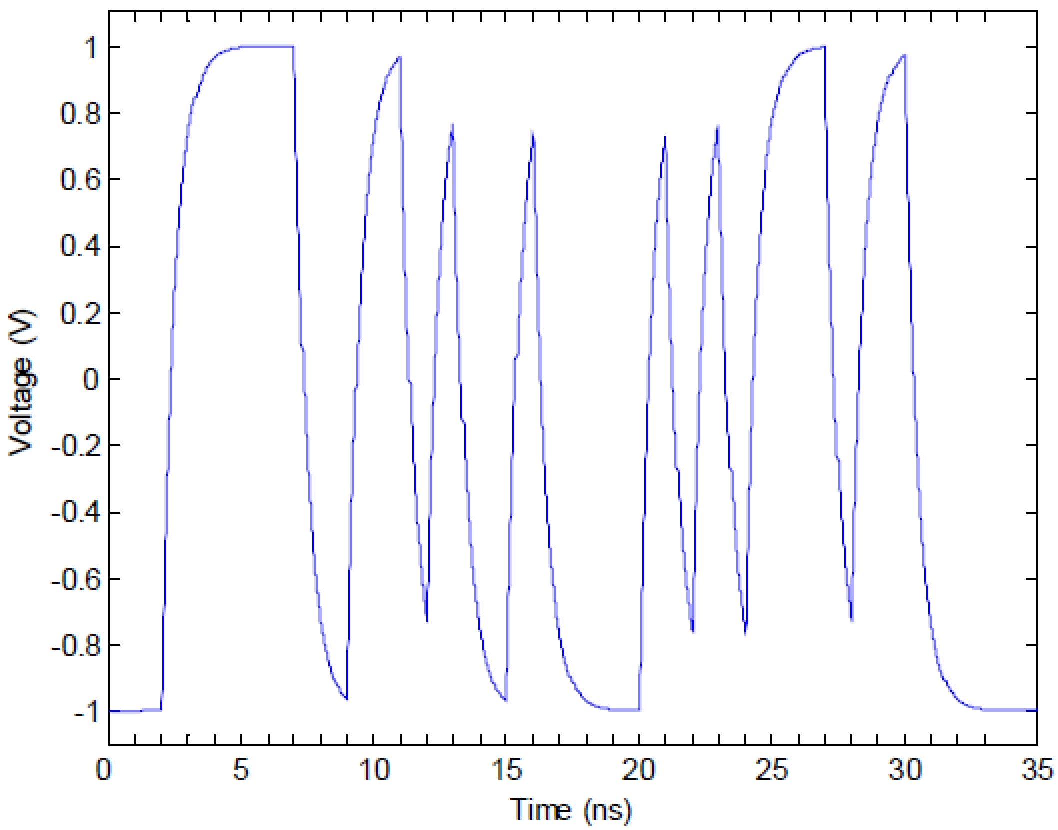


Figure 2: The RC Waveform as Simulated with Transient Analysis

QuickEye overlays the unit intervals on the intermediate waveform to generate the eye diagram. The input bit pattern can be a literal bit list specified in the netlist, a sequence of random bits, a bit pattern read from a file, a pseudo-random selection of bit patterns, or the worst-case bit pattern calculated by peak distortion analysis. (See Eye Source Bit Data and Peak Distortion Analysis).

QuickEye calculates the bit error rate by comparing the data received to the data sent, and counting the number of errors. For a given set of data, the QuickEye convolution algorithm is faster than full transient analysis. Like transient analysis, however, QuickEye requires gigabit run sizes to determine BER values in the 10^{-12} range. This practical limitation is the reason for VerifEye.

VerifEye Technical Notes

VerifEye analysis is a statistical eye-analysis algorithm, similar to the public domain Stat Eye (See QuickEye and VerifEye References for a citation). VerifEye calculates the bit error rate from an analysis of the probability density function for the receiver voltage. VerifEye incorporates the effects of intersymbol interference (ISI) by considering how transitions that preceded the current bit could affect the receiver voltage, given the step response.

For example, suppose the current receiver bit is a 1. This can happen in two ways: either the previous bit was a 1, in which case there was no transition, or the previous bit was a 0, in which case there was a low-to-high transition. In the first case, the receiver voltage is very near to V_{HI}, the voltage representing a “1” bit. In the second case, the voltage does not reach V_{HI} due to the slow rise time of the step response at the end of the channel. If the bits are independent, the probability of a transition here is 0.5. Statistically, therefore, the waveform has a 50% chance of having no deviation, and a 50% chance of having a negative deviation.

Chain this process forward and backward, considering each possible transition in turn. However, even if the bits are independent, the transitions are not. A high-to-low transition must be followed by a low-to-high transition and vice versa. VerifEye calculates conditional probabilities including all the possible cases.

The results are probability distribution functions (PDFs) for the voltage value of the rising and falling waveforms. The probability density function $p_V(v)$ gives the probability of obtaining a particular voltage at the receiver at some sample time tau (τ). Figure 1 shows an artificial example.

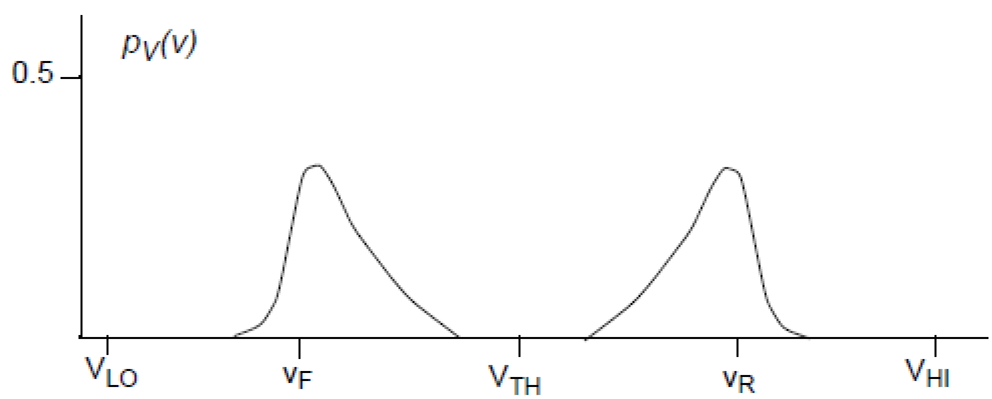


Figure 1. Probability Density Function for Receiver Voltage at Sample Time τ

The distribution of $p_V(v)$ is bimodal. The lobe on the left is the PDF for the falling waveform at the given sample time. The lobe on the right is the PDF for the rising waveform at the sample time.

If the channel are purely lossless, the two lobes is narrow peaks. The preceding figure,shows some spreading of the distributions due to intersymbol interference (ISI), jitter, and other factors that causes the voltage to vary around the ideal values. The effect of ISI is to spread the distribution toward the threshold voltage.

Represent the receiver voltage PDF with formulas. Let B_{TX} be the logic value (0 or 1) of the transmitted bit, and let B_{RX} be the logic value (0 or 1) of the received bit. The left lobe Figure 1 is the PDF of the receiver voltage when the transmitted bit was a 0, or in probability notation:

$$p_V(v)\big|_{B_{TX} = 0}$$

The right lobe is the receiver voltage PDF when the transmitted bit was a 1, or:

$$p_V(v)\big|_{B_{TX} = 1}$$

Since B_{TX} must be either 0 or 1, the total area under the PDF is unity:

$$p_V(v)\big|_{B_{TX} = 0} + p_V(v)\big|_{B_{TX} = 1} = 1$$

If the bit values are independent, the two lobes are equal in area.

The bit error rate (BER) depends on the threshold voltage. Let V_{TH} be the threshold voltage that determines whether the receiver bit B_{RX} is a 1 or a 0. The bit error rate at sample time τ is the area under the PDF that is above V_{TH} when the transmitted bit B_{TX} is a 0, plus the area of the PDF that is under V_{TH} when B_{TX} is a 1, all divided by the total area under both lobes. Since the total area under the PDF is unity, the area of the overlapping regions equals the bit error rate.

Consider just the left lobe of the PDF, the voltage lobe when the transmitted bit is a zero. Figure 2 shows a PDF for B_{TX}=0, with a tail that overlaps the threshold (the area of overlap is exaggerated for effect).

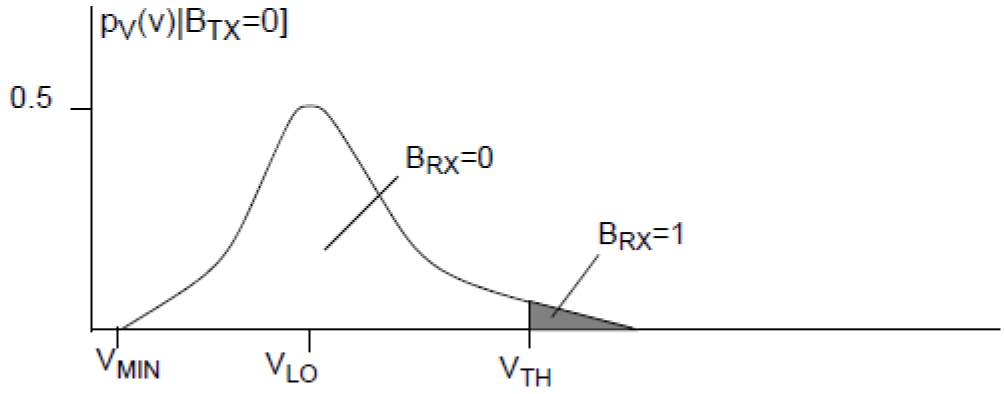


Figure 2. PDF with Bit Error Rate.

The area under the curve that lies under the threshold V_{TH} is the probability that the received bit is correctly received as a 0. The shaded region in Figure 2 represents the probability that the received bit is incorrectly read as a 1 when the transmitted bit is a 0.

The area of the shaded region represents the probability:

$$P\left(B_{RX} = 1\Big|_{B_{TX} = 0}\right) = P\left(v(\tau) > V_{TH}\Big|_{B_{TX} = 0}\right)$$

Where *P()* signifies a simple probability rather than a density function.

If the PDF lobe for B_{TX}=1 also has a tail that falls under the threshold, the area of the error region represents the following probability:

$$P\left(B_{RX} = 0\Big|_{B_{TX} = 1}\right) = P\left(v(\tau) < V_{TH}\Big|_{B_{TX} = 1}\right)$$

The sum of the two probabilities is the bit error rate at the specified threshold voltage V_{TH} and at the specified sample time τ.

$$BER_\tau = P\left(v(\tau) > V_{TH}\Big|_{B_{TX}=0}\right) + P\left(v(\tau) < V_{TH}\Big|_{B_{TX}=1}\right)$$

When BER_τ is plotted over all values of sample time τ from 0 to 1UI, the result is the familiar bathtub curve (Figure 3). Each point on the curve shows the probability that the received bit B_{RX} fails to equal the transmitted bit B_{TX} for some value of the threshold voltage V_{TH}, at sample time τ. In Figure 3, the Y-axis uses logarithmic scaling.

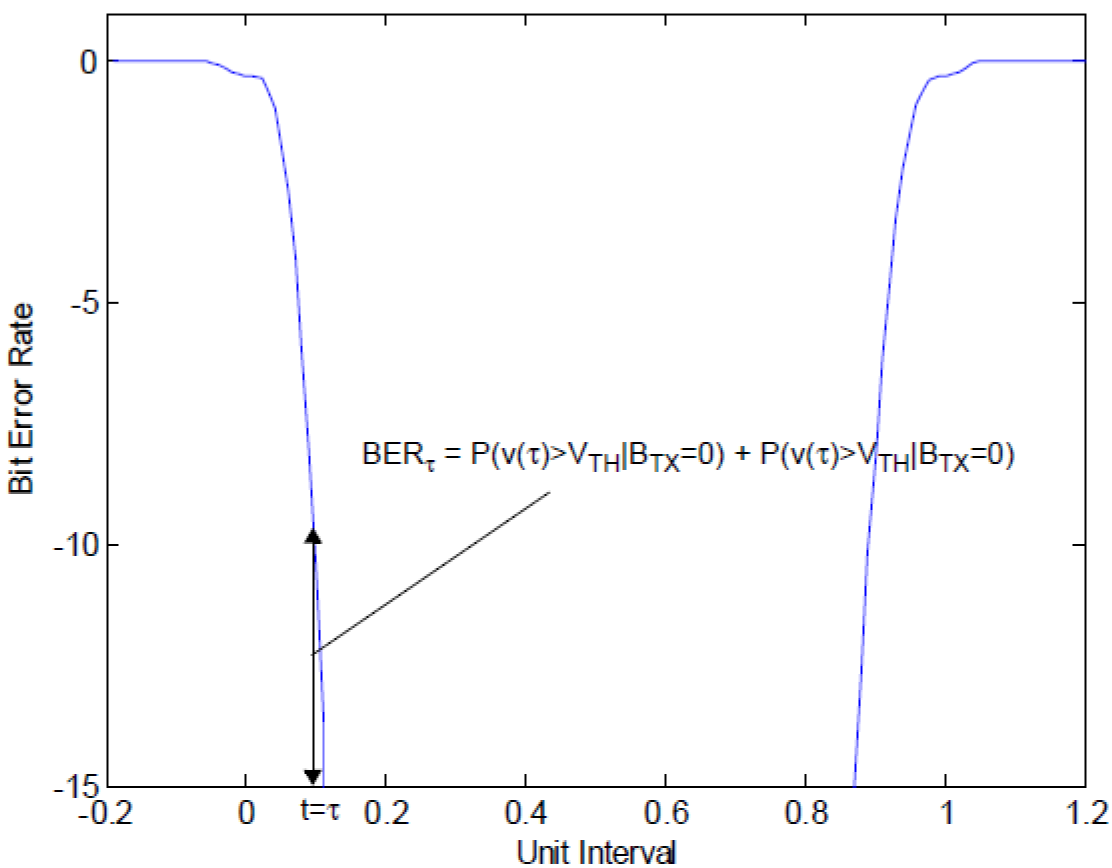


Figure 3. A Bathtub Curve of Bit Error Rate (BER) at Threshold Voltage V_{TH}.

The integral of the BER curve over threshold voltages ($V_{MIN} < V_{TH} < V_{MAX}$) produces the cumulative distribution function (CDF) of the receiver voltage. A plot of the CDF is the 3D bathtub or contour plot.

Feed-Forward Equalization

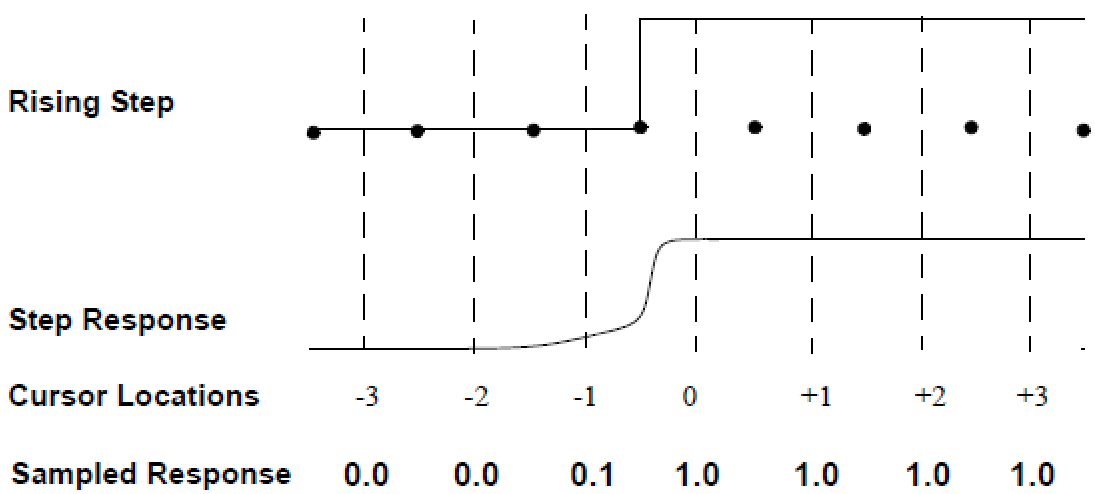
Equalization is one of the principal methods for improving the signal integrity of a channel. Equalization can be applied at the transmitter (Eye Source), the receiver (Eye Probe), or both. The Eye Source provides Feed-Forward Equalization (FFE). The Eye Probe provides Decision-Feedback Equalization (DFE) and Continuous-Time Linear Equalization (CTLE). This topic describes how Nexxim performs driver-side FFE. See Decision Feedback Equalization for details on DFE. See Continuous-Time Linear Equalization for details on CTLE.

Feed-forward equalization (FFE) is an extended form of frequency enhancement. Channels typically attenuate high-frequencies more than low frequencies. FFE adjusts the waveform being injected into the channel to compensate for frequency-dependent losses suffered during propagation.

The basic idea of FFE is to replace a single driver with a series of drivers, each one delayed by a set amount on the previous one. The delay in Nexxim is the unit interval (UI). These drivers are called *taps*. Each tap drives with a given strength, called the *tap weight*. The tap weights are set so as to reduce intersymbol interference (ISI).

Calculating the FFE Tap Weights

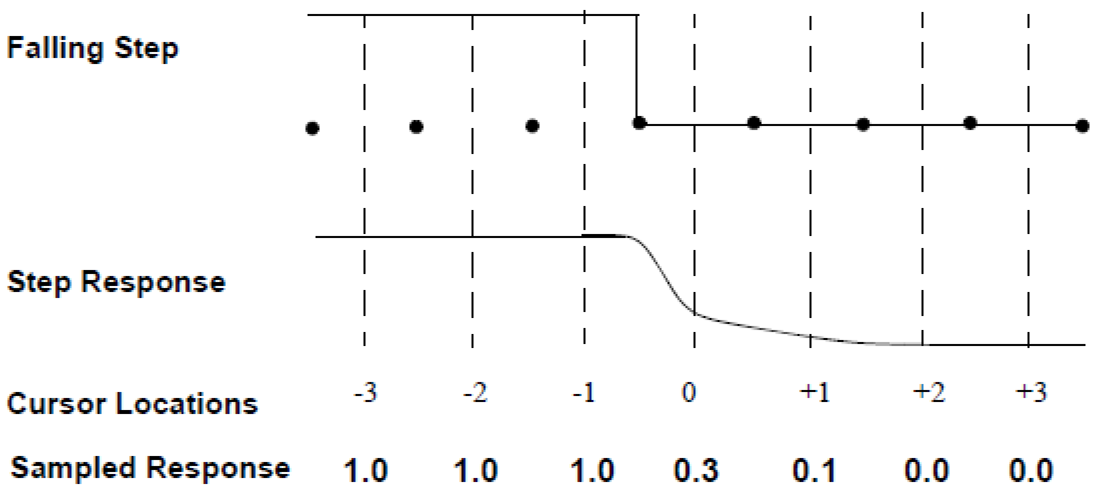
The algorithm for automatically calculating the tap weights in QuickEye and VerifEye is the Zero-Forcing Equalizer (ZFE). The ZFE algorithm is invoked when the FFE weights are not given but a non-zero number of taps is specified. The algorithm starts with the response to the channel of a rising step function.



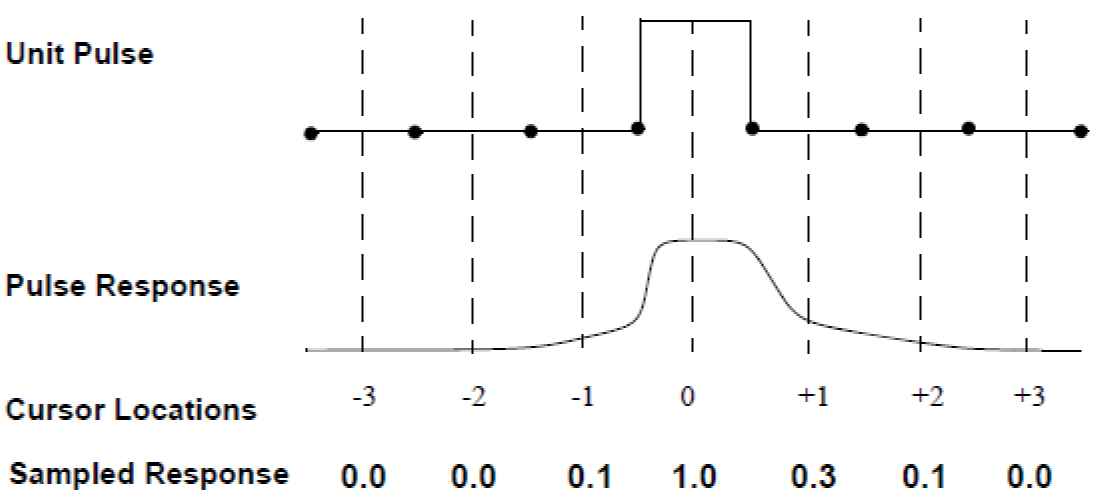
In the preceding figure, the dots on the Rising Step function plot delimit the unit intervals (UIs).

The dotted lines show the time locations of the *cursors* or sample points for the Step Response. The cursors are at the midpoint of each UI in this example. The numbers under the cursor lines are cursor *locations*. Location 0 is the *main cursor*, sampling the response at the center of the pulse. Cursors at locations -1, -2, -3, etc. are *precursors*, sampling the values of bits that occur earlier in time than the main cursor. Cursors at locations +1, +2, +3, etc. are *postcursors*, sampling the values of bits that occur later in time than the main cursor. The lowest line in the preceding figure,shows the value of the sampled step response at each cursor.

Inverting the rising step response produces the falling step response.



Adding the rising step response to the falling step response shifted one UI simulates the response of the channel to a unit pulse.



The goal is to make the total response zero at the time points corresponding to the center of the eye for ISI for a number of cursors in the pulse response equal to the number of FFE taps specified. The FFE setup window lets you specify how many taps to use and which cursor locations to equalize. Each sampled response is replaced by a weighted sum of the neighboring responses.

For example, a three-tap FFE with starting location -1 equalizes the main cursor, one precursor, and one postcursor. The equation for the weighted sum looks something like Equation 1:

$$E_C = w_1 R_{C-1} + w_2 R_C + w_3 R_{C+1}$$

(1)

Where:

E_C is the equalized value of the response at cursor *C*.

w_1, w_2, and w_3 are the weights associated with the three taps.

R_X is the sampled value of the response at the cursor locations *X* = *C-1*, *C*, and *C+1*.

Applying this formula over a number of cursors and setting each weighted sum to the corresponding value in the unit pulse produces a system of equations. Using the pulse response in the preceding figure, here are the equations for a three-tap FFE with one precursor (*C* = -1), the main cursor (*C* = 0) and one postcursor (*C* = +1).

$$E_{-1} = 0.0w_1 + 0.1w_2 + 1.0w_3 = 0$$

(2)

$$E_0 = 0.1w_1 + 1.0w_2 + 0.3w_3 = 1$$

(3)

$$E_1 = 1.0w_1 + 0.3w_2 + 0.1w_3 = 0$$

(4)

The sampled responses are known, and all other cursor values are forced to be zero. Solving for the tap weights yields w_1 = -0.308, w_2 = 1.063, and w_3 = -0.106.

> **Note:**
> 1. QuickEye and VerifEye normalize the automatically calculated tap weights such that the sum of all weights is 1.0. No calculation is done in this example.
> 2. The cursor locations in principle extend infinitely in both directions, and always exceed the number of taps. The response of cursors outside the tapped region can limit the accuracy of the tap weight calculations. Nexxim applies an error-minimizing algorithm to correct for this source of error. This calculation is not shown in this example.

Applying the FFE Tap Weights to Generate the Output Voltages

To generate the output, QuickEye and VerifEye apply the tap weights are applied to the step response $V_s(t)$ calculated using the transient analysis engine of the Circuit solver.

Begin with the default operation, no equalization. QuickEye and VerifEye apply the step response to the rising and falling edges in the bit pattern.

Here are some terms to be used in the formulas:

The step response is $V_s(t)$.

The subscript i references the cursor locations where bits are evaluated, $-\infty \leq i \leq \infty$.

b_i is the value (0 or 1) of the bit at cursor location i.

The step response is applied only at transitions. Function is defined α to be the exclusive-OR of the current bit and its immediate predecessor:

$$\alpha(b_i) = b_i \otimes b_{i-1}$$

Function α is 1 when the bits differ, indicating that a transition has occurred. Function α is 0 for non-transitions.

When the current bit is a one, the transition is a rising edge and the sign of the step response is positive. When the current bit is a zero, the transition is a falling edge and the sign of the step response is negative. Function is defined $\beta(b_i)$ to set the sign of the adjustment based on the bit value (0 or 1) of b_i.

$\beta(0) = -1, \beta(1) = +1$

With this notation, the formula for the unequalized output voltage is expressed at time t as follows:

$$V_{out}(t) = \sum_{i=-\infty}^{\infty} \alpha(b_i) \cdot \beta(b_i) \cdot V_s(t - i \times UI)$$

(5)

Now add *N* taps of FFE, indexed with variable *j*, which is swept on the Starting Location specified in the GUI, $1 - N \le Start \le 0$, to the Ending Location, $End = Start + N - 1$. The series of tap locations must always include location 0.

The tap weights W_j are the ones calculated earlier.

Using this notation, express the formula for the voltage output adjusted by FFE as follows:

$$V_{FFE}(t) = \sum_{i=-\infty}^{\infty} \sum_{j=Start}^{End} \alpha(b_{i-j}) \cdot \beta(b_{i-j}) \cdot W_j \cdot V[t - (i-j) \times UI]$$

(6)

FFE in the Frequency Domain

Channel attenuation is equivalent to adding a low-pass filter. To compensate, the FFE acts as a high-pass filter in the frequency domain.

The frequency response *H(s)* is the Laplace transform of the time domain pulse response *R(t)*. The frequency response of an N-tap equalized signal $H(s)_{FFE}$ can be calculated on the unequalized frequency response *H(s)*, the tap weights W_j, and the time delays T_j.

$$H_{FFE}(s) = H(s) \cdot \sum_{j=1}^{N} W_j \cdot e^{-j\omega T_j}$$

(7)

The frequency response on the typical FFE is that of a high-pass filter.

FFE Examples

Figures 1 and 2 show Quick Eye analyses without equalization and with FFE.

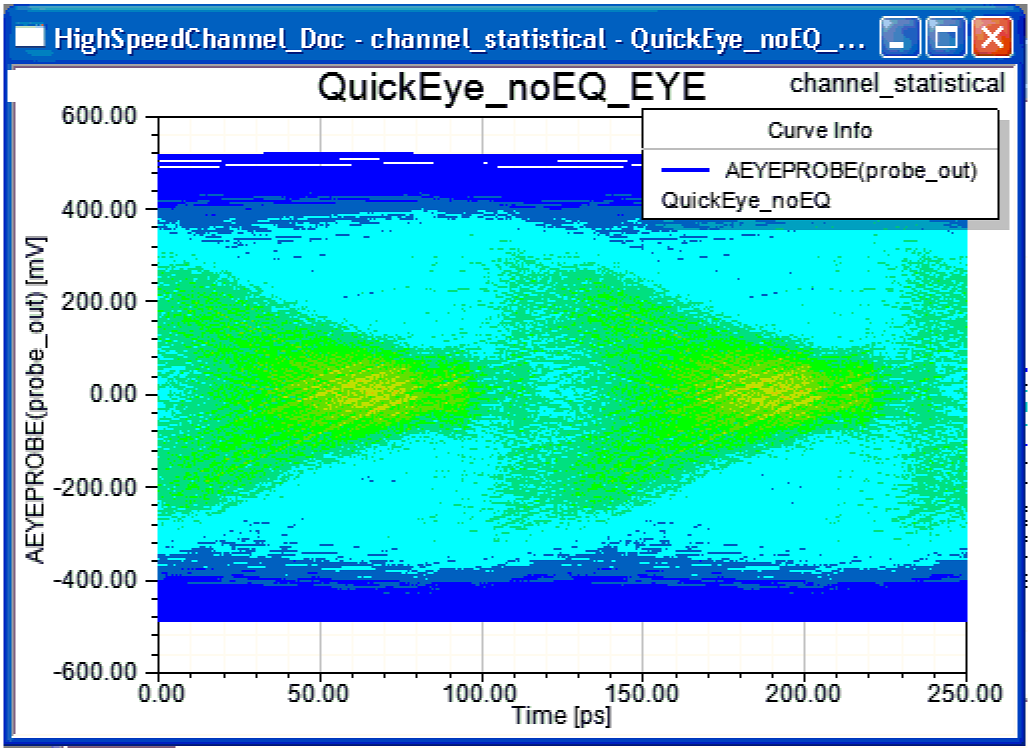


Figure 1. Quick Eye analysis of a high-speed channel with no equalization.

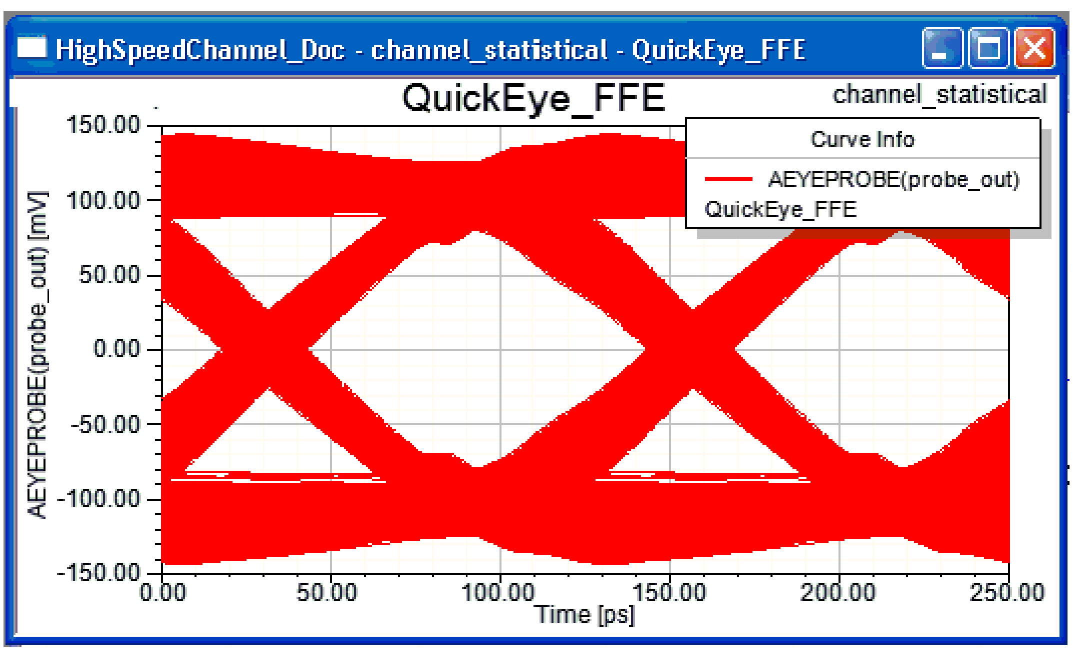


Figure 2. Quick Eye analysis of the high-speed channel with 4-tap FFE. The tap coefficients automatically calculated by the Circuit solver are (0.5797, -0.3481, 0.0386, and -0.0335).

Figures 3 and 4 show VerifEye analyses without equalization and with Feed-Forward Equalization.

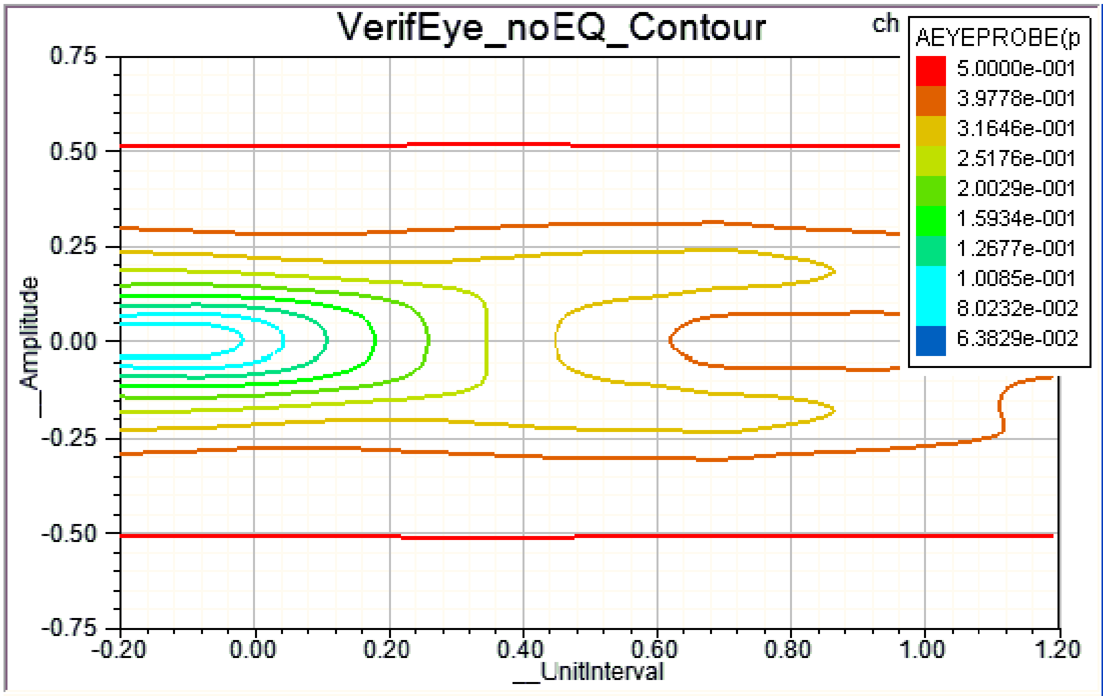


Figure 3. VerifEye analysis of a high-speed channel with no equalization.

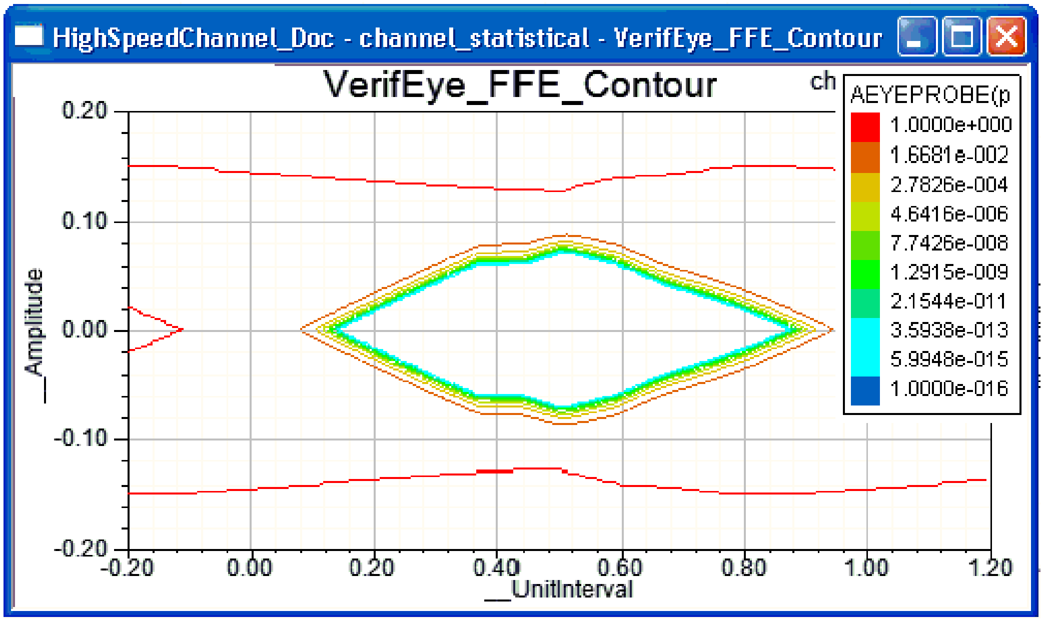


Figure 4. VerifEye analysis of the high-speed channel with 4-tap FFE. The tap coefficients automatically calculated by the Circuit solver are (0.59792, -0.3481, 0.03863, and -0.03355).

Feed-forward equalization is set up in the Eye Source, using the FFE_TAPS, FFE_LOCS, FFE_WEIGHTS, NORMALIZE_FFE_WEIGHTS, and FFE_COMPUTE_PROBE parameters.

Decision-Feedback Equalization

Decision-Feedback Equalization (DFE) is set up in the Eye Probe. Receiver DFE corrects for inter-symbol interference (ISI) from previous bits on the interpretation of the current bit.

Eye Probe parameters specify values for logic 1, logic 0, and the default threshold voltage. The decision-feedback equalizer applies the threshold to determine the logic value (0 or 1) of each incoming bit. The DFE keeps the results of the decision on the state of previous bits, and applies a set of tap weights to compensate for ISI at rising and falling transitions.

As in FFE, the number of DFE taps determines the number of unit intervals over which equalization is to operate. The DFE knows what the transmitted voltage level of a 0 or 1 should be. For each decoded bit, the DFE can calculate the deviation of each received bit voltage on the ideal voltage. The DFE weights are applied to compensate for the deviations. In a classical DFE, the sum of the tap weights multiplied by the corresponding bit errors is added to the received voltage of the current bit before it is sent to the decoder. Instead, the Circuit solver applies the sum of the weighted tap values to the threshold voltage.

The DFE weights can be automatically calculated using an algorithm similar to the algorithm used for FFE. Alternatively, supply the DFE weights directly. No normalization is applied to the DFE weights in either case.

Formula for DFE Output

While the FFE taps work toward canceling ISI both before the UI (precursors) and after the UI (post-cursors), DFE taps are limited to canceling post-cursors, because of the need to make a decision on a bit before its effect can be dealt with.

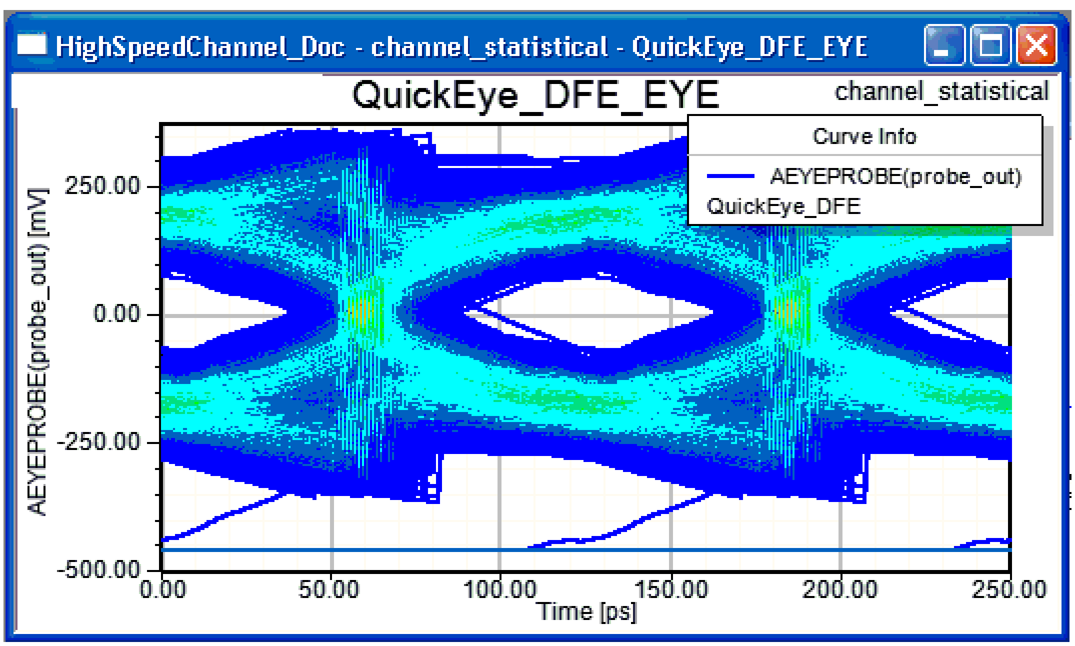
Like FFE, DFE is applied only at transitions (rising and falling edges). Using the notation introduced earlier for FFE, function α is 1 for transitions and 0 for non-transitions, while function β is +1 for rising edges and -1 for falling edges. DFE tap locations start at location 0, the current bit. The formula for the output voltage with *M* taps of DFE is as follows.

$$V_{DFE}(t) = \sum_{i=-\infty}^{\infty} \sum_{j=0}^{M} \alpha(b_{i-j}) \cdot \beta(b_{i-j}) \cdot W_j$$

(8)

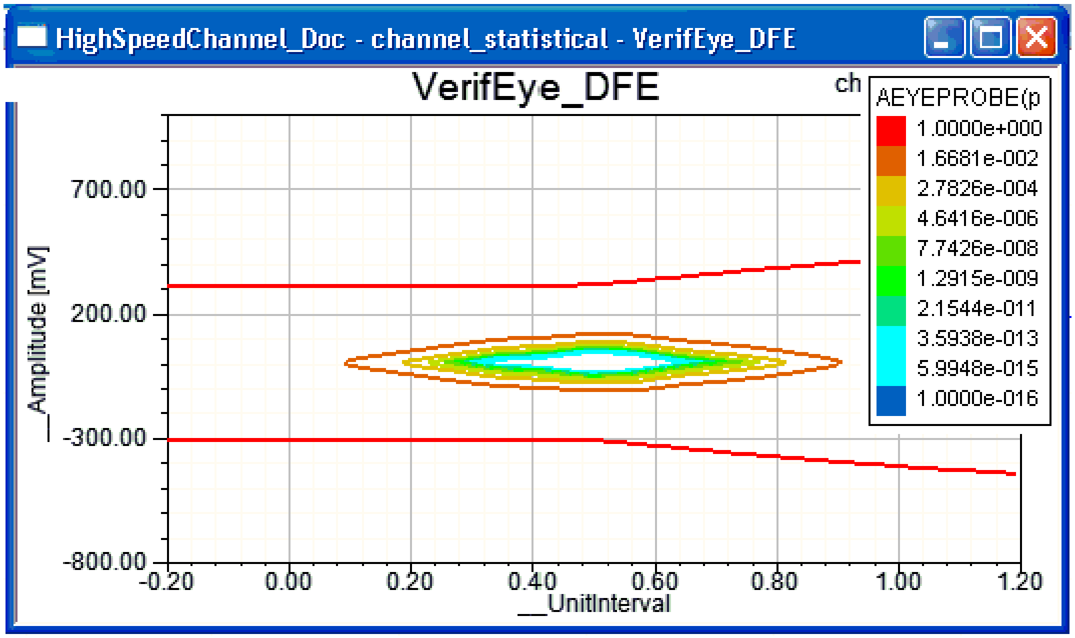
DFE Examples

Figure 1 shows the Quick Eye analysis of the same high-speed serial channel shown in Figure 8, but with 4 taps of DFE. The abrupt edges are produced when the DFE adjusts the decision threshold.



Weights automatically calculated for DFE by the Circuit solver are (0.10439, 0.05148, 0.03139, and 0.02215)

Figure 2 shows the VerifEye analysis of the same high-speed serial channel shown in Figure 10, but with 4 taps of DFE.

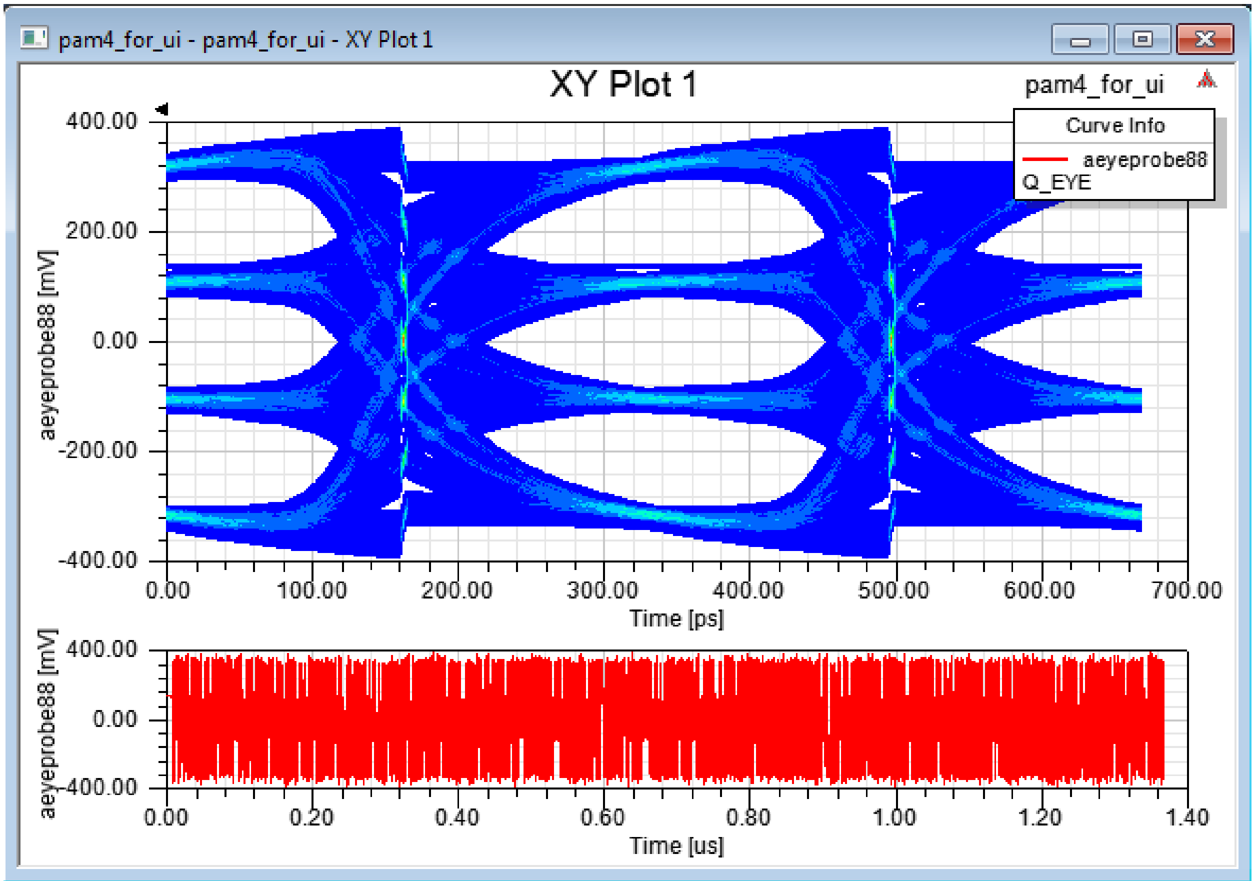


Weights automatically calculated for DFE by the Circuit solver are (0.10439, 0.05148, 0.03139, and 0.02215)

Decision-feedback equalization is set up in the Eye Probe, using the DFE_TAPS, DFE_LOCS, DFE_WEIGHTS (or DFE_WEIGHT_LBS and DFE_WEIGHT_UBS), DECISION_HIGH, DECISION_LOW, and DECISION_THRESHOLD parameters.

Decision-Feedback Equalization with PAM4

When PAM4 modulation is used, the eye diagram report shows three eyes, bounded by the four voltage levels. DFE can be applied to all three eyes in PAM4 transmissions, using a vector of three voltage thresholds to control the decision portion of the algorithm.



Continuous Time Linear Equalization

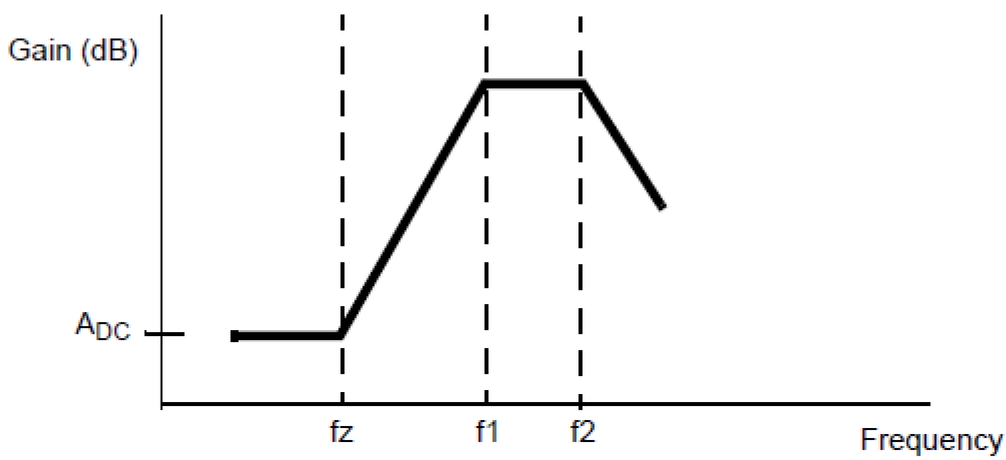
Communication channels typically attenuate the high-frequency components of the signal more than the low-frequency components. High-frequency roll-off degrades rise and fall times. Continuous-Time Linear Equalization (CTLE) compensates for this loss by inserting a high-pass filter at the receiver.

Nexxim offers four ways to specify the CTLE transfer function:

1. By specifying one or two poles and, optionally, one zero and the DC gain of the high-pass filter.
2. By specifying the tabulated values of the frequency-dependent transfer function directly with a data file.
3. By specifying any number of poles and zeros, the range of DC gain values, and the high-frequency gain, representing a generic rational function.
4. By specifying the coefficients of the numerator and denominator of a polynomial representing a generic rational function.

CTLE Data Parameters

The default, "legacy" implementation uses a first- or second-order filter model, depending on whether one or two poles (respectively) are specified. To enable CTLE, the user gives the linear frequency corresponding to the first pole (*f1*). Optionally, the user can supply a linear frequency corresponding to the second pole (*f2*), a zero frequency for the response curve, and a list of allowable DC *Gain* values. The first pole frequency *f1* typically is the lower bound of the high passband. The optional second pole frequency *f2* typically is the upper bound. The optional zero frequency *fz* typically is the beginning of the upslope in the response curve. Here is a simplified diagram:



The Circuit solver varies the *Gain* to find the optimum equalization (maximum eye height) for the channel. The list of gain values can contain one value or several (-infinity dB to 0 dB). If no list of *Gain* values is supplied, the Circuit solver calculates the optimum CTLE gain between the corresponding linear values, from 0 (equivalent to -infinity dB) to 1 (equivalent to 0 dB). For the legacy transfer function formulas, there are only two poles and one zero. Let:

$$\omega_1 = 2\pi f_1$$

$$\omega_2 = 2\pi f_2$$

$$\omega_z = 2\pi f_z$$

$$A_{DC} = 10^{(DCGain/20)}$$

The legacy transfer function with one pole frequency is:

$$H(s) = \frac{s + \omega_1 \cdot A_{DC}}{(s + \omega_1)}$$

(1)

The legacy transfer function with two poles but no zero is:

$$H(s) = \omega_2 \cdot \frac{s + \omega_1 \cdot A_{DC}}{(s + \omega_1) \cdot (s + \omega_2)}$$

(2)

The legacy transfer function with two poles and a zero is:

$$H(s) = \frac{\omega_1 \cdot \omega_2}{\omega_z} \cdot \frac{A_{DC} \cdot (s + \omega_z)}{(s + \omega_1) \cdot (s + \omega_2)}$$

(3)

CTLE Transfer Function from a File

Alternatively, define the CTLE transfer function over a appropriate range of frequencies, then put the tabulated data in a file. The format of the CTLE data file is described later in this topic.

Nexxim reads the data on the file specified by the **file** parameter in the Eye probe, and fits a rational function to the data. The **reltol** option allows you to specify the fitting error. The resulting rational function is applied in the simulation (QuickEye, VerifEye, Transient, or LNA).

The transfer function with a rational function has the following form. Let:

$P = \text{Number of poles } Z = \text{Number of zeros, Z < P } \omega_p = 2\pi f_p \ \omega_z = 2\pi f_z$

$$H(s) = \frac{\prod_{j=1}^{Z}(s+\omega_{z_j})}{\prod_{i=1}^{P}(s+\omega_{p_i})}$$

(4)

A rational function generally provides the best fit to CTLE data: the CTLE transfer function data may represent a lumped RC implementation with known poles and/or zeros. If the data is from a rational function originally, a pure rational function finds the poles and zeros exactly.

The data file can contain multiple transfer functions for the same set of frequencies, controlled by the Eye probe **tf_num** parameter. Transfer functions in the file are numbered from 1 to N. By default (**tf_num** =0), Nexxim tries each transfer function in turn, finally using the one that produces the maximum eye height. Setting parameter **tf_num** to an integer from 1 to N directs Nexxim to use only that one transfer function in the analysis. If the **tf_num** is less than zero or greater than the number of transfer functions in the file, an error occurs.

Here is a summary of the format for the CTLE transfer function data. The format is similar to the Touchstone 2.0 file format.

- The file must be in text format, and the file name must have a **.ctle** extension.
- A comment is any text after an exclamation point (!). Whitespace and blank lines are ignored.
- A line beginning with **[Complex format]***may* be included to specify the data format. Options are **RI** (Real/Imaginary) and **MA** (Magnitude/Angle in degrees). If no **[Complex format]** line is present in the file, **RI** is the default interpretation. All transfer functions in a given file must use the same data format (**RI or MA**).
- A line beginning with **[Number of frequencies]** must be present to specify the number of frequency points in the file.
- A line beginning with **[Number of transfer functions]** must be present to specify the number of transfer functions given for each frequency point in the file.
- A line with **[Data]** alone introduces the lines of data.
- Each line of data has the frequency in Hz as the first entry. The frequencies must be in ascending sequence, but do not need to be evenly spaced. There must be no repeated frequencies (same convention as Touchstone 2.0).

- After each frequency, the transfer function values appear, separated by commas, tabs, or spaces. For Real/Imaginary and Magnitude/Angle data formats, each transfer function uses two entries.

> **Note:**
>
> To guarantee that high frequencies remain unique after rounding, use high precision (e.g., for frequencies in the 10GHz range, a precision of 10 digits is recommended).
>
> For Magnitude/Angle data, angles are in *degrees* (as in Touchstone 2.0).

Here is an example:

```
! CTLE Transfer Function Data in Real/Imaginary Format
!
[Complex format] RI
[Number of frequencies] 30
[Number of transfer functions] 2
!

! frequency,r:1,i:1,r:2,i:2

[Data]

5.00E+07,1.3708,-0.0096,1.3708,-0.00849
1.00E+08,1.3708,-0.0196,1.3708,-0.0175
1.50E+08,1.3708,-0.0297,1.3708,-0.0265
2.00E+08,1.3708,-0.0397,1.3708,-0.0355
2.50E+08,1.3708,-0.0498,1.3708,-0.0445
3.00E+08,1.3708,-0.0599,1.3708,-0.0535
3.50E+08,1.3708,-0.0700,1.3708,-0.0626
4.00E+08,1.3708,-0.0802,1.3708,-0.0717
4.50E+08,1.3708,-0.0903,1.3708,-0.0809
5.00E+08,1.3708,-0.1002,1.3708,-0.0901
5.50E+08,1.3708,-0.1112,1.3708,-0.0992
6.00E+08,1.3708,-0.1212,1.3708,-0.1092
6.50E+08,1.3708,-0.1312,1.3708,-0.1182
7.00E+08,1.3708,-0.1422,1.3708,-0.1272
7.50E+08,1.3708,-0.1522,1.3708,-0.1372
8.00E+08,1.3708,-0.1622,1.3708,-0.1472
```

```
8.50E+08,1.3708,-0.1732,1.3808,-0.1562
9.00E+08,1.3708,-0.1842,1.3808,-0.1662
9.50E+08,1.3708,-0.1942,1.3808,-0.1762
1.00E+09,1.3708,-0.2052,1.3808,-0.1862
1.05E+09,1.3708,-0.2162,1.3808,-0.1962
1.10E+09,1.3708,-0.2272,1.3808,-0.2072
1.15E+09,1.3708,-0.2382,1.3808,-0.2172
1.20E+09,1.3708,-0.2492,1.3908,-0.2282
1.25E+09,1.3708,-0.2602,1.3908,-0.2382
1.30E+09,1.3708,-0.2722,1.3908,-0.2492
1.35E+09,1.3708,-0.2832,1.3908,-0.2602
1.40E+09,1.3708,-0.2942,1.3908,-0.2722
1.45E+09,1.3708,-0.3062,1.3908,-0.2832
1.50E+09,1.3708,-0.3182,1.3908,-0.2952
```

Specifying a Generic Equalizer Transfer Function, Pole-Zero Format

A generic equalizer transfer function can be specified using either the pole-zero format or the polynomial format.

With the pole-zero format, you specify the poles and zero frequencies for the generic equalizer rational function, as well as the allowable DC Gain values and a value for the high-frequency peak gain (maximum amplitude of the high-pass filter). As with the legacy CTLE, Nexxim varies the DC Gain to find the value that gives the maximum eye height given the other parameters.

$$P = \text{Number of poles}\ Z = \text{Number of zeros, Z < P}\ \omega_p = 2\pi f_p\ \omega_z = 2\pi f_z$$

$$A_{DC} = 10^{(DCGain/20)}\ A_{AC} = 10^{(ACGain/20)}$$

Case 1: The data includes one or more zero frequencies, Z>0. The rational function for this case has the form:

$$H(s) = A_{DC}\left[\frac{\prod_{j=1}^{Z}(s+\omega_{z_j})}{\prod_{i=1}^{P}(s+\omega_{p_i})}\right]$$

(5)

When P=2 and Z=1, Equation (5) reduces to the legacy case of Equation (3). This case generalizes the legacy formulation of Equation (3) from P=2 and Z=1 to any P>1 and any Z<P. This allows Nexxim to match industry-standard CTLE specifications such as VESA.

Case 2: The data does not have any zero frequencies explicitly specified, Z=0. The rational function has the form:

$$H(s) = A_{AC}\cdot\prod_{i=2}^{P}\omega_{p_i}\cdot\left[\frac{s+\frac{A_{DC}}{A_{AC}}\cdot\omega_{p_1}}{\prod_{i=1}^{P}(s+\omega_{p_i})}\right]$$

(6)

When P=2, Z=0, and A_{AC}=1, Equation (6) reduces to the legacy case of Equation (2). This case generalizes the legacy formulation of Equation (2) from P=2, Z=0, and A_{AC}=1 to any P>1 and any A_{AC}<1. This allows Nexxim to match industry-standard CTLE specifications such as USB3.1 Gen2.

Note that at DC (s=0, A_{AC}=1), $H(s) = A_{DC}$ as expected.

Specifying a Generic Equalizer Transfer Function, Polynomial Format

A generic equalizer transfer function can be specified using either the pole-zero format or the polynomial format. Standards such as MIPI use a polynomial format for the transfer function. Use the GUI entry fields to enter the coefficients for the numerator and denominator of the function.

For a polynomial with N+1 degrees in the numerator (coefficients a_0, a_1, ... $a_{N)}$, and M+1 degrees in the denominator (coefficients b_0, b_1, ... b_M), M >= N, the transfer function is:

$$H(s) = \frac{\sum_{i=0}^{N} a_i s^i}{\sum_{j=0}^{M} b_j s^j}$$

Transmit Jitter in QuickEye and VerifEye

Transmit jitter is variation in the location of edges. QuickEye and VerifEye can add the effects of transmit jitter, in combinations of Random (Gaussian), Uniform, Periodic, and Custom (user-defined) jitter.

QuickEye Jitter Calculation

QuickEye employs a Monte-Carlo method: each transition is displaced a random amount on the nominal time using a probability distribution that combines all specified Gaussian, Uniform, Periodic, and Custom jitter distributions. The Monte Carlo approach requires significant simulation time, since there is a different set of jittered transitions for each bit in the original stream.

VerifEye Jitter Calculation

The VerifEye method involves calculating the probability distribution function (PDF) for receiver voltage. VerifEye calculates a second probability distribution function that combines all specified Random (RJ), Uniform (UJ), Periodic (PJ), and Custom (user-defined, CJ) jitter distributions. The total jitter PDF (p_{TJ}) is the convolution of the individual PDFs:

$$p_{TJ}(tj) = p_{RJ}(rj) * p_{UJ}(uj) * p_{PJ}(pj) * p_{CJ}(cj)$$

VerifEye translates the total jitter probability distribution function to the corresponding voltage PDF at each sample time. The voltage PDF has the same distribution as the jitter PDF. The mean of the voltage PDF is offset on the mean of the jitter distribution (always 0) by the value of the rising or falling step response. The width of the voltage PDF is the width of the jitter PDF scaled by the rate of change of the response at the sample time.

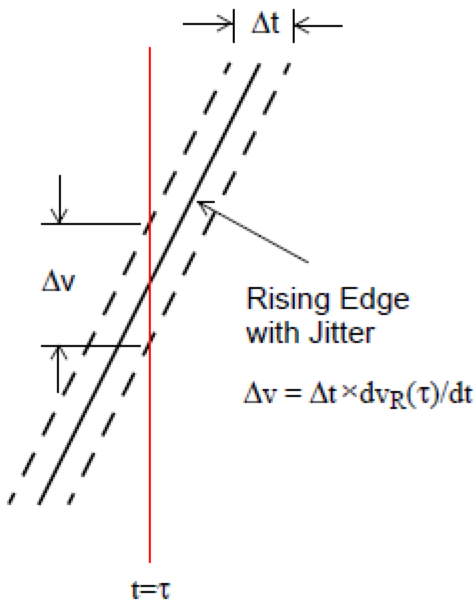


Figure 1. Transmit Jitter PDF Translated to Voltage PDF.

Figure 1 diagrams an arbitrary segment of a rising waveform, sampled at time τ. The dotted lines indicate the maximum displacement of the waveform on the time axis due to transmit jitter, Δt. The exact shape of the jitter distribution is not shown. The corresponding spread in the voltage PDF due to jitter is shown as Δv. The slope of the rising wave at sample time τ is:

$$\frac{\mathrm{d}}{\mathrm{d}t}v_R(\tau)$$

The spread in the voltage PDF due to the spread in the jitter PDF is:

$$\Delta v = \Delta t \times \frac{\mathrm{d}}{\mathrm{d}t}v_R(\tau)$$

The next topics provide details on the PDFs for random (Gaussian), uniform, periodic, and custom (user-defined) transmit jitter.

Random Transmit Jitter

The Gaussian probability distribution function for variable *x* with mean μ and standard deviation σ is:

$$\frac{1}{\sqrt{2\pi\sigma^2}}e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

For QuickEye and VerifEye, the variable is the time value of the random jitter (*t*), the mean μ is zero, and the standard deviation σ is set to Eye Source entry TXRJ. The resulting PDF for random jitter is:

$$p_{RJ}(rj) = \frac{e^{-(rj^2/2\times TXRJ^2)}}{\sqrt{2\pi\times TXRJ^2}}$$

This PDF has the familiar normal curve, as shown in Figure 2:

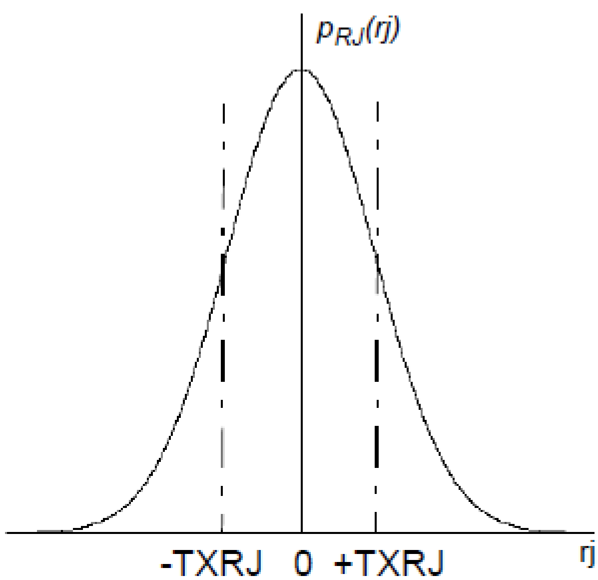


Figure 2. Random Jitter PDF.

Uniform Transmit Jitter

Uniform random jitter represents a distribution of offsets in time on the nominal edge location. The uniform distribution is centered on time *uj*=0 with bounds -0.5*TXUJ to +0.5*TXUJ, where TXUJ is the entry on the Eye Source (Figure 3).

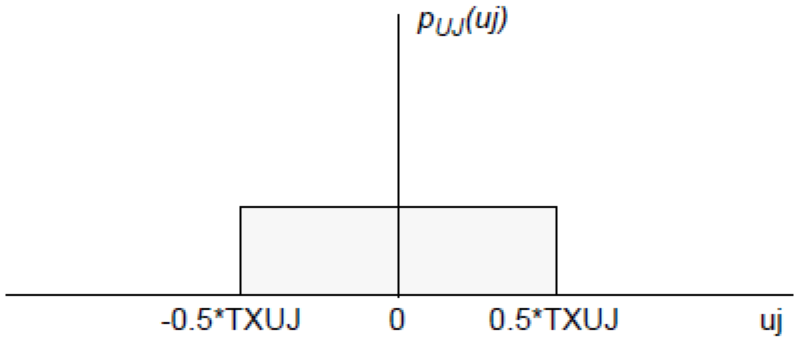


Figure 3. Uniform Jitter PDF.

Uniform jitter is not common by itself, but can be used in combination with the other jitter distributions to more closely match measured values.

Periodic Transmit Jitter

Periodic jitter simulates the effect of variations in voltage such as those produced by sinusoidal noise. The periodic variation in voltage translates into a periodic variation in the location of the edges of the signals generated.

In the QuickEye and VerifEye implementation of periodic jitter, the frequency of the periodicity is ignored. The noise frequency is assumed to be uncorrelated to the clock frequency, and the phase of the periodic noise is basically a uniform random variable. The noise voltage is the sine of this variable, giving the following probability distribution (TXPJ is the Eye Source entry):

$$p_{PJ}(pj) = \frac{1}{\pi\sqrt{1-\left(\frac{pj}{TXPJ}\right)^2}}$$

for |*pj*| < TXPJ, zero otherwise.

Figure 4 shows the shape of the periodic jitter PDF. This probability distribution has peaks at |*pj*| = TXPJ, and is sometimes approximated as a bimodal or dual-Dirac distribution (but not by QuickEye or VerifEye).

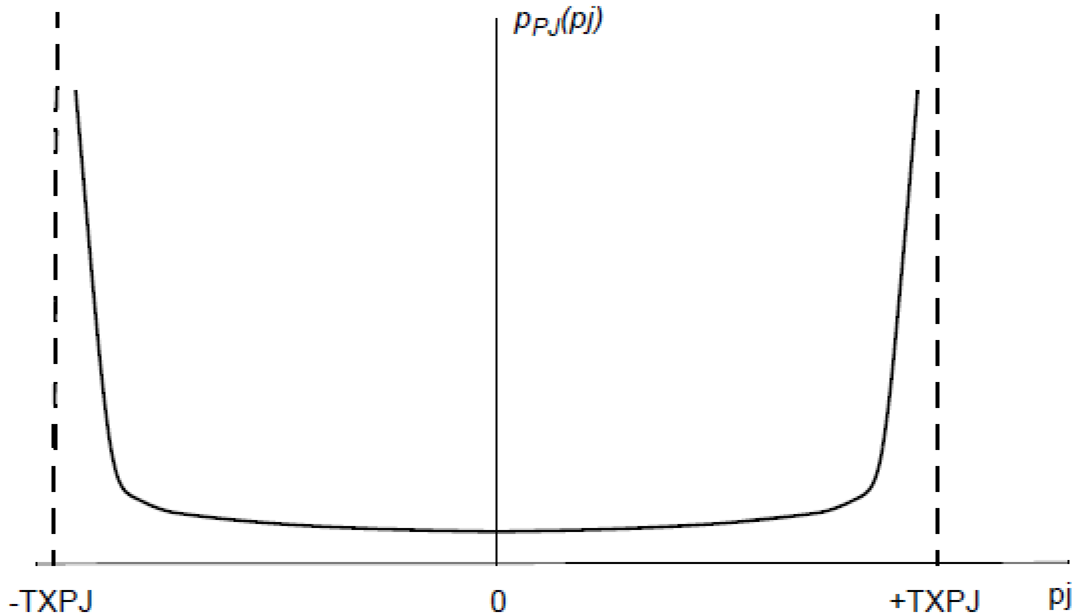


Figure 4. Periodic Jitter PDF

Custom or User-Defined Transmit Jitter

The user can specify a probability density function (PDF) for transmit jitter, to be combined with any other forms of jitter specified in the Eye source for both QuickEye and VerifEye. The PDF data is read on the file indicated by the **TXCJ** parameter on the Eye source.

The data in the jitter file are in two columns. The first column is the time in seconds. The time values may be positive or negative, and need not be centered around t=0. The Circuit solver automatically centers the mean value of the data at t=0 before doing the jitter calculation. The second column is the PDF value for each time. The PDF values must be greater than or equal to zero.

The Circuit solver automatically scales the PDF data such that the area underneath is unity, so no manual scaling is required.

The Circuit solver performs linear interpolation between the data points. Best results are obtained when the number of data points is large enough to minimize discontinuities of slope. The time axis should extend far enough to allow the PDF to drop to negligible values at both ends.

Duty Cycle Distortion

Duty cycle distortion or DCD is a systematic variation in the duty cycle of the transmitter clock, the proportion of time that the clock is high. By default, the duty cycle is 50% (upper waveform in

Figure 1). The DCD parameter on the Eye source specifies the duty cycle distortion as a fraction of the unit interval (UI). The transmitted bits can be numbered from 1 to N; bit numbers 1, 3, 5, etc. are odd bit positions, numbers 2, 4, 6, etc, are even bit positions. (Note that the odd and even positions do not in general correspond to particular bit values (0 or 1), except when the signal is a pure clock.)

For both QuickEye and VerifEye, the Circuit solver reads the DCD parameter (DCD as a fraction of the UI) or the DCD_TIME parameter (DCD as an absolute time in seconds.) A DCD_TIME parameter value is converted to DCD as a fraction of the UI. The solver computes the offset value (0.5×DCD×UI). The offset value is then subtracted from all even bit positions and added to all odd bit positions (lower waveform in Figure 1).

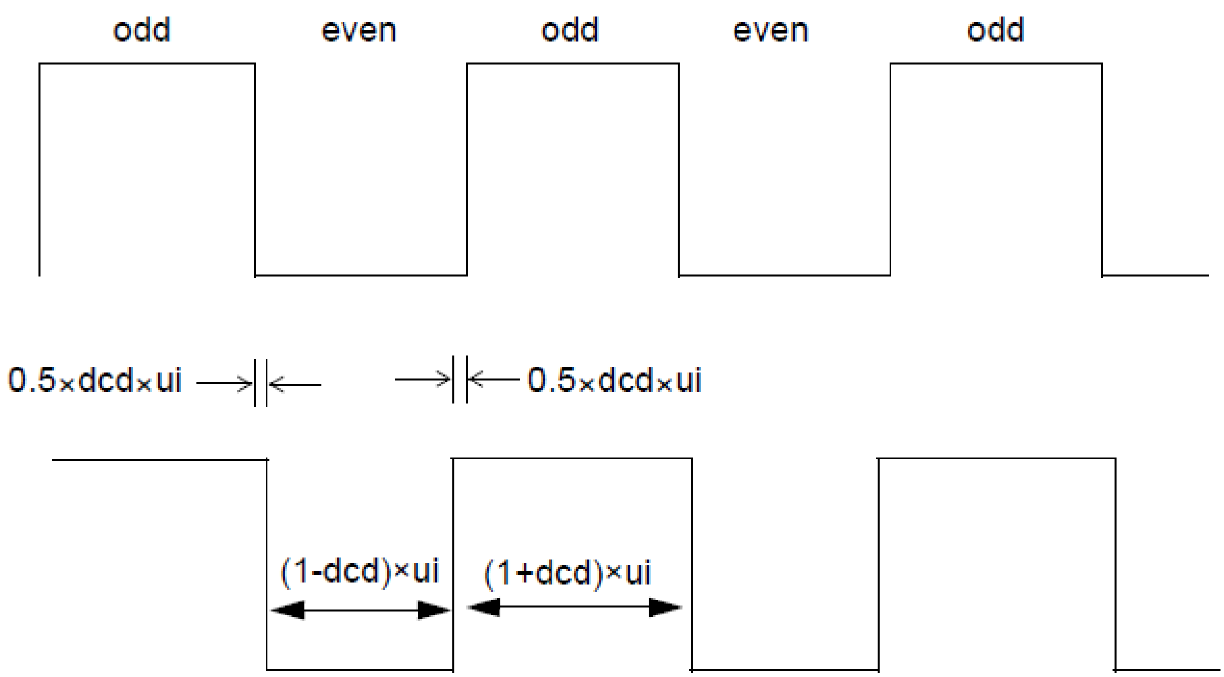


Figure 1. Clock Waveforms. The Lower Trace shows Duty Cycle Distortion.

Peak Distortion Analysis

Peak distortion analysis (PDA) calculates the worst case bit pattern on the step responses of the channel, for both VerifEye and Quick Eye analyses. Peak distortion analysis is calculated for each combination of Eye Source and Eye Probe.

For Quick Eye analyses, the worst case bit patterns can be pushed to all Eye Sources and repeated by setting the **PD_COUNT** parameter in the Quick Eye analysis setup. When the PDA data is enabled, any bit patterns set in the Eye Sources are ignored. Figure 1 shows an example of the eye diagram from Quick Eye resulting from one iteration of the worst-case bits (no repeats).

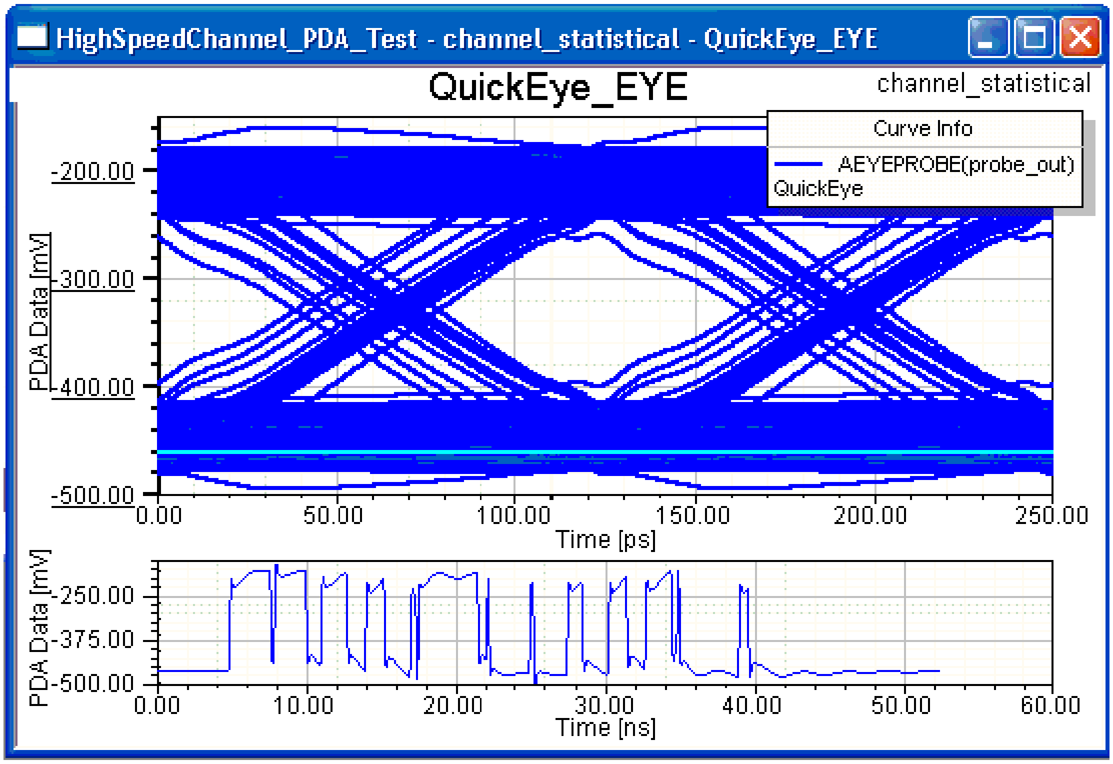


Figure 1. QuickEye Result using Worst-Case Bit Pattern from Peak Distortion Analysis

For both QuickEye and VerifEye analyses, the Peak Distortion domain data (bit list, error contribution) can be reported as a standard 2D plot or in a data table. Figures 2 and 3 show the 2D plot and data table for the bit pattern above. The data table shows just a portion of the data.

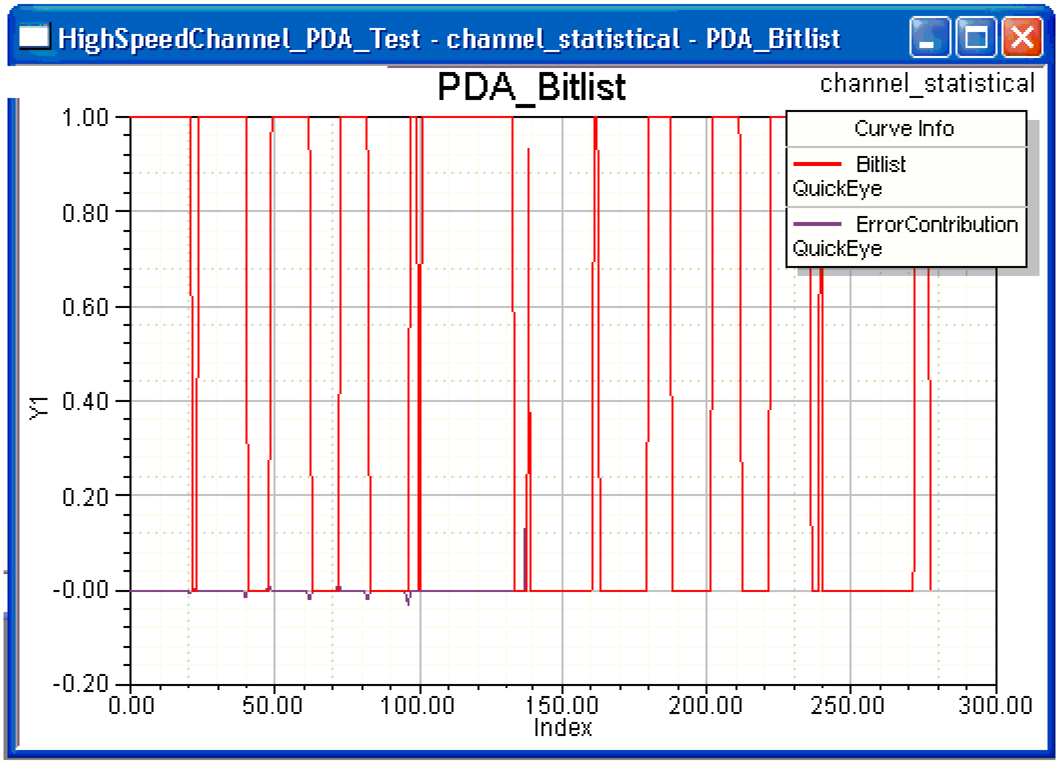
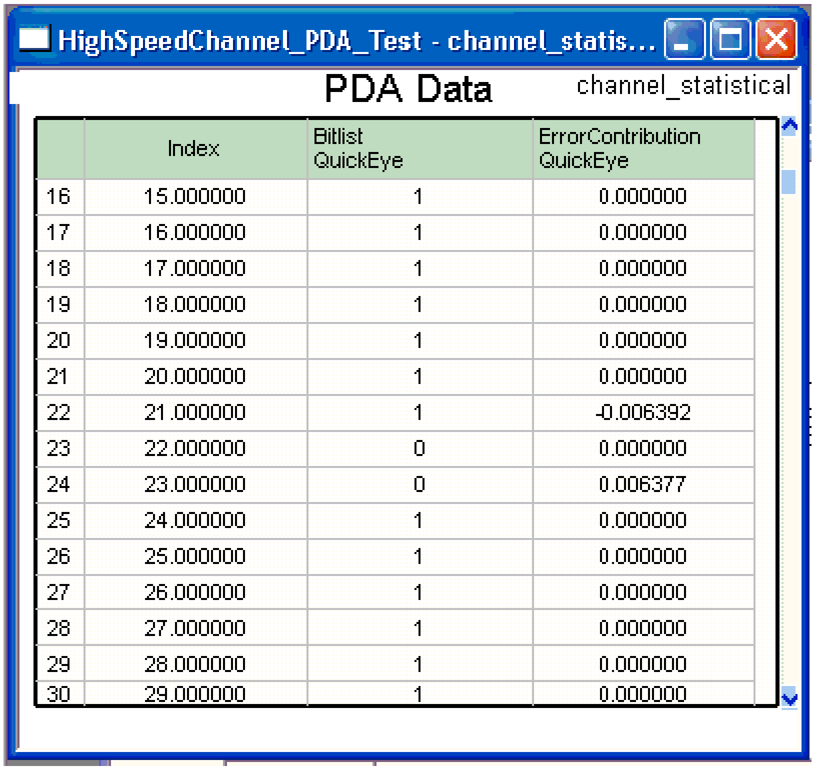


Figure 2. 2D Rectangular Plot of Worst-Case Bit Pattern from Peak Distortion Analysis



HighSpeedChannel_PDA_Test - channel_statis...

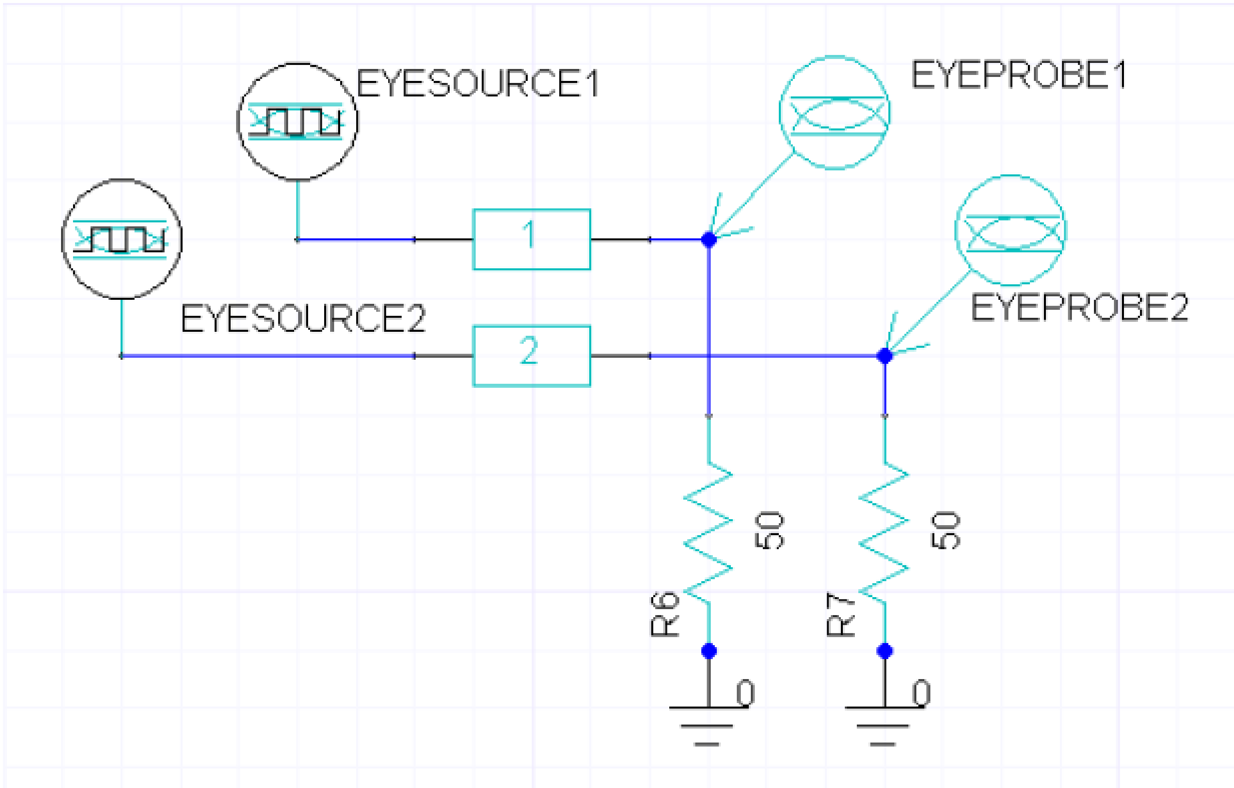
PDA Data channel_statistical

	Index	Bitlist QuickEye	ErrorContribution QuickEye
16	15.000000	1	0.000000
17	16.000000	1	0.000000
18	17.000000	1	0.000000
19	18.000000	1	0.000000
20	19.000000	1	0.000000
21	20.000000	1	0.000000
22	21.000000	1	-0.006392
23	22.000000	0	0.000000
24	23.000000	0	0.006377
25	24.000000	1	0.000000
26	25.000000	1	0.000000
27	26.000000	1	0.000000
28	27.000000	1	0.000000
29	28.000000	1	0.000000
30	29.000000	1	0.000000

Figure 3. Data Table for Worst-Case Bit Pattern from Peak Distortion Analysis

NOTE: Peak Distortion Analysis (PDA) considers any 8b10b or 64b66b encoding when generating the worst-case bit pattern.

The PROBE parameter in the PDA setup determines the target for worst-case analysis. Here is a reference diagram:



When PD_COUNT=-1, PDA calculates the worst-case bit pattern for EYESOURCE1 to EYEPROBE1 and from EYESOURCE2 to EYEPROBE2, but no bits are sent.

When PD_COUNT=0 or greater, the target PROBE must be specified. If PROBE=EYEPROBE1 in the example above, PDA calculates the worst case bit patterns from EYESOURCE1 and EYESOURCE2 to EYEPROBE1, and those bit patterns are sent to the analysis.

Encoding 8b10b 64b66b 128b130b 128b132b

8b10b, 64b66b, 128b130b, and 128b132b encoding are common methods for reducing errors caused by long strings of 1s with embedded isolated 0s, and long strings of 0s with embedded isolated 1s. By default, no encoding is performed (Eye Source option **do_encoding** =0). The VerifEye, QuickEye, AMI, or Transient Source can be enabled to do encoding on the transmitted data (**do_encoding** =1 for 8b10b, 2 for 64b66b, 3 for 128b130b, and 4 for 128b132b).

Notes:

1. Option **do_encoding** is valid only when **modulation=NRZ** on the Eye Source. **do_ encoding** is ignored when **modulation=PAM4**.
2. Peak Distortion Analysis (PDA) considers the encoding when generating the worst-case bit pattern.

Eye Source Bit Data

The Eye source can generate bit data for both QuickEye and transient analyses, using one of four methods.

- The source can link to a file that contains the data, using the BITFILE parameter. See QuickEye Bit Files for further information.
- The source can contain the sequence of 0 and 1 bits, using the BITLIST parameter. See QuickEye Bit Lists for further information.
- The source can generate a sequence of random bits, controlled by the RANDOM_BIT_COUNT and RANDOM_SEED parameters. See QuickEye Random Bit Data for further information.
- The source can generate a pseudorandom sequence of bit patterns, controlled by the PRBS_NO, PRBS_COUNT, PRBS_SEED, and PRBS_INVERT parameters. See "QuickEye Pseudorandom Bit Data" on page 25-316 for further information.

If both the BITFILE and the BITLIST parameters are set, the BITFILE is used.

Any of these bit sources can be repeated using the REPEAT_COUNT parameter. The default is one time through the list (REPEAT_COUNT=0).

- The bit stream can be the worst-case pattern from Peak Distortion Analysis (PDA). When the PD_COUNT parameter is set to 1 or higher in the Quick Eye analysis setup, the Eye Source uses the worst-case bit pattern with REPEAT_COUNT set to (PD_COUNT). See Peak Distortion Analysis for further information.

QuickEye Bit Files

The BITFILE parameter on the Eye Source contains the path to the file containing the test bits. The bit file for QuickEye can be a simple text file (**.txt** extension) containing a list of 0s and 1s without separators.

A QuickEye bit file can also use a Coded Bit Pattern (**.cbp** extension). A coded bit pattern file has two entries on each line: a *tag* and the number of *repeats*. The *tag* entry specifies the type of bit pattern to generate:

Tag	Bit Representation
Zero	0
One	1
Lonely_0	101
Lonely_1	010

Tag	Bit Representation
Random	Random 0s and 1s

The *repeats* entry tells how many instances of the given bit representation to generate. A line beginning with an exclamation point is a comment.

Here is an example of a Coded Bit Pattern file:

```
! example coded bit pattern file
zero 10
random 100
one 1
zero 42
lonely_1 3
```

QuickEye Bit Lists

The BITLIST parameter on the Eye Source specifies the sequence of 0s and 1s without separators. The bit list is preceded in the netlist by a pound sign (**#**) to indicate that it is binary data.

QuickEye Random Bit Data

The RANDOM_BIT_COUNT parameter on the Eye Source specifies the number of random bits to generate.

The RANDOM_SEED parameter controls the repeatability and uniqueness of the random sequence. The default RANDOM_SEED is 1000.

Repeatability. A positive RANDOM_SEED value generates the same sequence of random bits on every simulation.

Uniqueness. When two or more Eye Sources use the same positive RANDOM_SEED value, the random bit sequences is the same for all such sources, and each source generates the same sequence of bits on every run.

Negative values of RANDOM_SEED guarantee uniqueness among multiple random bit sources, with or without repeatability.

- Multiple sources with RANDOM_SEED set to **-1** receive unique seeds 1000, 1001, ... 1000+(N-1), and these seeds stay the same on every simulation. (The order in which multiple sources receive seeds is indeterminate, but is the same on every simulation.)

- Multiple sources with RANDOM_SEED set to **-2** receive unique seeds derived on the system clock, and these seeds is different on every simulation.

QuickEye Pseudorandom Bit Data

Setting the PRBS_NO parameter to an integer value between 2 and 31 specifies the length of the pseudorandom bit patterns. The Circuit solver randomly selects from bit pattens of the given length to generate the total number of bits requested.

The PRBS_BITLENGTH parameter specifies the total number of bits to be generated. The default is (2^{PRBS_NO} -1).

The PRBS_SEED parameter controls the starting value for the pseudorandom bit sequence. The default is 0 (zero).

The PRBS_INVERT parameter, when set to 1, inverts the bit values as they are generated. The default is 0 (no inversion).

QE and VE Crosstalk Analysis

For crosstalk analysis, the design incorporates multiple combinations of Eye Sources and Eye Probes. The typical crosstalk situation has one victim channel (the one whose integrity is being assessed) and one or more aggressor channels creating the crosstalk. Each channel has an Eye Source on the transmit end and an Eye Probe at the receive end; these two components link to each other via parameters. (A channel may have multiple Eye Probes, but only the probe at the receiver should be linked to the source.) To make the analysis meaningful, the design should use coupled transmission lines.

The basic idea of crosstalk analysis is to run the victim channel alone to obtain the victim's eye diagram with no crosstalk. Then run both the victim and the aggressor channels, compute the victim's eye diagram again, and compare the two eyes to assess the effect of the crosstalk.

Quick Eye and VerifEye treat correlation differently, affecting the treatment of crosstalk.

QuickEye is similar to transient in the way it deals with correlation. The algorithm inherently takes into consideration the time behavior at the output due to each source given by the step response. When convolving the step responses from various sources, the factors that contribute to correlation (bit sequence, phase delays, bps/UI) are taken into consideration.

By default, the VerifEye statistical methodology computes correlated crosstalk, but uncorrelated crosstalk can also be computed.

Uncorrelated Crosstalk

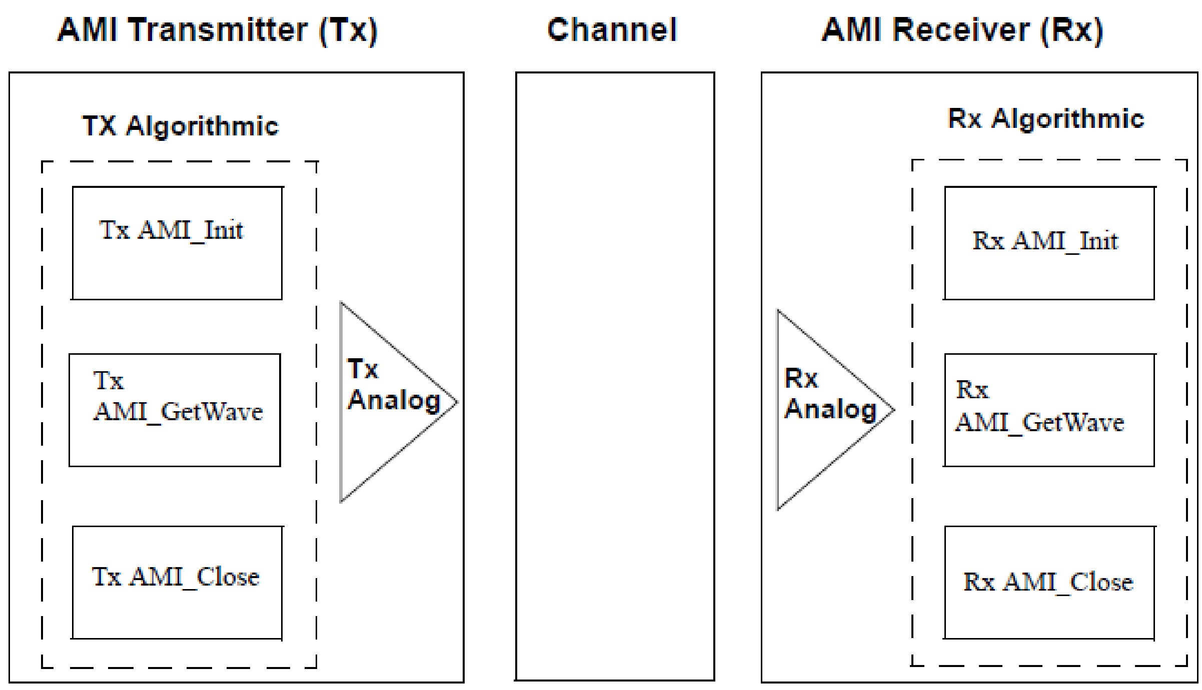
If the two lines (i.e., the victim and the aggressor channels) have the same UI, the crosstalk pulse occurs in the same place in each bit time. QE shows the pulse, but VE does not. However, if the two UIs differ, the crosstalk pulse is out of phase with the intended signal, and its effect shows up equally at any location in the eye. This circumstance is “uncorrelated crosstalk". To compute uncorrelated crosstalk, uncheck the **Calculated correlated crosstalk** box in the **VerifEye (Statistical Eye) Analysis** window, or set the solver option **eye.verifeye_correlated_crosstalk** to zero. Refer to "Set Up the VerifEye Analysis" on page 11-57 and "VerifEye and Quick Eye Analysis Options Reference" on page 25-180.

AMI Analysis Technical Notes

The IBIS Algorithmic Modeling Interface simulates user-specified transmitter and receiver models, furnished as C-interface compiled libraries, over a linear channel. The AMI Specification is published in the IBIS Specification (See AMI Analysis References).

AMI Analysis Components

Here is a diagram showing the internal structure of the AMI transmitter and receiver.

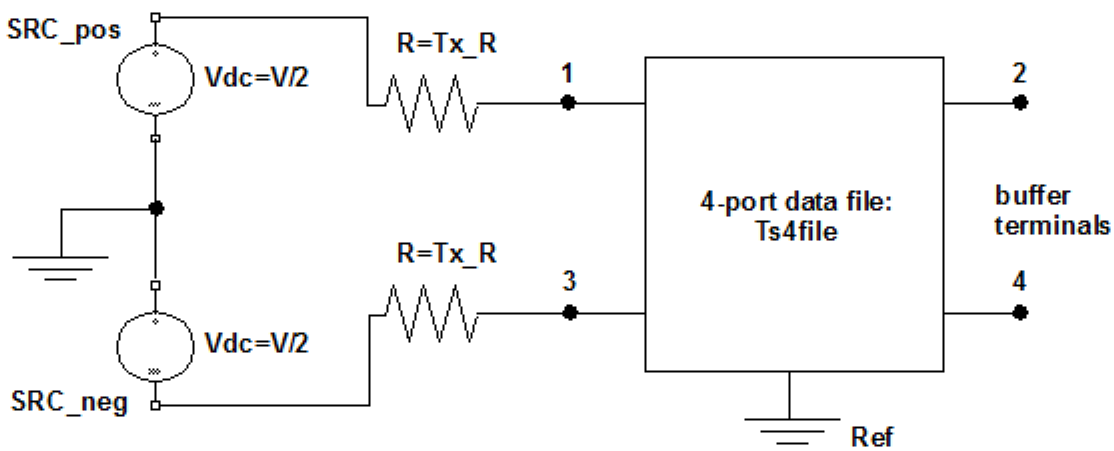


The following topics describe the options for the AMI transmitter, the Channel Impulse Response, and the AMI Receiver.

The AMI Transmitter Model

The AMI transmitter consists of an Analog output model and three Algorithmic processing functions, Tx **AMI_Init**, Tx **AMI_GetWave**, and Tx **AMI_Close** that model equalization behavior and are provided in a .dll or .so. The IBIS specification allows the transmitter to omit the TX **AMI_GetWave** function.

The Analog output model can be specified using traditional IBIS methods (including linearized I/V tables and voltage ramps), or with 4-port frequency-dependent data from a Touchstone-format file containing the transmitter behavior.



When the transmitter model uses 4-port data, the reserved parameters **Ts4file** (name of Touchstone file), **TX_V** (transmitter voltage swing), and **Tx_R** (series resistances) control the transmitter operation.

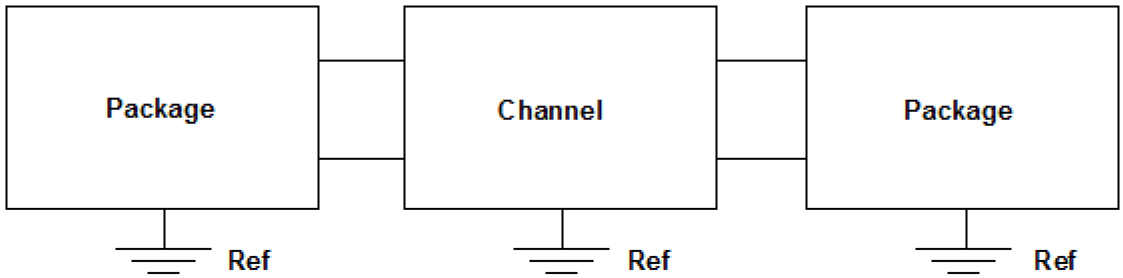
- The voltages of the voltage sources correspond to **V=Tx_V** for logic level 1 and **V=-Tx_V** for logic level 0.
- The series resistors have resistance **R=Tx_R**.

See "AMI Ts4 Analog Buffer Model Parameters " on page 11-150"AMI Ts4 Analog Buffer Model Parameters " on page 11-150 for details on these parameters.

The Channel Impulse Response

AMI Analysis assumes a linear channel. The channel is characterized by its impulse response function.

When 4-port Touchstone files are referenced for the AMI transmitter and receiver, the channel is assumed to include the package behavior in the impulse response:

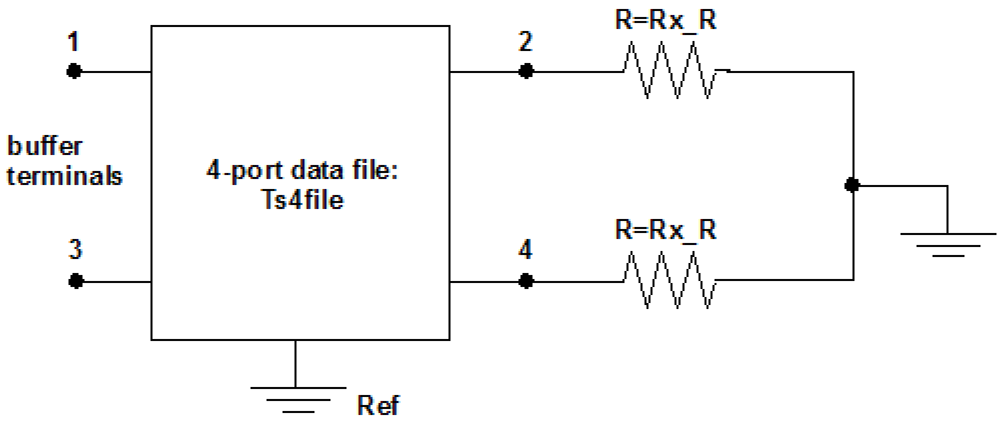


The AMI Receiver Model

The AMI receiver consists of an Analog input model and three Algorithmic processing functions that model equalization behavior and are provided in a .dll or .so: Rx **AMI_Init**, Rx **AMI_GetWave**, and Rx **AMI_Close**. The IBIS specification allows the receiver to omit the Rx **AMI_GetWave** function, unless it is part of an AMI retimer link (See AMI Repeaters).

The RX **AMI_GetWave** function in the receiver model can generate clock times internally, and return the clock ticks as parameters to the solver for post-processing. For AMI channel simulation, the results of clock tick sampling can be observed in the Statistical Eye Plot for the EyeAfterProbe quantity. For an AMI Retimer link, the clock times are used to generate a new digital waveform (See AMI Repeaters).

The Analog Receiver model can be specified using traditional IBIS methods or with 4-port frequency-dependent data from a Touchstone-format file containing the receiver behavior.



When the receiver model uses 4-port data, the reserved parameters **Ts4file** (name of Touchstone file), and **Rx_R** (terminating resistances) control the Receiver analog section operation. The terminating resistors have resistance **R=Rx_R**. See "AMI Ts4 Analog Buffer Model Parameters " on page 11-150 for details on these parameters.

AMI Time Domain Simulation

Ansys supports both Time Domain and Statistical types of AMI simulation as described in the IBIS specification. In **Electronics Desktop**, AMI Time Domain simulation is called AMI Analysis which models signal propagation through a transmitter, interconnect, and receiver. The AMI transmitter generates a waveform representing a stream of bit values and sends the waveform through the transmitter model to the channel. The wave on the transmitter is convolved with the impulse response of the channel to generate the input to the receiver.

AMI analysis is more resource intense than AMI Statistical simulation, which is called VerifEye Analysis in **Electronics Desktop**. VerifEye depends solely on statistical analysis based on impulse responses (IR) modified by the AMI models, which allows them to only return linear approximations. In contrast, AMI analysis allows AMI models to provide nonlinear, time varying algorithms that allow for more flexible simulations.

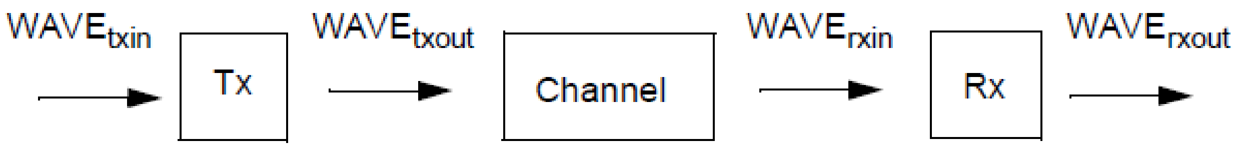
AMI Time Domain Simulation Phases

AMI Analysis proceeds in three phases: Initialization, Data Generation, and Close. The transmitter (Tx) and receiver (Rx) models must contain the functions **AMI_Init()** and **AMI_Close ()**. The transmitter and receiver models may contain **AMI_GetWave()** functions. Each of these functions is called during the appropriate phase.

At Initialization, the files are opened and checked for the presence or absence of the Tx **AMI_ GetWave()** and Rx **AMI_GetWave()** functions. The simulator calls the **AMI_Init()** functions to compute the impulse responses (IRs). The **AMI_Init()** functions are only called during the Initialization phase. The Intermediate and Final IRs are not used during AMI simulation if the Tx and Rx **AMI_GetWave()** functions exists, respectively.

The presence of the **AMI_GetWave()** functions determines the sequence of events during the Data Generation phase. If the AMI Tx or Rx model contains an **AMI_GetWave()** function, the **AMI_GetWave()** function processes the waveform. When no Tx **AMI_Getwave()** function is supplied, the simulator convolves the initial waveform with the Intermediate IR. The simulator then calls the Rx **AMI_GetWave()** function to equalize the effects of the channel on the modified waveform and determine the final waveform ($WAVE_{rxout}$). Because the transmitter has no **AMI_ Getwave()** function, the simulator uses deconvolution to determine any effects on the transmitter. The simulator uses these effects to create a pseudo-GetWave function which is then applied to the initial waveform to determine what comes out of the transmitter. The receiver's input is independent on the results of the pseudo-GetWave function.

All AMI analyses produce the same overall sequence of waveforms, all of which may be displayed as reports:



- The **Initial Wave**: The Piecewise Linear waveform generated by the bit source in the AMI transmitter and passed into the AMI transmitter algorithmic model ($WAVE_{txin}$ in the diagram).
- The **Wave After Transmitter**: The waveform that is output on the AMI transmitter model ($WAVE_{txout}$).
- The **Wave After Channel**: The waveform output by the channel and passed into the AMI receiver model ($WAVE_{rxin}$).
- The **Wave After Probe**: The waveform output by the AMI receiver model ($WAVE_{rxout}$).

Eye diagrams can be created for each of these 4 steps. See AMI Analysis for details.

AMI GetWave Functions

In a typical AMI analysis, the application provides an **AMI_GetWave()** function in both transmitter and receiver models. A detailed description follows of what happens when both, one, or neither **AMI_GetWave()** function is provided. See the IBIS Specification for more information on the cases where one or both of these functions are not present. (See AMI Analysis References.)

At Initialization:

The simulator opens the files and notes whether there are Tx **AMI_GetWave()** and Rx **AMI_GetWave()** functions.

The simulator computes the initial impulse response (IR) of the main channel. This includes the Tx's analog output, the channel, and the Rx's analog front end. Signals are sampled using a constant timestep, which is set in the UI. The simulator also computes the IRs of any aggressor channels. At the end of this phase, the simulator has the Initial IR, Intermediate IR returned by the Tx **AMI_Init()**, and Final IR returned by the Rx **AMI_Init()**.

> **Note:** When the Tx and Rx **AMI_GetWave()** functions exist, they modify the waveform to account for their effects, and the Intermediate and Final IRs are not used in the simulation. They are stored for post-processing.

During Data Generation:

The simulator processes each block of transmit data in this order:

1. The AMI Source generates a user-specified sequence of bits using one of several methods.
2. The AMI transmitter converts the stream of bits into a Piecewise Linear (PWL) waveform. The PWL waveform includes any transmit jitter specified in the AMI Source. This Initial Wave, $WAVE_{txin}$, is the input to the AMI transmitter.
3. If there is a Tx **AMI_Getwave()** function:
   a. The simulator calls Tx **AMI_GetWave()** to modify $WAVE_{txin}$ and create $WAVE_{txout}$, the Wave After Transmitter.
   b. The simulator convolves $WAVE_{txout}$ with the Initial IR to create $WAVE_{rxin}$. Contributions from aggressors are added linearly.
4. If there is no Tx **AMI_GetWave()** function:
   a. The simulator creates $WAVE_{txout}$ using information extracted on the Intermediate IR:
      1. The Intermediate IR is deconvolved with the Initial IR to determine what effects a waveform experiences on the transmitter.
      2. The simulator uses the deconvolution results to create a pseudo-GetWave function for the transmitter.
      3. This pseudo-GetWave function is applied to $WAVE_{txin}$ to simulate $WAVE_{txout}$.
   b. The simulator convolves $WAVE_{txin}$ with the Intermediate IR to create $WAVE_{rxin}$. Contributions from aggressors are added linearly.

   > **Note:** $WAVE_{txout}$ is not involved in creating $WAVE_{rxin}$.

5. If there is a Rx **AMI_Getwave()** function, $WAVE_{rxin}$ is passed to the receiver model's **AMI_Getwave()** function to create $WAVE_{rxout}$.
6. If there is no Rx **AMI_Getwave()** function and:
   a. There is no Tx **AMI_GetWave()** and there are no crosstalk aggressors, the Final IR contains the combined effects of the entire system. The simulator convolves $WAVE_{txin}$ with the Final IR to produce $WAVE_{rxout}$.
   b. There is a Tx **AMI_GetWave()** and/or crosstalk contributions must be accumulated in the $WAVE_{rxin}$, then deconvolution is required. The simulator creates $WAVE_{rxout}$ using information extracted on the Final IR:
      1. The Final IR is deconvolved with the Intermediate IR to determine what effects a waveform experiences on the receiver.
      2. The simulator uses the deconvolution results to create a pseudo-GetWave function for the receiver.
      3. This pseudo-GetWave function is applied to $WAVE_{rxin}$ to simulate $WAVE_{rxout}$.
7. The simulator saves clock times that receiver model internally generates for post-processing.
8. The data for each block on the receiver model's **AMI_GetWave()** function are formatted for post-processing.

At the Close:

Once all the blocks have been processed, the Circuit simulator calls the **AMI_Close()** functions in the transmitter and receiver models to close files and release memory.

AMI Statistical Simulation

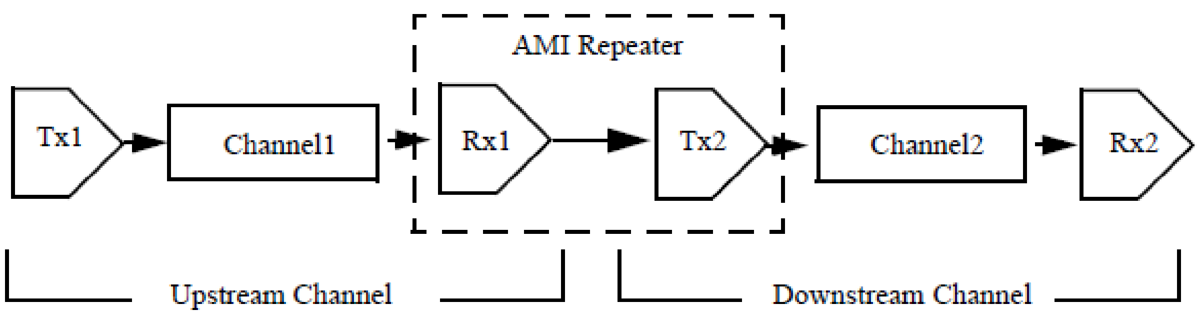
AMI statistical simulation calculates the impulse responses at various points in the channel. In **Electronics Desktop**, AMI statistical simulation is performed using the VerifEye simulator.

In AMI statistical simulation, the solver performs the following sequence:

1. Reads the IBIS model file and the AMI parameters file.
2. Calls the AMI_Init functions in the transmitter and receiver, and generates an impulse response which combines the channel impulse response with any filtering steps present in the AMI_Init functions.
3. Generates the impulse response at three stages: Initial (before any AMI_Init functions are applied), Intermediate, (after the transmitter AMI_Init is applied) and Final (after both AMI_Init functions are applied).

AMI Repeaters

A repeater is a device placed in the middle of a channel to compensate for loss. An AMI repeater model consists of a Receiver (Rx) model on one AMI channel setup and a Transmitter (Tx) model on a second channel setup. The pins on the Rx and Tx are paired to define the repeater. The repeater compensates for channel loss, reducing the bit error rate of the system.



In the preceding figure, the upstream channel of the repeater consists of:

- The upstream Transmitter model, both algorithmic and analog portions (Tx1).
- The physical channel (Channel1).
- The upstream Receiver model, both algorithmic and analog portions (Rx1).

A single impulse response characterizes the incoming (upstream) analog channel of the repeater, which includes the upstream Tx analog model, the physical channel, and the upstream Rx analog model.

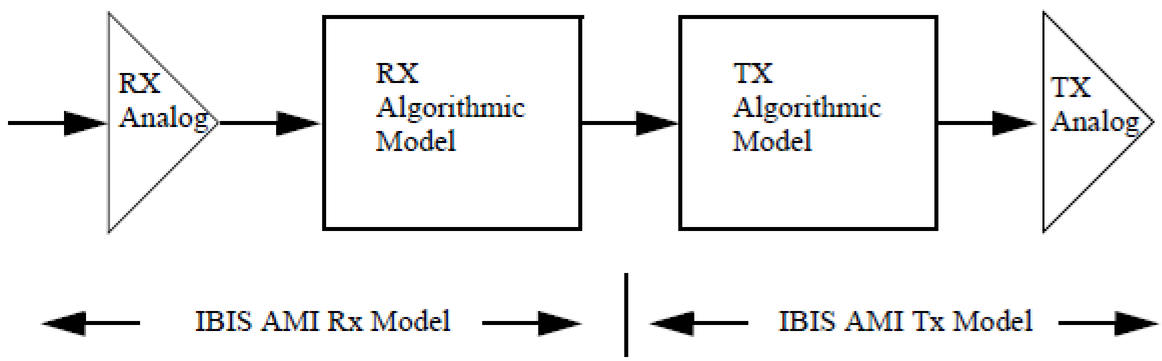
In the preceding figure, the downstream channel of the repeater consists of:

- The downstream Transmitter model, both algorithmic and analog portions (Tx2).
- The physical channel (Channel2).
- The upstream Receiver model, both algorithmic and analog portions (Rx2).

A second impulse response characterizes the outgoing (downstream) channel, including the downstream Tx analog model, the physical channel, and the downstream Rx analog model.

A repeater model is specified in a single IBIS library (.ibs) file that includes both the input and output models. The .ibs file identifies the pins on the Rx and Tx that are paired to define the repeater. See AMI Repeater Properties for information on setting the AMI Repeater properties.

Here are some details about the AMI repeater components:



The signal flow in a repeater model is from Rx Analog to Rx algorithmic to Tx algorithmic to Tx Analog.

The Rx and Tx algorithmic models represent equalizers, clock data recovery (CDR) circuits, or pre-emphasis inside the devices.

The analog part of the Rx model represents the input termination at the device input. Looking on the Rx analog portion, the Rx algorithmic block is assumed to have an infinite input impedance. The analog part of the Tx model represents the output impedance at the device output. Looking on the Tx analog portion, the Tx algorithmic block is assumed to have the output characteristics of an ideal voltage source. In AMI repeater simulations, both Rx and Tx analog models are assumed linear and time-invariant.

A Repeater can be either a Redriver or a Retimer.

AMI Redrivers

A redriver equalizes the upstream channel and retransmits it to the downstream channel. The output signal is continuously driven by the input signal and no retiming is performed when the redriver retransmits the signal. The redriver Rx passes its equalized output waveform directly to the Tx half of the repeater. The redriver boosts the signal, improving the quality of the signal at the link and at the output of the final receiver.

In a redriver, both algorithmic models can also implement the AMI_GetWave function.

See also:

AMI Retimers

AMI Time Domain Simulation with AMI Redriver or Retimer

AMI Statistical Simulation with AMI Redriver or Retimer

AMI Retimers

A retimer contains both a CDR and a generator for clock times in its AMI_GetWave function. The retimer CDR uses the clock times to sample the Rx equalized waveform and generate a digital stimulus for the Tx half of the repeater. The use of a retimer significantly improves the signal quality at each repeater stage.

In a retimer, the Rx algorithmic model must implement AMI_GetWave and the function must return clock times. The retimer Tx Algorithmic model can also implement AMI_GetWave.

For a retimer, the simulator generates a digital input to the retimer Tx by sampling the Rx AMI_GetWave output at each (tick + UI/2). the digital stimulus has **vlow** of -0.5V and **vhigh** of +0.5V.

AMI Time Domain Simulation with AMI Redriver or Retimer

In time domain simulation, the waveform from upstream channel 1 is propagated to downstream channel 2.

AMI Redriver Time Domain Simulation

Here is the sequence of operations for an AMI redriver link.

1. The solver obtains the impulse response of the upstream analog channel 1.
2. The impulse response is sent through the AMI_Init functions of the upstream transmitter Tx1 and receiver Rx1.
3. The solver obtains the impulse response of the downstream analog channel 2.
4. The impulse response is sent through the AMI_Init functions of the downstream transmitter Tx2 and receiver Rx2.
5. The solver performs AMI time domain simulation of upstream channel.

6. The simulator uses the output on the upstream receiver (Rx1out) as the input to the downstream transmitter (Tx2in). Any AMI_GetWave function in Rx1 or Tx2 is ignored.
7. The solver performs AMI time domain simulation of the downstream channel.

AMI Retimer Time Domain Simulation

Here is the sequence of operations for an AMI retimer link.

1. The solver obtains the impulse response of the upstream analog channel 1.
2. The impulse response is sent through the AMI_Init functions of the upstream transmitter Tx1 and receiver Rx1.
3. The solver obtains the impulse response of the downstream analog channel 2.
4. The impulse response is sent through the AMI_Init functions of the downstream transmitter Tx2 and receiver Rx2.
5. The solver performs AMI time domain simulation of upstream channel.
6. The solver samples the waveform output by upstream receiver Rx1 AMI_GetWave function at 0.5UI after each clock tick returned by the function. The parameter **Rx_ Receiver_Sensitivity** controls the interpretation of a sampled bit as logic 1 or logic 0; see the IBIS Specification (AMI Analysis References) for further information.
7. The sampled waveform is used to generate a new digital waveform as input to the downstream transmitter Tx1. Any AMI_GetWave function in the downstream transmitter Tx2 is ignored. The digital waveform has values of -0.5V for logic 0 and +0.5V for logic 1.
8. The solver performs AMI time domain simulation of the downstream channel.

AMI Statistical Simulation with AMI Redriver or Retimer

AMI statistical simulation calculates the impulse response of a channel. When the system includes one or more AMI Repeater links, statistical simulation differs between redrivers and retimers.

If the Repeater is a redriver, the output of the final **AMI_Init()** at the Receiver (Rx) end of the downstream channel must incorporate the impulse responses of both channels.

1. The solver calculates the Initial, Intermediate, and Final impulse responses of the upstream channel.
2. The solver calculates the Initial and Intermediate impulse responses of the downstream channel.
3. The solver convolves the Final impulse response of the upstream channel with the Intermediate impulse response of the downstream channel to calculate the impulse response input to the Rx **AMI_Init()** function in the downstream channel.
4. The final impulse response output by the downstream Rx **AMI_Init()** function is used to perform statistical simulation of the combined channels.

When multiple redrivers are chained, the final impulse response of any given downstream channel includes the effects of all previous upstream channels and redriver pairs. The solver treats the combination of all previously analyzed channels as the upstream channel.

1. The solver calculates the Initial and Intermediate impulse responses of the next channel in the chain, the downstream channel in the analysis.
2. The solver convolves the final impulse response output by the last Rx **AMI_Init()** function of the previously analyzed channels with the Intermediate impulse response of the downstream channel to calculate the impulse response input to the Rx **AMI_Init()** function in the downstream channel.
3. The final impulse response output by the downstream Rx **AMI_Init()** function is used to perform statistical simulation of the combined channels.

If the Repeater is a retimer, the two channels are independent.

1. The solver calculates the Initial, Intermediate, and Final impulse responses of the upstream channel.
2. The solver calculates the Initial, Intermediate, and Final impulse responses of the downstream channel.
3. The solver uses the Final impulse response for the upstream channel to perform a statistical simulation of channel 1.
4. The solver then uses the Final impulse response of the downstream channel to perform a statistical simulation of channel 2.

Harmonic Balance Technical Notes

The following topics provide technical information about Nexxim Harmonic Balance analysis.

Nonlinear and Linear Solvers

Harmonic balance applies Kirchhoff’s Current Law (KCL) at each node for each independent frequency, then solves the nodal equations supplied by the device models in the frequency domain, yielding the steady state frequency spectrum at every node and every branch. In this way the frequency spectrum for all the currents at each node is “balanced”. The algorithm combines a nonlinear solver with a linear solver. As illustrated in Figure 1, the nonlinear solver is an outer loop that arranges the equations so they can be solved with a linear solver in an inner loop.

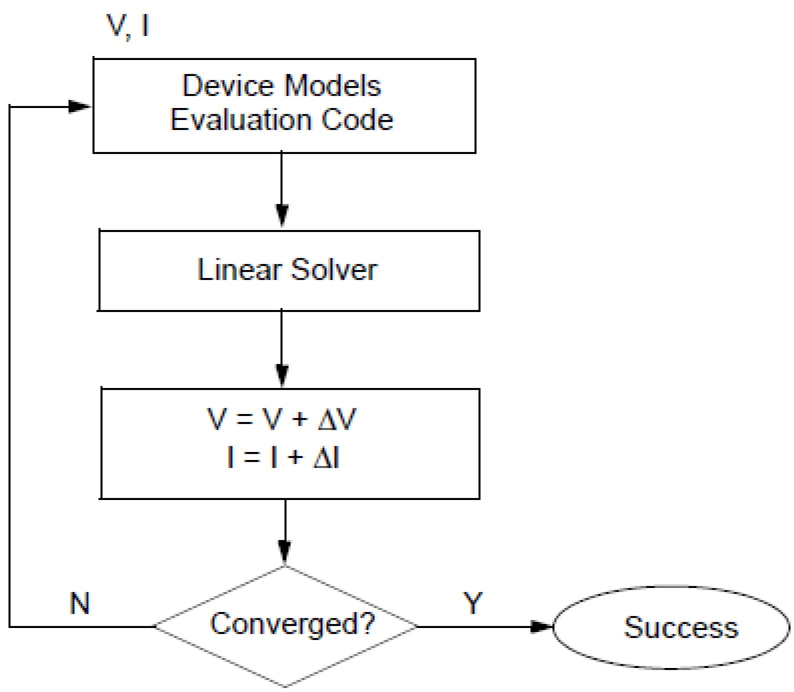


Figure 1. Nonlinear Solver Loop

Nexxim One-Tone Harmonic Balance Analysis

Single-tone or periodic harmonic balance computes the periodic steady-state response of a circuit. A time-dependent response $x(t)$ is periodic with period T when $x(t) = x(t + T)$ for all t, in response to a sinusoidal input with frequency $f = 1/(nT)$, where n is an integer.

A one-tone harmonic balance analysis uses a single tone frequency f with a specified number of harmonics, M.

> **Note:**
>
> When the option **hb.power_of_two_expansion=1** is set, the single-tone harmonic balance algorithm adjusts the maximum harmonic number MAXK so that (2*MAXK + 2) is always a power of 2 (i.e., MAXK = {1, 3, 7, 15, 31, 63, ...}). Power of two expansion is always off for multi-tone analysis.

Nexxim N-Tone Harmonic Balance Analysis

Multi-tone or N-tone harmonic balance analysis computes the steady-state response of a circuit to two or more input sinusoids. The output yields a spectrum with all the elements of the harmonic series of the two fundamental inputs and their intermodulation (sum and difference) frequencies. The Fourier coefficients in multi-tone harmonic balance represent harmonic and intermodulation distortion.

An N-tone mixer harmonic balance analysis uses one or more tone frequencies, f_1 through f_N, each with a specified number of harmonics M_1 through M_N to be used in the analysis. In the typical N-tone mixer analysis, one of the tones has greater power than the others and is analyzed over a greater number of harmonics. The harmonic series for tone $f_1$contains harmonic numbers k_1 = 1, 2, ... M_1. The series for tone $f_N$contains harmonic numbers k_N = 1, 2, ... M_N.

The harmonic balance analysis yields the Discrete Fourier coefficients of the output at the frequencies

$k_1 f_1 + k_2 f_2 + \ldots k_N f_N, \mid k_i \mid \leq M_i$

If the frequency content at harmonic k is [V, I], the frequency content at harmonic $-k$ is the complex conjugate [V*, I*].

Controlling Nexxim Harmonic Balance Analysis

The following topics describe options and statement parameters that control the harmonic balance analysis.

Controlling Nexxim Harmonic Balance Analysis

MAXK Expansion

Each **MAXK** entry M_i specifies the maximum harmonic number to use for the corresponding tone f_i. The analysis is performed at DC and with harmonics 1 through M_1 of tone f_1 (frequency f_1 itself is harmonic 1), the harmonics 1 through M_2 of tone f_2, up to harmonics 1 through M_N of tone f_N, plus their combinations (sums and differences of the harmonics).

> **Note:** The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

For a two-tone mixer analysis the total number of output frequencies is:

$M_1 + 2*M_1*M_2 + M_2$

In a two-tone analysis with two frequencies f_1=1 MHz, M_1=4 and f_2=27 kHz, M_2=2, harmonic balance computes the Fourier coefficient for each of the frequencies listed in the following table at each node in the circuit. The total number of frequencies in the output is 4 + 2*4*2 + 2 = 22.

When the option **power_of_two_expansion=1** is set, the single-tone harmonic balance algorithm adjusts the maximum harmonic number MAXK so (2*MAXK + 2) is always a power of

2 (i.e., MAXK = {1, 3, 7, 15, 31, 63, ...}). Set the option to 0 to disable expansion. Power of two expansion is not used for multi-tone analysis. The default for single-tone analysis is 1 (on).

Output Frequencies for Two-Tone Harmonic Balance Example

Formula	Frequency (Hz)	Formula	Frequency (Hz)
f_1	1.0e+6	$3f_1 + 2f_2$	3.054e+6
$2f_1$	2.0e+6	$4f_1 + f_2$	4.027e+6
$3f_1$	3.0e+6	$4f_1 + 2f_2$	4.054e+6
$4f_1$	4.0e+6	$f_1 - f_2$	0.973e+6
f_2	27e+3	$f_1 - 2f_2$	0.946e+6
$2f_2$	54e+3	$2f_1 - f_2$	1.973e+6
$f_1 + f_2$	1.027e+6	$2f_1 - 2f_2$	1.946e+6
$f_1 + 2f_2$	1.054e+6	$3f_1 - f_2$	2.973e+6
$2f_1 + f_2$	2.027e+6	$3f_1 - 2f_2$	2.946e+6
$2f_1 + 2f_2$	2.054e+6	$4f_1 - f_2$	3.973e+6
$3f_1 + f_2$	3.027e+6	$4f_1 - 2f_2$	3.946e+6

Initial Guess for Vector of Node Voltages

The option **hb.initial_guess** can be used to control the initial guess for the vectors of node voltages and currents used in the harmonic balance calculation.

> **Note:**
>
> The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

Fewer Tones Initial Guess

initial_guess=fewer_tones is the default for multi-tone analysis. For single-tone analysis, **fewer_tones** has the same effect as **initial_guess=dc** (performs a DC analysis, then starts harmonic balance with the DC values as the initial guess). For multitone analyses, **fewer_tones** starts with the DC operating point as the initial guess and solves the circuit with only the first tone enabled. The values with the 1-tone analysis are used as the initial guess for the analysis of the circuit with the first two tones enabled, et cetera until the circuit has been solved with all tones enabled. For example, if the netlist contains:

```
.OPTIONS hb.initial_guess=fewer_tones
.HB tones=(1e6, 1.1e7, 2.1e7) maxk=(3,4,5)
```

The simulator first calculates the DC operating point, then uses those values as the initial guess in solving the single-tone analysis using tone 1e6Hz. The values on the 1-tone analysis then become the initial guess for the two-tone analysis using tones 1e6Hz and 1.1e7Hz. The result on the two-tone analysis then become the initial guess for the full analysis using all three tones.

Fewer Tones Transient Initial Guess

initial_guess=fewer_tones_transient solves the 1-tone problem using transient analysis, then proceeds to solve the multitone problems with the **fewer_tones** algorithm described above. This option has two subordinate options.

- **initial_guess.fewer_tones_transient.tmax=***val* sets the maximum step size. There is no default.
- **initial_guess.fewer_tones_transient.tstop=***val* specifies the stop time.

DC Initial Guess

initial_guess=dc performs a DC analysis, then starts harmonic balance with the DC values as the initial guess. **DC** is the default for single-tone analysis.

Zero Initial Guess

initial_guess=zero starts harmonic balance with vectors of zero voltages and currents as the initial guess.

File Initial Guess

initial_guess=file uses vectors of node voltages and currents saved from a previous harmonic balance simulation. When **file** is the source for the initial guess, the netlist option statement must also include an **initial_guess_file=** [*path*]*filename* option to identify the file. The *filename* specifies the name of an SDF-format file generated by a previous run of HB. The *path* can be omitted if the file is in the directory where the netlist resides. The *path* can specify a subdirectory in the current directory; otherwise, the path must provide the full directory path.

See External Files in the Netlist File Format help topic for further information.

> **Note:**
>
> The path and filename of the initial guess file from a previous run of HB must not be the same as the path and filename of the output file to be generated by the current run of HB. That is, HB cannot read old values from and write new values to the same file.

Transient Initial Guess

initial_guess=transient performs a transient analysis, then uses the final values as the initial guess for the harmonic balance analysis. When **transient** is the source for the initial guess, the netlist option statement can also include two additional options to control the transient analysis.

- **initial_guess.transient.tmax=***val* suggests a maximum step size to use.
- **initial_guess.transient.tstop=***val* sets the stop time.

Initial Guess in Sweeps

By default, harmonic balance applies the initial guess strategy at the beginning of each iteration of a sweep. A separate option, **result_as_initial_guess**, controls this behavior.

- **result_as_initial_guess=0** turns off the use of previous results.
- **result_as_initial_guess=1** enables the use of previous results. However, the use of previous results is the default, so this option setting is never needed.

Nonlinear Solver Controls

The following options control the convergence of the outer, nonlinear solver loop.

> **Note:**
>
> The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

hb.abstol=*val* sets the absolute error tolerance for currents to be used by harmonic balance. Increase (loosen) to aid convergence with less accuracy. The default is 1e-12 Amp.

hb.vntol=*val* sets the absolute error tolerance for voltages to be used by harmonic balance. Increase (loosen) to aid convergence with less accuracy. The default is 1e-7 Volt.

hb.reltol=*val* sets the relative error tolerance for both currents and voltages to be used by harmonic balance. Increase (loosen) to aid convergence with less accuracy. The default is 1e-3 (dimensionless).

max_newton_iterations=*val* sets the maximum number of tries before the nonlinear iterative method gives up trying to converge. Increase if necessary to aid convergence.

limiting_step=*val* sets the maximum absolute update in voltage that can occur from one iteration to the next. Decrease if necessary to aid convergence.

memory_over_speed=*val* controls the use of memory resources. With the default of 0, the simulation emphasizes speed over memory, but switches to a memory-conserving scheme if too many memory resources are being consumed. With **memory_over_speed** set to 1, simulation emphasizes memory over speed. **memory_over_speed** cancels automatic adjustments such as the adjustment of MAXK.

cluster_tol=*val* clusters frequencies that are within a certain tolerance of each other, in order to conserve memory. The final solution is not affected, but the simulation speed may be slowed for certain nonlinear circuits. This option is applied for multitone problems in circuits with no frequency-dependent devices. The default value is 1e-2 Hz.

When **cluster_tol** is in effect, Nexxim saves a message such as the following to the log file:

```
analysis:hb(status): frequency compression = 7.14286
```

This message indicates the percentage reduction in the number of frequencies that results from clustering (about 7% in the example above).

> **Note:**
>
> Frequency-domain elements in the circuit prevent the application of clustering in the harmonic balance analysis. When one or more frequency-domain elements are present, Nexxim saves the following message to the log file:
>
> ```
> analysis:hb(status): presence
> of frequency domain elements: frequency clustering NOT utilized
> ```
>
> In particular, a port element contains a complex impedance, so the port element acts as a frequency-domain device. When clustering is appropriate in harmonic balance analysis, it may be possible to use a simple resistor in place of a port element.

Linear Solver Controls

The following options control the linear equation solver inside the nonlinear solver.

> **Note:**
>
> The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

max_linearsolver_iterations =*val* sets the maximum number of iterations for the solution of the system of linear equations at each Newton step.

linearsolver_restarts sets the maximum number of times the linear solver is restarted until a satisfactory step solution is obtained. The default is 1. (Most users do not need to adjust this option.)

HB Convergence Aids

The **alpha**, **beta**, and **cmin** options aid in harmonic balance convergence by adding conductances and capacitances to the circuit. The values can be increased to aid the convergence of circuits with rapidly changing circuit values.

> **Note:** The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

- **hb.alpha=***val* sets a conductance across all branches. The default value is 1e-12 mho.
- **hb.beta=***val* sets a conductance between every node and ground. The default value is 0.0 mho.
- **hb.cmin=***val* sets a capacitance between every node and ground. The default value is 0.0 Farad.

> **Note:** If floating nodes are present in the circuit, a non-zero **beta** conductance value is automatically added to those nodes. If a value of **beta** has been specified in an **.options** statement in the netlist, that value is used. Otherwise, if **alpha** is greater than zero, the added **beta** conductances have the value **alpha** /10. If **alpha** is not greater than zero, the added **beta** conductances have the value 1e-13 mho.
>
> Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

The **continuation** option controls the strategy to be used when convergence is not achieved with normal methods.

- **continuation=source_stepping_last_tone**

  Source stepping solves the last tone in the **TONES** argument to the **.HB** statement by reducing the input voltages and currents and adding resistances between all nodes and ground, solving this low-voltage case, then gradually increasing the inputs until a solution is found that uses the original input values. Source stepping is the default continuation strategy.

- **continuation=pseudo_transient**

  In the pseudotransient continuation strategy, harmonic balance is run with an additive transient term. Pseudotransient continuation is very useful for problems that are not strongly nonlinear but that have high memory requirements because of the size of the problem or of singularities in the circuit. If source-stepping does not provide a solution or if memory bounds are reached, continuation automatically switches to pseudotransient. This option has two subordinate options:

- **continuation.pseudo_transient.step_reltol_scale=***val*

  The harmonic balance pseudotransient continuation solver relies on the HB solver as the inner solver. This option specifies the relative tolerance of the HB solver with respect to the pseudotransient tolerance. The default for **step_reltol_scale** is 0.8.

- **continuation.pseudo_transient.pt_reltol_scale=***val*

  The harmonic balance pseudotransient continuation solver relies on the HB solver as the inner solver. This option specifies the relative tolerance of the HB solver with respect to the pseudotransient tolerance. Because HB is called repeatedly, this scale factor tightens the tolerances. The default for **pt_reltol_scale** is 0.8. If pseudotransient continuation does not converge in a reasonable number of iterations or shows convergence problems toward the end of the simulation, lowering **pt_reltol_scale** to 0.5 might help. (Most users do not need to adjust this option.)

- **continuation.pseudo_transient.max_h=***val*

  HB pseudotransient continuation has a time-stepping scheme with an internal variable, **h**, that can be thought of as a resistance value. The variable **h** is manipulated internally, but it cannot exceed a maximum value. This option sets the maximum value for **h**. Larger values of **h** cause HB pseudotransient continuation to more closely resemble standard HB. Smaller values of **h** lead to quicker and more accurate solutions, typically with lower memory overhead but with increased simulation time. The default is 5e3.

Handling Strongly Nonlinear Circuits

The **method** parameter in the .HB statement selects between the standard harmonic balance calculation and a shooting method that facilitates single-tone analysis of circuits with strongly nonlinear behavior. The **tstab** parameter in the .HB statement assists HB in finding a solution for difficult circuits, especially when the shooting method is in effect.

> **Note:**
>
> The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

method=hb, standard harmonic balance, is the default for single-tone analysis and is always used for multi-tone harmonic balance analysis.

method=shooting is available for single-tone analyses only. When **method** is set to **shooting**, single-tone harmonic balance uses a time-domain shooting method that is efficient for nonlinear circuits. The **method** option is ignored for multi-tone harmonic balance.

If single-tone harmonic balance fails or takes a long time to complete with the default, try setting **method=shooting** for subsequent runs. Example:

```
.HB TONES=(2.7e6) MAXK=(7) METHOD=SHOOTING
```

The **tstab** parameter in the .HB statement represents additional time needed for transient to settle before HB tries to use the result. A large **tstab** may be used to aid convergence for circuits that have initial nonlinearities, or to allow HB to find one out of several possible solutions. **Tstab** can be applied in both one-tone and multi-tone harmonic balance.

Increasing Accuracy or Speed

The HB statement parameters **auto_refine_solution** and **trim_tol** direct the simulator to automatically adjust the number of harmonics during the simulation to provide either increased accuracy (more harmonics) or faster simulation (fewer harmonics).

> **Note:**
>
> The prefix **hb.** before an option specifies that the value of the option applies only to harmonic balance. Without the prefix, the value applies to all analyses that use the option.

The auto_refine_solution Statement Parameter

When the simulation includes a sweep of power, accuracy can be improved by increasing the number of harmonics as power increases. The **auto_refine_solution** parameter increases the harmonics as needed. The syntax for this statement parameter is:

`auto_refine_solution=no|yes`

The default is **no**, specifying that no automatic increase is to be done.

With **auto_refine_solution=yes**, Nexxim examines the result at each sweep iteration to determine if the solution is sufficiently resolved. If it is not, Nexxim doubles the number of harmonics and repeats the iteration. Start the sweep with MAXK set to a low number, and have Nexxim automatically increase MAXK as needed to maintain accuracy. Example:

```
.HB TONES=(1.01e6,2.7e6) MAXK=(3,7) AUTO_REFINE_SOLUTION=YES
+ SWEEP pwr_val START=1.5 STOP=2.5 STEP=0.5
```

The trim_tol Statement Parameter

IP3 analyses and similar N-tone problems with N>2 may not require the full tensor product using all the harmonics of all the tones. To increase the speed of the analysis, it is useful to “prune” harmonics that have minimal contribution. The **trim_tol** parameter in the HB statement causes Nexxim to prune harmonics automatically after the initial guess calculation. The syntax is:

`trim_tol=``val`

The default value is 0.0, specifying that no pruning is to be done.

The parameter **trim_tol** is a voltage tolerance like **vntol**. The tolerance is applied to the harmonics calculated in the two-tone initial guess. Setting **trim_tol** to a value such as 1e-6 directs Nexxim to eliminate any harmonic whose coefficient has magnitude less than 1e-6. The pruning is then symmetrically applied to higher tones so the result has fewer frequencies than the full tensor product.

With the **trim_tol** parameter, start out with large MAXK values for each of the HB TONES frequencies. This is equivalent to oversampling the frequencies. The **trim_tol** parameter directs Nexxim to prune the harmonics in a way that depends on the circuit specifics. Use this parameter to avoid trying to determine the optimal MAXK values in advance. Example:

```
.HB TONES=(1.01e6,2.7e6,3.221e6) MAXK=(21,21,21) TRIM_TOL=1e-6
```

TONE Parameter on Sources

To ensure that the appropriate harmonic balance frequency is used with a SIN, PWL, or PULSE source, qualify the source instance statement by adding a **TONE** =*tone_value* parameter at the end of the source instance statement in the netlist. The *tone_value* is then used in a subsequent HB statement.

The **TONE** parameter is needed with a SIN voltage or current source only when the driving frequency of the source differs on the frequency at which the harmonic balance is to run. For example, to analyze a circuit driven by a single 1-KHz AC voltage source using harmonic balance at a test tone of 1 KHz over the first 31 harmonics, the netlist is:

```
V1 1 0 SIN(0 1v 1000)
.HB TONES=1000 MAXK=31
```

However, to specify a 1000-Hz tone for harmonic balance while driving the circuit with a 2000-Hz sinusoidal voltage, the voltage source statement should include the **TONE** parameter:

```
V1 1 0 SIN(0 1v 2000) TONE=1000
.HB TONES=1000 MAXK=31
```

The **TONE** parameter is *required* with any PWL or PULSE source. For example, to analyze a mixer driven by sources at 1 MHz and 27 kHz over the first four harmonics of a PWL source and the first two harmonics of a PULSE source, the netlist syntax is:

```
V23 20 0 PWL(0 0 0.1e-6 2.0 0.5e-6 5.0 1.0e-6 0 R) TONE=1.0e6
V1 1 0 PULSE (0 5 0 5e-3 5e-3 27e-3 27.0e3) TONE=27.0e3
.HB TONES=(1.0e+6, 27.0e+3) MAXK=(4, 2)
```

Restriction on Sources with Time Delay

Several voltage and current sources can specify a time delay parameter. However, such a time delay prevents harmonic balance analysis from correctly calculating the steady-state period. Therefore, sources with time delays should not be used in circuits for Nexxim harmonic balance analysis.

HB Output Notes

By default, all node voltages are output to the results file. For large circuits, simulation time can be significantly reduced by specifying a set of values to output, rather than the default of outputting values for every node.

In the **Schematic Editor**, Click the **Edit Output Quantities** feature in the **Solution Setup** window to specify particular outputs.

The netlist can use one or more .PRINT statements to specify the outputs from harmonic balance analysis. If you do not add a .PRINT statement, only node voltages are output. The netlist syntax is:

```
.PRINT HBoutput1 [output2] ...
```

See Controlling Nexxim Output for further information.

General Conversion Parameters

Nexxim can also use the results of HB to compute complex general conversion (GC) transfer parameters. General conversion parameters are also called nonlinear S-parameters, since they are the large-signal analogy to small-signal S-parameters. A GC parameter is the ratio:

$$GC_{ij}\langle m, n\rangle = \frac{B_i\langle m\rangle}{A_j\langle n\rangle}$$

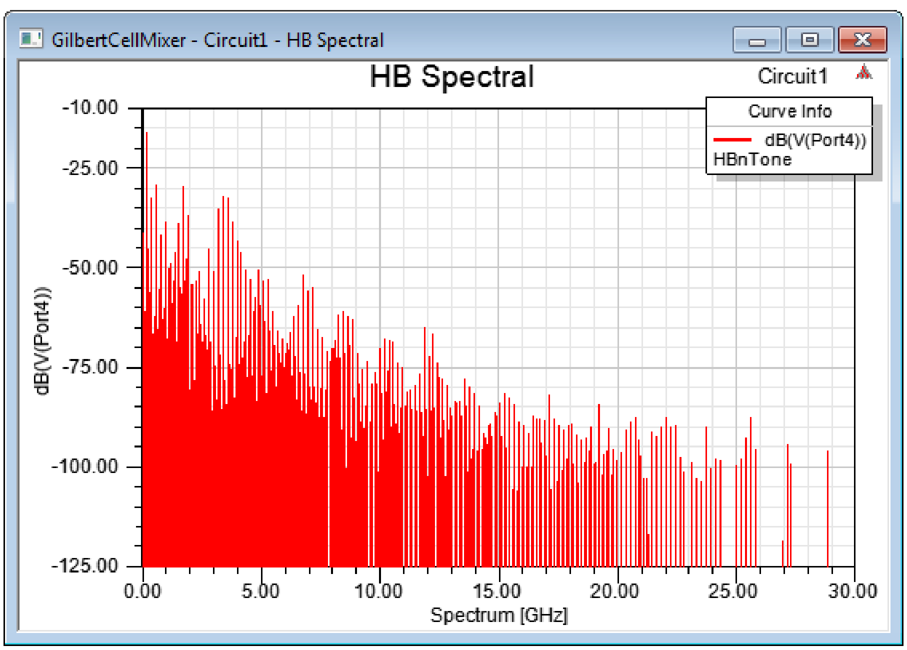
Where:

- *i* is the name of the output port.
- *j* is the name of the input port.
- <*m*> is the number of the harmonic of interest at the output port (+/-*m*)
- <*n*> is the number of the harmonic of interest at the input port (+/-*n*)
- A_j is the magnitude of the incident traveling wave at input port *j*.
- B_i is the magnitude of the incident traveling wave at output port *i*.

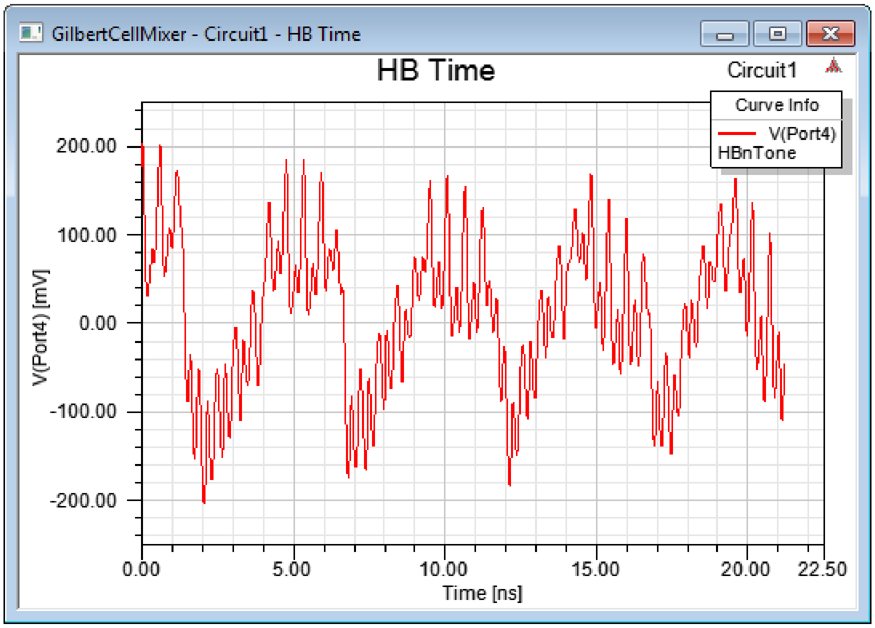
To display general conversion parameters in the Reporter window, select the Sweep domain, under category **General Conversion**. The responses are of the format GC(*output_port*,*input_port*)<*output_harmonic*,*input_harmonic*>.

Time Domain and Frequency Domain Response Formulas

As shown earlier, spectral domain results look like the following graph:



Time domain results look like the following graph:



This topic describes the relationship between the responses in the time and spectral (frequency) domains.

The lines in the spectral domain plot above represent the absolute value, in dB, of the phasors $\underline{V}_n$, corresponding to the response at the harmonic frequencies of the HB analysis, f_n. The complex values of the $\underline{V}_n$can be obtained by creating a data table report.

Each phasor is associated with a time-domain waveform:

$$V_n(t) = \mathrm{Re}\left\{ \underline{V}_n e^{j2\pi f_n t} \right\}$$

$$= \quad V_n^R \cos(2\pi f_n t) - V_n^I \sin(2\pi f_n t)$$

$$= \left| \underline{V}_n \right| \cos(2\pi f_n t + \phi_n),$$

where the rectangular and polar representations of the $\underline{V}_n$ are related by:

$$\underline{V}_n = V_n^R + j\,V_n^I = \left|\underline{V}_n\right| e^{j\phi_n}.$$

For example, a purely real phasor $\underline{V}_n$=5V represents the time-domain waveform:

$$V_n(t) = 5\cos(2\pi f_n t) \quad (\mathrm{V}),$$

and a purely imaginary phasor $\underline{V}_n$=-5*j*V represents the time-domain waveform:

$$V_n(t) = 5\sin(2\pi f_n t) \quad (\mathrm{V}).$$

The complete time-domain waveform is a superposition of the waveforms for all harmonics:

$$V(t) = \sum_n V_n(t) = \sum_n \left|\underline{V}_n\right| \cos(2\pi f_n t + \phi_n).$$

HB Load-Pull Analysis Notes

Nexxim load-pull analysis computes the frequency-dependent input impedance Z of an ideal tuner, using harmonic balance analysis. The harmonic balance analysis is computed over a spectrum of frequencies specified with the **TONES** and **MAXK** entries in the HB statement. The load-pull analysis setup matches the frequencies analyzed by HB to the frequencies of interest specified in the ideal tuner definition.

Impedances at Tuner Frequencies

The frequencies used to compute the tuner impedance are specified in the **TUNER_FREQS** entry in the HB statement. The impedances at these frequencies are calculated on the following formula:

$$Z = Z_r \times \frac{(1+\Gamma)}{(1-\Gamma)}$$

Where Γ (Gamma) is the reflection coefficient of the tuner, and Z_r is a reference impedance. The ideal tuner instance parameters define Γ and Z_r:

```
Rxxxn1 n2PORTNUM=val
+ GAMMA_MAG=val // Magnitude of Gamma
+ GAMMA_ANG=val // Angle of Gamma
+ REF_REAL=val // Real part of reference impedance
+ REF_IMAG=val // Imaginary part of reference impedance
```

The impedance calculation is performed over a combined sweep of **GAMMA_MAG** and **GAMMA_ANG**. **GAMMA_MAG** is swept over a range from zero to one, and **GAMMA_ANG** is swept over a range of zero to 2π (6.283185307) radians. At each sweep point, a harmonic balance is performed to solve for the circuit responses at the calculated tuner impedance.

A tuner frequency can be any appropriate value. If no tuner frequency matches the current HB frequency (or when no tuner frequency is specified), the default impedance parameters **RDEF** and **XDEF** are used as the tuner impedance.

Impedances at Cluster Frequencies

The analysis can incorporate tuner impedances at frequencies of interest other than the actual tuner frequencies. These “cluster” frequencies typically represent harmonics of the tuner frequencies and intermodulation products for multi-tone analyses, but can be any frequencies of interest. The cluster frequencies are specified in the **Z_FREQS** parameter in the tuner definition. The impedances at these frequencies are specified in the corresponding **Z_REAL** and **Z_IMAG** entries. If no cluster frequencies are specified, only the tuner frequencies are analyzed.

Default Impedances

The default tuner impedances are specified in the **RDC**, **RDEF** and **XDEF** entries in the tuner definition. The **RDC** entry specifies the resistance of the tuner at DC (frequency = 0). The **RDEF** entry specifies the default resistance of the tuner. The **XDEF** entry specifies the default reactance of the tuner. The default impedances are used at any frequency not specified in the **TUNER_FREQS** or **Z_FREQS** entries, and when no tuner frequency is supplied.

See *Ideal Tuner* in the Nexxim Components help topics for details on the netlist format for the ideal tuner element.

See Nexxim Load-Pull Analysis Netlist Syntax for an example of the ideal tuner netlist format.

Oscillator Analysis Technical Notes

The following topics provide technical details on Nexxim oscillator analysis.

Using the Oscillator Probe

The oscillator probe identifies a resonance to be analyzed, and provides initial values used in the frequency calculation. The oscillator probe represents an oscillating voltage source with amplitude given by the parameter **A** at the frequency given by the parameter **FREQ**. At all other frequencies, the probe current is zero. The oscillator probe is a frequency domain element; in the time domain (e.g., in transient and DC analyses) the oscillator probe behaves like an open circuit.

The oscillator probe applies a small voltage at a known frequency. Nexxim starts with the frequency specified in the **TONES** entry in the **.OSC** statement and also in the **FREQ** parameter on the oscillator probe, and applies one of several methods to calculate an initial guess at the oscillating frequency. The method used to calculate an initial guess at the circuit oscillating frequency is controlled by the **initial_guess** option (See Initial Guess for Oscillator Frequency). The voltage given by probe parameter **A** is always used as the initial probe voltage. The simulator adjusts the frequency and voltage of the probe until both the real and imaginary components of the probe current (I_{probe}) are zero, that is, $Re(I_{probe}) = 0$ and $Im(I_{probe}) = 0$. The frequency where this occurs is the oscillating frequency.

Placement of a probe is circuit-dependent. If the focus of the oscillation in the circuit is known in advance, place the probe accordingly. If not, try placing the probe across two nodes in the circuit that exhibits a voltage difference when the circuit is oscillating.

Note:

As analysis progresses, oscillator analysis may save messages like the following to the log file:

analysis:osc(status): Probe 0: selecting (*node_name1*,*node_name2*) as alternative location 1.

analysis:osc(status): Probe 0: selecting (*node_name3*, *node_name4*) as alternative location 2.

The *node_name*s are circuit-specific. If oscillator analysis fails, Nexxim automatically reruns the analysis after moving the probe to alternative location 1. If that analysis fails, Nexxim retries using alternative location 2.

Controlling Nexxim Oscillator Analysis

The options described in the following topics allow you to fine-tune the oscillator analysis.

Initial Guess for Oscillator Frequency

The initial guess for the resonant frequency can be controlled by the **initial_guess** option and its subordinate options. Settings for the **initial_guess** option are **transient**, **fewer_tones_**

transient, **freq_sweep**, and **dc**. For single-tone oscillator analysis (with no driven frequency), **transient** is the default (listed as **transient** on the Oscillator Options window in the schematic editor). For the two-tone oscillator analysis that includes a driven oscillation, **fewer_tones_ transient** is the default method (two-tone oscillator analysis is available only in netlists).

Another option, **osc.result_as_initial_guess**, allows Nexxim to use the result on the previous oscillator sweep step as the initial guess for the current step.

Transient Initial Guess

osc.initial_guess=transient obtains an initial guess for the resonant frequency by running a transient analysis.

- The subordinate option **osc.initial_guess.transient.num_cycles** specifies the number of cycles to use in the transient analysis (default is calculated).
- The subordinate option **osc.initial_guess.transient.tmax** specifies the maximum timestep to use in the transient analysis (default is calculated).
- The subordinate option **osc.initial_guess.transient.tstop** specifies the stop time to use in the transient analysis (default is calculated).
- The subordinate option **osc.initial_guess.transient.tstep** specifies the stop time to use in the transient analysis (default is calculated).
- The subordinate option **osc.initial_guess.transient.output_start** specifies an initial period of transient analysis to be ignored in the initial guess.

Fewer Tones Transient Initial Guess

initial_guess=fewer_tones_transient solves for the oscillator frequency using transient analysis, then proceeds to solve the full two-tone circuit. This option has two subordinate options.

- **initial_guess.fewer_tones_transient.tstop=***val* specifies the stop time. There is no default.
- **initial_guess.fewer_tones_transient.tmax=***val* sets the maximum step size. There is no default.

Frequency Sweep Initial Guess

osc.initial_guess=freq_sweep sweeps the frequency of the probe looking for a frequency where the real component of the probe current [Re(Iprobe)] crosses zero from positive to negative. If a zero-crossing is not observed in the sweep range, the initial frequency guess is the frequency at which the minimum value min{abs[Re(Iprobe)]} is obtained. The **osc.initial_ guess=freq_sweep** option has four subordinate options

- Subordinate option **osc.initial_guess.freq_sweep.start** sets the starting frequency. The default is 1e9 Hz.
- Subordinate option **osc.initial_guess.freq_sweep.stop** sets the ending frequency. The default is 5e9 Hz.
- Subordinate option **osc.initial_guess.freq_sweep.num_steps** sets the number of sweep frequencies to be swept. **num_steps** defaults to 50.
- Subordinate option **osc.initial_guess_freq_sweep.method** sets the sweep type. At present, the only method is **linear**.

> **Note:**
>
> The value of probe parameter **A** is always used as the initial probe voltage.

DC Initial Guess

osc.initial_guess=dc uses the value of the **FREQ** parameter on the oscillator probe as the initial guess for the resonant frequency. It also uses DC analysis to calculate initial voltage and current values.

Previous Result as Initial Guess

By default, oscillator analysis uses the initial guess strategy at the beginning of each iteration of a sweep. The option **osc.result_as_initial_guess** allows Nexxim to use the result on the previous oscillator sweep step as the initial guess for the current step. Use of this option can avoid repeated analysis runs.

The default for **osc.result_as_initial_guess** is false (0). Set this option to true (1) to enable the analysis to use the result from each previous step as the initial guess for each new step.

> **Note:**
>
> For **result_as_initial_guess** to be effective, the increments in the variable being swept must be closely spaced or in a region where the frequency does not vary much with each sweep step.
>
> If an analysis step fails while this option is in effect, Nexxim defaults to the normal guess based on transient analysis.

Controlling Convergence of Oscillator Analysis

Oscillator analysis involves circuits that may be highly nonlinear. The **nonlinear_method** option is provided to allow the user to select an alternative strategy if standard Newton-Raphson iteration fails to converge. Settings for the **nonlinear_method** option are **newton**, **damped_ newton**, and **descent**.

- **osc.nonlinear_method=newton** uses the standard algorithm, controlled by harmonic balance options. **Default** (standard Newton-Raphson) is the fastest and least robust (lowest probability of convergence)
- **osc.nonlinear_method=damped_newton** checks the ΔV value at each step. If ΔV exceeds an internally derived maximum, the iteration is rerun. The **damped_newton** method is intermediate in speed and robustness between the default and descent methods.
- **osc.nonlinear_method=descent** uses the slope of the derivative as a guide to the direction for the next iteration. **Descent** is the slowest and most robust of the three methods.

Setting Tolerances for Oscillator Analysis

Two options are provided to control the tolerances for oscillator analysis.

- **.osc.reltol** is the standard relative voltage tolerance used in Newton-Raphson. Specifying **osc.reltol** allows the user to tighten or loosen the voltage tolerance for accuracy vs. convergence, as applied just to oscillator analysis. The default is 1e-4.
- **osc.freq_abstol** is the relative frequency tolerance to be applied to the resonant frequency calculation. The default is 10.

Setting Tolerances for Oscillator Analysis

Two options are provided to control the tolerances for oscillator analysis.

- **.osc.reltol** is the standard relative voltage tolerance used in Newton-Raphson. Specifying osc.reltol allows the user to tighten or loosen the voltage tolerance for accuracy vs. convergence, as applied just to oscillator analysis. The default is 1e-4.
- **osc.freq_abstol** is the relative frequency tolerance to be applied to the resonant frequency calculation. The default is 10.

Floating Nodes Warning

Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

LNA Technical Notes

Linear network analysis performs a frequency-domain analysis of the circuit linearized around the DC operating point. You can also run DC noise analysis, group delay analysis, and AC transfer function analysis.

Linear network analysis can be used with all passive circuits, and with active circuits that operate under small-signal conditions. In linear analysis the signal level and termination values do not cancel the small-signal conditions, and the superposition principle holds.

After a successful linear-circuit analysis, the results reflect the electrical characteristics of the circuit under small-signal conditions. The results of the analysis are linear network parameters such as scattering (S), admittance (Y), hybrid (H), and impedance (Z) parameters, port parameters such as RHO and VSWR, and noise parameters such as NF and FMIN.

> **Note:**
>
> Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

The results also include transfer functions of each output current over each input voltage at each of the test frequencies. Transfer functions are typically complex values.

LNA Netlist Inputs and Outputs

In a netlist, inputs and outputs for LNA are identified by adding port impedance resistors. Additional inputs and outputs for LNA analyses can be specified by adding AC specifications, as discussed later.

Port impedance resistors include a **PORTNUM** parameter to specify the port number and additional parameters to specify the impedance. All resistors with **PORTNUM** values are included as inputs and outputs. LNA calculates the network parameters (S, Y, Z, etc.) and the noise parameters.

See *Resistor, Port Impedance* in the Nexxim Components help for further information.

ZERO_PORT_VALUES Option

Port impedance resistors specify the port impedances with their **PORTNUM** parameters. LNA needs to treat these impedances as external to the circuit. The option **ZERO_PORT_VALUES=1** is therefore required for LNA.

The **ZERO_PORT_VALUES** option is automatically enabled with any design entered through the **Schematic Editor**. It does not appear on the list of **Solution Options**. For designs entered with the Netlist Editor, the **ZERO_PORT_VALUES** option must be entered explicitly with a **.OPTIONS** statement.

See *Resistor, Port Impedance* in the Nexxim Components help for further information.

Nexxim Group Delay Analysis

Optionally, LNA can include group delay analysis. Group delay analysis determines the delay of the propagation of energy at a given frequency point, defined as the derivative of the phase of a network parameter with respect to frequency. The calculation is $GD_{ij} = d\phi_{ij} / d\omega$, where ϕ_{ij} is the phase of S_{ij}.

To compute group delay, numerical perturbation is used to compute the derivative. At each frequency, two analyses are performed, one at the nominal frequency *F*, and a second at the perturbed frequency *F'*. The perturbed frequency is:

$F' = F \times (1.0 + GROUP\_DELAY\_PERT)$

The default for GROUP_DELAY_PERT is 0.001 (0.1%) which is acceptable for most circuits. The typical range is 0.1 to 1.0e-7. However, smaller values may yield more accurate results for circuits with high-Q components, or those with sharp passbands-stopbands. Larger values may yield more accurate results for large circuits that use numerically complex models, where truncation errors can accumulate. Negative values for PERT use a perturbed frequency less than the nominal frequency.

Nexxim DC Noise Analysis

Optionally, LNA can include DC noise analysis.

- *DC noise analysis* calculates the noise spectral density at designated outputs due to thermal, flicker, and shot noise sources in a circuit that has been linearized around the DC bias operating point.
- *Thermal noise* is a variation in current due to random thermal motion of the electrons in physical resistors and bulk materials. Thermal noise is directly proportional to the absolute temperature.
- *Flicker noise* is a random variation in current associated with contamination and crystal defects in semiconductors. Flicker noise is associated with a flow of direct current. Flicker noise generates a noise signal with energy concentrated at low frequencies; because of this 1/*f* frequency dependence, flicker noise is also called 1/*f* noise.
- *Shot noise* is a fluctuation in the current across P-N junctions in diodes and transistors due to random barrier crossings by electrons and holes in the vicinity of the junction. Like flicker noise, shot noise is associated with a direct current flow.

Results of noise simulation include the Output Noise (Onoise) matrix and the Input-Referred (Inoise) matrix. The entries in the Onoise and Inoise matrices have units of Volt/Hz^{½} or Amp/Hz^{½}.

DC Output Noise Matrix

Noise analysis generates a matrix Syy of transfer functions that show the noise power spectral density (noise power at each output at each frequency), in units of Volt^2/Hz or Amp^2/Hz.

$$Syy = [TF(\omega) \times Sxx(\omega) \times TF(\omega)^H]$$

Where:

- Syy is the power spectral density at the output. The size of Syy is (number_of_outputs × number of outputs).
- Sxx is the power spectral density of the noise sources at frequency ω radians per second. Typically, Sxx is a diagonal matrix.
- TF is the transfer function matrix at the output due to the noise sources. TF^H is the Hermitian transpose of the TF-matrix.

The output noise (Onoise) matrix is calculated from Syy with the formula:

$$Onoise = \sqrt{Syy}$$

The interpretation of the output noise matrix depends on the numbers of outputs, and on the presence and numbers of input reference nodes.

- When there is just one output, the result is the square root of spectrum of noise power at that output over the specified range of frequencies.
- When there are N>1 outputs, the result is an N×N matrix. The diagonal elements of the matrix are the square root of the self-noise spectrum for each output port (Onoise_*n_n*). The off-diagonal elements are the square roots of the cross-noise spectrum between each pair of output ports (Onoise_*n1_n2*). Each element is actually a vector showing the transfer functions at all the test frequencies.

DC Input Noise Matrix

When there are one or more inputs, the result includes the input referred noise transfer functions for the noise power at each output referred to each specified voltage or current input. The input-referred noise matrix (Inoise matrix) is calculated on the power spectral density equation:

$$Inoise(out, in) = \frac{Onoise(out, out)}{\sqrt{TF(out, in) \times (TF(out, in))^*}}$$

Where Onoise is the output noise matrix and TF* is the complex conjugate of the transfer function matrix TF.

LNA-Associated AC Analysis

Additionally, the netlist can include voltage or current sources with an **.AC** entry to specify the magnitude and phase to be applied to the AC value. (See LNA Circuit Configuration for an example.) With a corresponding **.PRINT AC** statement, LNA computes AC transfer functions for any outputs listed in the **.PRINT AC** statement. If the **.PRINT AC** statement is omitted, no AC transfer functions are output.

TV Noise Technical Notes

The following topics describe the technical considerations regarding various TV noise issues such as sources, spectrum, and computations.

Sources of Noise

The noise output Onoise is a combination of thermal, shot, and flicker noise.

- *Thermal noise* is a variation in current due to random thermal motion of the electrons in physical resistors and bulk materials. Thermal noise is directly proportional to the absolute temperature.
- *Shot noise* is a fluctuation in the current across P-N junctions in diodes and transistors due to random barrier crossings by electrons and holes in the vicinity of the junction. Shot noise is associated with a direct current flow.
- *Flicker noise* is a random variation in current associated with contamination and crystal defects in semiconductors. Like shot noise, flicker noise is associated with a flow of direct current. Flicker noise generates a noise signal with energy concentrated at low frequencies; because of this 1/*f* frequency dependence, flicker noise is also called 1/*f* noise.

Noise Spectrum

The noise spectrum consists of the power spectrum of the coefficients for the harmonic frequencies.

The large-signal periodic steady-state of a circuit can be represented by the Fourier equation derived in harmonic balance. In a single large-signal circuit, the one-tone periodic steady state transfer function can be represented by the equation:

$$\sum_k C_k e^{j2\pi k f_1 t}$$

Where *k* is the harmonic number $-\infty < k < \infty$, f_1 is the large-signal steady-state frequency, and C_k is the coefficient giving the magnitude of the function at harmonic *k*.

The small-signal AC perturbation can be represented by the equation:

$$Ae^{j2\pi f_i t}$$

Where *A* is a constant and f_i is the small-signal frequency.

The output response of the circuit is the product of the small-signal perturbation and the large-signal transfer function. The response can be represented by the equation:

$$\sum_k A C_k e^{-j2\pi t(kf_1 + f_i)}$$

The frequency of the response has been shifted by f_i. When constant A is normalized to 1 (one), the coefficient C_k gives the magnitude of the response at harmonic *k*.

In a circuit with two large-signal tones f_1 and f_2, the periodic steady state can be represented by the equation:

$$\sum_k \sum_l C_{k,l} e^{j2\pi t(kf_1 + lf_2)}$$

Where *k* is the harmonic number for frequency f_1 and *l* is the harmonic number for frequency f_2. The response of this circuit to the small-signal perturbation above can be represented by the equation:

$$\sum_k \sum_l A C_{k,\,l} e^{-j2\pi t(kf_1 + lf_2 + f_i)}$$

The frequency of the response has again been shifted by f_i. When constant A is normalized to 1 (one), the coefficient $C_{k,l}$ gives the magnitude of the response at the frequency that combines harmonic k of f_1 and harmonic l (letter l) of f_2.

Noise Computation Formulas

The calculation of the noise frequencies is a multiple-step procedure.

First, a set of output frequencies is generated from the swept output frequencies by applying the offsets on the **RELHARMVEC** list to the (n) corresponding frequencies on the **SS_TONES** list.

$$F_{output} = F_{swept} + \sum_{t=1}^{n} (relharmvec[t] \times SSTONES[t])$$

Here is an example that shows how the offsets work.

Suppose the appropriate output frequencies are 1001Hz, 1010Hz, and 1100Hz. This sequence could be specified using the **POI** sweep specification. However, this example uses the **RELHARMVEC** offset to calculate the appropriate frequencies. (Since only one offset is required, the example could use **RELHARMNUM** instead of **RELHARMVEC**.)

```
.TV_NOISE
+ DEC 1 1 100
+ SS_TONES=[1000]
+ RELHARMVEC=[1]
...
```

The decade (DEC) sweep generates F_{swept} values of 1, 10, and 100. An input frequency is calculated for each swept value. In this example with one **SS_TONES** frequency, there is only one product:

```
(relharmvec[1] × SS_TONES[1]) = (1 × 1000) = 1000.
```

Then:

$F_{output}[1] = 1 + 1000 = 1001$

$F_{output}[2] = 10 + 1000 = 1010$

$F_{output}[3] = 100 + 1000 = 1100$

The default for **RELHARMVEC** is a vector of zeros; with this value, the output frequencies are just the swept frequencies.

For the gain calculation, Nexxim TV Noise computes the input frequency F_{input} at each of the values of the output frequency F_{output} by adding sum of the products of each of the (*n*) steady-state frequencies F_{SS_TONES} multiplied by the corresponding **REFSIDEBAND** coefficient:

$$F_{input} = F_{output} + \sum_{t=1}^{n} (refsideband[t] \times SSTONES[t])$$

> **Note:**
>
> When an **IPROBE** device is specified, the netlist must also have the **REFSIDEBAND** list of coefficients for the **SS_TONES** frequencies in order to perform the conversion gain calculation. When no **IPROBE** device is specified, the **REFSIDEBAND** entry is optional; the default coefficient value is 1.

Here is a simplified example. This example uses a single output frequency.

```
.TV_NOISE
+ POI 1 1000000 // Single frequency for the sweep
+ SS_TONES=[1000, 2000, 3000]
+ RELHARMVEC=[0, 0, 0] // This is the default
+ REFSIDEBAND=[-1, 1, 2]
```

The resulting calculation is:

$SUM1 = (0 \times 1000) + (0 \times 2000) + (0 \times 3000) = 0$

$F_{output} = F_{swept} + SUM1 = 1000000 + 0 = 1000000$

$SUM2 = (-1 \times 1000) + (1 \times 2000) + (2 \times 3000) = 7000$

$F_{input} = F_{output} + SUM2 = 1000000 + 7000 = 107000$

Typically, the sweep specification generates several output frequencies, perhaps using **RELHARMVEC** coefficients other than zero.

The Onoise output vector is calculated for the specified output devices at the output frequencies. Nexxim combines the harmonic frequencies (from -SS_MAXK to +SS_MAXK, plus 0) of each of the steady-state tones (specified by **SS_TONES**) with the output frequencies to calculate a vector of internal noise frequencies that can contribute to the Onoise output.

> **Note:**
>
> If any of these internal noise frequencies is zero, the flicker noise (1/*f*) component is set to 0.0, and a warning message is displayed.

The formula is too complicated to write explicitly. Here is a numerical example.

```
.TV_NOISE
+ POI 1 1000000 // Single frequency for the sweep
+ SS_TONES=[1001, 20000]
+ RELHARMVEC=[0, 0] // This is the default
+ SS_MAXK=[1, 2]+
```

Using the default for **RELHARMVEC**, F_{output} is just the swept frequency, one megahertz.

The F_{onoise} vector is calculated for every combination of harmonics (-1, 0, +1) for the 1001-Hz steady-state tone and harmonics (-2, -1, 0, +1, +2) for the 20000-Hz steady state tone. The resulting vector has (3 × 5) = 15 elements:

```
1000000 + (0 × 20000) + (0 × 1001) = 1000000
1000000 + (0 × 1001) + (1 × 20000) = 1020000
1000000 + (0 × 1001) + (2 × 20000) = 1040000
1000000 + (0 × 1001) + (-2 × 20000) = 960000
1000000 + (0 × 1001) + (-1 × 20000)= 980000
1000000 + (1 × 1001) + (0 × 20000) = 1001001
1000000 + (1 × 1001) + (1 × 20000) = 1021001
1000000 + (1 × 1001) + (2 × 20000) = 1041001
1000000 + (1 × 1001) + (-2 × 20000) = 961001
1000000 + (1 × 1001) + (-1 × 20000) = 981001
1000000 + (-1 × 1001) + (0 × 20000) = 998999
1000000 + (-1 × 1001) + (1 × 20000) = 1018999
1000000 + (-1 × 1001) + (2 × 20000) = 1038999
1000000 + (-1 × 1001) + (-2 × 20000) = 958999
1000000 + (-1 × 1001) + (-1 × 20000) = 978999
```

Nexxim then calculates the noise output at these frequencies.

Periodic Transfer Function Computation Formulas

The netlist must specify the combination of harmonics of the steady state **SS_TONES** frequencies, using either **MAXSIDEBAND** or **SIDEBAND**.

When the **SS_TONES** group box has only one frequency, click the **MAXSIDEBAND** entry to specify the maximum harmonic of that frequency. The input frequencies is calculated using the formula:

$F_{input} = F_{output} + \{-K \times f1 \text{ to } K \times f1\}$

Where F_{output} is the output frequency (possibly swept), K is the value specified for **MAXSIDEBANDS**, and *f1* is the frequency specified in the **SS_TONES** entry.

Alternatively, specify a set of specific sidebands to use for the single output frequency, using the SIDEBAND entry. For example, if the SS_TONES frequency is *f1*, the entry

```
SIDEBAND=[1,3,-5]
```

specifies a calculation of the input frequencies using the following set of frequencies:

$F_{input} = F_{output} + \{(1 \times f1), (3 \times f1), (-5 \times f1)\}$

When the **SS_TONES** group box specifies more than one frequency, you must Click the **SIDEBAND** entry to specify the combinations of harmonics that are of interest as input frequencies (**MAXSIDEBANDS** is not valid with multiple **SS_TONES** frequencies). Each combination of harmonic numbers (positive integers, negative integers, or zero) contains one value for each of the frequencies in the **SS_TONES** group box. For example, if the **SS_TONES** group box contains two frequencies *f1* and *f2*, the entry

```
SIDEBAND=[1,0,2,-1,1,1]
```

specifies three combinations of harmonics of the two SS_TONES frequencies, and the input frequencies is calculated using the formula:

$F_{input} = F_{output} + \{(1 \times f1 + 0 \times f2), (2 \times f1 - 1 \times f2), (1 \times f1 + 1 \times f2)\}$

Floating Node Warning

Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

Floating Node Warning

Nexxim issues a warning message if the circuit contains floating nodes or nodes with single connections. Only nodes with a single branch connection that is both conservative and constitutive are flagged. For example, the controlling nodes in a VCVS have a constitutive/behavioral/relational connection to the controlled nodes, but not a physical/structural/conservative connection, and therefore are not flagged.

26 - Glossary

This section defines the terminology used in the Ansys desktop help topics. Terms are listed in alphabetical order.

A B C
D E F
G H I
J K L
M N O
P Q R
S T U
V W X
Y Z

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.

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

C

D

E

I

O

P

S

U

V

W